DEVELOPING A KNOWLEDGE DISCOVERY FRAMEWORK IN HIGHWAY PROJECTS USING INDUSTRIAL FOUNDATION CLASS (IFC) EXTENSION

by

Zhenhua Shen

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Civil and Environmental Engineering

Charlotte

2016

Approved by:

____

Dr. Hyunjoo Kim

____

Dr. Jy Wu

____

Dr. Don Chen

____

Dr. Na Lu

____

Dr. Mona Azarbayjani

ABSTRACT

ZHENHUA SHEN. Developing a knowledge discovery framework in highway projects using Industrial Foundation Class (IFC) extension. (Under the direction of DR. HYUNJOO KIM)

Highway-project data are being accumulated at a dramatic pace with the advance of information technology; consequently, the need for discovering useful knowledge among the historical data to prevent highway-project construction delay is increasing. However, the efficiency and effectiveness of most current highway-project knowledge discovery approaches are very low. A new systematic approach integrating Industrial Foundation Class (IFC), data warehouse, and data mining technologies is needed to address this issue. The proposed highway IFC data schema adopts an object oriented design approach to integrate interdisciplinary highway information into one universal intelligent format. IFC data is machine readable and suitable for complex query and reasoning, thus the highway-project knowledge discovery process can be highly automated by deploying IFC in the data warehouse with data mining function. This above-mentioned approach is tested with a 3D highway data warehouse prototype system in a case study of KICT project. The prototype system stores the highway IFC data and automatically generate the feature data and target data according to user requirements. Two data mining models are built to find useful patterns among the data. The performances of the models are compared. A cost sensitive analysis is especially performed to take into consideration of the practical cost of a project manager’s decision. Overall, this research has developed a highly automated approach to discover highway-project knowledge from historical data in an effective and efficient manner, which can help highway-project managers make timely and wise decisions to avoid potential construction delay and the associated costs.

ACKNOWLEDGMENTS

First, I would like to thank my academic advisor Dr. Hyunjoo Kim who is a role model of a diligent and effective researcher for me. He led me into the interesting field of construction data modeling and data mining. Dr. Kim guided me to develop the thesis topic and helped me to improve my research. I really appreciate his encouragement and suggestion during my PhD study. This thesis would not have been possible without his valuable advice.

I also want to express my special gratitude for Dr. Jy Wu. As the doctoral program director, he continuously supported me throughout the course of my dissertation. His insightful ideas and inputs made my PhD study more productive. Dr. Wu's great sense of humor helped establish a relaxed atmosphere in my PhD presentation. Thank you very much.

Many sincere thanks to Dr. Don Chen (a knowledgeable professor in BIM and highway construction management field), Dr. Na Lu (a nice, wonderful professor, Good luck at Purdue!) and Dr. Mona Azarbayjani (a smart, helpful professor who always smiles). Thank you all for serving on my committee and providing constructive suggestions and helpful assistances.

This thesis is dedicated to my loving family. A special appreciation goes to my father and mother who value the importance of education. They always stand by my side. They give me love, support and understanding during my study abroad in the US. I love my family. God bless everyone.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	artificial intelligence
AEC	architecture, engineering, construction
AUC	area under ROC curve
BIM	building information modeling
CBR	case based reasoning
FNR	false negative rate
FPR	false positive rate
IFC	industrial foundation class
KDD	knowledge discovery in database
OLTP	online transactional processing
OLAP	online analytical processing
ROC	receiver operating characteristic
TPR	true positive rate
TNR	true negative rate

CHAPTER 1: INTRODUCTION

1.1 Background and Motivation

The highway industry is overwhelmed with the explosion of data collected and stored during the project life cycle (Woldesenbet, 2014). An entire highway project life cycle usually includes planning, design, bidding, construction, and operation & maintenance phases. Each phase involves various activities including project evaluation, land surveying, estimation, design alternative selection, earthwork calculation and other procedures. Professionals from different disciplines need to collaborate in a team and exchange data from each other intensively, which produces and circulates a large amount of interdisciplinary data with various formats across different domains (Pniewski, 2011). The amount of highway project data stored in database can add up to a large volume. Enterprise databases usually range from gigabytes to terabytes. Department of Transportation (DOT) also captures and preserves various data in database for data analysis purpose. For example, the pavement management division of Oklahoma DOT alone collects approximately 52 million pieces of data annually (Lewis, Jeong, & Woldesenbet, 2013). As more sensors are put into use in the highway industry, the data quantity is even accelerating.

Since the highway project data is being accumulated at a dramatic pace, the need for extracting useful knowledge from the data is also increasing. Interdisciplinary information is embedded in the highway project data that could be a rich source for knowledge extraction. Highway project managers are interested in finding patterns from the flood of highway project data (Woldesenbet, 2014). Knowledge can be obtained by interpreting the patterns. Without knowledge, it is hard for highway project managers to understand the project nature and make forecast. The lack of understanding of highway projects' nature may lead to undesirable management practices, which could cause serious a project delay or cost overruns (Kim, 2014).

However, discovering knowledge from the large amount of interdisciplinary highway project data effectively and efficiently is still difficult in practice. This is because manual feature/target data preparation from the large amount of interdisciplinary highway data is complex, time-consuming, and requires frequent professional meeting and collaboration. Highway project is information intensive. Besides, a highway project has the interdisciplinary nature. There are hundreds of different types of data for different disciplines. Each type of the data has its own format and standard. The data organization structure, naming convention may be various. Some data are even collected and recorded in ad-hoc manners (Soibelman & Kim, 2002). Usually, raw data is not appropriate for direct knowledge discovery. Instead, data mining needs to be applied on well-constructed feature data and target data. The attribute of interest is called target attribute. Feature attribute is the attribute that may have an important effect on target attribute. Both of them represent the abstract characteristics of the research subject. The research also points out that the preparation of feature/target data is found to be very important to the process of knowledge discovery. Feature generation is widely known and agreed to be the key to the success of knowledge discovery process. Manual feature generation usually consumes a large amount of the total effort invested in the entire knowledge discovery process. In order to extract useful information from a huge amount of data and then combine them into one abstract feature/target data, many professionals need to be gathered frequently. Traditional data modeling approach saves the data in a way that computers may not understand. The data saved in a computer are usually in the types of nominal names and numeric values. A computer will recognize the data as a series of numbers and characters in which, professionals are required to manually search, check, aggregate, manipulate the unorganized raw data. Significant time and efforts are needed to find and generate corresponding feature/target data. Decision makers often have to wait much time for responses from the professionals to make the feature/target data ready. Frequently, more time is spent in the feature/target data generation process instead of the real work of knowledge discovery (Hammad, AbouRizk, & Mohamed, 2014). The manual feature/target data generation process tends to be very costly.

1.2 Problem Statement

The amount of disciplinary highway project data is accumulating, which could provide a valuable resource for knowledge discovery and thus be used to facilitate highway project decision making. On the other hand, efficiently and effectively discovering knowledge from the highway project data is still very difficult in reality. Raw highway data is enormous in size and interdisciplinary. It usually consists of different formats. Manually generating feature/ target data from the raw data can be very costly. The explosion and chaos of highway data significantly decreases the efficiency and effectiveness of knowledge discovery process in the highway industry (Woldesenbet, 2014).

As a result, a new systematic approach is needed to address the current difficulty. This research proposes a new systematic approach integrating Industrial Foundation Class (IFC), data mining, data warehouse technologies to meet the research need. IFC is an object oriented data schema. It integrates disciplinary highway project data into one single intelligent format, thus automates the feature/target data generation process. Data mining methods such as decision tree and case based reasoning (CBR) are utilized to extract useful knowledge. Data warehouse serves as a data analytics repository system to increase the reusability of the whole system. A series of three objectives to meet the research need are presented in the following section.

1.3 Objectives

The overall goal of this research is to develop a new systematic approach to automatically discover knowledge from highway project data by using IFC, data mining, data warehouse technologies. Specific objectives to achieve this goal are detailed below.

- Objective 1: to expand IFC in highway projects

Objective 1 proposes to create an IFC data model for highway project elements. This research expands current IFC development approach to highway projects by following the international standards. Real project data will be created in the format of physical IFC file which can be used to achieve the goal shown in Objective 2.

- Objective 2: to develop a data warehouse framework for highway projects

Objective 2 proposes to develop a systematic framework for highway projects using a highway IFC data schema, data mining and data warehouse. The structure of the framework is designed and proposed in this research to implement a system. The mapping relationship between the highway IFC data schema and a data warehouse structure is investigated. A prototype system is implemented based on the designed framework to test expected functionalities. An increased level of automation for feature/target data generation is also demonstrated through the system.

- Objective 3: to perform a data mining analysis to discover knowledge from the implemented system

Objective 3 proposes to perform a data mining analysis in the prototype system and find interesting patterns or useful knowledge in a set of real highway project data. Decision tree and case based reasoning (CBR) methods are utilized to conduct the analysis.

1.4 Proposed Approach

The overall proposed research approach consists of four steps (Figure 1) such as Highway IFC data model design, data warehouse system design, data mining methods design and prototype system implementation.

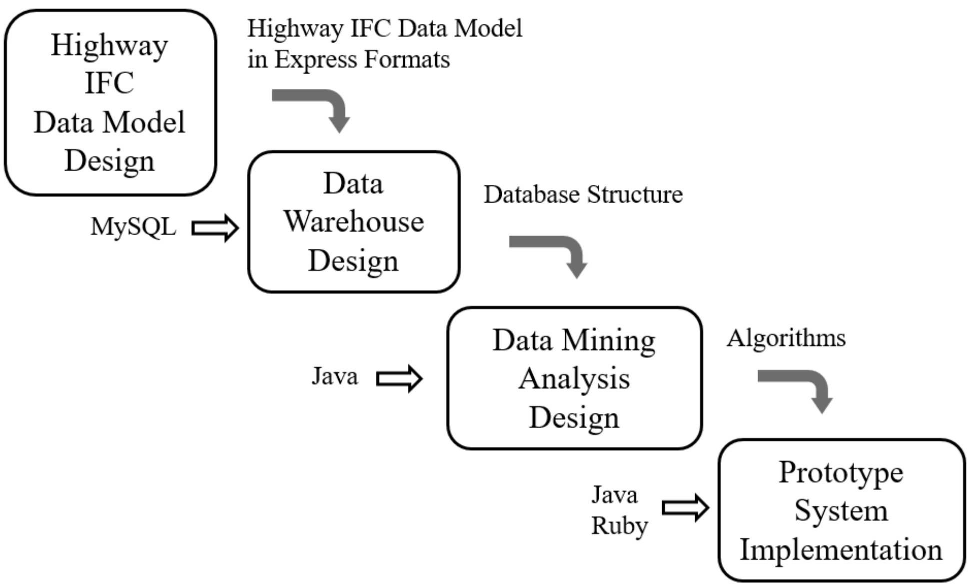


Figure 1: Overall research approach

In the highway IFC data model design, IFC schema is expanded from building a set of project data in highway project data. In the data warehouse design step, a data schema is designed to transform and store the highway IFC data in the database. The mapping relationship between highway IFC data and database tables is established. In the data mining methods, design step, decision tree and case based reasoning (CBR) algorithms embedded in prototype system are presented. These methods will be applied to highway project data to find patterns. In the last step, prototype system modules, architecture, and user interfaces are designed.

The main difference between conventional approach and the proposed approach is that the proposed approach describes data in an object oriented way. By following this approach, data warehouse can intelligently generate the feature/target data, thus the automation level of the whole process can be improved.

1.5 Research Scope

Highway project data covers a vast area of domains. However, the scope of this research is limited to highway structure data and construction activities/tasks data. The research focuses on the quantity, cost and schedule information. The data mining methods used in the research are limited to decision tree and case based reasoning (CBR). These two data mining methods require less domain knowledge or parameter setting, therefore especially appropriate for interdesciplinary knowledge extraction.

1.6 Research Significance

This research provides a systematic solution to extract knowledge in highway projects. Furthermore, it improves the efficiency and effectiveness of the highway knowledge discovery process by introducing IFC, data mining and data warehouse technology. This research would be beneficial to highway project managers as a proposed framework can help them efficiently find knowledge in making better decisions.

1.7 Thesis Organization

The thesis is organized as follows: Chapter 2 provides the literature review and the research gap. Chapter 3 presents the research design and methodology. Chapter 4 continues with a case study of proposed approach on KICT project data. Chapter 5 is the conclusion of the whole research.

CHAPTER 2: LITERATURE REVIEW

This section provides a literature review on the latest developments in the related areas. The theoretical background in the research can be classified in the following areas: data mining methods, data warehouse approach and Industrial Foundation Class (IFC) technology. Section 2.1 provides an overview of the current data mining methods. Section 2.2 reviews the development of data warehouse approach. Section 2.3 introduces the concept and practice of IFC technology. Research gap is discussed at the end of the chapter.

2.1 Data Mining

2.1.1 Data Mining Concept

This research will utilize data mining as an advanced data analysis method to identify a pattern(s) and extract knowledge in highway project data. Data mining is considered an interdisciplinary field involving concepts from machine learning, statistics, mathematics, and artificial intelligence (Shu-Hsien Liao, 2012). Fayyad defined data mining as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Fayyad, 1996). Data mining can be described as "*the union of historical and recent developments in statistics, AI, and machine learning. These techniques are then used together to explore data and find previously-hidden patterns within*" (Kumar & Bhardwa, 2011). A pattern is an expression describing facts in a subset of a set of facts or rules. The expression is called a pattern if it is simpler than the enumeration of all facts in the subset of facts (Batth, 2014). In highway projects, project managers could be assisted in his daily decision making through the pattern found. One discovered pattern might be “a construction activity performed by a certain team using excavator in rainy days that has a pattern of constant schedule delay by 10%”. Patterns might also be much more complex, taking into account other project factors such soil type, material, construction technique, etc. Generally, data mining can be categorized as descriptive method and predictive method. Predictive method is used to forecast future value or predict a categorical value based on input data (Chris Rygielskia, 2002). This research will focus on two predictive methods: decision tree and case based reasoning (CBR).

2.1.2 Decision Tree Method

The decision tree method is used in this research for the capability of predictive analysis. Studies show that the tree structure in the decision tree method is the most commonly used pattern for knowledge representation (Li-Yen Chang, 2005). Figure 2 illustrates an example of the result of a decision tree structure. It has a root node (decision node) and terminating nodes (result node). The decision node is divided by the splitting rules. The right and left arrows coming out of the decision nodes represent the branches of the tree. Both of them are connected to the result nodes (Witten, Frank, & A.Hall, 2011). Given a set of input data, the attribute values of the data are tested on the decision nodes. A path then can be traced from the root node to a result node, which stores the prediction value (Han, Kamber, & Pei, 2012). Decision trees can easily be transformed to a set of rules for future decision making.

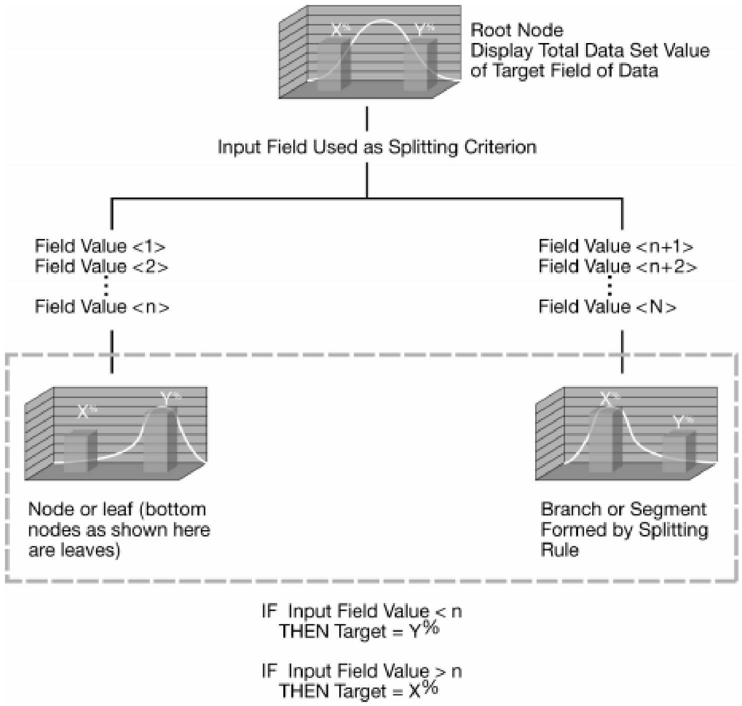
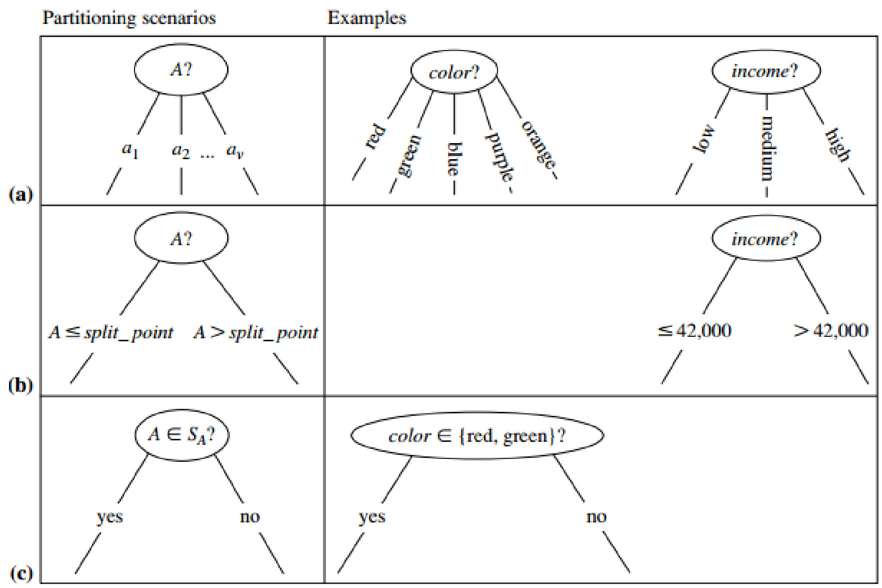


Figure 2: A simple example of decision tree structure (SAS, 2011)

There are three partition scenarios for a node in the decision tree (Table 1). If the splitting attribute on the node is discrete-valued. The outcomes of the test correspond directly to the known values of A shown in Table 1. Branches are created for each known value. If A is continuous-valued, the test has two possible outcomes. The two outcomes often correspond to two conditions: “A<=split_point” and“A>split_point”. Two branches are created from the node for the outcomes. If A is discrete-valued and a binary tree must be produced, the test at node N is of the form “$A \in S_A$?” where S_A is the splitting subset for A. It is a subset of the known values of A. Each outcome would be “Yes” or “No”. One branch is created from the node for each corresponding outcome. This research will build the tree structure to identify the relationship between input project data attributes (root nodes and decision nodes) and target result (terminating nodes).

Table 1: Partition of decision tree node (Han, Kamber, & Pei, 2012)



2.1.3 Case Based Reasoning Method

Case based reasoning (CBR) is another method used in this research to make a prediction. The principle of CBR is inspired by human reasoning and memory mechanism (Figure 3).

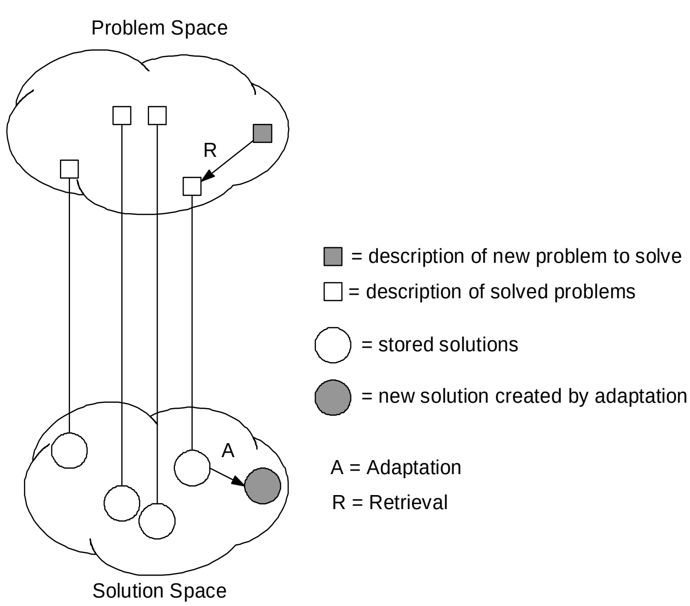


Figure 3: Human reasoning and memory mechanism (Aamodt & Plaza, 1994)

If a person successfully solves a problem, the description of the solved problem is then saved in his/her memory. If the person later meets a new similar problem as before, the description of previous solved problem is recollected and the person can try to follow the same strategies to retrieve a similar solution. Thus, reapplying previously successful solution in this new but similar context may provide a more practical solution (Pantic, 2000).

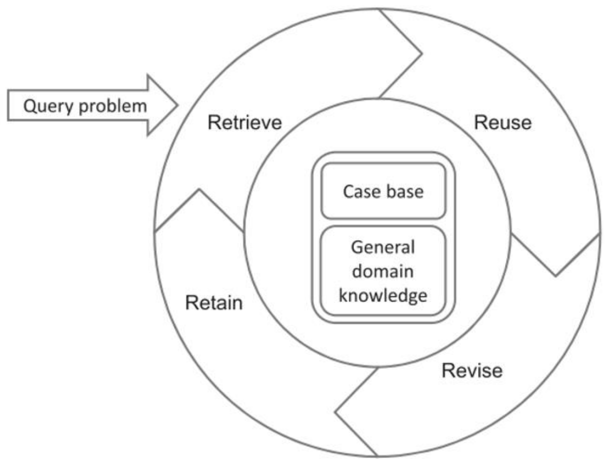


Figure 4: CBR working cycle (Aamodt & Plaza, 1994)

In this research, CBR is used to apply the human reasoning and memory mechanism in a data analysis process. A case is "*a contextualized piece of knowledge representing an experience that teaches a lesson fundamental to archiving the goal of the reasoner*" (Kolodner, 1993). A case describes one particular situation. A case record stores the valuses of several attributes associated with that situation. CBR is defined as reasoning by remembering previous cases: previously solved cases are used to suggest solutions for new but similar problems (López, 2013). This research will follow a four-step scheme of the CBR working cycle to construct a CBR system. Aamodt and Plaza proposed the scheme in 1994 (Aamodt & Plaza, 1994): "*Retrieve the most similar case(s); Reuse the case(s) to* *attempt to solve the current problem; Revise the proposed solution if necessary; Retain the new solution as a part of a new case*". The whole process is drawn in Figure 4.

2.1.4 Data Mining Application

Deployment of data mining techniques such as decision tree and CBR has the potential of allowing project managers to identify patterns within a collection of design and construction data (Soibelman & Kim, 2002). Data mining has been used in several applications within construction in the past. Akhavian (2013) applied data mining techniques to improve the process of construction fleet operations (Akhavian, 2013). The technology has also been used to predict accuracy of concrete compressive strength (Chou, 2011) and structural systems (Wang & Ghosn, 2006). Data mining has been used to solve highway related problems too. Tah et al. explored the use of data mining in the construction planning of highway bridge projects (Tah, Carr, & Howes, 1999). Data mining and neuro-fuzzy methods were utilized to predict a short-term highway travel time. (Coufal & Turunen, 2003). Data mining algorithms were put forward to extract a model to describe highway traffic flow state of similar region section (Chen, Zhang, & Liu, 2009). Data mining were also applied to the area of traffic forecast (Lin Yu, 2011), crash analysis, pavement management (Zhou, Wang, Wang, & Reichle, 2009).

2.2 Data Warehouse

2.2.1 Data Warehouse Concept

This research will utilize the data warehouse technology to increase the efficiency and reusability of traditional knowledge extraction process. Data warehouse is a concept usually connected with data storage and management system specifically designed for a data analysis. It can be seen as a central repository of data that is created by integrating data from one or more disparate sources (Ahmad, 2000). It also facilities to extract the data in the data mining process and stores the data in a form appropriate for complex query and future analysis. It was been reported that data warehouses store current as well as historical data and are used for creating trending reports for management reporting such as weekly and monthly comparisons (Devlin, 1996). in the construction industry where the project manager is daily confronted with various types of project data (drawings, invoices, documents, specifications and so on) and faces challenging tasks daily in which data warehouse models could provide him/her with a type of knowledge extraction system. Data warehouse is a flexible and powerful environment for problem identification, and pattern discovery, which could result in better management decisions (Dayi, 2008).

2.2.2 Differences between Data Warehouse and Traditional Database

Data warehouse is an online analytical processing (OLAP) system while traditional database is an online transactional processing system (OLTP). OLTP is very common in commercial applications such as online banking system, student management system, etc. They are designed for day-to-day operations including insert, delete, update and search (Park & Kim, 2013). An OLTP system is known to be customer-oriented and is used for transaction and query processing by clerks, clients, and information technology professionals (Han, Kamber, & Pei, 2012). Most state department of transportation (DOT) highway information system is based on OLTP database (Table 2). The databases range from in-house spreadsheet to conventional online transactional programs such as Microsoft SQL server and Oracle server. For example, highway inventory system, Proposal and Estimates System (PES), Pavement Management information System (PMIS) are all based on SQL server or Oracle database (Woldesenbet, 2014). An OLTP system consists mainly of short atomic transactions because the efficiency of transactional operation is its biggest concern. An OLTP database focuses on optimizing daily data transaction performance rather than providing in-depth data analysis. As mentioned in previous section, performing complex queries in OLTP databases would substantially degrade the performance of other operational tasks (Han, Kamber, & Pei, 2012). The running speed of complex queries would be very low. OLTP database is not an appropriate data storage and management system for data mining.

Table 2: Highway databases used in DOT (Woldesenbet, 2014)

Division	Database	Category	Type of Data	Method
System Planning/ Research	Grip lite/ Highway Inventory	Roadway Inventory	Functional Class, Right of Way, Route Classification, Terrain Area Type, right-of-way, railroad crossing, etc.	Manual / Semi-Automated
		Traffic	Average Annual Daily Traffic (AADT), signals, lightings, traffic control, crash statistic, etc.	
		Bridge Inventory	Bridge span, width, length, load limit, inspection reports, etc.	
Pre-Construction	In-house Spreadsheets/	Preliminary Engineering Cost Data	Work efforts (engineering hours, number of sheets, etc.)	Manual
Bidding	PES/ LAS	Contract Documents	Bid information, award contracts proposal holders, advertisement, pre-bid, etc.	Semi-Automated
Construction Division	SiteManager	Construction Data	Reported quantity, material, change order contractor payment etc.	Manual
Pavement Management	Pavement Management System (PMS)	Pavement History	Pavement surface type, thickness, composition, etc.	In-house - Automated
		Structural Data (Distress Data)	Longitudinal Cracking, Transverse Cracking, Patching, Spalling, Fatigue, etc.	Consultant
		Functional Data	Average Roughness, Ride, Rut etc.	In-house
		Other (structural)	Friction, Deflectometer (FWD), ESAL	In-house

This research adopts the data warehouse technology rather than a traditional approach. One main difference in the database structure of the data warehouse is that the tables in OLAP database are de-normalized while the tables in OLTP database are normalized. The de-nomalized organization of data makes complex query process more efficient in OLAP database, which significantly improves the efficiency of advanced data analysis such as data mining process. Data warehouse is a kind of OLAP database which is designed to support advanced data analysis. According to William H. Inmon, "*A data warehouse is a* *subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision making process*" (Inmon, 1996). The subject-oriented means data warehouse is organized around subjects in a business. A data warehouse typically provides a view of particular subject issues and its related factors (Qian & Qing, 2009). The word "integrated" means data warehouse usually stores different kinds of heterogeneous data which should be compatible with various data formats (Han, Kamber, & Pei, 2012). Time-variant means time sequence is emphasized when data warehouse stores data. Organization of data depends upon time. Nonvolatile means data warehouse does not require transaction processing. in terms of data management operation, Other differences between OLTP and OLAP are listed in Table 3.

Table 3: OLTP and OLAP (Han, Kamber, & Pei, 2012)

Feature	*OLTP*	*OLAP*
Characteristic	operational processing	informational processing
Orientation	transaction	analysis
User	clerk, DBA, database professional	knowledge worker (e.g., manager, executive, analyst)
Function	day-to-day operations	long-term informational requirements decision support
DB design	ER-based, application-oriented	star/snowflake, subject-oriented
Data	current, guaranteed up-to-date	historic, accuracy maintained over time
Summarization	primitive, highly detailed	summarized, consolidated
View	detailed, flat relational	summarized, multidimensional
Unit of work	short, simple transaction	complex query
Access	read/write	mostly read
Focus	data in	information out
Operations	index/hash on primary key	lots of scans
Number of records accessed	tens	millions
Number of users	thousands	hundreds
DB size	GB to high-order GB	$\geq$ TB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time

2.2.3 Data Warehouse Architecture

The data warehouse structure in this research implements three-tier architecture (Figure 5). The bottom tier includes the interface tools to extract, clean, transform, and import external data (ETL process) (Inmon, 1996). The main purpose is to merge various data from different sources into a unified format and store it in the database. Based on the data content, data marts are constructed in database. Each data mart is confined to specific selected subjects. The middle tier is an OLAP server which processes user's data query (Han, Kamber, & Pei, 2012). The top tier is a front-end client layer, which contains analysis tool such as data mining tools so the user can perform analysis on the data and control the whole process (Han, Kamber, & Pei, 2012).

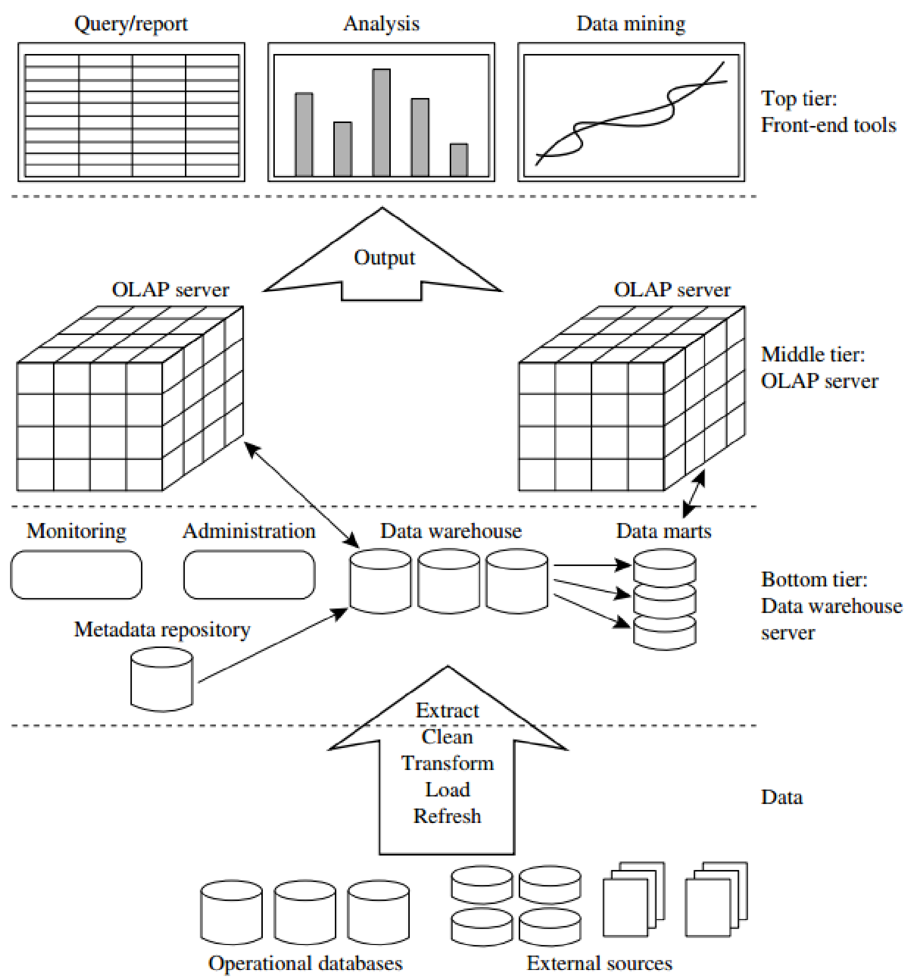


Figure 5: Data warehouse three tier architecture (Han, Kamber, & Pei, 2012)

In this research, the star schema and snowflake schema are used in data warehouse to organize the highway project data (Figure 6). The data in highway project data warehouse is multidimensional. Dimensions are the perspectives to describe a subject. Most highway project data are multidimensional data. In this research, data warehouse uses star schema or snowflake schema to organize multidimensional data. Star schema contains a large central table and a set of smaller attendant tables, one for each dimension (Han, Kamber, & Pei, 2012). The schema graph resembles a starburst, with the dimension tables displayed in a radial pattern around the central table (Shu-Hsien Liao, 2012). The central table is called fact table which contains the names of the facts, measures and keys to each of the related attendant tables. The attendant tables are called dimension tables (Thammasak Rujirayanyonga, 2006). A dimension table further describes the dimensional details of the subject. The snowflake schema is a variant of the star schema. The dimension tables in snowflake schema are further split into additional tables.

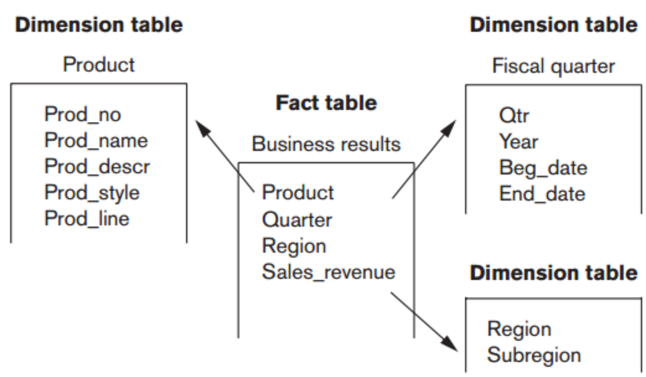


Figure 6: An example of star schema (Thammasak Rujirayanyonga, 2006)

2.2.4 Data Warehouse Application

The architectural engineering and construction (AEC) industry demands more applications of advanced data managing approaches like data warehouse due to the increasing need of data analysis. Construction documents are presented in a variety of ways, such as digital databases, text documents, and CAD drawings, and there is a need to improve organization and access to the data (Simoff & Maher, 1998). Zhu et al. explains that construction documents are typically separate collections of data that users are required to compile in identifying the desired data, which can lead to misunderstandings of construction documents (Zhu, Issa, & Cox, 2011). Lowe (2003) states that engineering designers spend up to 30% of their time searching for design information and often do not find useful information due to access difficulties and time requirements (Lowe, McMahon, & Culley, 2004). Accessing construction document data in an efficient manner is essential in construction management (Caldas & Soibelman, 2003). Significant amounts of data are generally stored in structured data, but the limitation is the ability to analyze the data for useful knowledge (Soibelman & Kim, 2002). Data warehouse has been applied to many areas in the AEC industry to address these issues. The information across several projects can be shared through data warehouse access (Kimball, 2002). For example, the material inventory information for the project in which the database is being accessed is not the only information available, but also the other projects in different locations. The data can be input in one project and accessed from another project. That data is accessible instantly to identify records and make decisions (inmon, William, 1996). Data warehousing is introduced to the construction project management domain in 2002 (K.W. Chau, 2002). In this research, a prototype system of Construction Management Decision Support System (CMDSS) was built to show the qurery function of the data warehouse. Later in 2003, Rujirayanyonga et. al. designed a similar data warehouse system for contractors (Thammasak Rujirayanyonga, 2006). The two systems presented querying and reporting functions of data warehouse while ignoring data mining. Other statistical models were combined with data warehouse to performed a cost estimation analysis in 2007 (S.W. Moona, 2007). Data warehouse was applied to a construction contract administration

(Chong et al., 2013) and it was used as a tool to avoid project disputes by performing case studies (Chong & Phuah, 2013). Qian et al. discussed the architecture design of data warehouse for highway management and on-line analytical processing technology (Qian & Qing, 2009). Ma et al. (2013) identified that the lack of data with tunnel projects could cause safety issues and saw the need to apply a data warehouse to the tunnel construction (Ma, Luo, & Chen, 2013). Data warehouse was also used as a data collection and organization tool for sewer infrastructure management (Park & Kim, 2013). This sort of database information can be applied to the architectural engineering and construction industries. Within these industries there is an abundance of data which is not being used or applied to daily practices (Chong et al 2010). Although data warehouse has been gradually adopted by the industry, few researches have been done to combine data warehouse with in-depth data mining methods for a systematic approach to extract and manage knowledge.

2.2.5 Integration of Data Warehouse and Data Mining

This research integrates data warehouse and data mining in a systematic approach to increase the data mining efficiency and effectiveness in highway projects. The systematic approach has four steps as illustrated in Figure 7. Data preparation is the fundamental step of collecting and storing raw data. Data Warehousing is the step of transforming raw data into forms that specifically appropriate for query and future data mining (Kimball, 1998). It serves as a highway knowledge database platform. Data mining step is the direct extraction of hidden pattern/knowledge from the large amount of data stored in data warehouse. The data warehousing and data mining steps are also optimized with information technology to increase the automation level thus improve the analysis efficiency in highway projects.

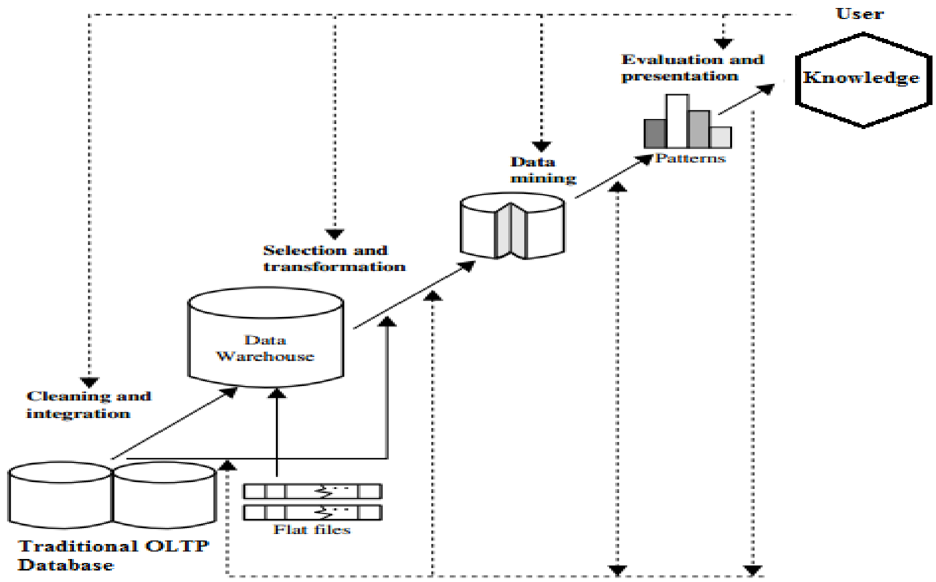


Figure 7: Knowledge discovery approach (Han, Kamber, & Pei, 2012)

2.3 IFC Technology

2.3.1 IFC and interoperability

This research uses Industrial Foundation Class (IFC) technology to integrate interdisciplinary highway project data. The architecture, engineering, and construction (AEC) industry as well as highway industry is experiencing challenges in its capability to manage interdisciplinary data. The project life cycle has planning, designing, building, manufacturing, occupying, maintenance and demolition phases (Pniewski, 2011). Various data are transferred and exchanged by interdisciplinary professionals among different domains. Collaborating teams have the need to efficiently manage and communicate various project data. For example, a mechanical engineer may want to use his software to read and modify an electrical engineer’s data file. However, the mechanical engineering drawing data format may be very different from the electrical engineering drawing data format. Table 4 shows an example of common 2D/3D drawing data formats involved in AEC industry project. One format may not be compatible with another.

Table 4: 2D/3D data formats in construction project (McGraw-Hill Construction, 2012)

Type of Format	Exchange Format
Image Formats	JPG, GIF, TIF, BMP, PNG, RAW, RLE
2D Vector Formats	DXF, DWG, AI, CGM, EMF, IGS, WMF, DGN, PDF, ODF, SVG, SWF
3D Surface and Shape Formats	3DS, WRL, STL, IGS, SAT, DXF, DWG, OBJ, DGN, U3D PDF (3D), PTS, DWF
3D Object Exchange Format	STP, EXP, CIS/2, IFC
XML Schemas	AecXML, Obix, AEX, BCxml, AGCxml
Game File Formats	V3D, X, U, GOF, FACT, COLLADA
GIS formats	SHP, SHX, DBF, TIGER, JSON, GML

In reality, in order to exchange the AEC industry and highway industry project data, each individual software application must develop and implement direct translators back and forth for all other pieces of software (Laakso & Kiviniemi, 2012). More time is spent in data translation instead in the real work of design and construction. Data exchange is very frequent in AEC industry. Almost 80/% AEC organizations are involved in frequent data exchange activities (Figure 8). The ability to effectively exchange interdisciplinary data is called interoperability. Interoperability is defined as: "*the ability of two or more systems or components to exchange information and to use the information that has been exchanged.*" (Laakso & Kiviniemi, 2012). Challenges related to data interoperability have existed in the AEC industry for as long as computers have been involved (Bloor & Owen 1995). Nowadays, data exchanges have become increasingly complex as more and more disciplines are involved in the AEC industry. Lack of data interoperability causes serious problems as shown in Figure 9.

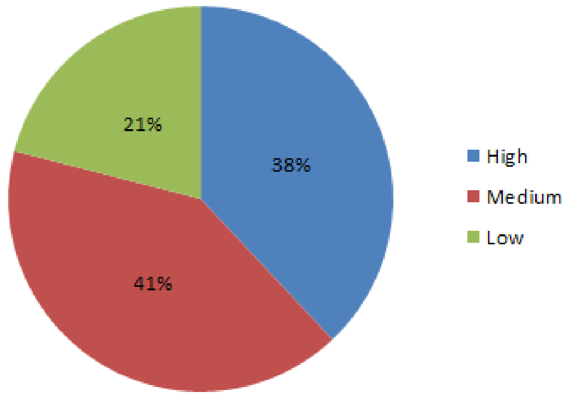


Figure 8: Frequency of data exchange in AEC industry (Autodesk, 2012)

Low level data interoperability leads to huge financial loss in AEC industry. institute of Standards (NIST) estimates that insufficient interoperability among information technology tools costs the US capital facilities industry USD 15.8 billion annually (McGraw-Hill, 2007). Another recent US survey suggested that software non-interoperability costs on average 3.1% of total project budgets (Woldesenbet, 2014).

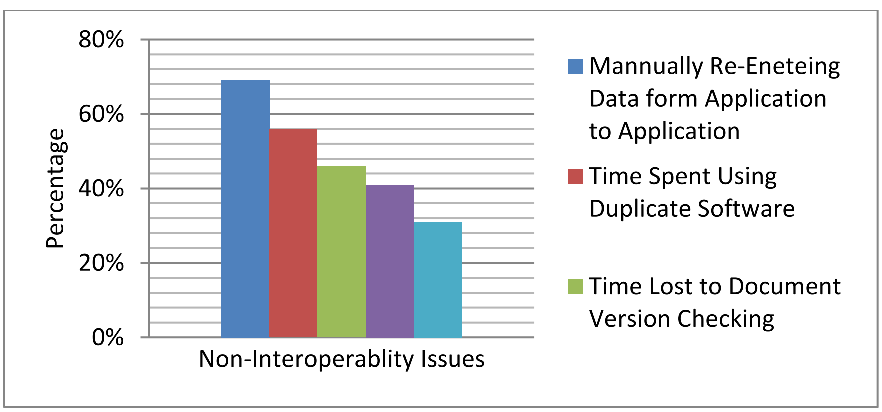


Figure 9: Issues due to lack of data interoperability (Autodesk, 2012)

2.3.2 IFC Concept

This research used IFC to improve the interoperability level in the highway industry. As discussed in previous sections, highway interdisciplinary data needs to be presented in a common interpretable form so that data interoperability can be improved (Autodesk, 2012). The initial goal of Building Information Modeling (BIM) was to address the interoperability issue in the AEC industry by proposing this common form. The preliminary concept of BIM has existed since the 1970s with the fast development of computer technology that makes computerized information system possible (Eastman C. , 1974). Several BIM approaches were proposed but they lack the international standardization. In 1994/95, the first international standard of BIM was released. It is called Industrial Foundation Class (IFC). The goal of IFC is to "*improve the use and exchange of industrial data throughout the whole AEC project lifeycle, globally, across disciplines and technical applications*" (Froese, 1999). This standard has been developed and promoted by buildingSMART. It is already registered by international Organization for Standardization (ISO) and is now an official international Standard ISO 16739:2013. It is becoming the dominant standard for BIM approach worldwide (McGraw-Hill Construction, 2012). The US National Building information Model Standard Project Committee (NBIMS-US) defines BIM as: "*Building information Modeling (BIM) is a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life-cycle; defined as existing from earliest conception to demolition."* According to this definition, BIM is an approach that develops and uses one single data format to store all aspects and all levels of project information (Shannon Barnes, 2009). Without BIM, each individual software application must translate data formats back and forth to communicate all other pieces of software. Many data formats and standards are involved. With BIM, all applications share that same standard (IFC), there is no complicated transformation needed

(Laakso & Kiviniemi, 2012). A comparison of these two conceptual scenarios is shown in Figure 10.

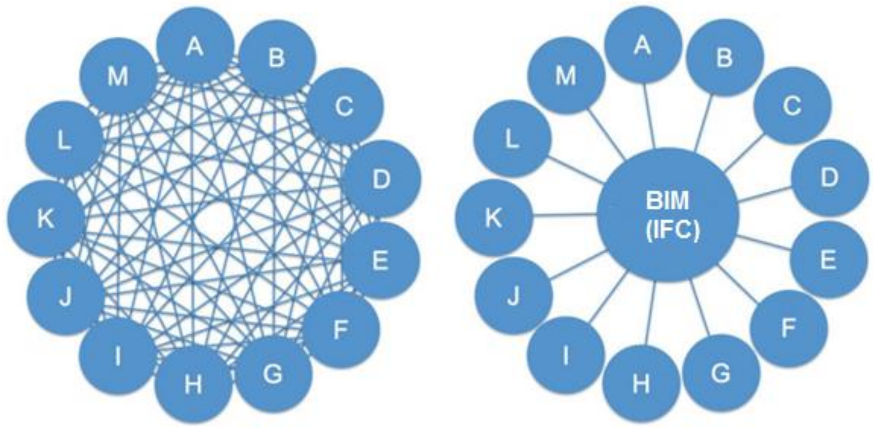


Figure 10: Comparison of interoperability (Laakso & Kiviniemi, 2012)

2.3.3 IFC International Standard

This research will use a series of international standards to expand IFC to highway industry. Most key concepts of IFC/BIM are already formalized in the ISO international Standard. The highly standardization of IFC/BIM approach greatly improves the data interoperability. Following the same international stand, BIM can be used anytime and anywhere. The main concepts and standards are listed as follows:

- Data modelling: A data model organizes data elements and standardizes how the data elements relate to one another. Since data elements document real objects and the relationships between them, the data model represents reality. BIM is a systematic approach of developing and utilizing data models for objects and relationships in AEC industry. The process of creating data models is called data modeling. The data modelling process should follow IFC specification, EPRCESS language style and STEP format,

- IFC specification: It is standardized as ISO 16739:2013. IFC standard defines the common data models for AEC industry. IFC has been "*directed to address all building information, over the whole building lifecycle, from feasibility and planning, through design (including analysis and simulation), construction, to occupancy and operation*" (Eastman C. , 2006). IFC adopts an object-oriented data modeling approach including multiple inheritance concepts (Eastman C. , 2006). IFC defines all real world objects and relationships in AEC industry. The system architecture of IFC is illustrated in Figure 11. Each shape represents an entity. The bottom level includes twenty six base entities, defining elements like Geometry, topology, Materials, Measurements, Actors, Roles, Presentations and Properties (BuildingSmart, 2013). These entities are further used to define shared objects including shared building elements, shared component elements, shared building elements, shared management elements, and shared facility elements (BuildingSmart, 2013). The top level uses all the entities in middle and bottom levels to describe specific discipline information in various domains such as HVAC, architecture, plumbing domain, etc (BuildingSmart, 2013). Now the coverage of IFC is limited to building industry (McGraw-Hill Construction, 2012). infrastructure and highway industry are not covered.
- EXPRESS language: It is standardized as ISO 10303-11. Express is the human readable language to describe the structure of data models. IFC uses EXPRESS language to describe AEC industry data models.
- STEP format: It is standardized as ISO 10303. STEP is a computer readable language to describe IFC. It is the implementation method for IFC digital file.

In summary, BIM is standardized in IFC specification, described in EXPRESS language for human, implemented in STEP format for computer. This research follows this scientific approach to expand BIM to highway industry.

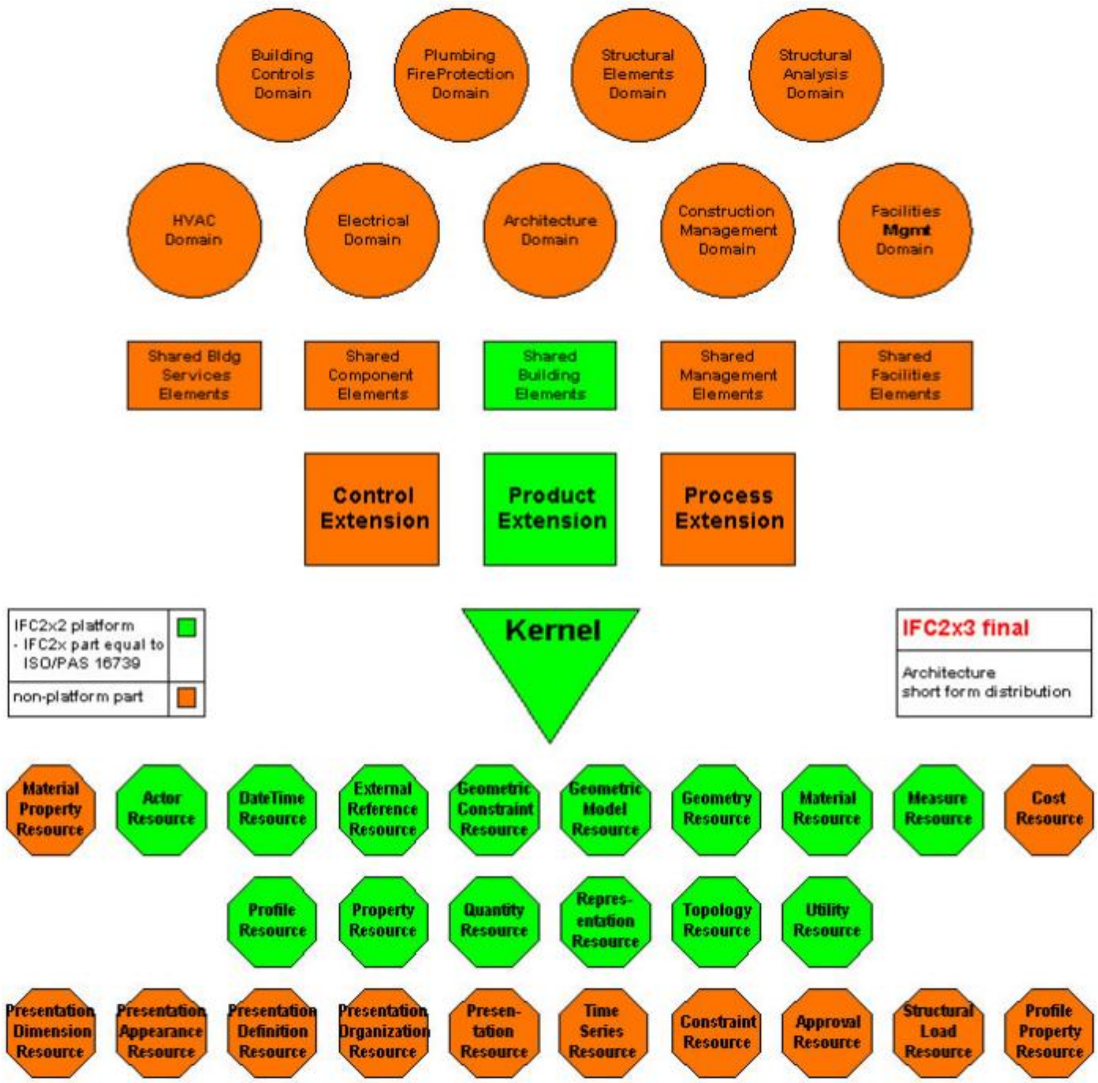


Figure 11: IFC system architecture (Eastman C. , 2006)

2.3.4 IFC Benefits

There are lots of benefits associated with IFC/BIM approach. Long-term benefits include: Reduce project duration, increased profits, reduced construction cost and fewer claims. Short-term benefits include: reduced documents errors and omissions, reduced rework, market new business, offer new services and reduced cycle time of specific workflows and high Return on investment (ROI) value (McGraw-Hill Construction, 2012). NBIMS-US also lists other benefits: improving collaboration between information suppliers and users; providing standard way of storing information; building data collection and sustainment of information into the business processes; providing more accurate complete information to support advanced data analysis (NBIMS, 2015).

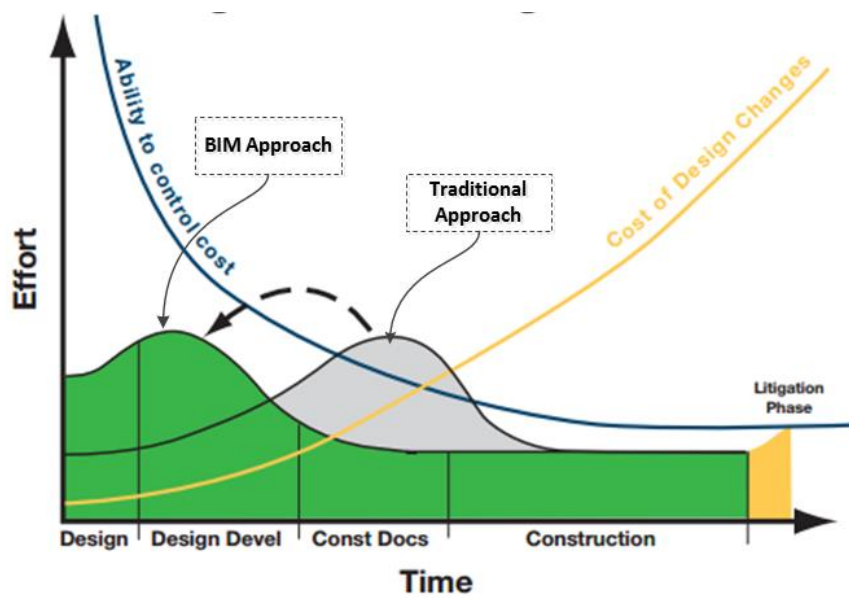


Figure 12: Improvement of cost control ability (McGraw-Hill, 2007)

Figure 12 shows how IFC/BIM approach improves the ablity of cost control. The ability to effectively control costs decreases as the AEC project progresses into later stages. So it is critical to encourage data interoperablity and team collaboration at the beginning of the project and to shift analysis and decision-making to the planning and design phase (McGraw-Hill, 2007). Compared to trandition approach, IFC/BIM provides opportunities to explore design alternatives and to analyze whole life-cycle cost, which significantly reduces the potential project risks of duration and budget overruns.

2.3.5 IFC Application

More companies have adopted IFC/BIM systems into their daily practices as IFC/BIM has become a widely used standard within the building industry (Sacks & Pikas, 2013) . BIM has been used to improve general design/construction practices such as energy analysis, quantity take-off estimations, cost estimations, and construction planning (Bynum, Issa, & Olbina, 2013) . IFC/BIM has also been applied to building energy simulations to provide faster and more accurate energy estimates (Kim & Anderson, 2013).

Table 5: Applications of IFC/BIM (Kreider, Messner, & Dubler, 2010)

BIM USE	Frequency	Rank	Benefit	Rank
	%	1 to 25	-2 to +2	1 to 25
3D Coordination	60%	1	1.60	1
Design Reviews	54%	2	1.37	2
Design Authoring	42%	3	1.03	7
Construction System Design	37%	4	1.09	6
Existing Conditions Modeling	35%	5	1.16	3
3D Control and Planning	34%	6	1.10	5
Programming	31%	7	0.97	9
Phase Planning (4D Modeling)	30%	8	1.15	4
Record Modeling	28%	9	0.89	14
Site Utilization Planning	28%	10	0.99	8
Site Analysis	28%	11	0.85	17
Structural Analysis	27%	12	0.92	13
Energy Analysis	25%	13	0.92	11
Cost Estimation	25%	14	0.92	12
Sustainability LEED Evaluation	23%	15	0.93	10
Building System Analysis	22%	16	0.86	16
Space Management / Tracking	21%	17	0.78	18
Mechanical Analysis	21%	18	0.67	21
Code Validation	19%	19	0.77	19
Lighting Analysis	17%	20	0.73	20
Other Eng. Analysis	15%	21	0.59	22
Digital Fabrication	14%	22	0.89	15
Asset Management	10%	23	0.47	23
Building Maint. Scheduling	5%	24	0.42	24
Disaster Planning	4%	25	0.26	25

Through the use of IFC/BIM, multiple simulations could be run to identify the optimum energy output for a building in a more rapid method than typical energy analysis simulations. Safety aspects of building construction have also been analyzed using IFC/BIM. Identifying safety hazards in temporary structures for construction can be tedious and time consuming, but IFC/BIM can be applied to this process to more easily and rapidly identify safety risks during the planning phase (Kim H. A., 2011). In December of 2009, a study was performed to help determine the situation of BIM application. 3D coordination and Design Reviews are both the most frequently used and most beneficial according to the survey results (Kreider, Messner, & Dubler, 2010). The detailed information is shown in Table 5.

IFC/BIM has been widely adopted in the AEC industry. The percentage of companies using IFC/BIM has jumped dramatically from 28% in 2007, to 49% in 2009, and to 71% in 2012 in North America (Figure 13).

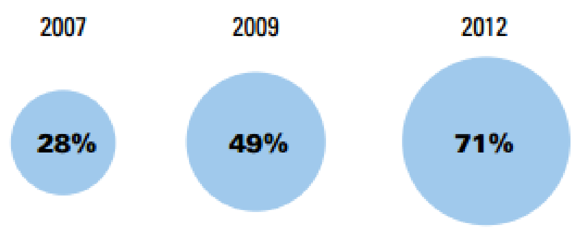
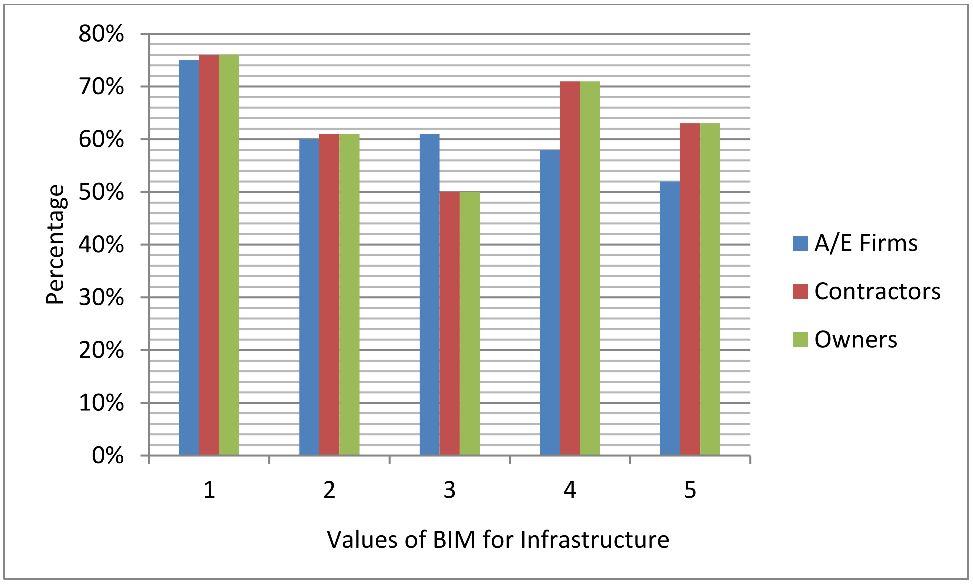


Figure 13: Levels of IFC/BIM adoption in North America (McGraw-Hill, 2012)

2.3.6 IFC in Highway Industry

Although IFC/BIM is already widely recognized in the building industry, it has not been fully expanded to the infrastructure industry as well as highway industry (BuildingSmart, 2013). IFC is limited to building industry. infrastructure industry and highway industry are not fully covered. A lot of benefits for building industry are also true for infrastructure and highway industry. The advantages of BIM approach encourage wider adoption and accelerate implementation of IFC standard in infrastructure industry and highway industry (Autodesk, 2012). A study conducted by McGraw-Hill Construction lists the major potential values that BIM can provide for infrastructure industry (McGraw-Hill Construction, 2012). In the study, all stakeholders believe the top benefit using IFC/BIM for infrastructure is the reduced conflicts and changes during construction. This result corresponds to current benefit of using IFC/BIM for building industry. Reducing conflicts and changes can improve the infrastructure schedule and productivity and lower down the risk of cost and schedule overruns. Other major values of IFC/BIM for infrastructure are shown in Figure 14.



NO.	Value
1	Reduced Conflicts & Change during Construction
2	Lower Risk & Better Predictability of Outcomes
3	Higher Quality, Better-Preforming Completed infrastructure
4	Improved Productivity
5	Improved Review & Approval Cycles
6	Reduced total Project Cost

Figure 14: Potential values of IFC/BIM for infrastructure (McGraw-Hill Construction, 2012)

This research uses IFC standard and BIM approach to integrate data in highway projects because highway industry shares the same interdisciplinary nature with building industry. Specifically, road, bridge and tunnel are closely connected to natural environment which has complex properties and is changing all the time. The complexity is increasing to store those properties and adapt the changes. There is a need to integrate natural environment data with traditional structure data. For example, geology data is very important in highway projects. IFC/BIM can integrate both road structure data and topo surface data in a single IFC file. So structure engineer and geologist can work together on the same model to estimate earthwork, which reduces potential conflicts and changes. Highway project data can be effectively communicated and utilized with IFC/BIM.

In addition, IFC/BIM has the potential to facilitate knowledge extraction in highway projects. IFC/BIM ensures if all interdisciplinary data follow the same IFC standard. By using IFC, the mapping relationship between input BIM data and data warehouse only needs to be established once. Based on the mapping relationship, the IFC data can be translated to populate the data warehouse structure. The translation process can be highly automated by computer programming. The manual effort can be minimized. More time can be spent in the real work of meaningful knowledge extraction rather than data preparation. Thus the efficiency and effectiveness of knowledge extraction in highway projects can be improved.

2.4 Research Gap

Although IFC have been widely adopted, little research has been done to apply IFC to facilitate data mining.

Data mining highly depends on well-constructed feature/target data. However, currently the generation process of feature/target data in data warehouse is still manual. Research points out that preparing feature/target data usually consumes almost 60% of the total effort invested in the entire knowledge extraction process (Figure 15). Data preparation step is considered as one of the most important, time-consuming, and hard parts of the whole knowledge discovery process (Lucio Soibelman, 2002). The data preparation process in highway project data warehouse is especially complex. The highway project data is often captured and persevered in ad-hoc manner. Furthermore, current data formats save data as nominal strings and numeric numbers instead of an object. The data formats cannot be easily interpreted by computer. The process of feature/target generate can be highly automated.

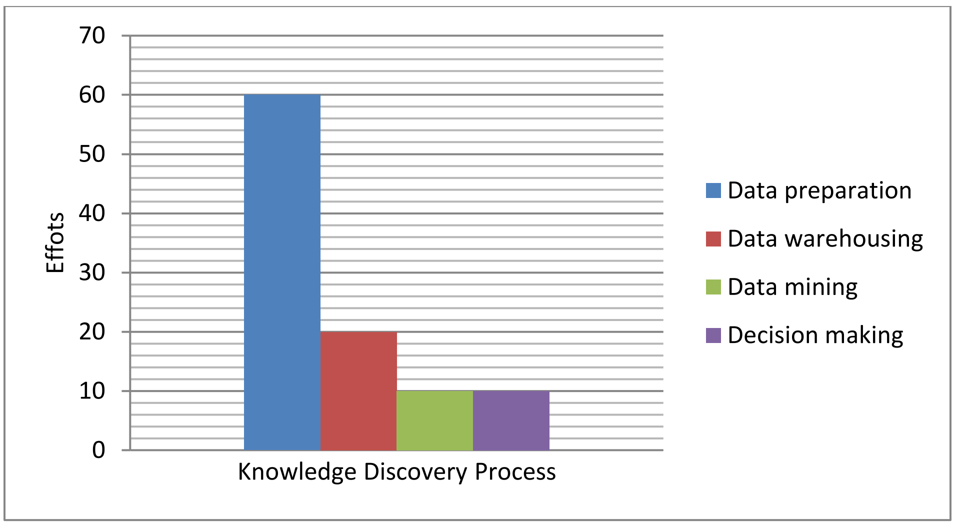


Figure 15: Efforts spent in knowledge discovery process (Soibelman & Kim, 2002)

IFC data schema is machine interpretable. It defines everything in an object-oriented approach, which assigns the objects attributes and establish relationships between them. IFC also supports inheritance. Computer recognizes IFC entities as an object instead of a string. It is suitable for automated complex querying and reasoning, thus facilitate automated feature/target data generation.

Moreover, IFC has the ability to describe any tangible objects and non-tangible relationships, which implicit it can integrate any interdisciplinary data among different domains. By considering more interdisciplinary data, more meaningful results can be found and studied.

This research aims to fill the gap between IFC and data mining by building a data warehouse to combine all the state-of-art technologies. IFC has the potential to automate the process of feature/target data generation and include more interdisciplinary data into data mining analysis, which further strengthen the depth and breadth of highway construction data analysis.

CHAPTER 3: RESEARCH DESIGN AND SYSTEM IMPLEMENTATION

Details of research design and methodology are presented in this section. . Section 3.1 provides the design process for Objective 1, which prepares IFC data model for highway project. Once the highway IFC data model is designed, the focus shifts to Objective 2, as outlined in section 3.2, which designs a star schema data warehouse system. In Section 3.3, the data mining methods used in this research are specified for Objective 3. In section 3.4, the implementation of prototype system is presented.

3.1 Highway IFC Data Model Design

A data model is a representation of data structures. It organizes data elements and standardizes the relationship between them. The data structures include three components: the data objects, the relationships between data objects, and the rules which govern operations on the objects (Liberty univ. Inc, 2015). There are three steps to create a highway IFC data model in this research. Table 6 shows the input information and corresponding deliverables for each step. The terms "conceptual", "logical", and "physical" differentiates levels of abstraction versus detail in the design step. A conceptual data model is summary-level data model identifying what objects need to be included in the data model. A logical data model is a fully-attributed data model. A physical model specifies how the data model is persisted in the physical computer environment. Several international standards are adopted to highly standardize the generated highway data model.

Table 6: Data model design steps

input information	Design Step	Deliverables
• Project nature	Step 1 Conceptual data model design	• Domains • Entities • Relationships
• Design requirement	Step 2 Logical data model design	• Data types • Entity attributes • Super/Sub types • Constrains
• Technical environment • Performance consideration	Step 3 Logical data model design	• Physical file

3.1.1 Conceptual Data Model

This step establishes the scope of work for a highway IFC data model design. It specifies the spatial elements that are expressed in the design and contains the least granular detail.

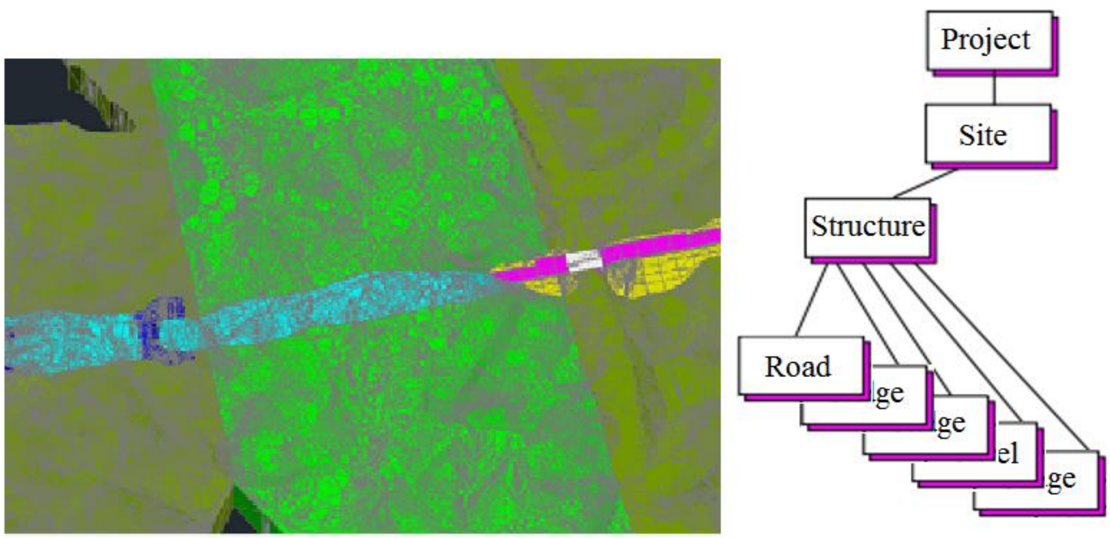


Figure 16: Project hierarchy

Spatial elements are seen as the primary structures for highway projects and required for the data exchange. These different entities are contained by each other such as they provide a clear hierarchical structure for the highway project. The figure shows such a structure. The four entities are Project, Site, Structure, and Structure Elements (Road, Bridge, Tunnel, etc.) are used to represent the levels of the spatial structure. These are the basic mandatory levels for the exchange of complex highway project data.

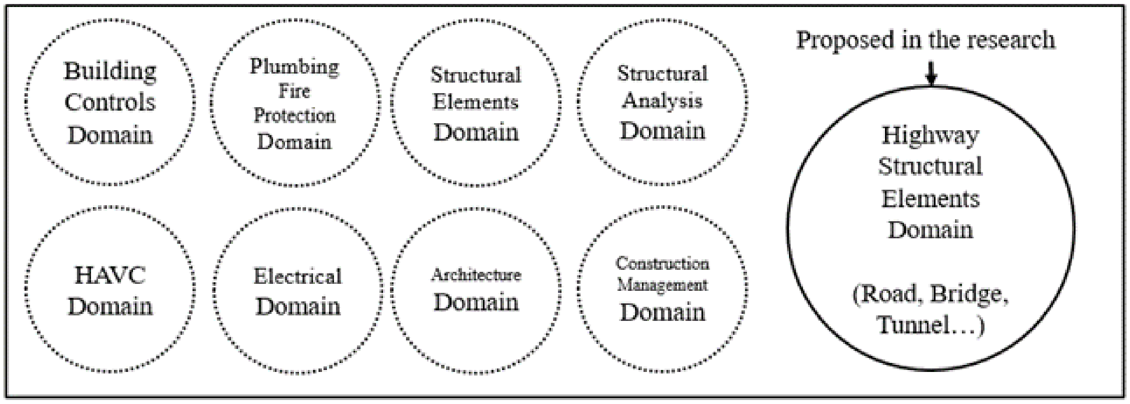
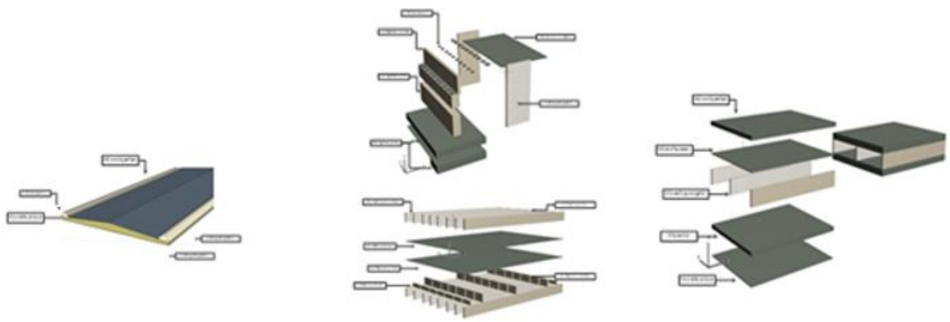


Figure 17: Current and proposed IFC domains

The entities that describe the data and the relationships between those entities are the only information shown through the conceptual data model. Figure 17 shows the current and proposed conceptual IFC domains in this research.

In a highway project domain, spatial structure elements (construction structures) are the most important elements. Specifically, in IFC data schema, a spatial structure element is the generalization of all spatial elements that might be used to define a spatial structure (BuildingSmart 2014). That spatial structure is often used to provide a project structure to organize a construction project (BuildingSmart 2014).

For the highway spatial elements, three kinds of structures are described by IFC entities in this research: road, bridge and tunnel. Figure 18 shows the list of construction components for each highway structure. Eleven road IFC entities, thirteen bridge IFC entities and five tunnel IFC entities are especially designed for this research. The proposed highway IFC data model together with existing BIM data model are combined to intelligently describe interdisciplinary whole life cycle highway information.



Road	Bridge	Tunnel
IfcRoad	IfcBridge	IfcTunnel
IfcRoadSurfaceCourse	IfcBridgeConnectingSlab	IfcTunnelUpperSlab
IfcSolidLine	IfcBridgeSupport	IfcTunnelPavement
IfcBrokenLine	IfcBridgeWallParapet	IfcTunnelSupportingWall
IfcRoadBase	IfcBridgeWallFrame	IfcTunnelSlab
IfcRoadBaseCourse	IfcBridgeWingWall	
IfcRoadSubbase	IfcBridgeProtectionWall	
IfcRoadBed	IfcBridgeRoadSeparator	
IfcRoadEmbankment	IfcBridgePavement	
IfcRoadFoundation	IfcBridgeUpperPlate	
IfcRoadCurb	IfcBridgeSteelBeam	
	IfcBridgeBracingBeam	
	IfcBridgeFoundation	

Figure 18: List of proposed highway IFC entities

The highest-level relationships between the current IFC entities and these proposed highway IFC entities are identified. The proposed highway conceptual model includes highway structural element IFC entities and the relationships between those entities. Each newly designed entity needs to locate its place in the whole IFC hierarchy structure. for example, IfcRoad entity contains all the information of the road structure in highway project. The relationship between existing entities and newly designed IfcRoad entity is illustrated in Figure 19. The entities in IFC hierarchy are nested within a deep sub entity tree. The IfcRoad has a trace down the path: IfcRoot -> IfcObjectDefinition -> IfcProduct

-> IfcBuildingElementProxy-> IfcRoad. in this tree hierarchical structure, sub entity inherits all the properties from its super entity. IfcRoad inherits all the attributes and relationship from its supertype entity IfcBuildingElementProxy.

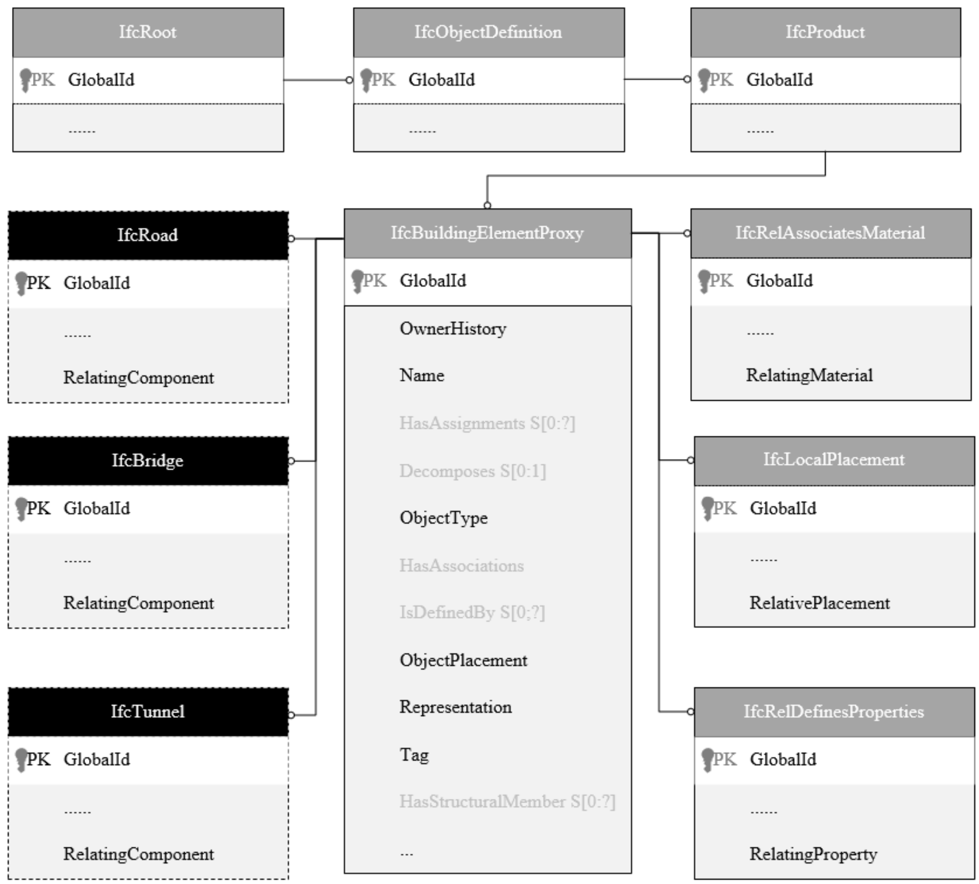


Figure 19: Relationship between current and proposed entities

3.1.2 Logical Data Model

A logical data model describes all the information of the data, without regard to the way they are physically written in a file. Features of a logical data model include the relationships among the entities, all attributes for each entity, a primary attribute and foreign attributes for each entity.

EXPRESS language is used to describe each newly designed entity's attributes and relationship. Express is a data modeling language which is standardized as international standard ISO 10303-11. According to this standard, there are four main types of elements that need to be considered for entity design: entity attributes, super types and subtypes, and algorithmic constraints (ISO, 2004). These elements can be represented in textual form and graphical form with EXPRESS language. Table 7 shows the main entity elements in EXPRESS graphical form. Each entity has attributes. Some attributes are mandatory while others are not. This relationship is defined by attribute/optional attribute elements. Each attribute is assigned with a simple or complex data type. Each entity has its own supertype and subtype. This relationship is defined by super type/subtype elements.

Table 7: Express symbols for entity elements (ISO, 2004)

Entity Element	EXPRESS Symbol
Simple Data Type	NUMBER, INTEGER, REAL, BOOLEAN, LOGICAL, STRING, BINARY
Complex Data Type	Entity_name, Enumerated_type_name, Define_type_name, Selectded_type_name
Super type/Subtype	Supertype_entity_name —Relation_type_name— Subtype_entity_name
Attribute	From_entity_name —Relation_type_name— attribute_name
Optional Attribute	From_entity_name - - Relation_type_name - - attribute_name

Take the IfcRoadbed entity for example (Figure 20), there are multiple attributes associated with it. Some of them are must-have attributes like GlobalID while others are optional like Tag. Each attribute has a type. for instance, the data type of value for GlobalID must be string. IfcRoadbed also has its own supertypes and subtypes. All proposed IFC entities are designed through this approach.

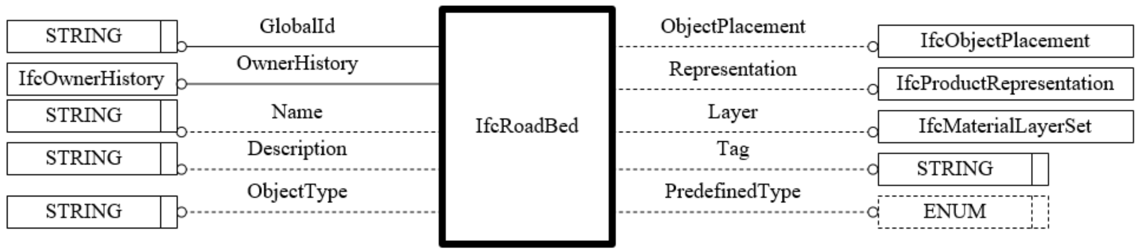


Figure 20: IfcRoadBed entity attributes and inheritance relationships

This research designs 29 highway structure elements IFC entities which are delivered in Express graphical format (Figure 21). 341 attributes are specified and 398 relationships are established for them. The details of proposed IFC entities is presented in Appendix A.

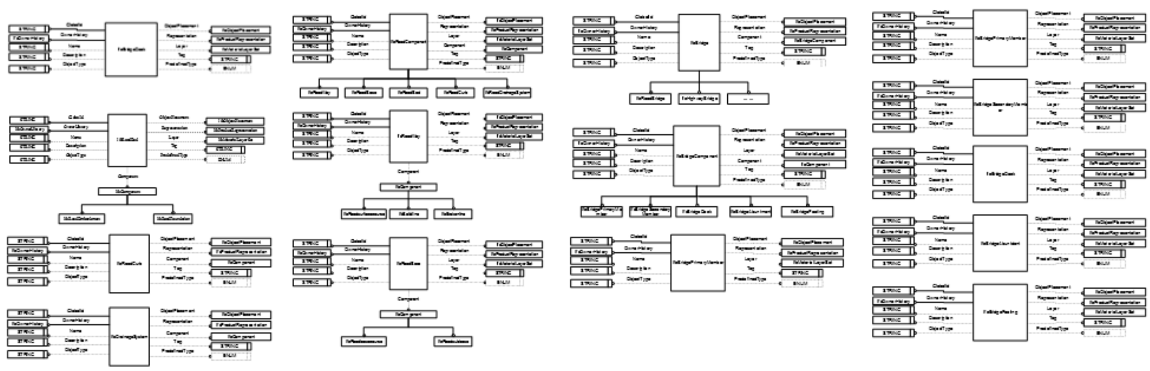


Figure 21: Screenshot of designed IFC entities

3.1.3 Physical Data Model

The designed entities are written in the format of STEP. STEP represents “Standard for the Exchange of Product model data” ISO 10303 international standard. Express language is for humans to read while STEP format is a computer readable language. The STEP format is especially optimized for computer processing. In this format, IFC entities are written line by line in a file. Each line represents an entity including ID, entity name, and attribute value information. The generated physical file has “ifc” as its extension name. It is called IFC file. The file is stored on the computer hard disk. Figure 22 shows an example of the mapping relationship between construction components “IfcBridgeWingWall”, its IFC entity and corresponding physical IFC file in STEP format.

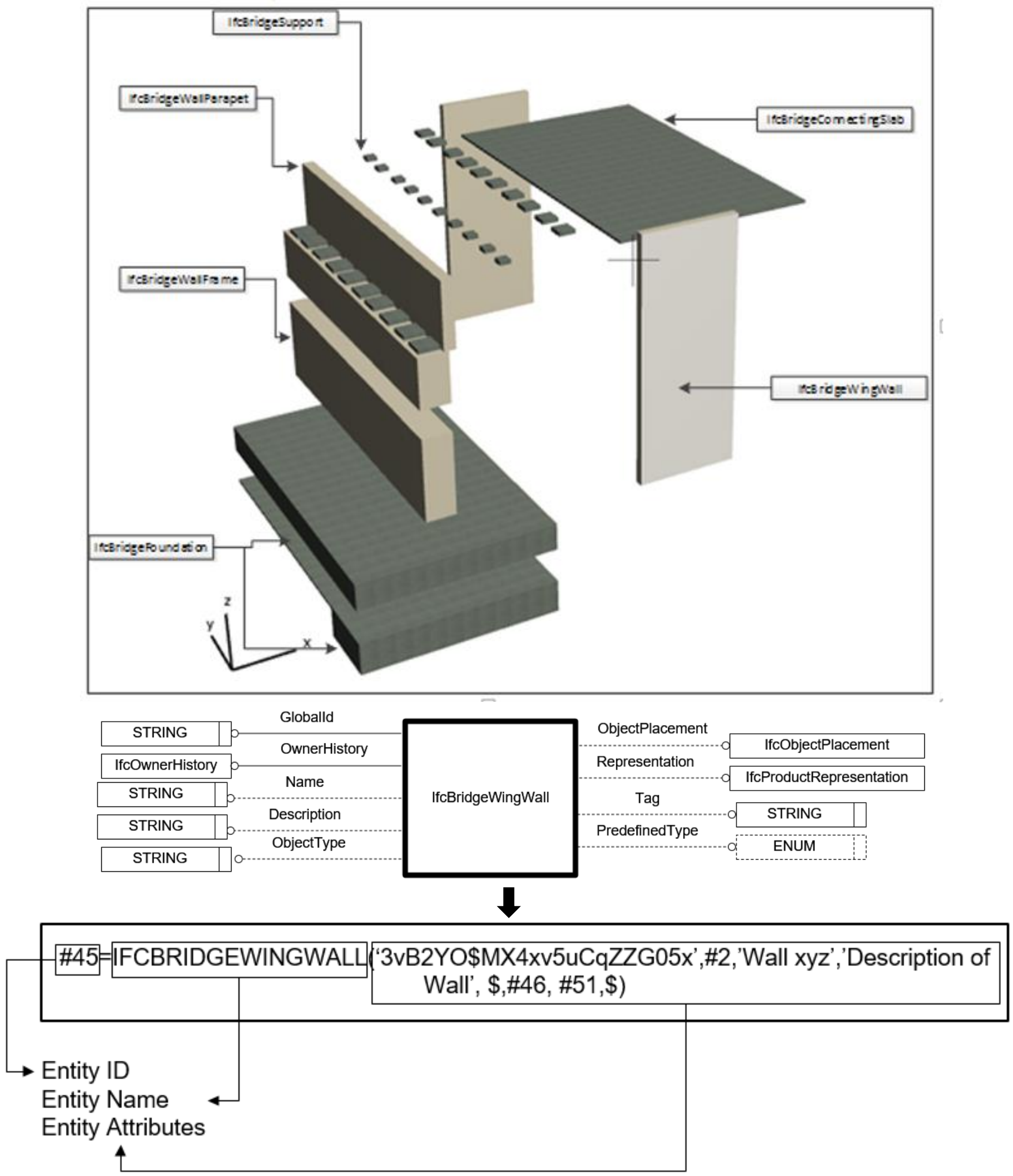


Figure 22: Physical IFC file for IfcBridgeWingWall

The IFC entity IfcRelAggregates is used to link the instances of IfcProject, IfcSite, IfcSpatialElement, IfcBuildingElementProxy and establishes an hierarchical structure.

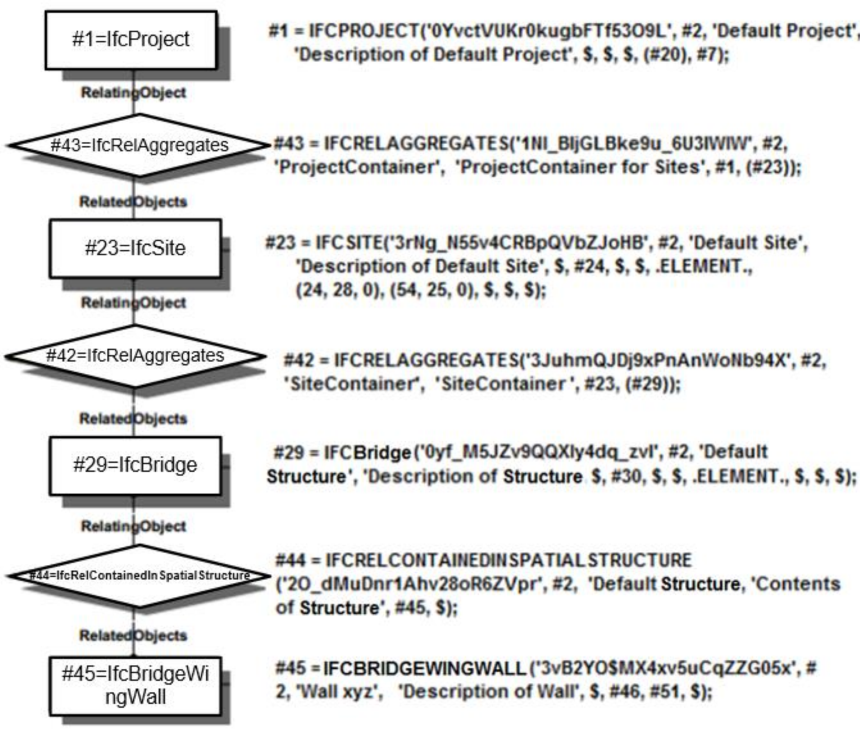


Figure 23: Hierarchy structure of IFC file

Each instance of IfcProject, IfcSite, IfcSpatialElement, IfcBuildingElementProxy is connected to other instances of the spatial structure by an instance of IfcRelAggregates, where the single relating element points to the element at the higher level and the 1 to many related elements point to the elements at the lower level of the hierarchy.

The Figure 23 shows the use of the IfcRelAggregates to define a spatial structure of a highway project having a single site with one bridge. IfcRelAggregates creates the hierarchical project structure: project -->site --> bridge --> bridge element (bridgewillwall). IfcRelContainedInSpatialStructure creates the hierarchical assignment of construction elements to the relevant spatial element. A path can be traced from IfcProject, IfcSite, IfcBridge to IfcBridgeWillWall, which stores bridge wingwall information.

3.2 Data Warehouse Design

The data warehouse is desgined to store, consolidate and manage data in multidimensional space (Figure 24). The physical IFC file is imported to data warehouse and then parsed by the IFC parser engine. IFC entites are extracted from physical IFC file and stored in respective IFC tables in the data warehouse. Feature data and target data are further generated from IFC tables. Dimension tables contain feature data while fact table contains target data. Data mining methods can be applied to feature data and target data to make prediction or find interesting patterns.

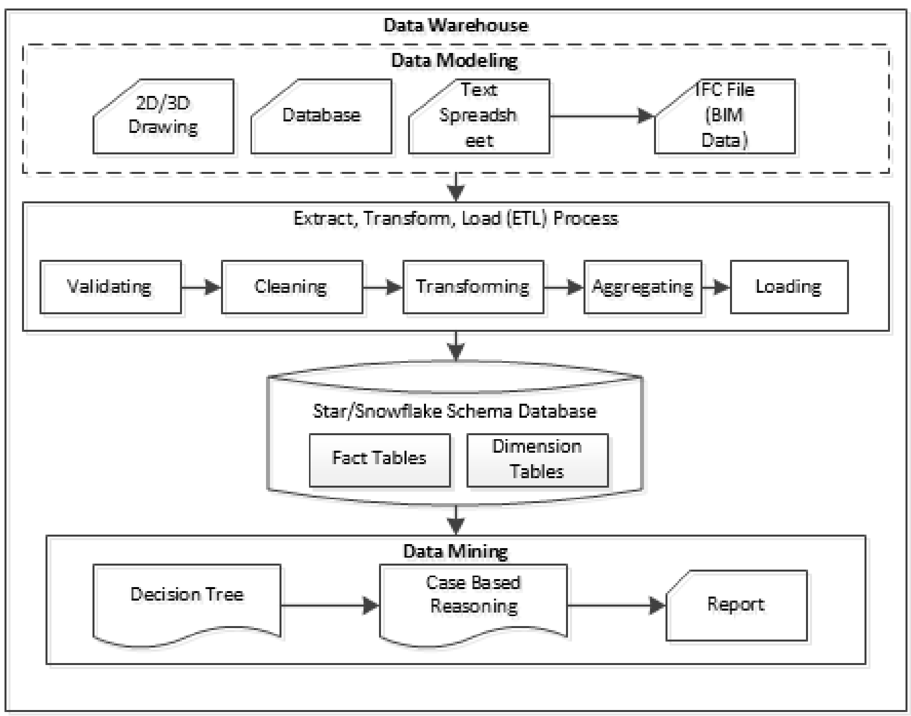


Figure 24: Data warehouse workflow

3.2.1 Feature Generation

Feature generation is the process of transforming raw data into features that represent the underlying nature of the studied subject, resulting in improved model perfomance on new data. In this research, raw IFC data can not be directly mined. In order to find interesting meaningful highway contruction delay patterns, abstract features need to be generated from the raw data.

There are sixteen feature attirubes and one target attribute defined in this research. Sixteen feature attributes include: season, ground condition, permit approval, construction type, road, bridge, tunnel, free float, milestone, incomplete site survey, incomplete drawing, design modification, change order, shortage of equipment, quantity and cost. One target attriubte is the construction activity delay status. The relationship between Physical IFC file and feature/target data is illustrated in Figure 25.

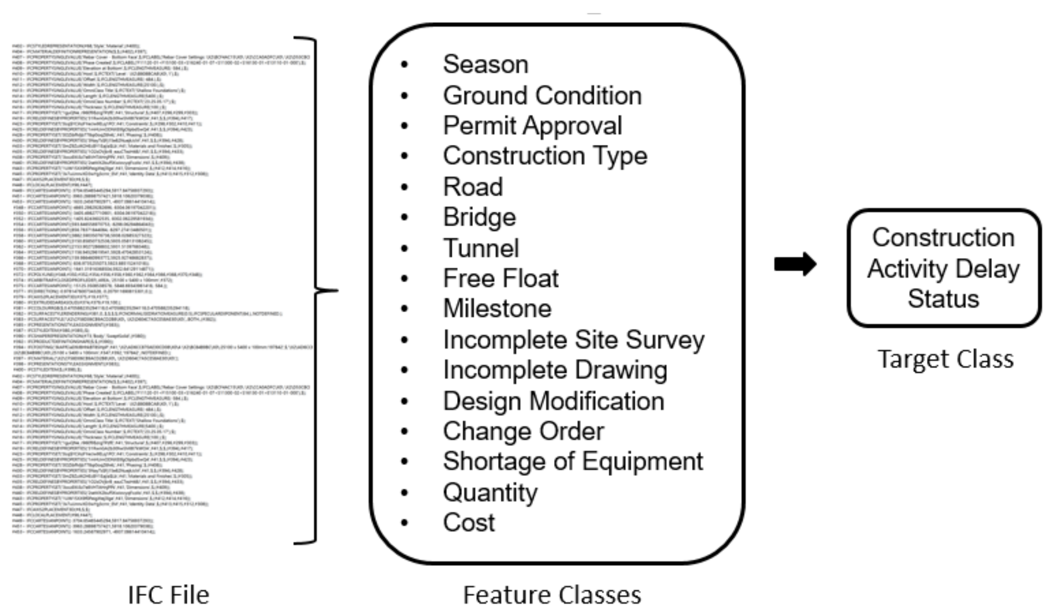


Figure 25: IFC file, feature data and target data

The sixteen feature attirubes are chosen according to previous studies. For example, the reason why “Change Order” is selected is that in reality highway project contractors usually frequently requested change orders for time extensions and cost overruns.

Table 8 lists the possbile values for each the feature attribues. Most of the feature attributes are nomial. Among them, “Road”, “Bridge” and “Tunnel” feature attributes represent if a construction activity is related to respective structure. “Cost” is the only one numeric feature attributes. The values of feature attribues are obtained by performing complex querys and calculations on the IFC entity tables.

Table 8: Possible values of feature attributes

#	Name	Data Type	Possible Value
1	Season	Nominal	Spring, Summer, Fall, Winter
2	Ground Condition	Nominal	Rocky, Not-Rocky
3	Permit Approval	Nominal	Yes, No, Unknown
4	Construction Type	Nominal	Construction, Destruction
5	Road	Nominal	Yes, No, Unknown
6	Bridge	Nominal	Yes, No, Unknown
7	Tunnel	Nominal	Yes, No, Unknown
8	Free Float	Nominal	Yes, No, Unknown
9	Milestone	Nominal	Yes, No, Unknown
10	Incomplete Site Survey	Nominal	Yes, No, Unknown
11	Incomplete Drawing	Nominal	Yes, No, Unknown
12	Design Modification	Nominal	Yes, No, Unknown
13	Change Order	Nominal	Yes, No, Unknown
14	Shortage of Equipment	Nominal	Yes, No, Unknown
15	Quantity	Nominal	Small, Large
16	Cost	Numeric	Real

For example, the activity delay status value can be obtained by comparing, aggragating and decomposing schedule information and actual activity information. There are many aspects of information associated with these two kinds of ifc entities. Among them, only “finish date and time” need to be queried, calculated and compared (Figure 26).

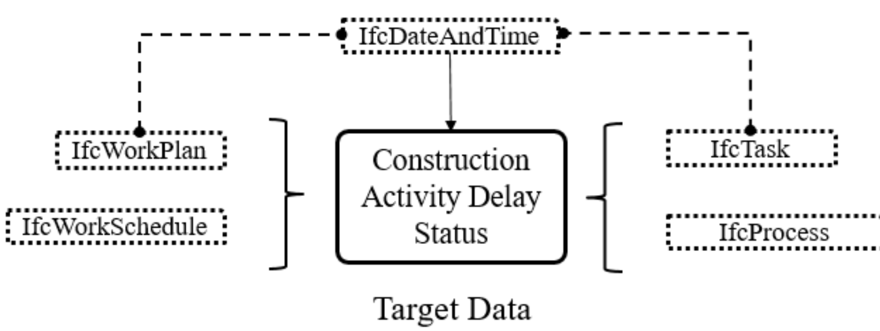


Figure 26: Query for construction delay status

Similarly, other feature class data can also be obtained by retrieving and processing corresponding ifs entities (Figure 27). For instance, if the any additional information is needed by the construction team during the activity. The construction team will send certain requests or orders the design team or other teams. This situation can be captured by IfcEvent and IfcProjectOrder entities. The IfcAssociateDocument can specify what kind of document is needed or changed. By setting up some simple rules, a program can automatically retrieve, search, analyze IFC entities based on the user's business needs and then prepare the corresponding feature attribute data.

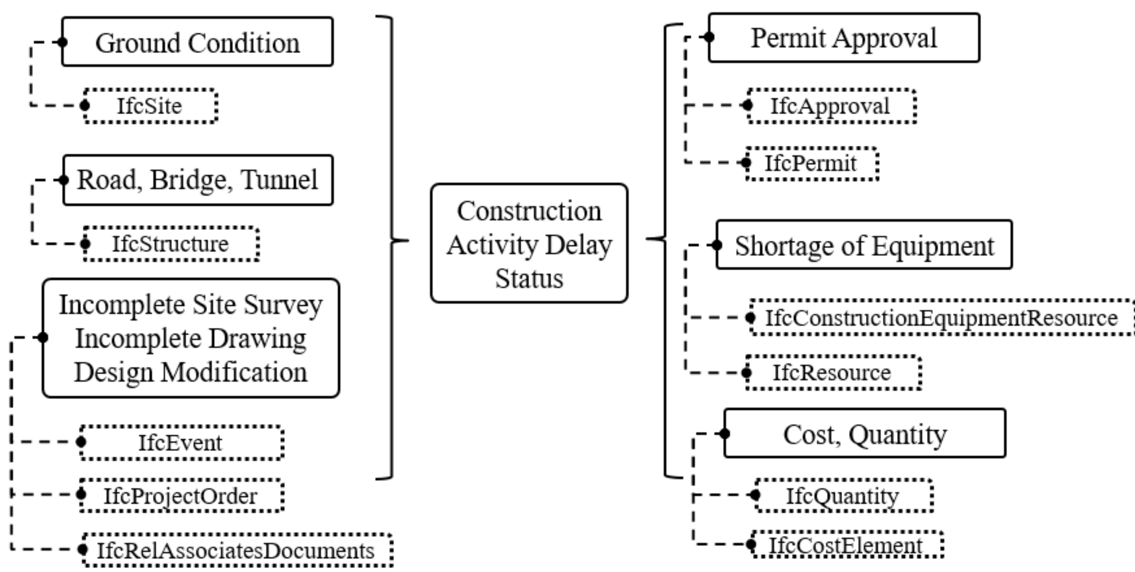


Figure 27: Feature attributes and corresponding IFC entities

3.2.2 Star Schema

The most common modeling paradigm in data warehouse is star schema. It is used to contain the multidimensional feature data and target data. The star schema consists two types of tables: fact table and dimension tables. Fact table specifies the relationship between feature data and target data. Dimension tables store the values of all the features.

In order to perform complex queries, the mapping relationship is established (Figure 28). Physical IFC file is imported to data warehouse to populate IFC entity tables based on the mapping relationship. Then the query process is carried out in data warehouse database. This database is especially designed to store retrieve, query and manage all these data. Once the query process is finished, dimension tables and fact tables are filled with data. These feature attribute data and target attribute data are in the form appropriate for further data mining.

IFC data specifically support ad-hoc querying. With IFC, star schema deals with raw data at the object level. Once mapping relationship is set up, star schema can automatically retrieve and process corresponding IFC data to generate feature/target data. User can quickly create a new knowledge discovery framework within data warehouse as the business need arises.

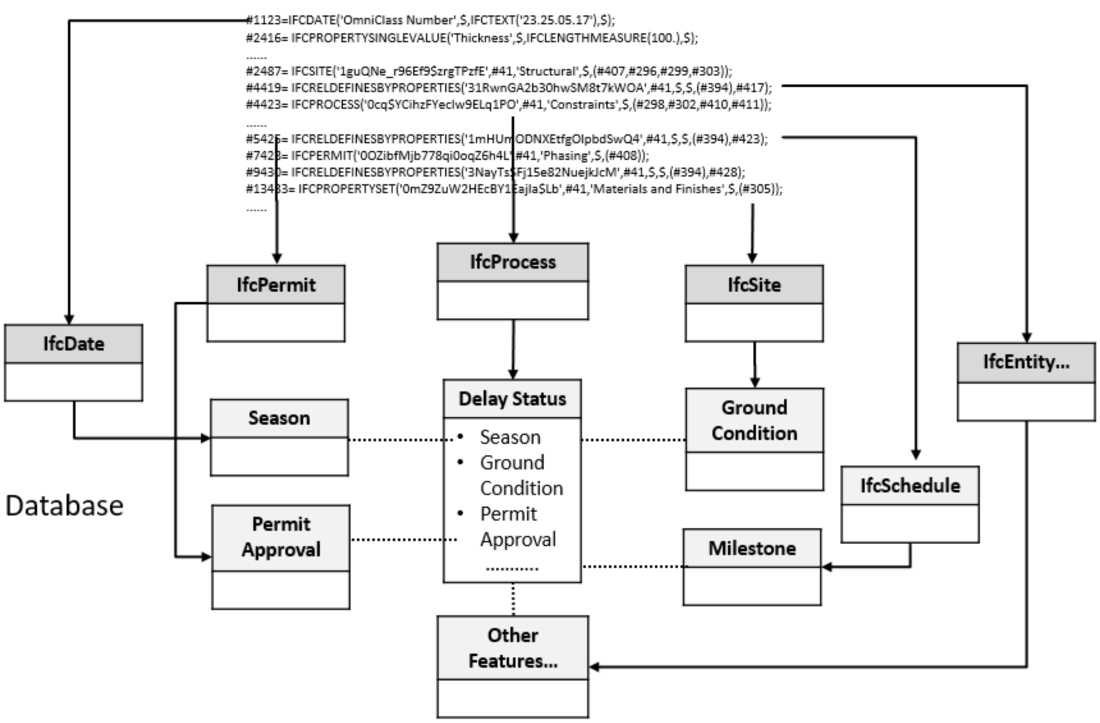


Figure 28: Example of mapping relationship between star schema and IFC file.

3.3 Data Mining Method Design

The purpose of this design is to perform a data mining analysis in the developed prototype system to find interesting patterns and extract useful knowledge. This reseach focuses on decesion tree and Case Based Reasoning (CBR) techniques. These techqunies can explore in-depth relathionship between attributes, but require less domain knowledge or parameter setting, therefore are appropriate for interdisciplinary exploratory knowledge discovery (Han, Kamber, & Pei, 2012). These techniques are used in this research to explore implicit data relationships in highway project dataset. Section 3.3.1 discusses the use of decision tree. Section 3.3.2 continues with case based reasoning. Section 3.3.3 presents the evaluation method of two models.

3.3.1 Decision Tree Model

Decision tree is a flowchart style tree structure (Han, Kamber, & Pei, 2012). This research aims to generate a decision tree from highway project data.

Output: A decision tree.

Method:

(1) create a node N;
(2) **if** tuples in D are all of the same class, C, **then**
(3)   return N as a leaf node labeled with the class C;
(4) **if** *attribute_list* is empty **then**
(5)   return N as a leaf node labeled with the majority class in D; // majority voting
(6) apply **Attribute_selection_method**(D, *attribute_list*) to **find** the "best" *splitting_criterion*;
(7) label node N with *splitting_criterion*;
(8) **if** *splitting_attribute* is discrete-valued **and**
    multiway splits allowed **then** // not restricted to binary trees
(9)   *attribute_list* ← *attribute_list* − *splitting_attribute*; // remove *splitting_attribute*
(10) **for each** outcome j of *splitting_criterion*
// partition the tuples and grow subtrees for each partition
(11)   let D_j be the set of data tuples in D satisfying outcome j; // a partition
(12)   **if** D_j is empty **then**
(13)   attach a leaf labeled with the majority class in D to node N;
(14)   **else** attach the node returned by **Generate_decision_tree**(D_j, *attribute_list*) to node N;
endfor
(15) return N;

Figure 29: General decision tree algorithm (Han, Kamber, & Pei, 2012)

The project dataset has many lines (tuples). Each data tuple stands for one instance. The attributes of data tuples are denoted by tree nodes. Among these attributes, predictor attributes are represented by internal nodes while response attribute is represented by leave node. The set of instances are trained with decision tree algorithms to generate the decision tree.

Figure 29 illustrates the workflow for the general decision tree algorithm. The input is a dataset. Each line of the dataset D includes feature attributes and target attribute. The target attribute may have multiple classes. The algorithm starts with creating an empty node N. If the dataset D has only one class, then there is no need to split the dataset anymore. If not, the algorithm chooses an attribute to split according to a certain Attribute_selection_method. Node N is labeled with the splitting attribute. Then new branches are attached to node N. for each branch, there is a new empty node. The same algorithm is applied to each branch recursively until there are no more attributes left.

The Attribute_selection_method in this research is based on information entropy concept. Equation 1 is used to calculate the information entropy for a dataset T. T is partitioned by n values of the a feature attribute into i subsets T_1, T_2…, T_i . Let T_i be one of these subsets. Thus Freq(C_i, T_i) is the number of instances (tuples) in T_i that belongs to feature attribute class C_i. | T_i | denotes the number of instances in the subset T_i.

$$\mathrm{Info}(T) = -\sum_{i=1}^{k} ((\mathrm{freq}(C_i, T) / |T|) \cdot \log_2(\mathrm{freq}(C_i, T) / |T|))$$

Equation 1: Information entropy equation

After T is partitioned to a sub set based on a feature attribute value X. Then the information entropy of this sub set can be obtained through equation 2 which is a weighted sum of entropies over the subsets.

$$\mathrm{Info}_x(T) = \sum_{i=1}^{n} ((|T_i|/|T|) \cdot \mathrm{Info}(T_i))$$

Equation 2: Information entropy over subsets

Based on the equations above, the information gain achieved by partitioning on feature attribute value X can be given by Equation 3. Attribute_selection_method in this research chooses a feature attribute to maximize Gain(X), which means this method always prefers the highest information gain.

$$\mathrm{Gain}(X) = \mathrm{Info}(T) - \mathrm{Info}_x(T)$$

Equation 3: information gain equation

This research will use open source software Weka to follow this algorithm to generate decision tree on real highway project data to find interesting patterns between data attributes.

3.3.2 Case Based Reasoning (CBR) model

CBR finds a solution to a new case by retrieving and reusing the historical cases stored in the database. In addition, CBR can provide feedback to the new case so a new solution can be made.

The reasoning and decision making process highly depends on similarity assessment between cases. Similarity refers to numerical measure of how alike two data objects are. If the objects are more alike, the value of measure is higher. Dissimilarity refers to numerical measure of how different two data objects are. If the objects are more alike, the value of measure is lower. Similarity is usually defined as a value in [0,1] for simplicity reasons. Maximum similarity is 1 while minimum similarity is 0.

Measure of similarity should fulfill the following basic properties: 1) Identity: Sim(X, X) = Sim(Y, Y). 2) Monotonic: The addition of common features (attribute values) and/or the deletion of distinctive features both increase the similarity. 3) Independence: The order of the joint effect of any two components is independent of a third factor (López, Case-Based Reasoning-A Concise Introduction, 2013).

$$\text{Similarity}(\text{Case}_I, \text{Case}_R) = \sum_{i=1}^{n} w_i \times \text{sim}(f_i^I, f_i^R)$$

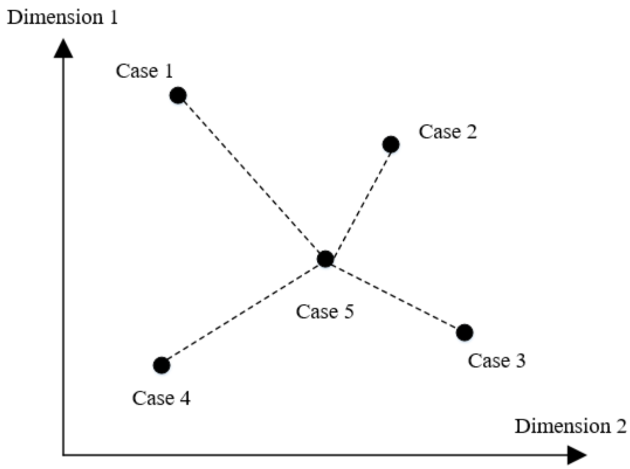


Figure 30: Similarities and distances between cases (Pantic, 2000)

Figure 30 shows the function of similarity between two cases. in the function, n is the total number of attributes, wi is the importance weight of an attribute, sim is the similarity function of attributes, and fiI and fiR are the values for attribute i in the input and retrieved cases respectively (Pantic, 2000). All cases correspond to points in n dimensional space. The distance between the points is determined by similarity. Obviously, more similarity between cases means shorter the distance between the two respective points in the multidimensional space. The similarity is usually defined as the inverse of distances. This research will calculate the distances of construction activity cases to find a most similar solution.

Different equations are used to calculate the distance between two cases. for nominal data, m denotes the numbers of matches while p denotes the total number of attributes. Then the similarity can be given by Equation 4.

$$d(case_i, case_j) = \frac{p-m}{p}$$

Equation 4: Distances between nominal data

For binary attributes, contingency table can be draw to show the match situation (Table 9).

Table 9: Contingency table for binary data

	Object$_j$			
Object$_i$		1	0	Sum
	1	q	r	q+r
	0	s	t	s+t
	Sum	q+s	r+t	p

$$d(case_i, case_j) = \frac{r+s}{q+r+s+t}$$

Equation 5: Distance between symmetric binary data

$$d(case_i, case_j) = \frac{r+s}{q+r+s}$$

Equation 6: Distance between asymmetric binary data

For numeric data, Minkowski distance is a popular distance measure (Equation 5). The distance is defined by the following formula. In the formula, i = (x_{i1}, x_{i2}, …, x_{ip}) and j = (x_{j1}, x_{j2}, …, x_{jp}) are two p-dimensional data objects, and h is the order (the distance so defined is also called L-h norm).

$$d(case_i, case_j) = \sqrt[h]{|x_{i1} - x_{i2}|^h + |x_{j1} - x_{j2}|^h + \cdots + |x_{ip} - x_{ip}|^h}$$

Equation 5: Minkowski distance

When h=2, the L-h norm becomes L-2 norm. The distance becomes typical Euclidean distance (Equation 6).

$$d(case_i, case_j) = \sqrt[2]{|x_{i1} - x_{i2}|^2 + \left|x_{j1} - x_{j2}\right|^2 + \cdots + \left|x_{ip} - x_{ip}\right|^2}$$

Equation 6: Euclidean distance

In this research, the feature data may contain mixed types which can be nominal, symmetric binary, asymmetric binary, numeric or ordinal. In this case, a weighted formula can be used to combine their effects (Equation 7).

$$d(i,j) = \frac{\Sigma_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\Sigma_{f=1}^{p} \delta_{ij}^{(f)}}$$

f is binary or nominal:

$d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$, or $d_{ij}^{(f)} = 1$ otherwise

f is numeric: use the normalized distance

f is ordinal: compute ranks r_{if} and treat z_{if} as interval-scaled

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

Equation 7: Distance between mixed type data

For n cases, a distance matrix is generated based on above-mentioned formulas (Figure 31). The solution of a new case can be retrieved by comparing these distances.

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ : & : & : & & \\ d(n,1) & d(n,2) & \ldots & \ldots & 0 \end{bmatrix}$$

Figure 31: Distances between multiple cases

3.3.3 Model Evaluation Method

3.3.3.1 Possible Outcomes

There are four possible outcomes for each binary classification (Table 10). Similarly, if positive refers to not delayed, negative refers to delayed. For each delay prediction, we will have the similar possible outcomes (Table 11):

True positives (TP): These refer to the positive instances that were correctly classified. TP is the number of true positives.

True negatives (TN): These are the negative instances that were correctly classified. TN is the number of true negatives.

False positives (FP): These are the negative instances that were incorrectly classified as positive (e.g., instances of class delay_status = no for which the classifier predicted delay_status = yes). Let FP be the number of false positives.

False negatives (FN): These are the positive instances that were mislabeled as negative (e.g., instances of class delay_status = yes for which the classifier predicted delay_status = no). Let FN be the number of false negatives.

Table 10: General binary classification outcomes

		Classification Results	
		Positive	Negative
Actual Results	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Table 11: Construction delay classification outcomes

		Classification Results	
		Not delayed	Delayed
Actual Results	Not delayed	True Positive (TP)	False Negative (FN)
	Delayed	False Positive (FP)	True Negative (TN)

3.3.3.2 Performance Measurements

Several performance measurements can be derived from the four possible outcomes.

3.3.3.2.1 Accuracy and Error Rate

An intuitive measure of the model performance is overall recognition rate calculated by the outcome values (Equation 8).

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{error rate} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Equation 8: Overall recognition rate

The overall recognition rate of the classifier reflects how well the classifier recognizes instances of all possible classes. The overall recognition rate has two measures. The accuracy of a classifier is the percentage of all tuples that are correctly classified. The error rate is the misclassification rate of a classifier, which is simply $1 - \text{accuracy}$. It also can be computed based on possible outcomes. This accuracy/error estimate is optimistic of the true corresponding accuracy/error rate.

3.3.3.2.2 Sensitivity and Specificity

It is important to note that accuracy is most effective when the distribution of target class is relatively balanced. When it comes to imbalance class problem, overall performance measure is not sufficient. The imbalance class problem refers to the problem where the main class of interest is rare. In reality, the classes of delay-status is also unbalanced. Usually, there are fewer delayed activities than not delayed activities. The problem in this research is an imbalance problem.

Let us consider this hypothetical situation. We may have a classification result like this: there are 100 predictions, among them, 99 are TP, only 1is FP (Figure 32). The accuracy is very high and error rate is very low. It seems that this model is excellent.

However, we cannot use the model to recognize any delayed activity. There is only one delayed activity and the model misses it. For this model, the recognition ability for delay class is very weak.

		Classification Results	
		Not delayed	Delayed
Actual Results	Not delayed	99	0
	Delayed	1	0

$$\text{accuracy} = \frac{99 + 0}{99 + 0 + 1 + 0} = 99\%$$

$$\text{error rate} = \frac{1 + 0}{99 + 0 + 1 + 0} = 1\%$$

Figure 32: An example of overall accuracy/error rate

The hypothetical example indicates that measuring only overall recognition ability is not enough. The recognition ability for each possible target class also needs to be measured separately. Two indicators are used to measure respective recognition ability (Equation 9). Sensitivity(recall) is the true positive (recognition) rate which the proportion of positive instances that are correctly classified. Specificity is the true negative (recognition) rate which the proportion of negative instances that are correctly classified. These measures are defined measure how well a certain classifier recognizes each class.

$$\text{sensitivity(recall)} = \text{True Positive Rate} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{specitivity} = \text{True Negative Rate} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

Equation 9: Overall recognition rate

Theoretically, a perfect model should classify the instances with highest sensitivity (recall) and highest specificity, which means the abilities to recognize both classes are very strong (Figure 33). This model can identify both positive and negative classes without any error.

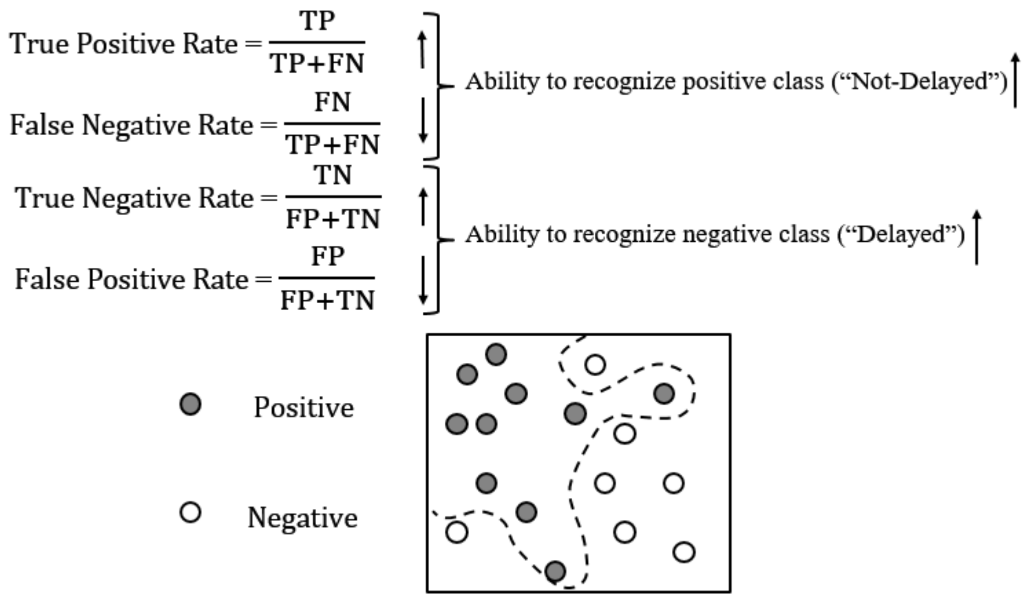


Figure 33: Perfect model performance

In reality, a model cannot perfectly identify all the classes. A model always makes mistakes, gives error. It cannot have both highest sensitivity (recall) and specificity values at the same time.

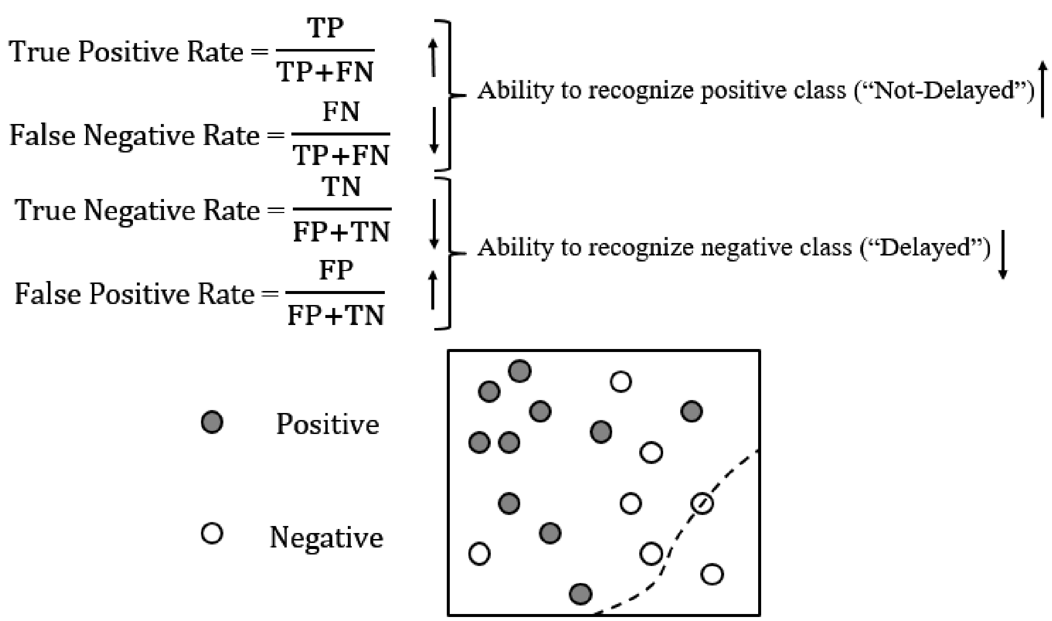


Figure 34: Overfitting for positive class

If model increases the ability to recognize positive class, then it will make more mistakes when recognizing negative class, more actual negative instances are labeled as positive (Figure 34). Vice versa, if model increases the ability to recognize negative class, then it will make more mistakes when recognizing positive class, more actual positive instances are labeled as negative (Figure 35).

$$\left.\begin{array}{l}\text{True Positive Rate} = \dfrac{TP}{TP+FN} \downarrow \\ \text{False Negative Rate} = \dfrac{FN}{TP+FN} \uparrow\end{array}\right\} \text{Ability to recognize positive class (“Not-Delayed”)} \downarrow$$

$$\left.\begin{array}{l}\text{True Negative Rate} = \dfrac{TN}{FP+TN} \uparrow \\ \text{False Positive Rate} = \dfrac{FP}{FP+TN} \downarrow\end{array}\right\} \text{Ability to recognize negative class (“Delayed”)} \uparrow$$

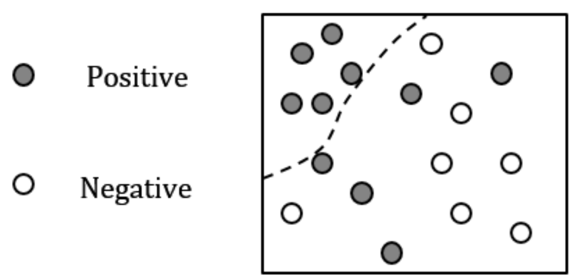


Figure 35: Overfitting for negative class

There tends to be an inverse relationship between sensitivity (recall) and specificity, where it is possible to increase one at the cost of reducing the other.

3.3.3.2.3 Construction Delay Decision Cost

According to previous discussion, given a test data, it will be very rare that a model give you both high sensitivity and specificity values. For two different models it may give you a hypothetical result like this (Figure 36): The accuracy is the same but sensitivity (recall) and specificity are very different.

For the first model, sensitivity is low, specificity is high. Which means the ability to recognize positive class ("Not-Delayed") is high. However, for negative classes it makes more errors. More actual delayed activities are predicted as "Not-Delayed" (90%).

For the second model, sensitivity is high, specificity is low. Which means the ability to detect negative class ("Delayed") is high. However, for positive classes it makes more errors. More actual not delayed activities are predicted as "Delayed" (90%).

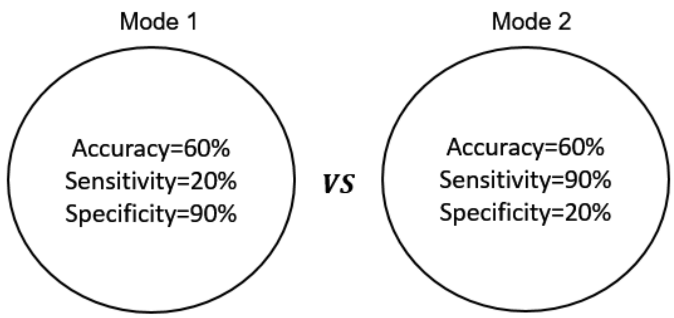


Figure 36: An example of two models with different sensitivity and specificity

When a project manager wants to make a decision, he or she needs know the kind of consequences associated with the decision result. Some decision (prediction) result can bring very serious consequences. We call this decision cost. Each decision result is associated with certain potential cost for construction delay predication.

With the model 1, a project manager tends to predict an actual delayed activity as "not delayed" .Certainly, there is a cost for this wrong decision. Things will get out of control. He may have a big trouble. He needs more time and resource to handle this unexpected situation. In addition, there may be some chain effect. Therefore, there is a very high potential cost with the decision.

With the model 2, a project manager tends to predict an actual not delayed activity as "delayed" .Yes. There is also a cost for this wrong decision. However, things are still in control. This cost is not as high as previous one.

Generally speaking, the cost of predicting an actual delayed activity as “Not-Delayed” (Model 1) is far greater than the predicting an actual not-delayed activity as “Delayed” (Model 2).

In this research’s setting, we prefer a model with relatively lower delay prediction cost, which means the model should have stronger ability to recognize negative class, which also means the value of specificity should be higher, the value of sensitivity should be lower.

For the imbalance problem in this research, in addition to overall accuracy and error rate, specificity is also used as a main measure to evaluate the model. The higher specificity, the better.

3.3.3.2.4 ROC Curve and AUC

Although specificity is used and we especially focus on increasing the ability to recognize negative class, we still want to balance the tradeoff effect of these two abilities. We don’t want the ability to recognize positive class lose much. For a binary classification problem, given different portions of the test data set, a receiver operating characteristic (ROC) curve allows visualizing the trade-off between the rate at which the model can accurately recognize positive class instances (TPR) versus the rate at which it wrongly recognize negative class instances as positive class (FPR). An increase in TPR usually occurs at the cost of an increase in FPR.

Given a dataset, given a scenario (a set threshold), we can calculate the TPR and FPR value, and then plot in this coordinate system. This point shows the tradeoff level for this model in this specific scenario. This procedure can be done many times for different scenarios. A series of points can be generated. By connecting the points, We can get a ROC curve (Figure 37). The area under roc curve is called AUC. AUC is used to measure the model's ability to balance the tradeoff effect. The value of AUC is large means the balance ability of the model is strong.

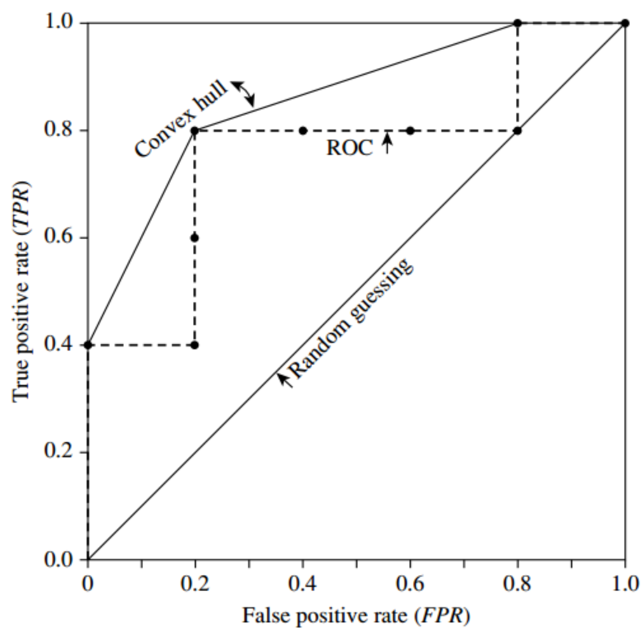


Figure 37: ROC curve and AUC (Han, Kamber, & Pei, 2012)

3.3.3.3 Cross Validation

After the measures are chosen, next step will be run the model on the test data to get evaluation results. The amount of data used in this research is limited. A challenge is how to get a more accurate evaluation result with limited amount of data is an issue.

To fix this issue, 10 times tenfold cross validation technique is utilized (Figure 38). Each time the whole data is randomized and then be divided into 10 pieces. Each piece is used in turns to as test data, it is processed by Decision tree and CBR to generate TP, FN, TN,FP values. This process is repeated 100 times, which produces 100 sets of values. These values are averaged to compute final evaluation results.

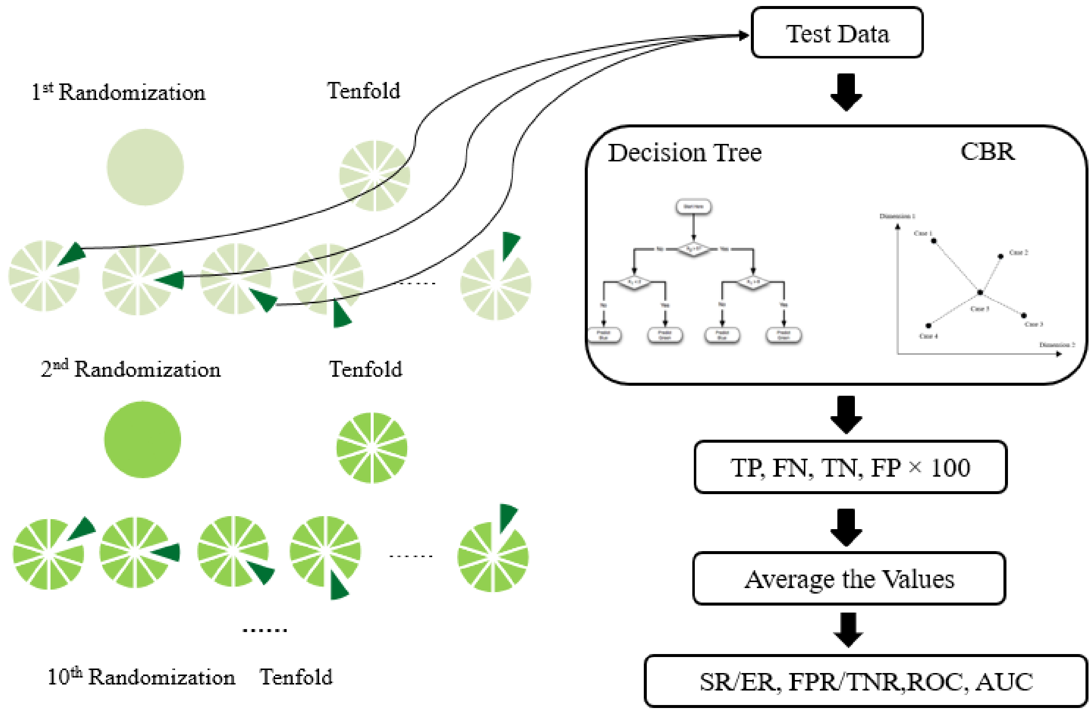


Figure 38: Ten times ten fold cross validation

3.4 Prototype System Implementation

3.4.1 System Module

The main modules, their functions and corresponding developing tools for these sub systems are illustrated in Figure 29. IFC generator and parser, ETL processor and system connectors are in written in Ruby and Java programming languages. MySQL server is the tool to build the physical data warehouse. Weka is a popular data mining software package. The Weka workbench has the state-of-the-art machine learning algorithms and data preprocessing tools (Witten, Frank, & A.Hall, 2011). The system is written in Java and distributed under the terms of the GNU General Public License. in the prototype system, it is used to connect data warehouse in SQL server to retrieve data and then perform analysis. Sketchup is used to provide necessary 3D representation.

One of the most important modules is IFC parser. The detailed code structure is illustrated as follows (Figure 39):

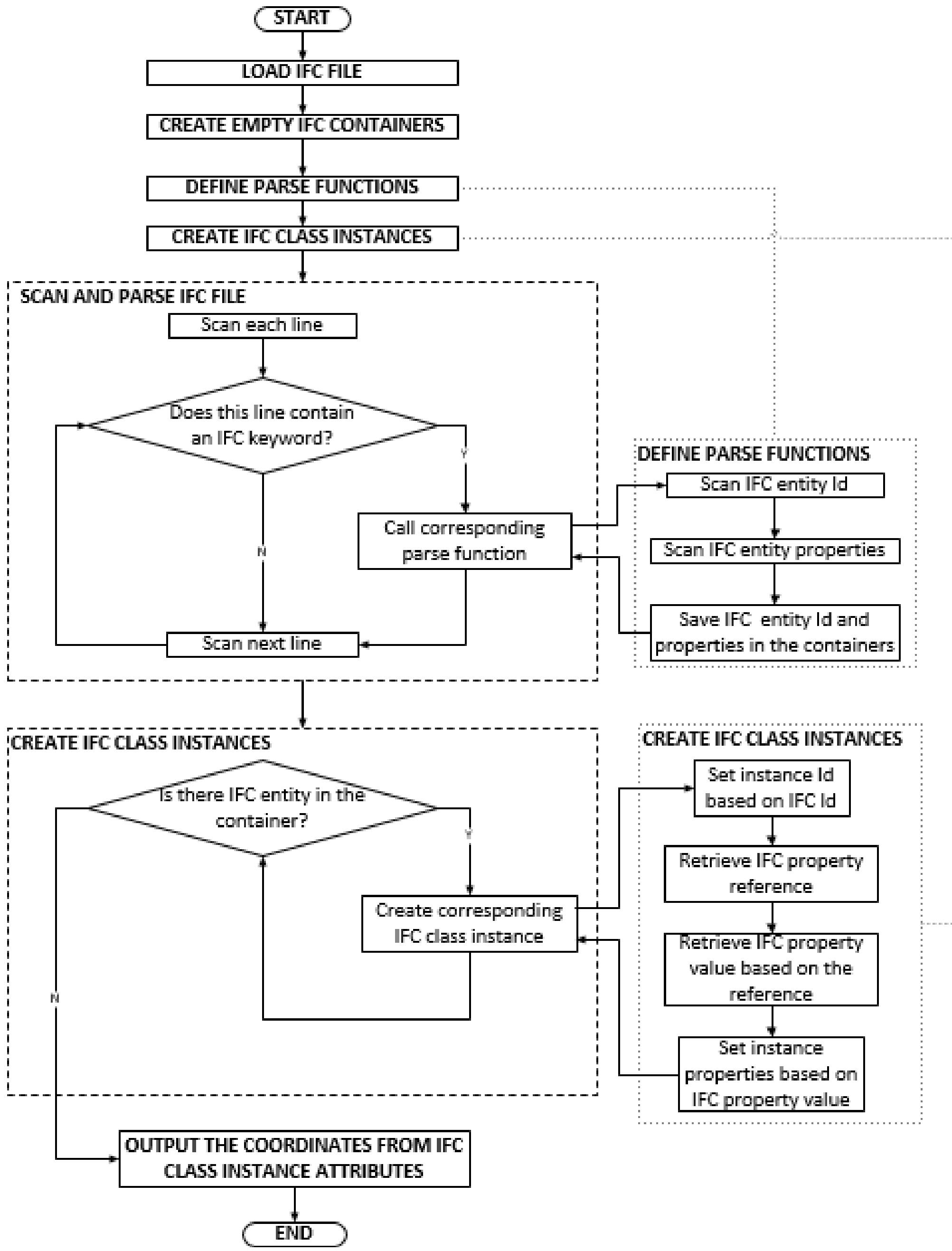


Figure 39: IFC parser flowchart

This module is created to read the physical IFC file and parse it into IFC entities (objects) in computer memory. The code starts by loading the IFC file from hard disk into computer memory. Empty arrays are created to temporarily store IFC entity information.

They are called IFC containers. Functions are defined to extract IFC entity information from each line of IFC file. They are called parse functions. Abstract IFC entity class is also defined for future use. Then the code begin scanning and parsing each line of IFC file. Corresponding IFC entity information will be saved in temporary IFC entity containers. IFC entity class instances will be further created based on the information in IFC entity containers and predefined IFC entity class.

3.4.2 System Architecture

The system architecture follows a client-server style (Figure 40). The processing is partitioned into three tiers: 1st client tier which initiates query request, 2nd application tier which performs analysis, 3rd database tier which stores and provides data. in this system, user first initiates the data mining request through frontend user interface system. The request is transferred to application server via internet. Application server retrieves data from backend data warehouse database according to the use's requirement. Then application server runs the built-in data mining algorithms on the retrieved data. The final result is transferred back to the frontend user interface.

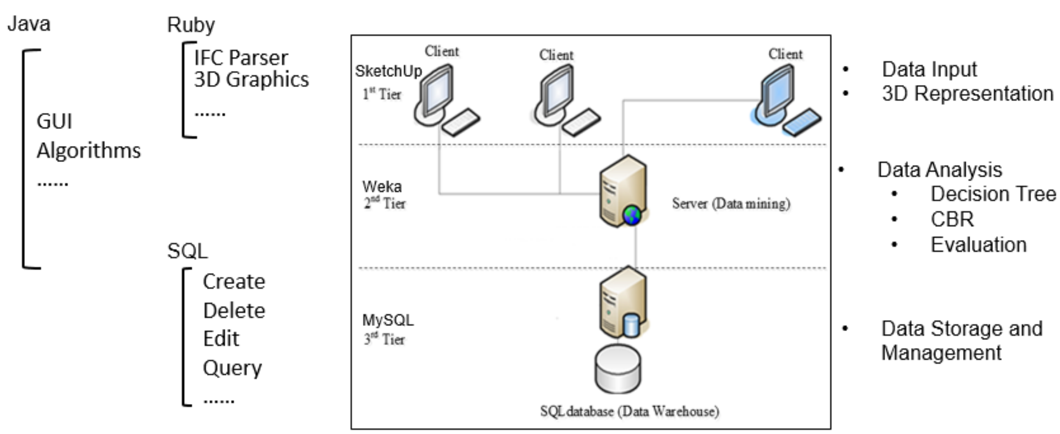


Figure 40. Client-Server style system architecture

3.4.3 User Interface

A series of pictures below shows the graphical representation of user interface and system workflow.

First, a user can import the IFC file by clicking the "Load IFC File" button (Figure 41). The file is read and parsed inside the system. A 3D representation of construction components are displayed on the screen. User can check and verify the construction components visually.

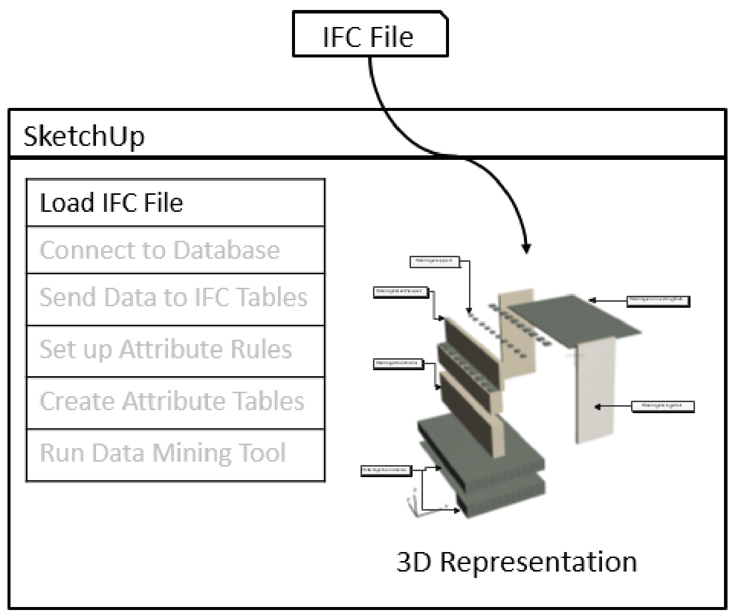


Figure 41: User interface - loading file

The frontend system connects to the backend database (Figure 42). IFC entity information are collected and transferred to database. The tables are created in database to store IFC entity information. IFC attributes are added to the fields of a table. A table consists of field values. GlobalId is set to be an IFC table's primary key.

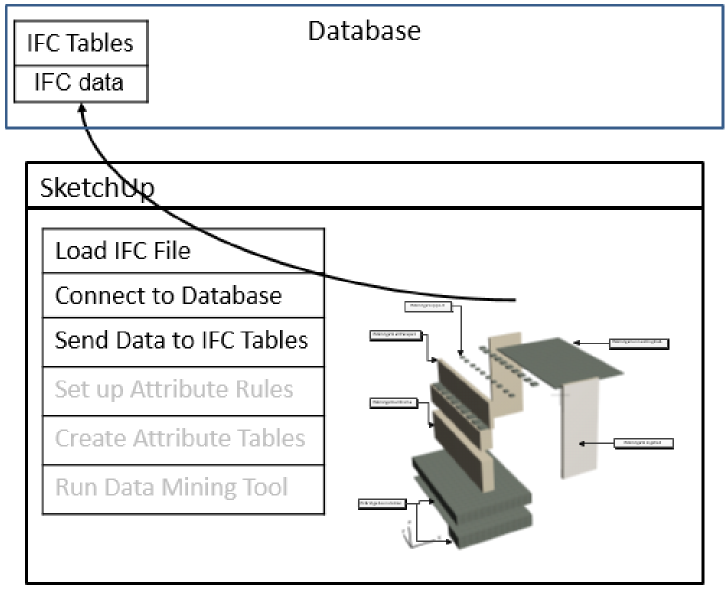


Figure 42: User interface – connecting database

User input some rules with Ruby and SQL scripts in the editor window of front end (Figure 43). The scripts are based on the user requirements. They specify what the feature attributes and target attribute are and how to generate these attribute values from IFC tables.

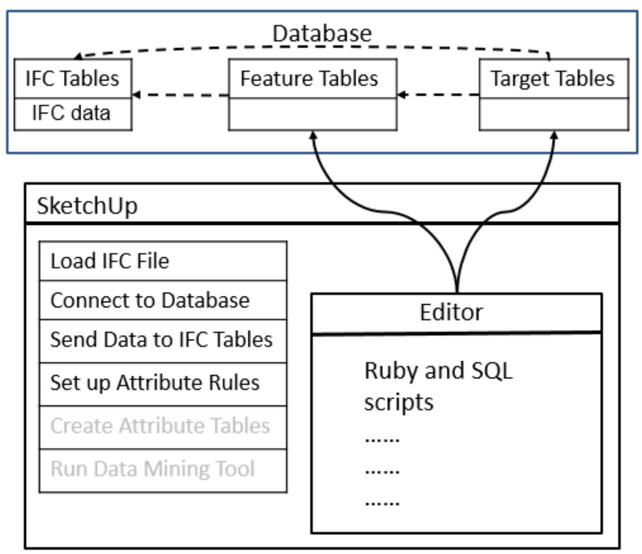


Figure 43: User interface – setting up rules

Remote database creates the feature attribute tables and target attribute table (Figure 44). Feature data and target data are then automatically populated to corresponding tables in database based on these rules.

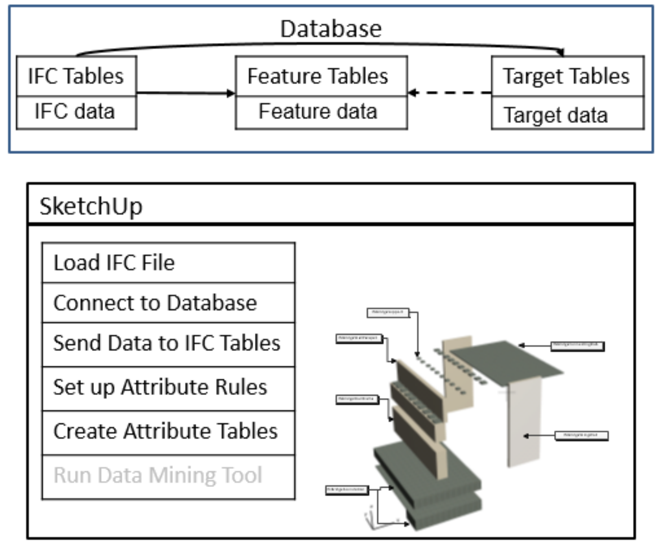


Figure 44: User interface – creating attribute tables

Finally, a data mining tool can be run to discover some patterns or make prediction based on the data (Figure 45). One part of the output gives estimates of the model's predictive performance.

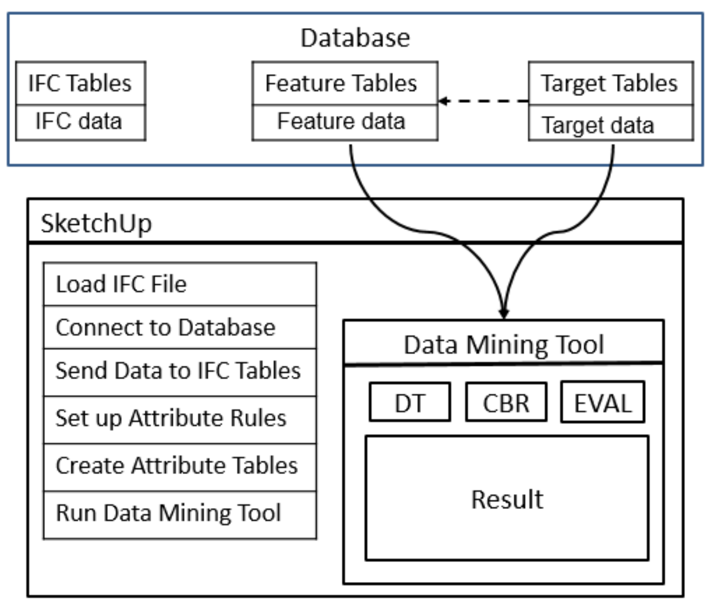


Figure 45: User interface – data mining tools

CHAPTER 4: CASE STUDY – KICT HIGHWAY PROJECT ANALYSIS

4.1 Case Study Process

The case study process as shown in Figure 46 is composed of four phases: 1) IFC file preparation, 2) Data storage and management, 3) Data mining, and 4) Model evaluation. The process starts by writing KICT highway project information into a Physical IFC file. Second step is to store the IFC entities in data warehouse, extract the feature and target data from these entities. Next step is to run the decision tree and case based reasoning tools on the feature and target data to explore some useful patterns. In the last step, the performances of the data mining models are compared to help project manager in predicting future highway construction delay.

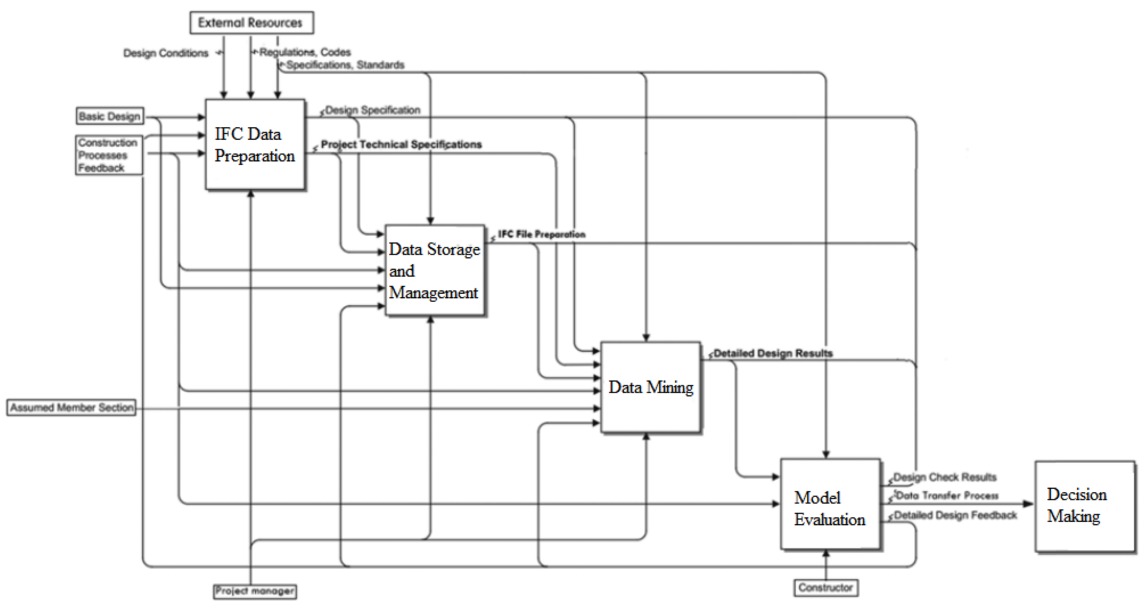


Figure 46: Case study process

4.2 IFC File Preparation

The case study data comes from Korea institute of Civil Engineering Technology (KICT). It is from a real highway construction project located in a mountain area of South Korea consisting of two bridges, one tunnel and two segments of road (Figure 47). The total length of the project is 947.5 meters and the total construction cost was $96,222,115. The project started on May 1, 2012 and completed on October 22, 2013. The project data is collected in ad-hoc manner. The project file formats include plain txt, spreadsheet, CAD drawing, PDF MS project and MS word files. The data contains the geometry information, site survey information, quantity and cost estimation information, project schedule information and actual activity information, etc. There are 314 construction activities included in the data.

Component	Length (m)	Coordinates	Structure
Bridge 1	15	-0+227.500~-0+212.500	Rigid frame-Rahmen structure
Road 1	213	-0+212.500~0+005.000	Asphalt paved road
Bridge 2	40.15	0+005.000~0+045.150	P.S.C. e-BEAM bridge
Road 2	474.85	045.150~0+520.000	Asphalt paved road
Tunnel	200	0+520.000~0+720.000	Cast in place concrete, R.C. Box

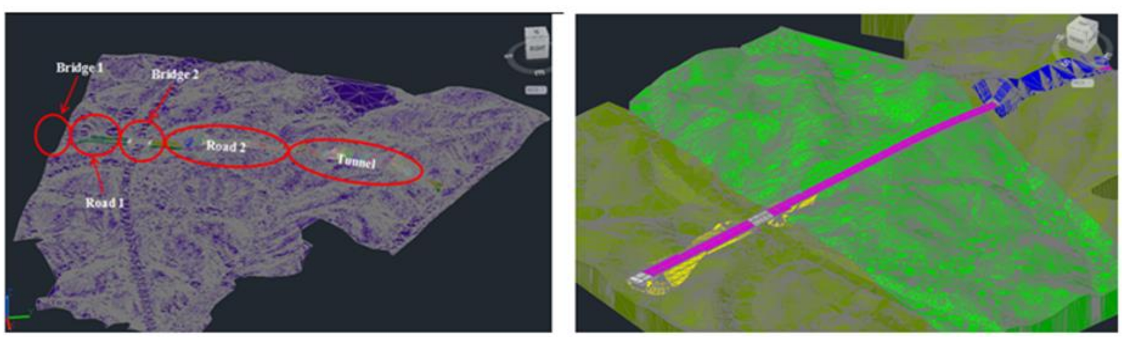


Figure 47: Summary of KICT project information

Since the raw project data is in various formats and recorded in ad-hoc manner, data preprocessing is required to improve the data quality. Several techniques such as data cleaning and data discretization are employed. During this process, missing values are identified, noise are smoothed out while outliers are removed, and inconsistencies in the data are corrected.

The raw project data is then transformed to corresponding IFC entities according to previously designed highway IFC data schema (Figure 48). Each object is converted to an IFC entity. Subtypes of IfcRelationship entity define all objectified one-to-one and one-to-many relationships between these IFC entities. The attributes of the entity are also specified with associated properties of respective object. The designed highway IFC schema combines data from multiple sources to form a coherent data. All the IFC entities are written line by line in a physical IFC file.

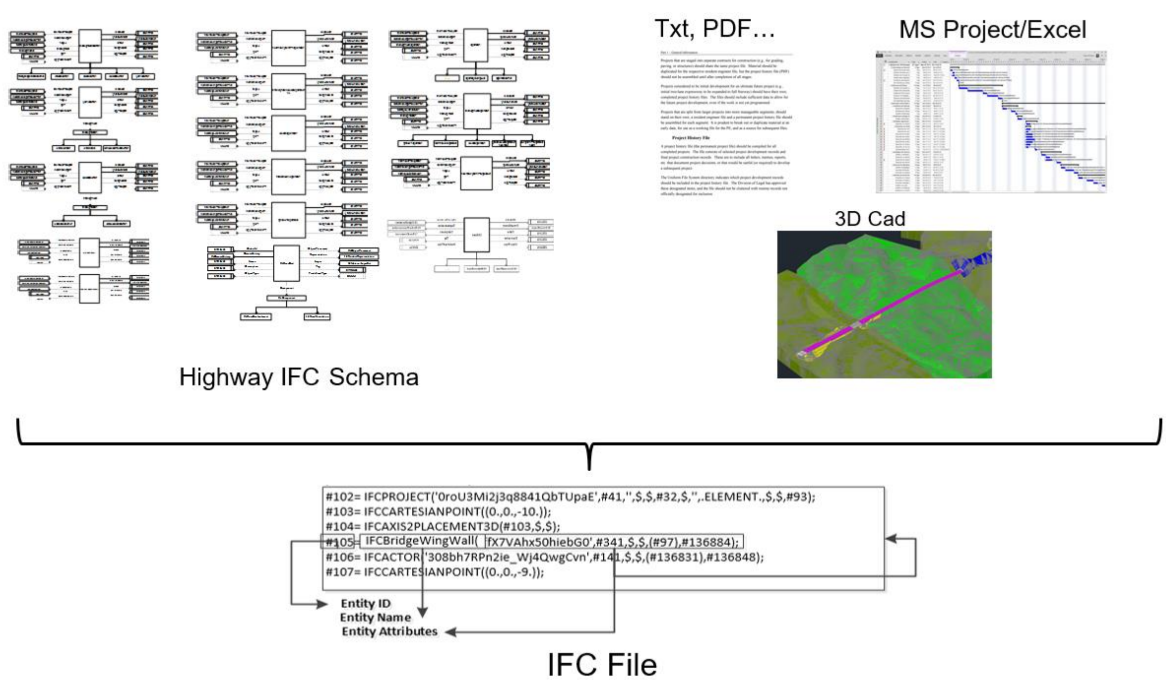


Figure 48: Conversion of project information

The generated IFC file following ISO 10303-21 encoding mechanism has 81,118 lines. It includes 81,118 entity instances, 156, 246 attributes and 12,735 relationships (Figure 49). Each line of this IFC file consists of annotations and attributes that define various project activity specifications. This IFC file provides the project information to extract feature class and target class data.


```
#296= IFCPROPERTYSINGLEVALUE('Rebar Cover - Top Face',$,IFCLABEL('Rebar Cover Settings: \X2\BCF4AC15\X0\ \X2\CCA0ADFC\X0\ \X2\D53CBCF5\X0\ 1'),$);
#297= IFCPROPERTYSINGLEVALUE('Offset',$,IFCLENGTHMEASURE(0.),$);
#298= IFCPROPERTYSINGLEVALUE('Level',$,IFCLABEL('Level: \X2\B808BCA8\X0\ 1'),$);
#299= IFCPROPERTYSINGLEVALUE('Enable Analytical Model',$,IFCBOOLEAN(.F.),$);
#300= IFCPROPERTYSINGLEVALUE('Phase Created',$,IFCLABEL('F11120-01+F15100-03>S16240-01-07+S11300-02+S16130-02>E13110-02-000'),$);
#301= IFCPROPERTYSINGLEVALUE('Elevation at Bottom',$,IFCLENGTHMEASURE(-1099.99999999997),$);
#302= IFCPROPERTYSINGLEVALUE('Moves With Grids',$,IFCBOOLEAN(.T.),$);
#303= IFCPROPERTYSINGLEVALUE('Rebar Cover - Other Faces',$,IFCLABEL('Rebar Cover Settings: \X2\BCF4AC15\X0\ \X2\CCA0ADFC\X0\ \X2\D53CBCF5\X0\ 1'),$);
#304= IFCPROPERTYSINGLEVALUE('Host',$,IFCTEXT('Level : \X2\B808BCA8\X0\ 1'),$);
#305= IFCPROPERTYSINGLEVALUE('Structural Material',$,IFCLABEL('Concrete - Cast-in-Place Concrete'),$);
#306= IFCPROPERTYSINGLEVALUE('Length',$,IFCLENGTHMEASURE(5200.),$);
#307= IFCPROPERTYSINGLEVALUE('OmniClass Title',$,IFCTEXT('Shallow Foundations'),$);
#308= IFCPROPERTYSINGLEVALUE('Assembly Description',$,IFCTEXT(''),$);
#309= IFCPROPERTYSINGLEVALUE('Width',$,IFCLENGTHMEASURE(24900.),$);
#310= IFCPROPERTYSINGLEVALUE('Thickness',$,IFCLENGTHMEASURE(1100.),$);
#311= IFCPROPERTYSINGLEVALUE('OmniClass Number',$,IFCTEXT('23.25.05.17'),$);
#312= IFCPROPERTYSINGLEVALUE('Assembly Code',$,IFCTEXT(''),$);
#313= IFCPROPERTYSET('3K$2PJ6699gwzUn8O12DDQ',#41,'Structural',$,(#296,#299,#303));
#318= IFCRELDEFINESBYPROPERTIES('288wP$KMX9jRo8pgYXSAu4',#41,$,$,(#172),#313);
#322= IFCPROPERTYSET('3ISZP8W_j6iubnIpuJTKY0',#41,'Constraints',$,(#297,#298,#302,#304));
#324= IFCRELDEFINESBYPROPERTIES('3MZeZ9umj0GQDAY34STp2Q',#41,$,$,(#172),#322);
#327= IFCPROPERTYSET('1$D9Jd34b1leybbSkDtuPr',#41,'Phasing',$,(#300));
#329= IFCRELDEFINESBYPROPERTIES('03I1EmLjPA8AZ8XlsXBJQG',#41,$,$,(#172),#327);
#332= IFCPROPERTYSET('2zoKEoFEz2UO8GSR6c05Tj',#41,'Dimensions',$,(#301));
#334= IFCRELDEFINESBYPROPERTIES('1ZbxzMpsX9tO5FGMVGqHqu',#41,$,$,(#172),#332);
#337= IFCPROPERTYSET('3EnWwCQs59v8UrCDjYRFu2',#41,'Materials and Finishes',$,(#305));
#339= IFCRELDEFINESBYPROPERTIES('0f4Cl0Tu9Ccu8ajihGANHH',#41,$,$,(#172),#337);
#342= IFCPROPERTYSET('0SBC0sTjH1ExwHwRh5Dpzv',#41,'Identity Data',$,(#307,#308,#311,#312));
#344= IFCPROPERTYSET('0OZTJrXErEYxDkucT4zqC$',#41,'Dimensions',$,(#306,#309,#310));
#346= IFCAXIS2PLACEMENT3D(#6,$,$);
```


Figure 49: Physical KICT project IFC file

4.3 Data Storage and Management

The IFC file is then loaded into the prototype system (Figure 50). First, Sketchup is opened up. Click the *Plugin* Button to run the Highway Knowledge Discovery System Demo Program. The popup menu has eight options. *Load IFC File, Show Site, Show Structure* are used to load IFC data and display the 3D model. *Connect to database, Send Data to IFC Table, Set Up Class Rules, Create Class Rules* are used to interact with remote database. *Run Data Mining tool* is used to call the decision tree and case based reasoning tools.

A standard dialog can be brought up by clicking the Load IFC File button. User can select an IFC file through this dialog. Choose the *KICT Highway Project.ifc* file. The IFC file is then parsed into IFC entities.

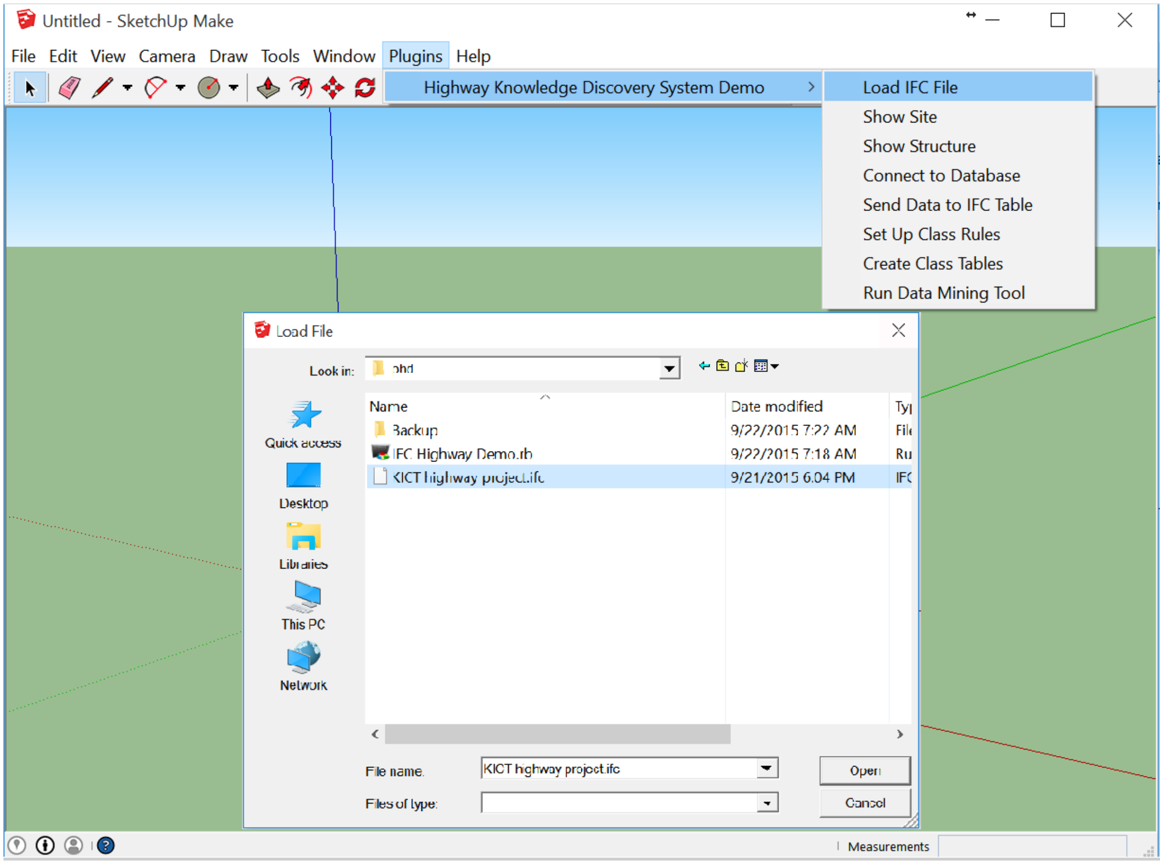


Figure 50: Loading KICT project IFC file

Figure 51 shows the screen after the user loads the IFC file into the system. It shows the 3D representation of construction structures. It enables user to check the details of the 3D model by rotating or zooming in and zooming out. The main part of the road is in gray color. The main part of the bridge and tunnel is in white color. Different colors of the site stand for different soil type within a certain area.

By clicking a 3D object, a window can be popped up to display some basic information of that object. The prototype system uses an interactive way so the user can visually explore the project activity information associated with a certain 3D spatial element (eg, site, structure element, etc).

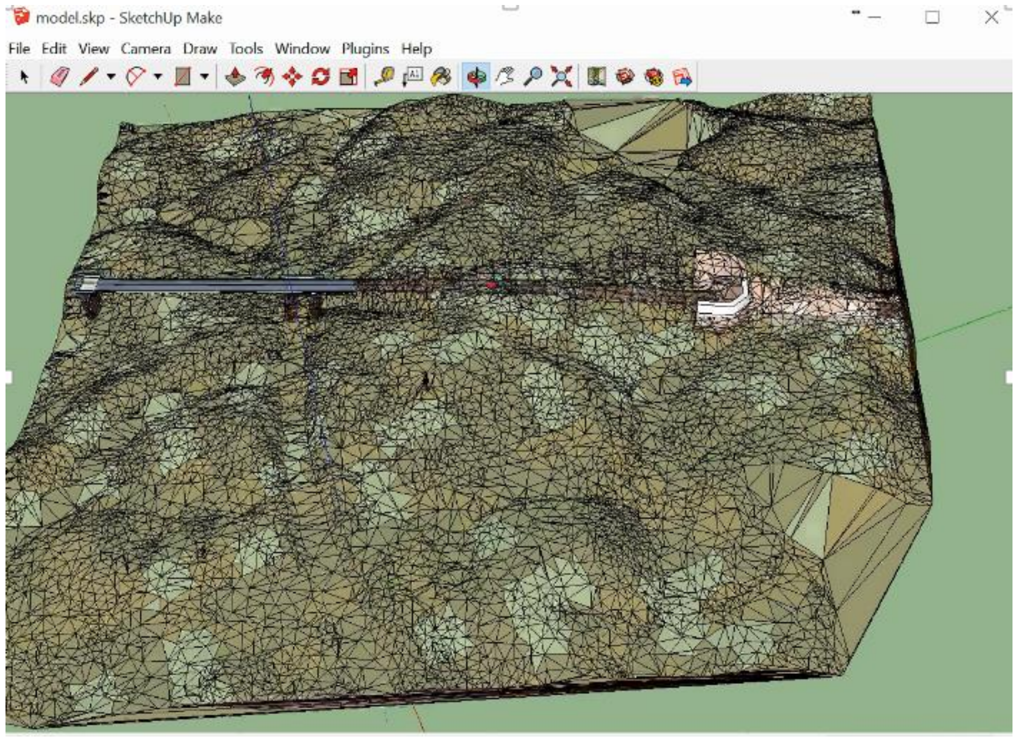


Figure 51: 3D representation of KICT project data

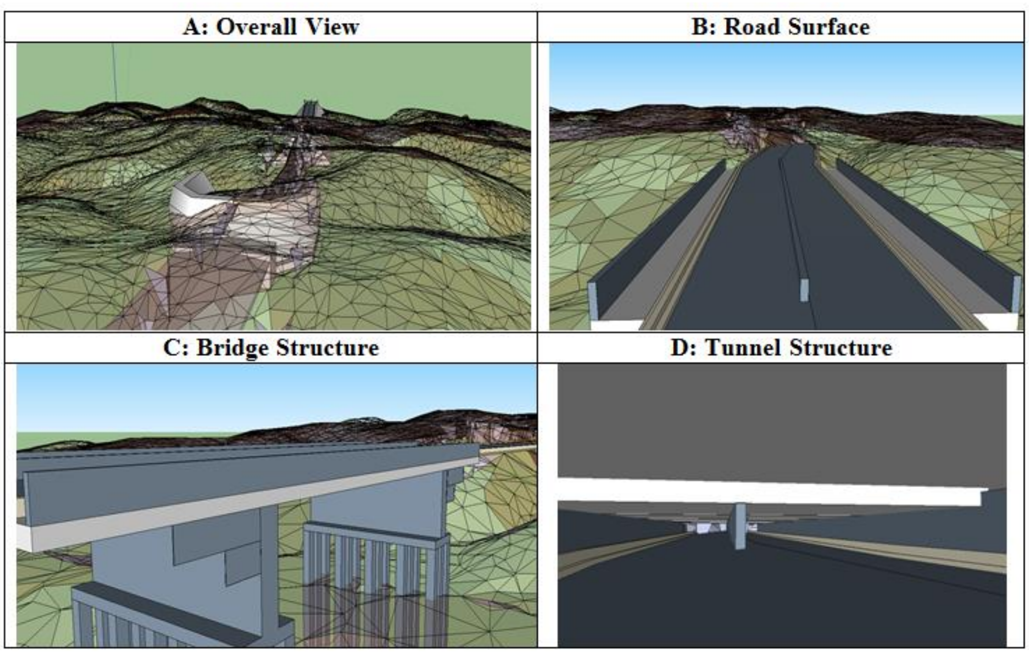


Figure 52: Details of 3D model

By clicking the *Send Data to IFC Tables* button, the IFC entities are populated to IFC tables in remote data warehouse (Figure 53). 215 IFC tables are created. Once IFC tables are filled with data, 16 feature attribute tables and 1 target attribute table are created. User inputs the SQL scripts to set up some rules. Data in IFC tables are then processed and queried to extract the values for feature and target attributes.

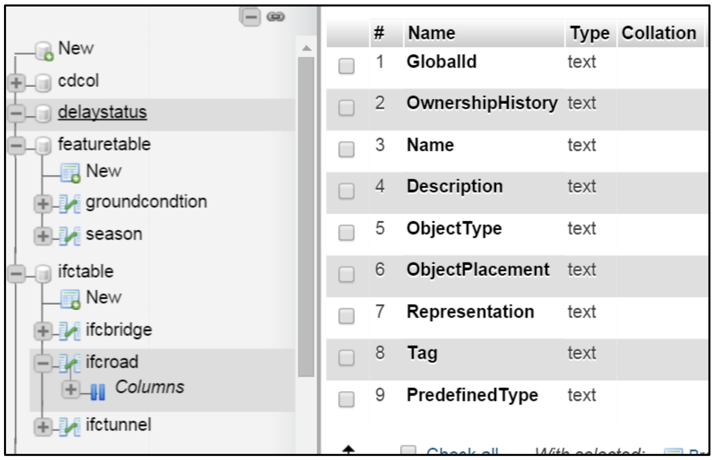


Figure 53: Construction of database

4.4 Data Mining

After the attribute data is ready, user can click the *Run Data Mining tool* button and *Decision Tree* tab to build a decision tree model. Figure 54 shows the textural tree model in the case study. The system generally chooses sensible defaults for decision tree generation. It is embedded with the pruning function so unnecessary branches of the tree can be simplified to avoid overfitting issue. Figure 55 illustrates a graphical tree generated in the prototype system. It provides a convenient way for the user to check the details of the decision tree structure.

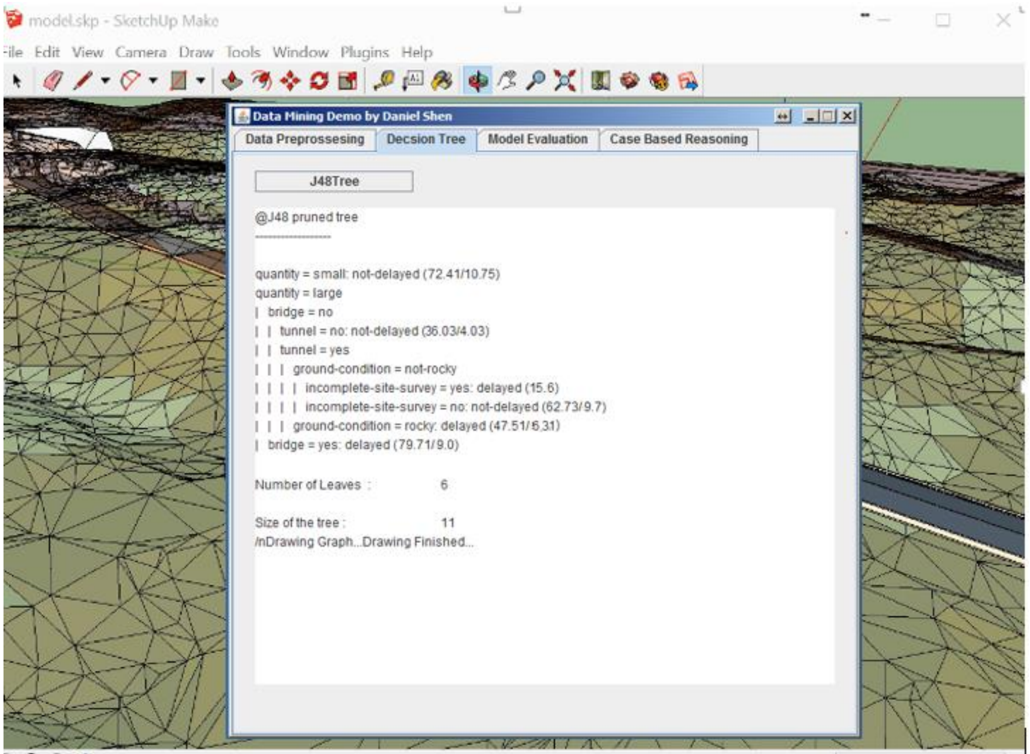


Figure 54: Textural decision tree

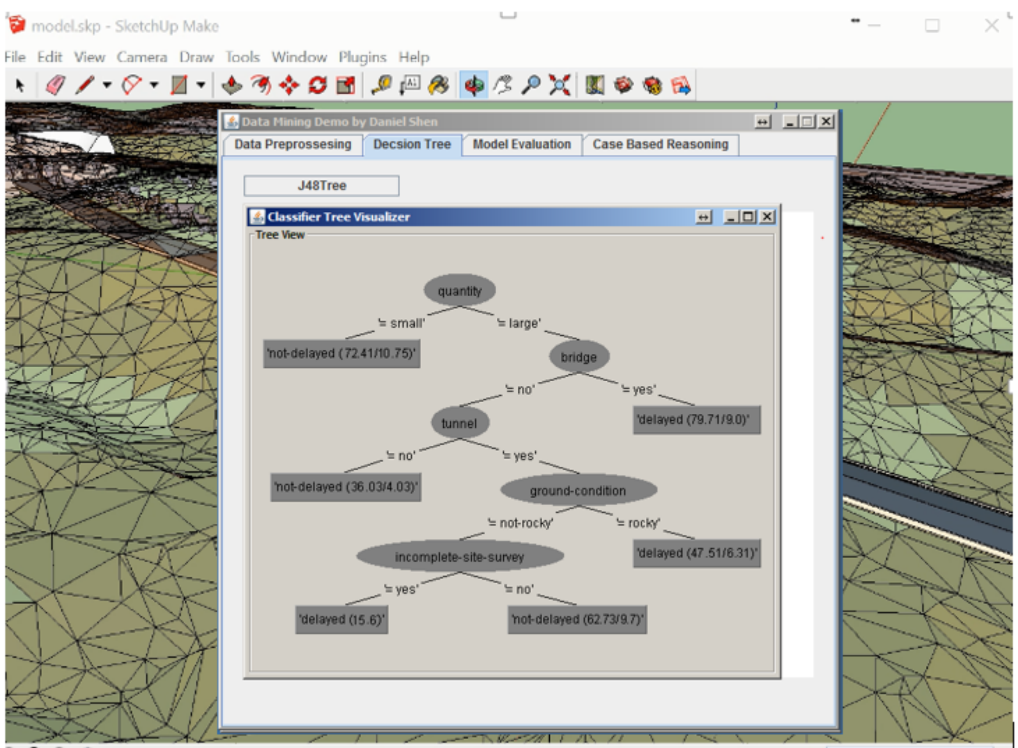


Figure 55: Graphical decision tree

In the tree structure, class label for target attribute has been assigned to a leaf. It is introduced by a colon and followed by the number of data instances which reach that particular leaf. The number is decimal due to the way decision tree algorithm utilize fractional instances to handle missing values. The number of incorrectly classified instances is also shown. For example, (72.41/10.75) means 72.41 of instances are classified as “not-delayed”, out of which 10.75 of instances are misclassified. The total number of leaves and the number of size of tree are displayed below the tree structure.

The first split is on the quantity attribute, and then, at the second level, the splits are on bridge attribute, the third level split is on tunnel attribute, the fourth level split is on ground condition attribute, the fifth level split is on incomplete site survey attribute. Binary splitting is enforce to simply the hierarchy of the tree structure.

Some interesting patterns can be found in the tree:

1) Quantity, bridge, tunnel, ground condition and incomplete site survey attributes are far more important than any other attributes in the training dataset. These five attributes already have enough information for us to predict the construction delay result.
2) Quantity and bridge attributes are more important factors to determine the construction delays. This is because the decision tree algorithm always prioritizes its splits by splitting the most significant attributes first. These two attributes can explain 51% of the cause of construction delay.
3) If a construction activity is related to tunnel construction in some aspects, the ground condition and site survey will be very important. Ground condition is especially important for the tunnel construction activity. If the ground is rocky, then this activity is usually delayed.

The tree structure can be used as a tool for project delay prediction. A prediction can be made by testing the attribute value of the instance on the nodes. From top to bottom, follow a certain path, a prediction can be made. Specifically, a set of rules can be extracted from the tree:

R1: IF quantity = small

THEN delay_status = not-delayed

R2: IF quantity = large

AND bridge = no AND tunnel = no

THEN delay_status = not-delayed

R3: IF quantity = large

AND bridge = no AND tunnel = yes AND ground_condition = rocky

THEN delay_status = delayed

R4: IF quantity = large

AND bridge = no AND tunnel = yes AND ground_condition = not-rocky

AND incomplete-site-survey = yes

THEN delay_status = delayed

R5: IF quantity = large

AND bridge = no AND tunnel = yes AND ground_condition = not-rocky

AND incomplete-site-survey = no

THEN delay_status = not-delayed

R5: IF quantity = large

AND bridge = yes

THEN delay_status = delayed

Case based reasoning is another tool for project delay prediction. Each instance in the database is treated as a case. Figure 56 shows the table editor for KICT project historical cases.

The first row shows the attribute type and name. Each column of the corresponding attribute lists the different values for different cases. Dark color field means the value of the attribute for that specific case is missing. User can manually modify the cases by directly clicking on each row. The stored cases are utilized as query and case values for case retrieval. The last column saves the target attribute value. Other columns save the feature attribute value.

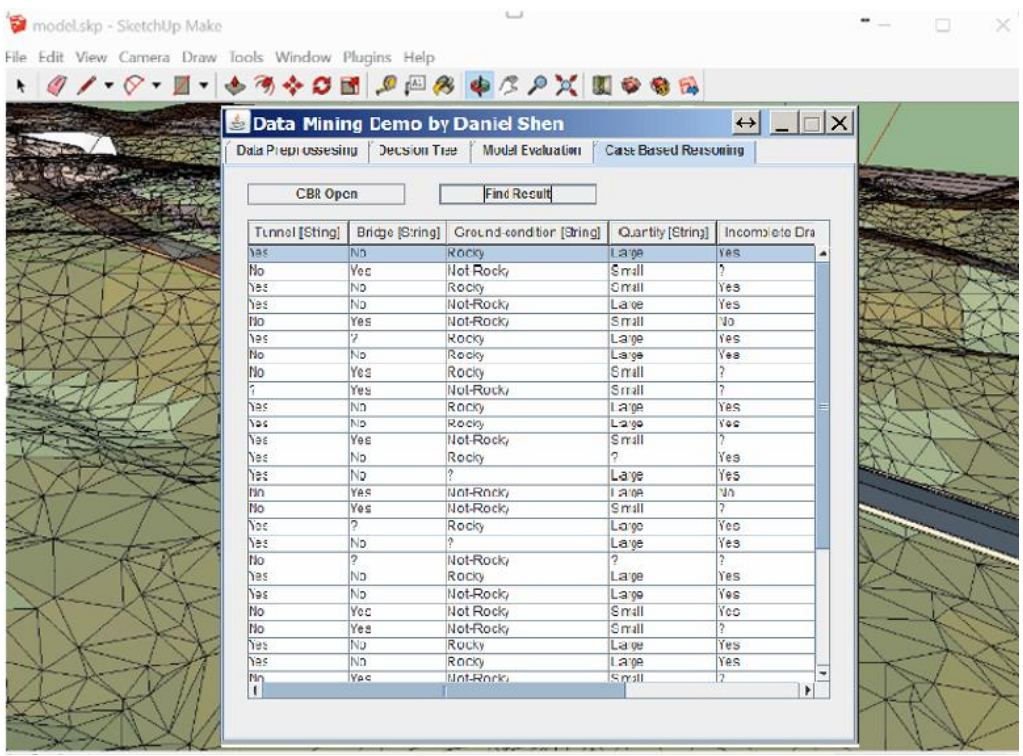


Figure 56: CBR case table

Retrieval engine can be provoked by pressing *Find Result* button, a query table window can be popped up for the user to submit the test case value manually (Figure 57). The weight of the attribute should be in the range of 1 to 5. In this case study, the weight is always set equal to 5. The scale option let the user chose the kind of match algorithm the system uses to retrieval the similar case. A fuzzy linear algorithm is adopted by default.

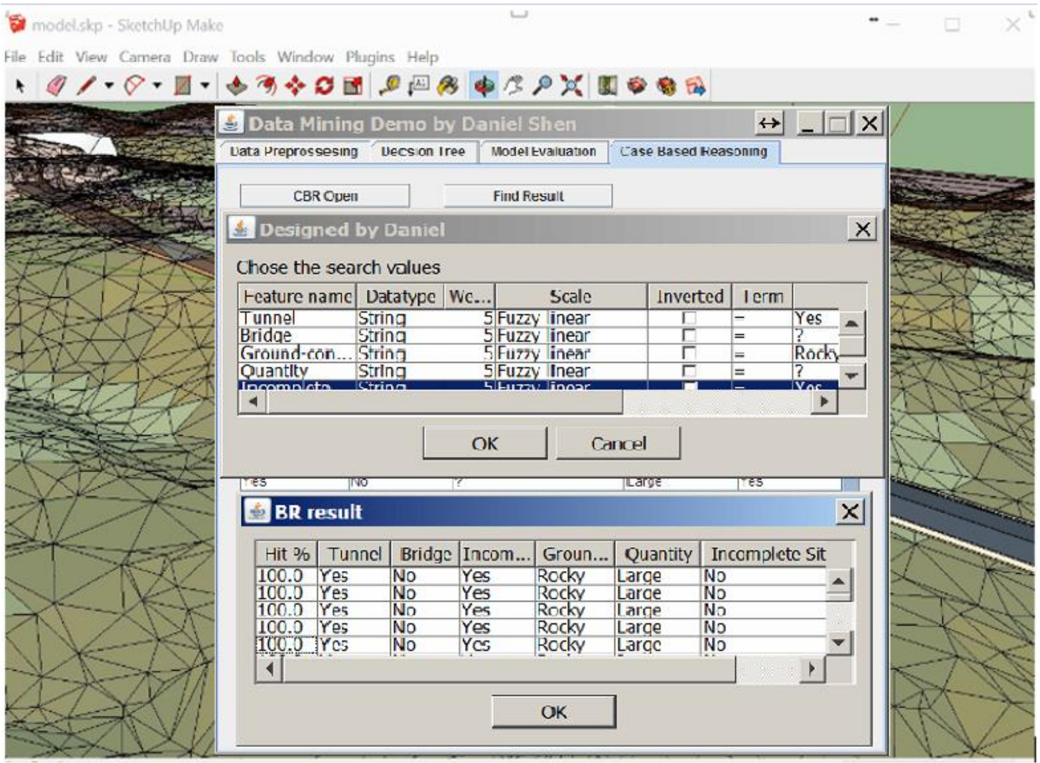


Figure 57: CBR query engine

User can input the query values for each cell in the table. Click *OK* button below. The table then list the historical cases that are most similar to the query values according to the fuzzy linear algorithm. Figure shows the list of cases ordered by similarity. The target attribute of the most similar case is usually the prediction value.

4.5 Model Comparison

The prototype system provides a module that evaluates the classifiers on test data and outputs performance statistics. Figure 58 shows the basic performance evaluation result for one test prediction in the case study. Success rate follows correctly classified instance number. Error rate follows incorrectly classified instance number. A confusion matrix is displayed at the bottom of the output screen. For example, 16 is the number of true positive instances, 2 is the number of false positive instances, 3 is the number of false negative instances, 10 is the number of true negative instances.

Other performance measures are derived based on the basic values above. 100 sets of test result can be obtained by ten times ten fold cross validation. A series of averaged values is calculated from these results. ROC curve is plotted and AUC is measured.

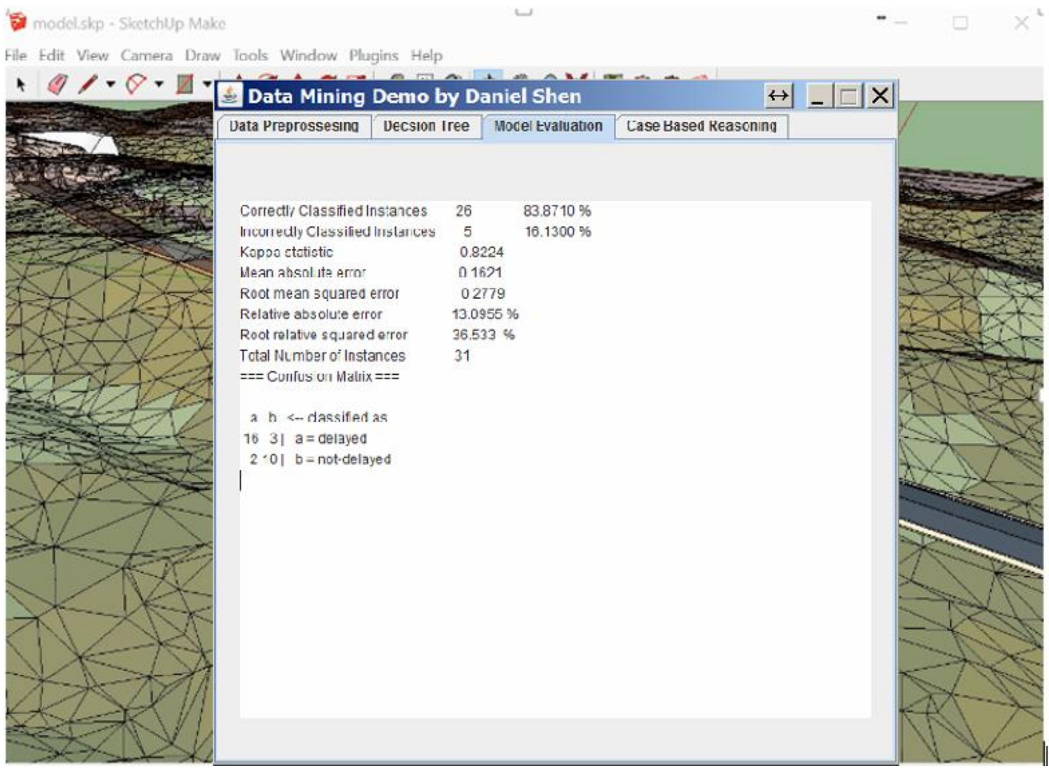


Figure 58: Model evaluation output

The final result is shown in Table 12. Several conclusions for this case study can be drawn.

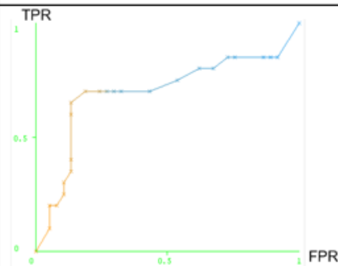
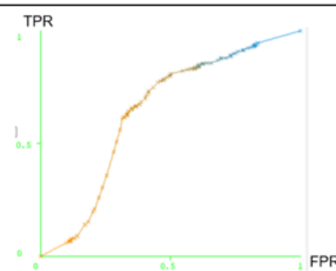
1) Decision Tree and CBR have similar overall predictive performance for a general project delay prediction (Average accuracy: 85.49>84.12).

2) When predicting delayed construction activity, Decision Tree is more sensitive (making less errors) than CBR (Average specificity: 87.09%>78.92%).

3） Compared to CBR, even though Decision tree is more sensitive to delayed activity, its ability to predict not-delayed activity doesn't lose much (AUC: 0.752>0.621).

A project manager should adopt decision tree model to predict highway construction delay according to the case study result.

Table 12: Comparison of model evaluation

	Decision Tree	Case Based Reasoning
Average Accuracy	85.49%	84.12%
Average Error Rate	14.51%	15.88%
Average Specificity	87.09%	78.92%
Average FNR	12.91%	21.08%
ROC Curve		
AUC	0.752	0.621

CHAPTER 5: CONCLUSION

This chapter presents the summary of research, review of contribution and discussion of future work.

5.1 Research Summary

This thesis describes an IFC based highway project knowledge discovery approach. Highway IFC data schema is developed to integrate interdisciplinary data into an intelligent format. A data warehouse prototype system is constructed to present the proposed approach. The system consists of IFC data import module, 3D representation module, database module and data mining module. Using this system, two machine learning models such as decision tree and case based reasoning are applied to historical IFC highway project data. A decision cost analysis is also introduced into the model evaluation process to help the highway project managers make a more practical decision.

The result of the case study shows that the highway project knowledge discovery approach can be highly automated by combining these technologies.

The proposed approach introduce IFC schema as the machine interpretable languages to describe whole life cycle highway project information. By integrating IFC, data warehouse, data mining, actually the whole system becomes a highway whole life cycle knowledge management system (Figure 59). Project history data are store in this system. It is highly automated, easy to manage, can minimize manual efforts, thus increases the efficiency.

What's more, IFC is object oriented, which implicates it is has the ability to describe any tangible object or non-tangible relationship. It shows the potential to handle multi domain information. By taking more interdisciplinary information into consideration, this approach can help user explore more complex patterns, make a more accurate prediction, gain more insights in highway project nature.

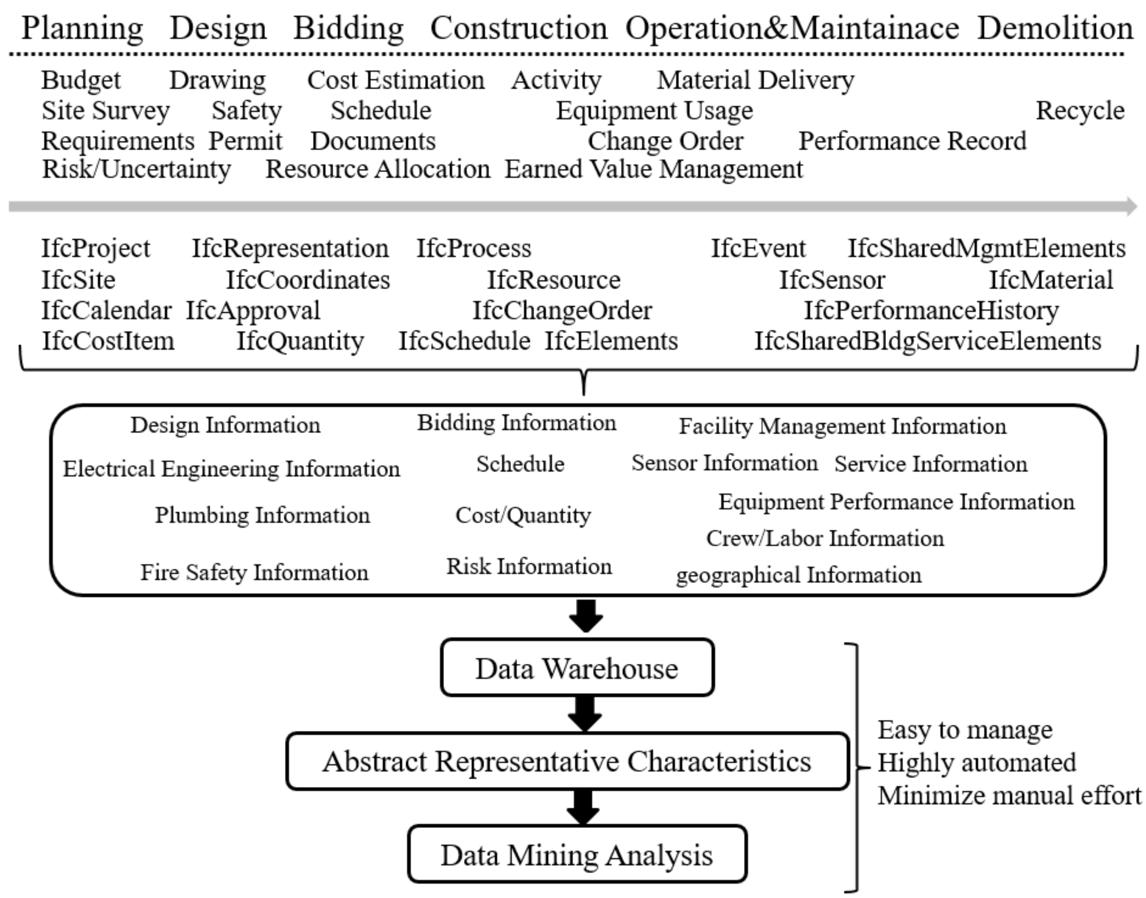


Figure 59: Highway whole lifecycle knowledge management system

5.2 Research Contribution

Research contributions are as follows:

- Proposed an intelligent machine interpretable IFC schema data structure for a highway project.

- Developed a highly automated framework for highway project knowledge discovery process to facilitate project manager's decision-making.
- Applied data mining methods to discover highway project construction delay knowledge
- Compared information entropy based on decision tree method performance and case similarity based CBR performance on highway project delay data.
- Introduced prediction/decision cost analysis in the highway construction activity delay prediction process.

5.3 Future Work

Future work is proposed in this section:

- Developing in-depth highway-specific IFC data schema.

  In this research, little study has been done on other highway project elements such as traffic, speed limit, slope protection and trench, etc. Full highway IFC data schema should cover a boarder range of highway design, planning and construction elements. Specific domain knowledge is needed to construct these corresponding IFC entities. It is important to work with professional highway engineers to create a plan for this IFC extension development.

- Applying more sophisticated knowledge discovery methods.

  Currently only decision tree and case based reasoning are applied. These two methods are not sufficient because highway project is becoming more complex. Another promising data mining method is artificial neural network, specifically, deep learning. Deep learning focuses on high-level abstraction of characteristics of data by setting up multiple processing layers. Deeper project nature can be explored by this neuron like structure.

- Integrating with distributed computing technique.

  Highway project involves multiple stakeholders. Each stakeholder manages part of the project files. These interdisciplinary project files are distributed in separate remote locations. Project participants usually modify and update the files simultaneously. Transferring data between these distributed servers is difficult. Building one centralized server to store and manage all the data is not practical. In order to solve this problem, distributed cloud computing system like Hadoop can be utilized to constitute a virtual server. It allows the distributed processing of large scale of project data across the clusters of servers. The project data is managed and analyzed in each cluster. And then all the results are transferred to a central server. The combination of the results efficiently gives a more quick result.

REFERENCES

Aamodt, A., & Plaza, E. (1994). CBR: foundational issues. *AI Communications*.

Ahmad, I. (2000). Data Warehousing in Construction Organizations. *Construction Congress*.

Akhavian, R. &. (2013). Knowledge-Based Simulation Modeling of Construction Fleet Operations Using Multimodal-Process Data Mining.

Autodesk. (2011). *Realizing the Benefits of BIM.*

Autodesk. (2012). *BIM for infrastructure: A vehicle for business transformation.*

Autodesk. (2012). *Making the Grade.*

Autodesk. (2012). *The Business Value of BIM for infrastructure.*

Batth, A. (2014). *Mining is The Process of Extracting Knowledge Hidden From Large Volumes of Raw Done.*

Bazjanac, V. (2008). *IFC BIM-based methodology for semi-automated building energy performance simulation.*

Brocklebank, J., & Dickey, D. A. (2003). *SAS for forecasting Time Series.*

Brockwell, P. J. (2002). ntroduction to Time Series and forecasting.

BuildingSmart. (2013). *industry Foundation Classes Release 4 (IFC4)*.

Bynum, P., Issa, R., & Olbina, S. (2013). Building information Modeling in Support of Sustainable Design and Construction. *Journal of Construction Engineering and Management*.

Chen, H., Zhang, Y., & Liu, F. (2009). The State Evaluation of Highway Traffic Flow Based on Observed Data. *international Conference on Transportation Engineering.*

Chou, J. C.-T. (2011). Optimizing the Prediction Accuracy of Concrete Compressive Strength Based on a Comparison of Data-Mining Techniques.

Chris Rygielskia, J.-C. W. (2002). Data mining techniques for customer relationship management.

Coufal, D., & Turunen, E. (2003). Short term prediction of highway travel time using data mining and neuro-fuzzy methods.

Dayi, Q. H. (2008). Application of Data Warehouse in intelligent Public Transportation System.

Devlin, B. (1996). Data Warehouse: From Architecture to Implementation.

DK Smith, M. T. (2009). *Building information modeling: a strategic implementation guide for architects, engineers, constructors, and real estate asset managers.*

Eastman, C. (1974). An Outline of the Building Description System.

Eastman, C. (2006). *IFC Review*.

Fayyad. (1996). *Advances in knowledge discovery and data mining.*

Froese, T. (1999). industry Foundation Classes for Project Management.

G. Piatetsky-Shapiro, W. F. (1991). Knowledge discovery in databases.

Gantz, J., & Reinsel, D. (2011). *Extracting Value from Chaos.*

Gantz, J., & Reinsel, D. (2012). *Big Data Bigger Digital Shadows and Biggest Growth in the Far East.*

Gantz, J., & Reinsel, D. (2012). Digital Shadows and Biggest Growth in Far East.

Gene Bellinger, D. C. (2011). Data, information, Knowledge, and Wisdom .

Hammad, A., AbouRizk, S., & Mohamed, Y. (2014). Application of KDD Techniques to Extract Useful Knowledge from Labor Resources Data in industrial Construction Projects. *J. Manage. Eng.*

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques.*

Hanke, J. E., & Wichern, D. W. (2009). *Bussiness forecasting.*

Hergunsel, M. F. (2011). *Benefits of Building information Modeling for Construction Managers and BIM Based Scheduling.*

Hongqin Fan, S. A. (2008). Assessing Residual Value of Heavy Construction Equipment Using Predictive Data Mining Model. *Journal of Computing in Civil Engineering*.

Inmon, W. (1996). The data warehouse and data mining.

ISO. (2004). *ISO 10303-11:2004.*

Jayasudha, & Chandrasekar, C. (2009). An Overview of Data Mining in Road Traffic and Accident Analysis. *Journal of Computer Applications*.

Jian Wang, M. G. (2007). Hybrid Data Mining/Genetic Shredding Algorithm for Reliability Assessment of Structural Systems. *Journal of Structural Engineering*.

K.W. Chau, Y. C. (2002). Application of data warehouse and Decision Support System in construction management. *Automation in Construction*.

Karina, G., Miquel, S.-M., & Víctor, C. (2010). Choosing the Right Data Mining Technique:Classification of Methods and intelligent Recommendation.

Kayhaniana, M., Suverkroppb, C., Rubyb, A., & Tsayc, K. (2006). Characterization and prediction of highway runoff constituent event mean concentration. *Journal of Environmental Management*.

Khandare Manish A., V. G. (2012). Project Duration forecasting Using Earned Value Method and Time Series.

Kim. (2014). Highway Alignment Construction Comparison Using Object-Oriented 3D Visualization Modeling. *Journal of Construction Engineering and Management*.

Kim, H. A. (2011). 3-D Temporary Facility Visualization Using BIM (Building information Modeling) Technology. *Journal of Construction Engineering and Project Managemen*.

Kim, H., & anderson, K. (2013). Energy Modeling System Using Building information Modeling Open Standards. *Journal of Computing in Civil Engineering*.

Kimball, R. (1998). *The data warehouse lifecycle toolkit: expert methods for designing, developing, and deploying data warehouses.*

Kolodner, J. L. (1993). An introduction to Case-Based Reasoning. *Artificial intelligence Review*.

Kreider, R., Messner, J., & Dubler, C. (2010). Determining the Frequency and Impact of Applying BIM for Different Purposes on Building Projects.

Kumar, D., & Bhardwa, D. (2011). Rise of Data Mining: Current and Future Application Areas. *international Journal of Computer Science Issues*.

Laakso, M., & Kiviniemi, A. (2012). The IFC Standard - A Review of History, Development, and Standardization. *Journal of information Technology in Construction*.

Lewis, M. P., Jeong, “. H., & Woldesenbet, A. (2013). *Data and information integration Framework for High Way Project Decision Making.*

Li Wang, F. L. (2013). Knowledge Discovery of Spatial Conflict Resolution Philosophies in BIM enabled MEP Design Coordination using Data Mining Techniques: a Proof-of-Concept.

Liberty univ. inc. (2015). *Data Modeling.* Retrieved from www.liberty.edu/media/1414/%5B6330%5DERDDataModeling.pdf

Liebowitz, M. (2003). A set of frameworks to aid the project manager in conceptualizing and implementing knowledge management initiatives.

Lin Yu, Z. P. (2011). Research on Urban Expressway Traffic Data Mining System and Its Application.

Li-Yen Chang, ,. W.-C. (2005). Data mining of tree-based models to analyze freeway accident frequency.

López, B. (2013). Case-Based Reasoning: A Concise introduction.

Ma, L., Luo, H. B., & Chen, H. R. (2013). Safety risk analysis based on a geotechnical instrumentation data warehouse in metrotunnel project. *Automation in Constrution*.

Marko Grobelnik, D. M. (2005). *Automated Knowledge Discovery in Advanced Knowledge.*

McGraw-Hill. (2007). *Interoperablity in Construction industry*.

McGraw-Hill Construction. (2012). *The Business Value of BIM in infrastructure.*

McGraw-Hill Construction. (2012). *The Business Value of BIM in North America.* Retrieved from images.autodesk.com/adsk/files/mhc_business_value_of_bim_in_north_america_2007-2012_smr.pdf

Miaou, S.-P., & Lum, H. (1993). Modeling vehicle accidents and highway geometric design relationships. *Accident Analysis & Prevention*.

NBIMS. (2015). *Frequently Asked Questions About the National BIM Standard-United States™.*

Noland, R. B., & Cowart, W. A. (2000). Analysis of Metropolitan Highway Capacity and the growth in vehicle miles of travel. *Transportation*.

Pantic, M. (2000). *introduction to Machine Learning & Case Based Reasoning.*

Park, T., & Kim, H. (2013). A data warehouse-based decision support system for sewer in frastructure management. *Automation in Construction*.

Persaud. (2001). STATISTICAL METHODS IN HIGHWAY SAFETY ANALYSIS.

Pniewski, V. (2011). *Building information Modeling: interoperability Issues.*

Qian, Z., & Qing, X. (2009). The Study on Data Warehouse Modelling and OLAP for Highway Management.

Qu Dayi, J. P. (2008). Application of Data Warehouse in intelligent Public Transportation System.

S.W. Moona, J. K. (2007). Effectiveness of OLAP-based cost data management in construction cost estimate. *Automation in Construction*.

Sacks, R., & Pikas, E. (2013). Building information Modeling Education for Construction Engineering and Management. I: industry Requirements, State of the Art, and Gap Analysis. *ournal of Construction Engineering and Management*.

SAS. (2011). *Decision Trees-What are they.*

Shannon Barnes, D. C.-L. (2009). BIM-enabled integrated Optimization tool for LEED Decisions. *Computing in Civil Engineering*.

Shu-Hsien Liao, P.-H. C.-Y. (2012). Data mining techniques and applications – A decade review from 2000 to 2011. *Expert Systems with Applications*.

Simoff, S. J., & Maher, M. L. (1998). DATA MINING IN HYPERMEDIA CASE LIBRARIES.

Simsion, & Graeme. (2007). *Data Modeling Theory and Practice.*

Soibelman, L., & Kim, H. (2002). Data Preparation Process for Construction Knowledge Generation through Knowledge Discovery in Databases. *JOURNAL OF COMPUTING IN CIVIL ENGINEERING*.

Tah, J., Carr, V., & Howes, R. (1999). information modelling for case-based construction planning of highway bridge projects. *Advances in Engineering Software*.

Tarapanoff, K., Quoniam, L., Júnior, R. H., & Alvares, L. (2001). intelligence obtained by applying data mining to a database of French theses on the subject of Brazil. *information Research*.

Thammasak Rujirayanyonga, J. J. (2006). A project-oriented data warehouse for construction. *Automation in Construction*.

Tho Manh Nguyen, A. M. (2005). *Data Warehousing and Knowledge Discovery:A Chronological View of Research Challenges.*

Usama Fayyad, G. P.-S. (1997). *From Data Mining to Knowledge Discovery in Databases.*

Wang, J., & Ghosn, M. (2006). Hybrid Data Mining/Genetic Shredding Algorithm for Reliability Assessment of Structural Systems.

Washington, S., Karlaftis, M., & Mannering, F. (2012). *Statistical and econometric methods for transportation data analysis.*

Watson, I. (1999). Case-based reasoning is a methodology not a technology. *Knowledge based systems*.

William J. Frawley, G. P.-S. (1992). Knowledge Discovery in Databases: An Overview. *AI Magazine*.

Witten, I. H., Frank, E., & A.Hall, M. (2011). *Data Mining: Practical Machnie Learning tools and Techniques.*

Woldesenbet, A. K. (2014). *Highway infrastructure Data and information integration & Assessment Framework: A Data-Driven Decision-Making Approach.*

Xu, T., Sun, X., He, Y., & Xie, C. (2009). Highway Accidents Analysis with Panel Data Model.

Zhang, J. (2002). Application of Data Warehouse and Data Mining in Construction Management.

Zhou Qian, X. Q. (2009). The study on data warehouse modelling and OLAP for highway management.

Zhou, G., Wang, L., Wang, D., & Reichle, S. (2009). integration of GIS and Data Mining Technology to Enhance the Pavement Management Decision Making. *Journal of Transportation Engineering*.

Zhu, Y., Issa, R., & Cox. (2011). Web-Based Construction Document. *Journal of Computing*.

APPENDIX A: PROPOSED HIGHWAY IFC SCHEMA EXTENTION

The proposed highway IFC schema defines basic spatial structure (road, bridge and tunnel) and their corresponding construction components. The proposed highway IFC entities together with existing BIM IFC entities provide a set of data schema for highway information storage and exchange.

This research shows that IFC has the potential to cover highway domain information. This is one of the initial efforts to expand IFC in highway industry. In the future, more sophisticated development is needed.

The spatial structures and construction components designed in this research are listed in the table below:

Road	Bridge	Tunnel
IfcRoad	IfcBridge	IfcTunnel
IfcRoadSurfaceCourse	IfcBridgeConnectingSlab	IfcTunnelUpperSlab
IfcSolidLine	IfcBridgeSupport	IfcTunnelPavement
IfcBrokenLine	IfcBridgeWallParapet	IfcTunnelSupportingWall
IfcRoadBase	IfcBridgeWallFrame	IfcTunnelSlab
IfcRoadBaseCourse	IfcBridgeWingWall	
IfcRoadSubbase	IfcBridgeProtectionWall	
IfcRoadBed	IfcBridgeRoadSeparator	
IfcRoadEmbankment	IfcBridgePavement	
IfcRoadFoundation	IfcBridgeUpperPlate	
IfcRoadCurb	IfcBridgeSteelBeam	
	IfcBridgeBracingBeam	
	IfcBridgeFoundation	

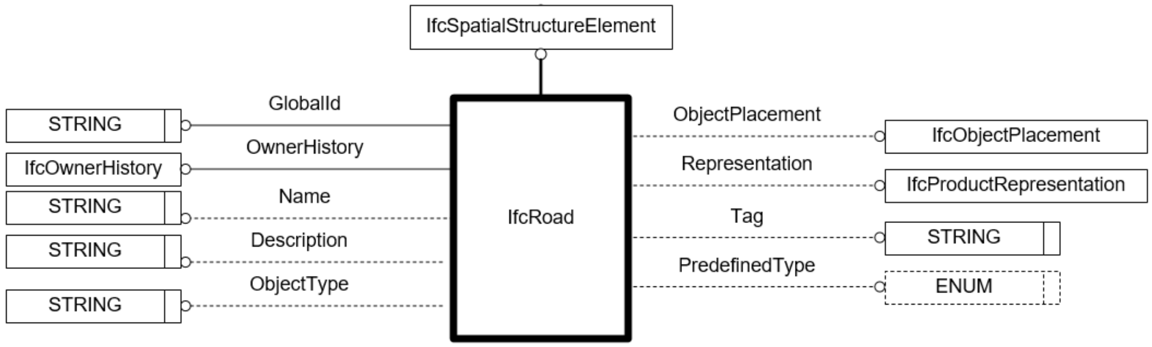
The detailed IFC specification, EXPRESS diagrams and construction components are listed as follows:

1) IfcRoad

- EXPRESS Specification:

```
ENTITY IfcRoad
SUBTYPE OF (    IfcSpatialStuctureElement);
WHERE
WR1        :        SIZEOF (QUERY(temp <*
SELF\IfcObjectDefinition.HasAssociations |
'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN
TYPEOF(temp) )) <= 1;    :        OPTIONAL IfcLengthMeasure;
END_ENTITY;
```

- EXPRESS diagram:



- Inheritance graph:

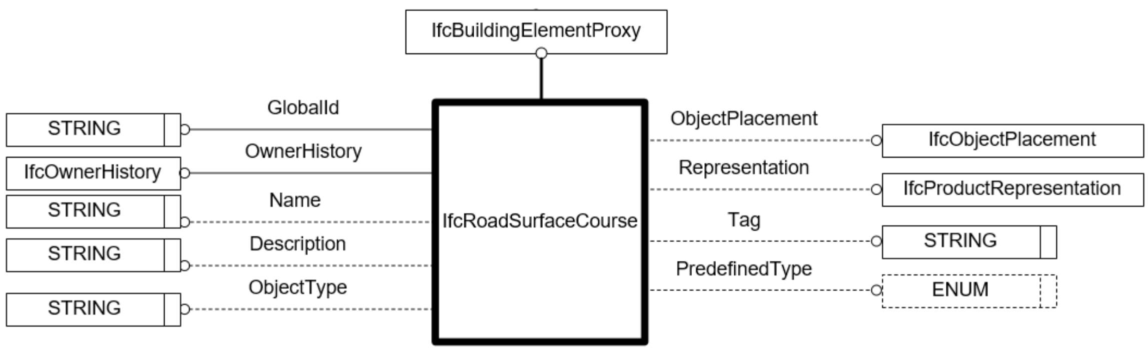
ENTITY IfcRoad;
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcSpatialStructureElement;
LongName : OPTIONAL IfcLabel;
CompositionType : IfcElementCompositionEnum;
INVERSE
ReferencesElements : SET OF IfcRelReferencedInSpatialStructure FOR RelatingStructure;
ServicedBySystems : SET OF IfcRelServicesBuildings FOR RelatedBuildings;
ContainsElements : SET OF IfcRelContainedInSpatialStructure FOR RelatingStructure;
ENTITY IfcSpatialStructureElement;
ENTITY IfcRoad;
END_ENTITY;

2) IfcRoadSurfaceCourse

- EXPRESS Specification:

ENTITY IfcRoadSurfaceCourse
SUBTYPE OF (IfcBuildingElementProxy);
WHERE
WR1 : SIZEOF (QUERY(temp <* USEDIN(SELF, 'IFCKERNEL.IFCRELASSOCIATES.RELATEDOBJECTS') | ('IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN TYPEOF(temp)) AND ('IFCMATERIALRESOURCE.IFCMATERIALLAYERSETUSAGE' IN TYPEOF(temp.RelatingMaterial)))) = 1;
END_ENTITY;

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcRoadSurfaceCourse;
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
Tag : OPTIONAL IfcIdentifier;

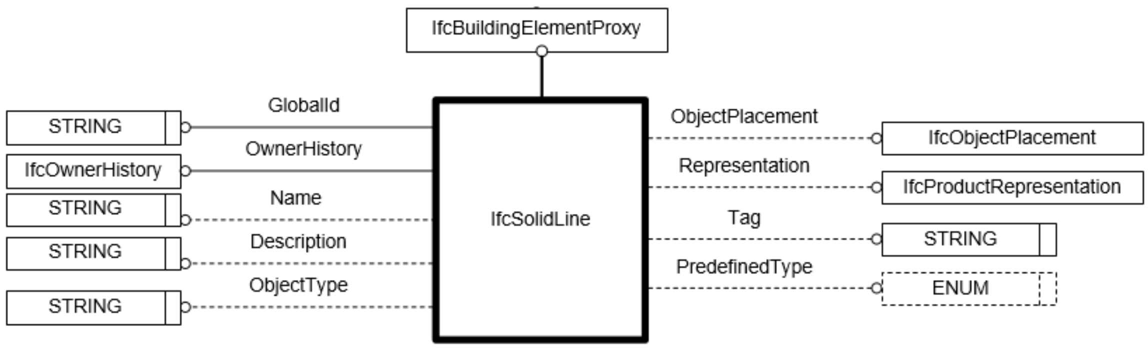
INVERSE
HasStructuralMember : SET OF IfcRelConnectsStructuralElement FOR RelatingElement;
ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;
HasCoverings : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;
END_ENTITY;

3) IfcSolidLine

- EXPRESS Specification:

ENTITY IfcSolidLine
SUBTYPE OF (IfcBuildingElementProxy);
WHERE
WR1 : SIZEOF (QUERY(temp <* SELF\IfcObjectDefinition.HasAssociations | 'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN TYPEOF(temp))) <= 1;
END_ENTITY;

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcSolidLine;
ENTITY IfcRoot;

GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
Tag : OPTIONAL IfcIdentifier;
INVERSE
HasStructuralMember : SET OF IfcRelConnectsStructuralElement FOR RelatingElement;
FillsVoids : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;
HasCoverings : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
HasOpenings : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;

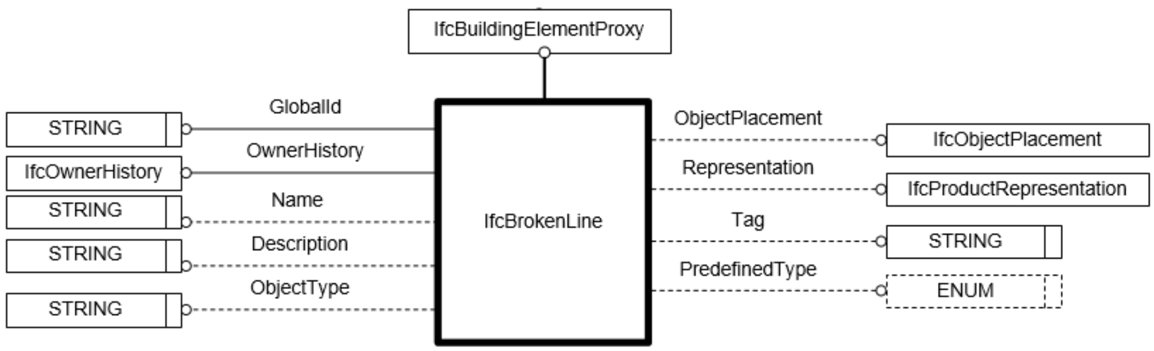
ENTITY IfcSolidLine;
END_ENTITY;

4) IfcBrokenLine

- EXPRESS Specification:

ENTITY IfcBrokenLine
SUBTYPE OF (IfcBuildingElementProxy);
WHERE
WR1 : SIZEOF (QUERY(temp <*
SELF\IfcObjectDefinition.HasAssociations |
'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN
TYPEOF(temp))) <= 1;
END_ENTITY;

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcBrokenLine;
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;

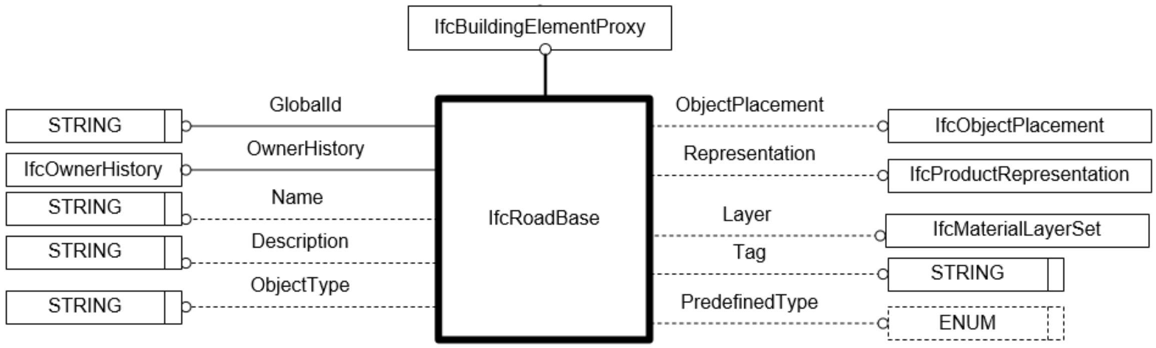
Tag : OPTIONAL IfcIdentifier;
INVERSE
HasStructuralMember : SET OF IfcRelConnectsStructuralElement FOR RelatingElement;
FillsVoids : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;
HasCoverings : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
HasOpenings : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;
ENTITY IfcBrokenLine;
END_ENTITY;

5) IfcRoadBase

- EXPRESS Specification:

ENTITY IfcRoadBase
SUBTYPE OF (IfcBuildingElementProxy);
WHERE
WR1 : SIZEOF (QUERY(temp <* SELF\IfcObjectDefinition.HasAssociations | 'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN TYPEOF(temp))) <= 1;
END_ENTITY;

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcRoadBase;
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
Tag : OPTIONAL IfcIdentifier;
INVERSE
HasStructuralMember : SET OF IfcRelConnectsStructuralElement FOR RelatingElement;
FillsVoids : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;
HasCoverings : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;

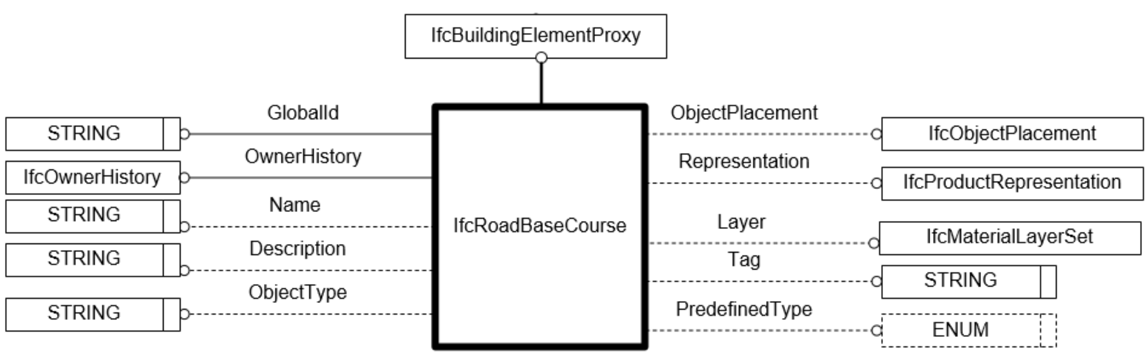
HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
HasOpenings : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;
ENTITY IfcRoadBase;
END_ENTITY;

6) IfcRoadBaseCourse

- EXPRESS Specification:

ENTITY IfcRoadBaseCourse
SUBTYPE OF (IfcBuildingElementProxy);
WHERE
WR1 : SIZEOF (QUERY(temp <* SELF\IfcObjectDefinition.HasAssociations | 'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN TYPEOF(temp))) <= 1;
END_ENTITY;

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcRoadBaseCourse;
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;

IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
Tag : OPTIONAL IfcIdentifier;
INVERSE
HasStructuralMember : SET OF IfcRelConnectsStructuralElement FOR RelatingElement;
FillsVoids : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;
HasCoverings : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
HasOpenings : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;
ENTITY IfcRoadBaseCourse;
END_ENTITY;

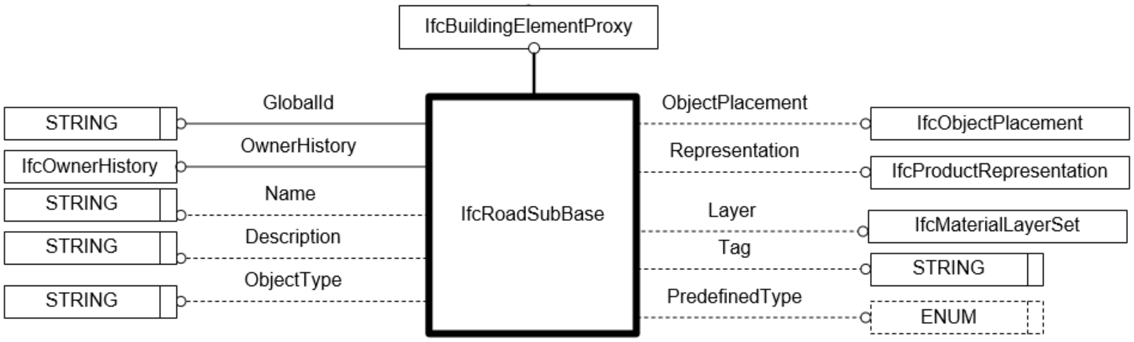
7) IfcRoadSubBase

- EXPRESS Specification:

ENTITY IfcRoadSubBase
SUBTYPE OF (IfcBuildingElementProxy);
WHERE

WR1 : SIZEOF (QUERY(temp <*
SELF\IfcObjectDefinition.HasAssociations |
'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN
TYPEOF(temp))) <= 1;
END_ENTITY;

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcRoadSubBase;
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
Tag : OPTIONAL IfcIdentifier;
INVERSE
HasStructuralMember : SET OF IfcRelConnectsStructuralElement FOR
RelatingElement;
FillsVoids : SET [0:1] OF IfcRelFillsElement FOR
RelatedBuildingElement;
ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;

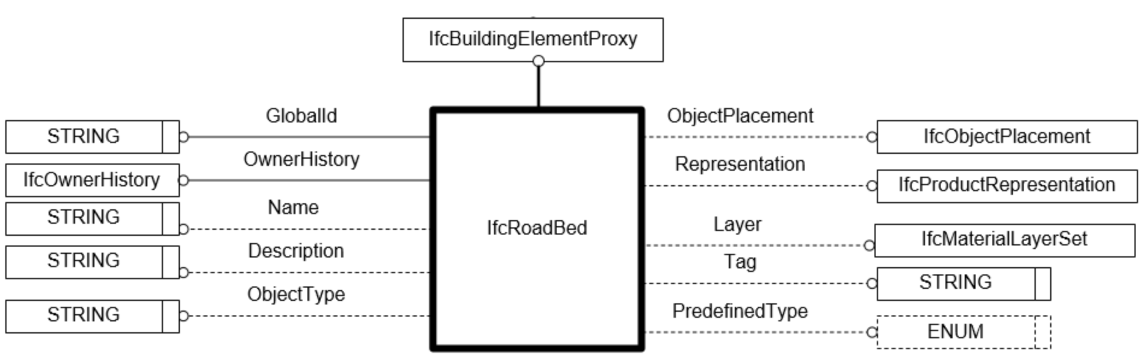
HasCoverings : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
HasOpenings : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;
ENTITY IfcRoadSubBase;
END_ENTITY;

8) IfcRoadbed

- EXPRESS Specification:

ENTITY IfcRoadbed
SUBTYPE OF (IfcBuildingElementProxy);
WHERE
WR1 : SIZEOF (QUERY(temp <* SELF\IfcObjectDefinition.HasAssociations | 'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN TYPEOF(temp))) <= 1;
END_ENTITY;

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcRoadBed;
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;

OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
Tag : OPTIONAL IfcIdentifier;
INVERSE
HasStructuralMember : SET OF IfcRelConnectsStructuralElement FOR RelatingElement;
FillsVoids : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;
HasCoverings : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
HasOpenings : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;
ENTITY IfcRoadBed;

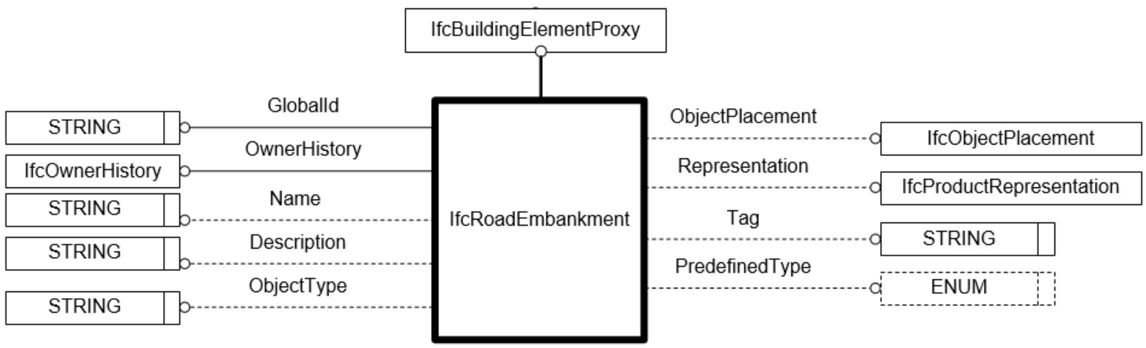
END_ENTITY;

9) IfcRoadEmbankment

- EXPRESS Specification:

ENTITY IfcRoadbed
SUBTYPE OF (IfcBuildingElementProxy);
WHERE
WR1 : SIZEOF (QUERY(temp <* SELF\IfcObjectDefinition.HasAssociations | 'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN TYPEOF(temp))) <= 1;
END_ENTITY;

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcRoadEmbankment;
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
Tag : OPTIONAL IfcIdentifier;
INVERSE

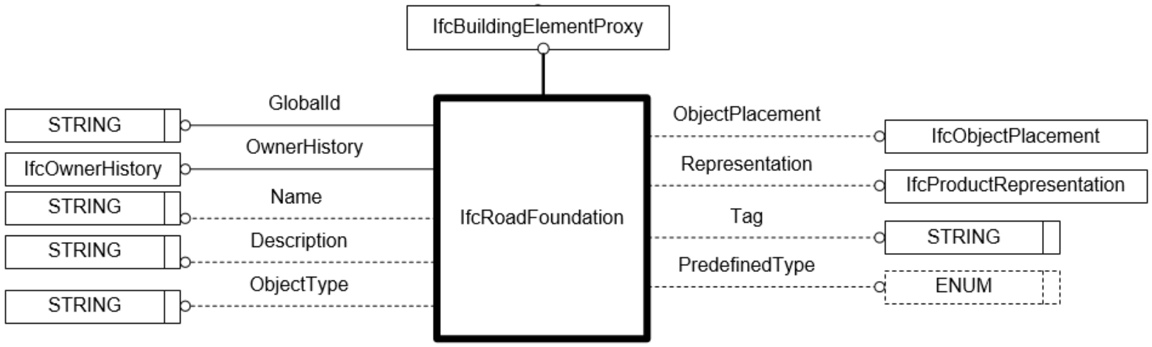
HasStructuralMember : SET OF IfcRelConnectsStructuralElement FOR RelatingElement;
FillsVoids : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;
HasCoverings : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
HasOpenings : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;
ENTITY IfcRoadEmbankment;
END_ENTITY;

10) IfcRoadFoundation

- EXPRESS Specification:

```
ENTITY IfcRoadFoundation
SUBTYPE OF (    IfcBuildingElementProxy);
WHERE
WR1    :    SIZEOF (QUERY(temp <*
SELF\IfcObjectDefinition.HasAssociations |
'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN
TYPEOF(temp) )) <= 1;
END_ENTITY;
```

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcRoadFoundation;
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
Tag : OPTIONAL IfcIdentifier;
INVERSE
HasStructuralMember : SET OF IfcRelConnectsStructuralElement FOR RelatingElement;
FillsVoids : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;
HasCoverings : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;

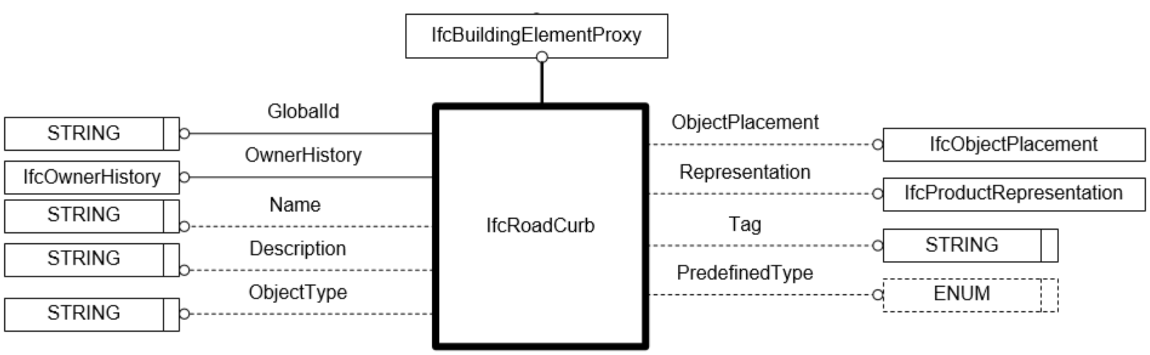
HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
HasOpenings : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;
ENTITY IfcRoadFoundation;
END_ENTITY;

11) IfcRoadCurb

- EXPRESS Specification:

ENTITY IfcRoadCurb
SUBTYPE OF (IfcBuildingElementProxy);
WHERE
WR1 : SIZEOF (QUERY(temp <* SELF\IfcObjectDefinition.HasAssociations | 'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN TYPEOF(temp))) <= 1;
END_ENTITY;

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcRoadCurb;
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;

IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
Tag : OPTIONAL IfcIdentifier;
INVERSE
HasStructuralMember : SET OF IfcRelConnectsStructuralElement FOR RelatingElement;
FillsVoids : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;
HasCoverings : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
HasOpenings : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;
ENTITY IfcRoadCurb;
END_ENTITY;

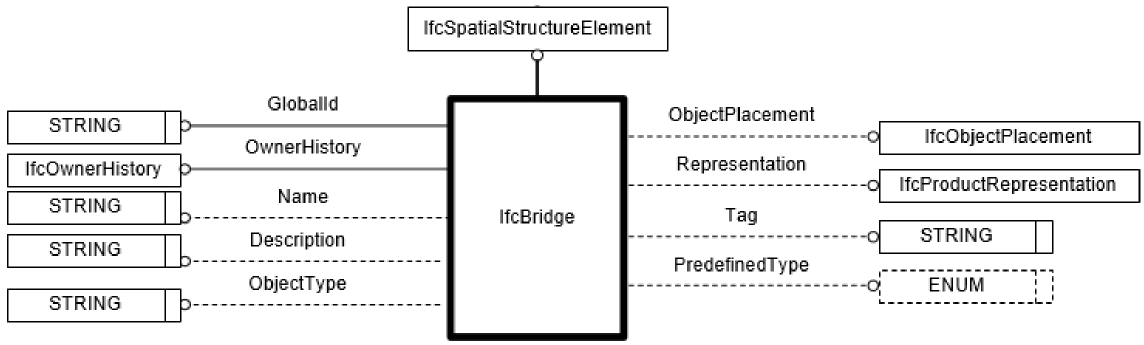
12) IfcBridge

- EXPRESS Specification:

ENTITY IfcBridge
SUBTYPE OF (IfcSpatialStuctureElement);

WHERE
WR1 : SIZEOF (QUERY(temp <* SELF\IfcObjectDefinition.HasAssociations | 'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN TYPEOF(temp))) <= 1;
END_ENTITY;

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcBridge;
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcSpatialStructureElement;
LongName : OPTIONAL IfcLabel;
CompositionType : IfcElementCompositionEnum;
INVERSE
ReferencesElements : SET OF IfcRelReferencedInSpatialStructure FOR RelatingStructure;
ServicedBySystems : SET OF IfcRelServicesBuildings FOR RelatedBuildings;

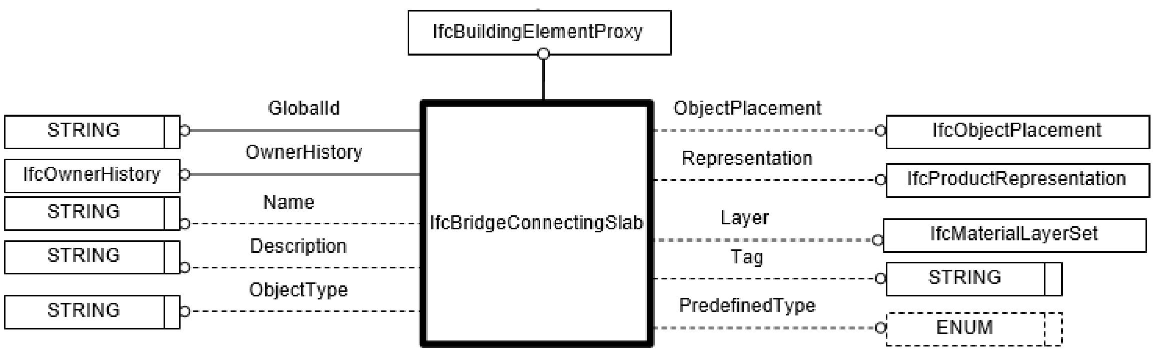
ContainsElements : SET OF IfcRelContainedInSpatialStructure FOR RelatingStructure;
ENTITY IfcSpatialStuctureElement;
ENTITY IfcBridges;
END_ENTITY;

13) IfcBridgeConnectingSlab

- EXPRESS Specification:

ENTITY IfcBridgeConnectingSlab
SUBTYPE OF (IfcBuildingElementProxy);
PredefinedType : OPTIONAL IfcSlabTypeEnum;
WHERE
WR61 : NOT(EXISTS(PredefinedType)) OR (PredefinedType <> IfcSlabTypeEnum.USERDEFINED) OR ((PredefinedType = IfcSlabTypeEnum.USERDEFINED) AND EXISTS (SELF\IfcObject.ObjectType));
END_ENTITY;

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcBridgeConnectingSlab;
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;

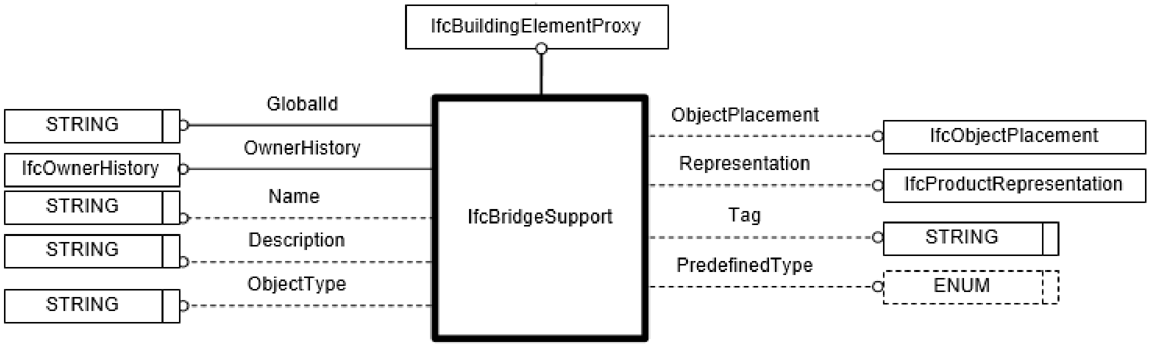
```
ObjectPlacement	:	OPTIONAL IfcObjectPlacement;
Representation :	OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy :	SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
Tag	:	OPTIONAL IfcIdentifier;
INVERSE
HasStructuralMember :	SET OF IfcRelConnectsStructuralElement FOR RelatingElement;
FillsVoids	:	SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
ConnectedTo	:	SET OF IfcRelConnectsElements FOR RelatingElement;
HasCoverings	:	SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections	:	SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures	:	SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
HasPorts	:	SET OF IfcRelConnectsPortToElement FOR RelatedElement;
HasOpenings	:	SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization	:	SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries	:	SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom	:	SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure	:	SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;
ENTITY IfcBridgeConnectingSlab;
PredefinedType	:	OPTIONAL IfcSlabTypeEnum;
END_ENTITY;
```

14) IfcBridgeSupport

- EXPRESS Specification:

```
ENTITY IfcBridgeSupport
SUBTYPE OF (	IfcBuildingElementProxy);
WHERE
WR1	:	SIZEOF (QUERY(temp <* SELF\IfcObjectDefinition.HasAssociations | 'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN TYPEOF(temp) )) <= 1;
END_ENTITY;
```

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcBridgeSupport;
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
Tag : OPTIONAL IfcIdentifier;
INVERSE
HasStructuralMember : SET OF IfcRelConnectsStructuralElement FOR RelatingElement;
FillsVoids : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;
HasCoverings : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;

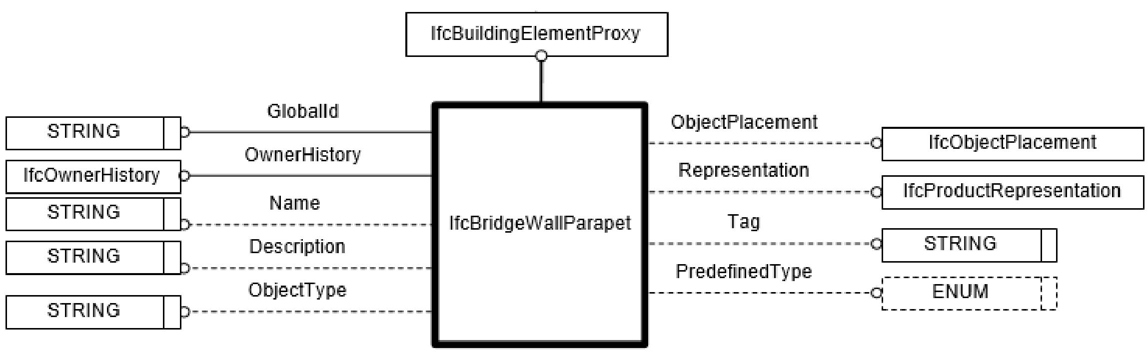
HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
HasOpenings : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;
ENTITY IfcBridgeSupport;
END_ENTITY;

15) IfcBridgeWallParapet

- EXPRESS Specification:

ENTITY IfcBridgeWallParapet
SUBTYPE OF (IfcBuildingElementProxy);
WHERE
WR1 : SIZEOF (QUERY(temp <* SELF\IfcObjectDefinition.HasAssociations | 'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN TYPEOF(temp))) <= 1;
END_ENTITY;

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcBridgeWallParapet
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE

HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
Tag : OPTIONAL IfcIdentifier;
INVERSE
HasStructuralMember : SET OF IfcRelConnectsStructuralElement FOR RelatingElement;
FillsVoids : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;
HasCoverings : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
HasOpenings : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;
ENTITY IfcBridgeWallParapet;
END_ENTITY;

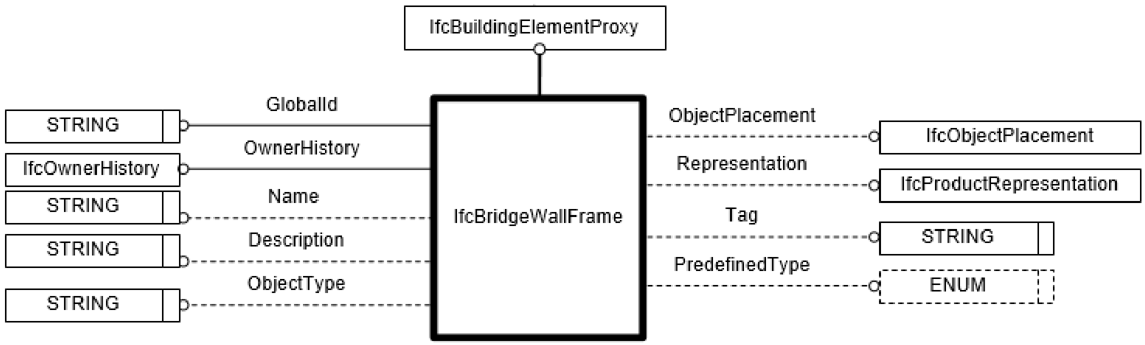
16) IfcBridgeWallFrame

- EXPRESS Specification:

ENTITY IfcBridgeWallFrame

SUBTYPE OF (IfcBuildingElementProxy);
WHERE
WR1 : SIZEOF (QUERY(temp <*
SELF\IfcObjectDefinition.HasAssociations |
'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN
TYPEOF(temp))) <= 1;
END_ENTITY;

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcBridgeWallFrame
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
Tag : OPTIONAL IfcIdentifier;
INVERSE
HasStructuralMember : SET OF IfcRelConnectsStructuralElement FOR
RelatingElement;

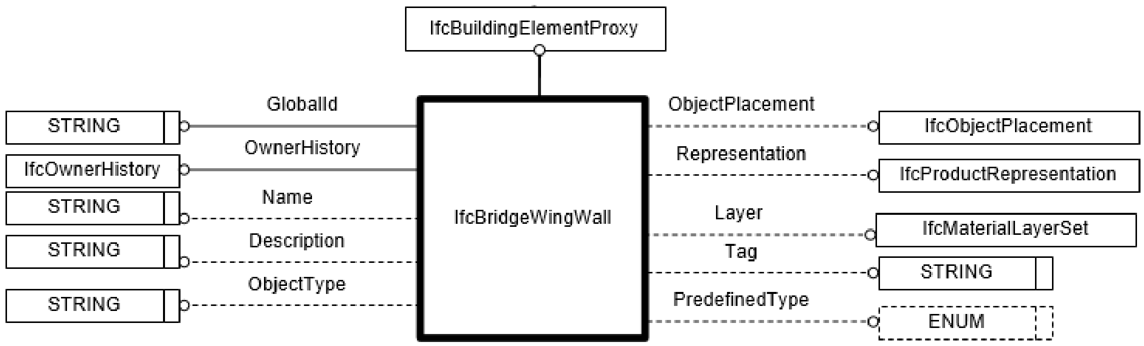
FillsVoids : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;
HasCoverings : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
HasOpenings : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;
ENTITY IfcBridgeWallFrame;
END_ENTITY;

17) IfcBridgeWingWall

- EXPRESS Specification:

ENTITY IfcBridgeWingWall
SUBTYPE OF (IfcBuildingElementProxy);
WHERE
WR1 : SIZEOF (QUERY(temp <* SELF\IfcObjectDefinition.HasAssociations | 'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN TYPEOF(temp))) <= 1;
END_ENTITY;

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcBridgeWingWall;
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
Tag : OPTIONAL IfcIdentifier;
INVERSE
HasStructuralMember : SET OF IfcRelConnectsStructuralElement FOR RelatingElement;
FillsVoids : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;
HasCoverings : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;

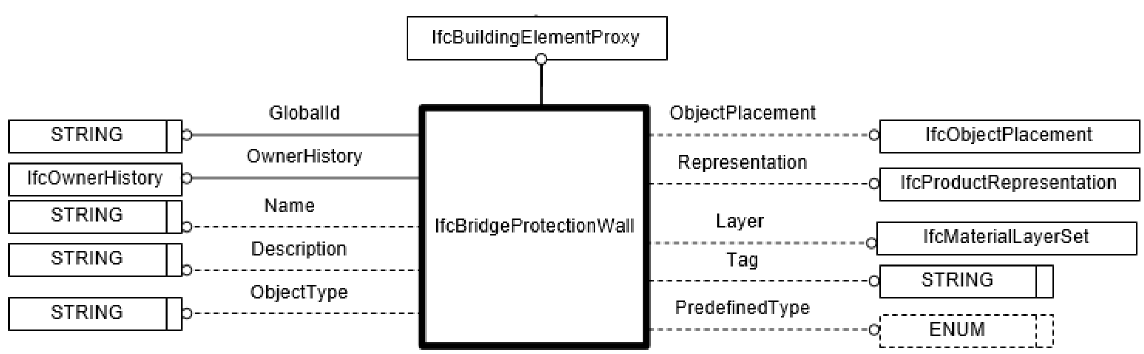
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
HasOpenings : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;
ENTITY IfcBridgeWingWall;

18) IfcBridgeProtectionWall

- EXPRESS Specification:

ENTITY IfcBridgeProtectionWall
SUBTYPE OF (IfcBuildingElementProxy);
WHERE
WR1 : SIZEOF (QUERY(temp <* SELF\IfcObjectDefinition.HasAssociations | 'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN TYPEOF(temp))) <= 1;
END_ENTITY;

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcBridgeProtectionWall;
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;

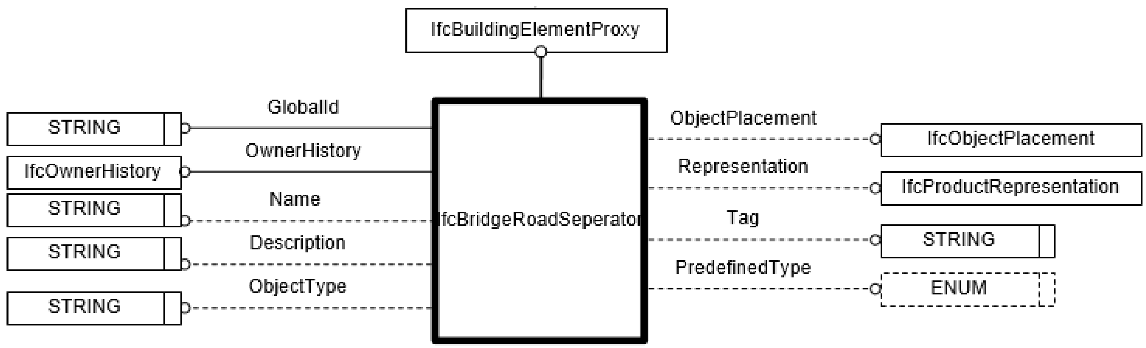
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
Tag : OPTIONAL IfcIdentifier;
INVERSE
HasStructuralMember : SET OF IfcRelConnectsStructuralElement FOR RelatingElement;
FillsVoids : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;
HasCoverings : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
HasOpenings : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;
ENTITY IfcBridgeProtectionWall;

19) IfcBridgeRoadSeperator

- EXPRESS Specification:

ENTITY IfcBridgeRoadSeperator
SUBTYPE OF (IfcBuildingElementProxy);
WHERE
WR1 : SIZEOF (QUERY(temp <* SELF\IfcObjectDefinition.HasAssociations | 'IFCPRODUCTEXTENSION.IFCRELASSOCIATESMATERIAL' IN TYPEOF(temp))) <= 1;
END_ENTITY;

- EXPRESS diagram:



- Inheritance graph:

ENTITY IfcBridgeRoadSeperator;
ENTITY IfcRoot;
GlobalId : IfcGloballyUniqueId;
OwnerHistory : IfcOwnerHistory;
Name : OPTIONAL IfcLabel;
Description : OPTIONAL IfcText;
ENTITY IfcObjectDefinition;
INVERSE
HasAssignments : SET OF IfcRelAssigns FOR RelatedObjects;
IsDecomposedBy : SET OF IfcRelDecomposes FOR RelatingObject;
Decomposes : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
HasAssociations : SET OF IfcRelAssociates FOR RelatedObjects;
ENTITY IfcObject;
ObjectType : OPTIONAL IfcLabel;
INVERSE
IsDefinedBy : SET OF IfcRelDefines FOR RelatedObjects;
ENTITY IfcProduct;
ObjectPlacement : OPTIONAL IfcObjectPlacement;
Representation : OPTIONAL IfcProductRepresentation;
INVERSE
ReferencedBy : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
ENTITY IfcElement;
Tag : OPTIONAL IfcIdentifier;

INVERSE
HasStructuralMember : SET OF IfcRelConnectsStructuralElement FOR RelatingElement;
FillsVoids : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
ConnectedTo : SET OF IfcRelConnectsElements FOR RelatingElement;
HasCoverings : SET OF IfcRelCoversBldgElements FOR RelatingBuildingElement;
HasProjections : SET OF IfcRelProjectsElement FOR RelatingElement;
ReferencedInStructures : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
HasPorts : SET OF IfcRelConnectsPortToElement FOR RelatedElement;
HasOpenings : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
IsConnectionRealization : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
ProvidesBoundaries : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
ConnectedFrom : SET OF IfcRelConnectsElements FOR RelatedElement;
ContainedInStructure : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
ENTITY IfcBuildingElementProxy;
ENTITY IfcBridgeRoadSeperator;