An Intelligent Decision Support System for Business IT Security Strategy

Yuanchen Xu

Faculty of Computing Engineering and Media

May 2021

A thesis submitted in fulfilment of the University's requirements for the Degree of Doctor of Philosophy

Abstract

Cyber threat intelligence (CTI) is an emerging approach to improve cyber security of business IT environment. It has information of an affected business IT context. CTI sharing tools are available for subscribers, and CTI feeds are increasingly available. If another business IT context is similar to a CTI feed context, the threat described in the CTI feed might also take place in the business IT context. Businesses can take proactive defensive actions if relevant CTI is identified. However, a challenge is how to develop an effective connection strategy for CTI onto business IT contexts. Businesses are still insufficiently using CTI because not all of them have sufficient knowledge from domain experts. Moreover, business IT contexts vary over time. When the business IT contextual states have changed, the relevant CTI might be no longer appropriate and applicable. Another challenge is how a connection strategy has the ability to adapt to the business IT contextual changes.

To fill the gap, in this Ph.D project, a dynamic connection strategy for CTI onto business IT contexts is proposed and the strategy is instantiated to be a dynamic connection rule assembly system. The system can identify relevant CTI for a business IT context and can modify its internal configurations and structures to adapt to the business IT contextual changes.

This thesis introduces the system development phases from design to delivery, and the contributions to knowledge are explained as follows.

A hybrid representation of the dynamic connection strategy is proposed to generalise and interpret the problem domain and the system development. The representation uses selected computational intelligence models and software development models.

In terms of the computational intelligence models, a CTI feed context and a business IT context are generalised to be the same type, i.e., context object. Grey number model is selected to represent the attribute values of context objects. Fuzzy

sets are used to represent the context objects, and linguistic densities of the attribute values of context objects are reasoned. To assemble applicable connection knowledge, the system constructs a set of connection objects based on the context objects and uses rough set operations to extract applicable connection objects that contain the connection knowledge.

Furthermore, to adapt to contextual changes, a rough set based incremental updating approach with multiple operations is developed to incrementally update the approximations. A set of propositions are proposed to describe how the system changes based on the previous states and internal structures of the system, and their complexities and efficiencies are analysed.

In terms of the software development models, some unified modelling language (UML) models are selected to represent the system in design phase. Activity diagram is used to represent the business process of the system. Use case diagram is used to represent the human interactions with the system. Class diagram is used to represent the internal components and relationships between them. Using the representation, developers can develop a prototype of the system rapidly.

Using the representation, an application of the system is developed using mainstream software development techniques. RESTful software architecture is used for the communication of the business IT contextual information and the analysis results using CTI between the server and the clients. A script based method is deployed in the clients to collect the contextual information. Observer pattern and a timer are used for the design and development of the monitor-trigger mechanism.

In summary, the representation generalises real-world cases in the problem domain and interprets the system data. A specific business can initialise an instance of the representation to be a specific system based on its IT context and CTI feeds, and the knowledge assembled by the system can be used to identify relevant CTI feeds. From the relevant CTI data, the system locates and retrieves the useful information that can inform security decisions and then sends it to the client users. When the system needs to modify itself to adapt to the business IT contextual changes, the system can invoke the corresponding incremental updating functions and avoid a

time-consuming re-computation. With this updating strategy, the application can provide its users in the client side with timely support and useful information that can inform security decisions using CTI.

Declaration

The content of this submission was undertaken in Faculty of Computing, Engineering and Media, De Montfort University, and supervised by Prof. Yingjie Yang and Dr. Ying He during the registration period. I hereby declare that the materials of this submission have not previously been published for a degree or diploma at any other university or institute. All the materials submitted for assessment come from my own research. Part of the research work presented in this submission has been published or has been prepared for publication in the first three papers as below.

Y. Xu, Y. Yang and Y. He, "A Business Process Oriented Dynamic Cyber Threat Intelligence Model," 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), 2019, pp. 648-653, doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00147.

Y. Xu, Y. Yang and Y. He, "A Representation of Business Oriented Cyber Threat Intelligence and the Objects Assembly," 2020 10th International Conference on Information Science and Technology (ICIST), 2020, pp. 105-113, doi: 10.1109/ICIST49303.2020.9202271.

Y. Xu, Y. Yang and Y. He, "An Incremental Updating Strategy for Cyber Threat Intelligence in Dynamic Business IT Environment". (In Preparation)

R. Wang, Y. Xu, and L. Chen, "GazeMotive: A Gaze-based Motivation-aware E-learning Tool for Students with Learning Difficulties", In: Lamas D., Loizides F., Nacke L., Petrie H., Winckler M., Zaphiris P. (eds) Human-Computer Interaction, INTERACT 2019, Lecture Notes in Computer Science, Springer, Cham, doi: 10.1007/978-3-030-29390- 1_34, vol 11749, pp 544-548, Aug. 2019.

By
Yuanchen Xu
May 2021

Acknowledgements

I would like to express sincere gratitude to all who have provided help for the research project. Particularly, I am grateful to my first supervisor, Prof. Yingjie Yang, and my second supervisor, Dr. Ying He, for their continuous supports, guidance and encouragements throughout my PhD study at De Montfort University.

I would like to thank the students and staff at De Montfort University who inspired me with their comments.

I would also like to thank my friends for their encouragements and my whole family for their unconditional love, trust and understanding.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

Organisations are still challenged by cyber threats. Cyber threats can probably disrupt businesses, steal commercially sensitive information, jeopardise brand reputation, etc., leading to financial loss [1]. As cyber attacks are evolving and becoming more sophisticated, businesses are suffering more from the defence. Advanced persistent threats (APT) are today's common attack attempts. While traditional cyber attacks using individual components, such as exploiting only one vulnerability, are considered as "less advanced", APTs use more advanced techniques that combine multiple sophisticated methods and tools to reach the victims and maintain the access to it. Causing ongoing damage to the targets, APTs keep phishing, plugging and probing until they achieve their objectives.

Cyber threat intelligence (CTI) has proved effective in enhancing information security of businesses. CTI provides information of cyber threats and the corresponding countermeasures mitigating the cyber threats. It is time to use CTI because it is increasing available and the techniques of formatting and sharing CTI provide more convenience. Earlier, threat indicators were shared by human in the format of unstructured or semi-structured texts through web portals or encrypted emails. Better than before, CTI now has more efficient formatting and sharing techniques. One of them is Structured Threat Information eXpression (STIX)/Trusted Automated eXchange of Indicator Information (TAXII) platform [2]. STIX defines a number of components that aim at expressing sets of indicators and other full-spectrum CTI information. STIX formats its data using Extensible Markup Language (XML) or JavaScript Object Notation (JSON) whereby machines can transport, read and process the information automatically. Working together with STIX, TAXII is a community-driven exchange framework. Based on Representational State Transfer

(REST), a software architecture, TAXII defines its RESTful services for exchanging STIX messages between TAXII servers and clients, and then its developers can reuse the services to develop business logics. Besides, CTI sources are explosively increasing. They support a wide range of data retrieval techniques, including STIX/TAXII. CTI users can not only access and consume CTI feeds, but also contribute to them. Thanks to the public contributions, more and more CTI feeds are available from the sources.

The relevance, between a CTI feed context and a business IT context, can be used to determine whether the CTI feed is useful for the business environment. A CTI feed has information describing an affected business IT context, and a business has a similar business IT context. If the two contexts are reasoned to be relevant, the business can believe that the cyber threats, provided in the CTI feed, might also happen in the business environment, and the business then can take defensive actions, which might also be provided in the CTI, before the actual cyber threats take place. This representative research work prompting the connection between CTI and businesses includes [3–5].

This chapter is organised as follows. Section 1.1 explains the motivation of undertaking the research and poses the research questions. Section 1.2 presents the aims and objectives of the research. Section 1.3 presents the research methodology. Section 1.4 talks about the contributions to knowledge the research made. Section 1.5 outlines of the thesis.

1.1. Motivation

Even though CTI can inform security decisions and CTI is increasingly available, CTI is still insufficiently used by organisations [6]. Because not all the businesses have sufficient domain experts, it is still a big challenge regarding how to help businesses identify CTI relevant to the IT contexts.

Furthermore, businesses IT contexts, however, are varying over time. The states of the business IT contexts can change over time. Therefore, another challenge is

that how the connection strategies can adjust themselves to adapt to these contextual changes timely. In business IT contexts, massive amount of operational related information is coming from different business units. As the new business models and technologies, e.g., Internet of Things (IoT), mobiles and cloud computing, are emerging, services based on them are explosively generating usage data. Numerous point solutions, e.g., industrial control systems (ICS) and enterprise resource planning (ERP) software, firewalls, intrusion detection systems (IDS) and access-control techniques, also generate large numbers of log messages. Business IT information generated and collected from various business units at a time moment is impossibly the same as the one at another time moment. After the business IT contextual states change, the connection strategies might be no longer useful and effective and the relevant CTI, identified by these strategies, might be no longer appropriate and applicable.

Research Questions

The research question of the research project is then posed:

How to dynamically connect CTI onto business IT contexts?

To deal with the question, the sub-questions are posed as follows:

- Q1. How to identify CTI relevant to business IT contexts?
  - Q1.1. How to interpret CTI contexts and business IT contexts for computations?
  - Q1.2. How to collect CTI contextual data and business IT contextual data?
- Q2. How to dynamically update the connection strategy when the business IT contextual states change?
  - Q2.1. How to interpret the dynamics in the system?

  - Q2.2. How to monitor business IT contextual changes and trigger the changes of the connection strategy?

- Q3. How to develop a feasible way to help businesses use this dynamic connection?

Q1 aims to develop a static connection strategy between CTI and business IT contexts, and Q2 aims to dynamically modify the connection strategy. As sub-questions of Q1, Q1.1 and Q2.1 deal with the interpretation of the CTI contexts and business IT contexts, and the interpretation will help the system understand the data used by the computations. Q1.2 deals with the data collection of CTI and business IT contexts. Q2.2 concerns the development of a monitor-trigger mechanism that can inform the strategy the business IT contextual changes and then trigger the changes of the connection strategy. Q3 concerns the development of the system, which aims to identify a feasible method of helping the users use this system.

1.2. Aims and Objectives

The main aim of this research project is to develop a system that has a dynamic connection strategy for CTI onto a business IT context. Specifically, according to the research questions, the objectives of this research project are listed as follows.

- Objective 1. to investigate feasible data collection methods involved in the problem domain, to develop a fundamental method of collecting CTI feeds and business IT contextual data, and then to use the method to collect CTI feeds from a representative CTI source and business IT contextual data

- Objective 2. to develop a feasible connection strategy for identifying CTI relevant to business IT contexts

- Objective 3. to develop a feasible updating strategy used to dynamically update the connection strategy according to the business IT contextual changes

- Objective 4. to develop a feasible monitor-trigger mechanism for the updating strategy for the connection strategy

- Objective 5. to develop a representation that can interpret the data used by the computations in the system. Specifically,
  - Objective 5.1. interpret the CTI feed contexts and business IT contexts
  - Objective 5.2. interpret the dynamic connection between the CTI feed contexts and business IT contexts
  - Objective 5.3. interpret the design of the application of the system
- Objective 6. to implement the connection strategy

Regarding Objective 4, the monitor will continuously monitor business IT contextual changes, and when a contextual change is captured, the monitor will inform these contextual changes to the updating strategy to invoke the consequent updating responses.

Once the objectives are achieved, the system can timely update its connection strategy and connected CTI and timely inform the businesses the up-to-date security decision supports.

1.3. Research Methodology

This section introduces the research methodology. Object-oriented analysis and design (OOAD) is a technical approach for analysing and designing a business or a system by applying object-oriented programming and visual modelling throughout the development process. A significant view of OOAD is "Abstraction". Abstractions represent something else, and they are logical, rather than physical, and less detailed than what they represent. A representation of a business or a system can be then initialised to be an instance for a specific environment. Therefore, the system can have an abstracted representation and the primary aim of the project is the development of the representation.

To identify what should be represented, the problem domain is analysed, and four elements are identified from the domain. They are listed as follows.

- E1. CTI contexts

- E2. business IT context
- E3. a dynamic connection between CTI contexts and the business IT context
- E4. how to apply the connection strategy to a business IT context

These four elements will be the objectives of the development of the representation. To develop the representation, some computational intelligence models and software development models are selected. The selected computational intelligence models are used for the representation of E1, E2 and E3. E4 concerns how to help a business IT context uses the system. The selected software development models are used for the representation of E4, and E4 concerns an implementation.

First of all, E1, E2 and E3 are regarded as objects. These objects are abstractions and generalisations of the corresponding concepts. An object has attributes, their corresponding attribute values and operations that are common behaviour shared by all the object instances.

Specifically, E1 and E2 are abstracted and defined to be the same type context object, and a context object can then be instantiated to be an instance for a specific CTI feed or a business IT context. E3 is abstracted and defined as connection objects, and the dynamics are abstracted to be the changes of the attribute values. A connection object is generated using a CTI context for a business IT context. The generated connection objects can be used to reason the relevance between the contexts, and the connection dynamics are reflected in the changes of the attribute values.

The selected computational models are grey numbers, fuzzy set theory and rough set theory. All the attribute values of the context objects are represented by grey numbers. A context object is represented by different groups of fuzzy sets, Group A and Group C. One Group A fuzzy set and multiple Group C fuzzy sets are used to represent a context object. A Group A fuzzy set stores the values after performing a normalisation of the whitenised grey numbers, and a Group C fuzzy set stores the membership degrees of the attributes to a linguistic variable indicating densities of the attribute values. A CTI feed has an affected business IT context, so the

connection objects has a special attribute that is similarity indicating the relevance between the CTI context and a business IT context. The other attributes of the connection objects are constructed based on the attributes of the context objects. For all the similarities, a Group B fuzzy set and multiple Group C fuzzy sets are defined. The Group B fuzzy set is defined on all the pairs of context objects, and its membership function is to obtain the numerical similarity values. A Group C fuzzy set stores the membership degrees of the numerical similarity to a linguistic variable indicating the discrete similarity density. Fuzzy set operations are used to select appropriate densities for the Group C fuzzy sets.

Afterwards, the connection objects can be generated based on the context objects. The attribute and their values of the connection objects are placed into a two dimensional table that is also called an information system according to rough set theory. Rough set operations are used to extract the applicable connection objects. These extracted instances have useful connection knowledge reflected in the attributes and their values. A pair of attribute and its value models a condition of the connection. If another further CTI feed meets the condition modelled by an applicable connection object, the similarity of the feed to the business IT context can then be reasoned.

Regarding how to continuously monitor the changes, the system uses a timer, which is also abstracted to be an object, to periodically monitor the changes of the attribute values of the context objects. After the changes of the attribute values, the system will then modify the corresponding attribute values of the connection objects in the information system and extract the applicable connection objects one more time.

Regarding E4, some software development models are selected to represent the design and development of the system. Use case diagrams are used to represent the human interactions with the system. Class diagrams are used to represent the the relationship between the classes and their functions. Activity diagrams are used to represent the business processes of the system. E4 also concerns an implement of the system. Modern programming languages and mainstream software architectures

are used for the implements. Regarding how the system supports business users, the system locates and retrieves the analysis results that can inform security decisions from the applicable CTI feeds and sends the results to the user clients.

1.4. Contributions to Knowledge

CTI carries valuable information that can inform business security decisions. However, organisations are still insufficiently using CTI, so it is promising to develop a feasible approach to assisting their businesses to identify relevant CTI, and few existing connection strategies have the ability to dynamically modify itself to adapt to the business IT contextual changes. Therefore, a business IT context oriented dynamic CTI system is then described, designed and developed.

The conducted research has led to the following contributions.

- C1. A novel hybrid representation with grey numbers, fuzzy set theory and rough set theory and the selected software development models is designed and developed. The representation abstracts the problem domain and generalises the data and functions of the system. The representation can be then instantiated for a specific business IT context and a set of CTI feeds.
  - C1.1. A novel view is identified that is using grey numbers to represent the attribute values of the context objects because an attribute in one context object might have multiple values at the same time. As the representation uses a IT component quantity-based method for the attributes, it is easy to know the minimum and maximum quantities of the IT components inside a context object. Grey numbers naturally suit the case.
  - C1.1. Dedicated use case diagrams with the primary use cases are created for the system, instructing the users how to interact with the system.
  - C1.2. Dedicated class diagrams describing the internal objects, the functions and their associations are created, instructing a rapid system implementation.

  - C1.3. Dedicated activity diagrams are created to describe the main business processes of the system.

- C2. Using an instantiation of the representation of the computational intelligence models, a novel connection rule assembly system is developed. The system can be used to assemble applicable connection rules for identifying CTI data relevant to a business IT context.

- C3. A novel and dedicated rough set based incremental updating approach with multiple updating operations is developed. These operations incrementally update the connection rules without re-computing the entire model to decrease the computation time. The approach ensures the connection rule assembly system up-to-date, so the CTI data identified as relevant by the system will also be kept up-to-date. The information that can inform security decisions in the CTI data then will be provided in time.

- C4. Instructed the software development models, an implementation of business IT context oriented dynamic CTI system is developed. The application locates and retrieves the useful information, such as risk assessment results, from the applicable connection rules based on the CTI feeds and a business IT context, and the application periodically sends the information to the business IT context.

1.5. Thesis Outline

The thesis is outlined as follows.

Chapter 2 Related Work

In this chapter, the prior research investigation is delineated, including a brief history of how CTI is developed, what the business IT contextual dynamics are, the existing strategies how businesses use CTI, some computational intelligence models that are suitable for this research context, some software development models that are suitable for this research context and a summary of the emphasis on the research investigation for the proposed system.

Chapter 3 Methodology: A System Architecture of Business IT Context Oriented Dynamic Cyber Threat Intelligence

In this chapter, the system architecture of business IT context oriented dynamic CTI is introduced. The problem domain is firstly analysed and four elements are identified, and these elements are the objectives of the representation. Secondly, the high-level system structure and the internal components are listed and explained. Thirdly, an initial hybrid representation using some selected computational intelligence models, including grey numbers and fuzzy sets, and software development models, including activity diagram, use case diagram and class diagram, is developed and illustrated.

Chapter 4 A Connection Knowledge Assembly System for Cyber Threat Intelligence onto Business IT Context

In this chapter, a connection knowledge assembly system for CTI onto business IT context is developed and introduced. Firstly, a detailed representation using grey number theory, fuzzy set theory and rough set theory is defined. Secondly, the representation is instantiated to be an instance of a rule based system. Finally, a case study applying the system to assembling the connection rules for a business IT context from National Vulnerability Database [7] is demonstrated.

Chapter 5 An Incremental Updating Strategy for Cyber Threat Intelligence in Dynamic Business IT Context

In this chapter, a rough set based incremental updating strategy for the system in Chapter 4 is developed and introduced. The strategy will incrementally update the equivalence classes and approximations when the information system changes. The computation complexity and efficiency is then analysed.

Chapter 6 An Application of Business IT Context Oriented Dynamic Cyber Threat Intelligence

In this chapter, an application of business IT context oriented dynamic CTI is developed. The detailed representation of software development models and an implementation of the system are introduced.

Chapter 7 Discussions and Conclusions

This chapter summarises the work and the contributions to knowledge made in

the thesis. Furthermore, potential future work provides new interesting challenges followed by a short conclusion of highlights in this research work.

Chapter 2 Related Work

2.1. Cyber Threat Intelligence

CTI proves effective and useful in improving the security posture of businesses. Any information about threats that can inform decisions is arguably CTI [8]. A CTI feed has the information of an affected business and the countermeasures.

It is the time to use CTI now. CTI representation methods and CTI sharing methods are available and better than before. Earlier, represented by structured or semi-structured descriptions, potential indicators are shared by human through web portals or encrypted emails. Structured Threat Information eXpression (STIX) [2, 9], a relatively new representation, defines a set of components meant to represent more expressive sets of indicators and other full-spectrum CTI information. It uses Extensible Markup Language (XML) or JavaScript Object Notation (JSON) to store and transport data, whereby machines can read and process the information automatically. Working with STIX, Trusted Automated eXchange of Indicator Information (TAXII) [2] is the community-driven exchange framework. Based on Representational State Transfer (REST), TAXII defines its own RESTful services for exchanging messages between TAXII clients and servers. Another contribution is from Burger et al. [10]. They proposed a layered taxonomy model that classifies CTI sharing technologies.

Another advantage of CTI is that CTI sources are explosively increasing. Based on STIX/TAXII, many sources are available and they have massive available CTI due to many contributions from the public, including Hail a TAXII [11], Automated Indicator Sharing (AIS) [12], ThreatConnect [13], etc. Most CTI sharing platforms

also have communities for sharing CTI, such as ThreatConnect [13] and MISP [14]. The users can both contribute to and consume the CTI feeds from other contributors. Other sources include National Vulnerability Database (NVD) [7] that has a large number of vulnerability feeds, US-CERT notes [15], etc.

However, even though CTI is increasingly available, the organisations are still insufficiently using CTI [6].

2.2. Dynamic Business Contexts and the Corresponding Dynamic Threats

Business environments involve miscellaneous dynamics. Threats associated with the contextual risk factors, correspondingly, can also change at the same time.

In business contexts, large amount of operational related information is coming from different business units. As the new business models and technologies, e.g., Internet of Things (IoT), mobiles and cloud computing, are emerging, services based on them are explosively generating usage data. Numerous point solutions, e.g., industrial control systems (ICS) and enterprise resource planning (ERP) software, firewalls, intrusion detection systems (IDS) and access-control techniques, also generate large numbers of log messages. The contexts of businesses are highly dynamic, rather than static. The information generated and collected from the business units today is impossibly same as the one yesterday. The dynamics can also be reflected in operations of businesses forming the business processes. According to the traditional definitions of business process, transformation is a characteristic, ensuring that business processes can be transformed as long as the outputs can add value to customer or market.

Additionally, the dynamics are defined as the changes by any role at any time with low latency [16]. Every time the business contexts change, it is highly possible for the threats associated with the contextual risk factors to change at the same time.

Suitable representations for business contexts are necessary and important to analyse and process business contextual information. It is defined by [17] that a

business model is a conceptual tool containing a set of elements with relationships and allows to express the business logic of a specific firm. However, process view is more important to align the operational practices with the changing business requirement [18]. Reference [19] claims that in an organisation, the security tasks can be determined by business context and technical context. Ontology can be useful for modelling business processes for the generalisation and reusability, and for the improved distribution, integration and interoperability. Another tool is Business Process Management Notation (BPMN) [20] that can model business processes as diagrams with the defined notations. To represent the dynamics of the concepts, multiple forms [20] can be used to represent the changing phenomena over time. The current trends are how to manipulate information of business processes and what information of business processes needs to be stored [18], [21], even in real-time. However, the organisations might not be willing to share all the business information as it is sensitive.

2.3. CTI Based Defensive Strategies for the Businesses

Organisations are still facing the challenge of defence against cyber threats. Cyber threats will cause serious consequences for businesses, such as disrupting businesses, stealing commercially sensitive information, jeopardise brand reputation, etc., leading to financial loss [1]. As cyber threats keep evolving and become more sophisticated, defence against them costs businesses more effort. Today's common attack scenarios are called advanced persistent threats (APT) that have two aspects *advanced* and *persistent*. Traditional threats use individual components, such as exploiting only one vulnerability of target victim systems. Comparing with them, *advanced* aspect of APT means that APTs use more advanced techniques combining multiple methods and tools to reach and access the victims. *persistent* aspect of APT mean that APTs do not stop after they access the victim systems and APTs instead keep phishing, plugging and probing until they achieve their objectives, which can cause ongoing and persistent impact on the victim systems.

Kill chain [22] is a useful perspective to analyse APTs. It deconstructs the

attack scenarios into different steps, and security actions done for one step can break the chain. The steps are “Reconnaissance”, “Weaponisation”, “Delivery”, “Exploitation”, “Installation”, “C2” and “Actions on Objectives”, and these steps describes the life cycle of an APT. An example of an analysis using kill chain is presented in Fig. 2.1. A set of IT components are mapped onto the steps of the kill

Phase	Intrusion 1	Intrusion 2
Reconnaissance	[Recipient List] Benign File: tcnom.pdf	[Recipient List] Benign File: MDA_Prelim_09.pdf
Weaponization	Trivial encryption algorithm: Key 1	
Delivery	Downstream IP: 60.abc.xyz.215 Subject: AIAA Technical Committees [Email body]	Downstream IP: 216.abc.xyz.76 Subject: 7th Annual U.S. Missile Defense Conference [Email body]
	dn...etto@yahoo.com	
Exploitation	CVE-2009-0658 [shellcode]	
Installation	C:\...\fssm32.exe C:\...\IEUpd.exe C:\...\IEXPLORE.hlp	
C2	202.abc.xyz.7 [HTTP request]	
Actions on Objectives	N/A	N/A

Figure 2.1: A Example Kill Chain Analysis

chain. The most useful part of the technique is that if a threat is identified in one kill chain step the next steps can be cut and the threat can be stopped. However, to find the threats in APT steps, businesses need powerful intelligence to support their decisions. In this case, CTI becomes important. Informed by CTI, the defenders are motivated to respond to incidents from reactively and passively to proactively.

To use CTI, dedicated personnel allocation has been enhancing information security strength of organisations. The case in [23] reveals that different teams are responsible for incidents with high or low impact. However, dissemination of incident information among the teams is not usually effective. A centralised unit that can integrate and process information and then respond to incidents might be necessary and helpful. Security operations centre (SOC) is such a solution, run by security

experts to monitor the entire information domain in organisations and then to make security recommendations. The state of the art of SOCs is studied in [24]. The functions of SOCs combine the analyses of the two sides, i.e., CTI and the organisations. The functions of SOC can be those in [25], whereby automated and manual interventions can be made based on the analysis of CTI and the information collected from the organisations. The functions can also be those in the security governance model [26], including security monitoring, CTI and vulnerability management, incident response and forensics, and data loss prevention. Various tools are supporting security teams such as SOCs. Security information and event management (SIEM) is one of them, allowing the analysis of information collected from various sources of the organisations to generate alerts. One of the mature products is AlienVault Unified Security Management [14]. Some have included user and entity behaviour analytics (UEBA) [27] and security orchestration automation and response (SOAR) [28]. UEBA allows detecting the behaviours of users and entities deviated from the patterns, then to alert. SOAR enables the organisations to collect information from various sources, then to define, prioritise and drive the security activities based on a standard workflow.

Another type of the tools focuses on helping organisation find relevant CTI in a more automatic way. A business has a specific context, and CTI has an affect business context. If a relevance decision between the two contexts can be made, threats described in CTI can be determined as the threats which will potentially occur in the organisations, and the defensive knowledge in the CTI could be applied to the potential threats. MISP [14] is such a product, allowing the organisations to create and add their contextual information as the attributes of a MISP event defined by MISP. For this created event, other relevant events will then be mapped automatically based on the comparisons for the attributes. Reference [5] promotes the relevance computation based on the contextual measures for CTI and organisations and then the corresponding security decisions can be made. References [29] [30] [31] promote various analyses using CTI.

However, to perform a more accurate relevance computation, the relevance computation needs more comprehensive and effective taxonomies for the contexts. More-

over, business contexts are highly dynamic. The dynamic operations of the businesses may cause the dynamic threats. Every time the dynamics take place, the connection strategies may not be useful and effective any more. Relevant CTI already found by the connection strategies may not be appropriate and applicable any longer, and consequent actions based on the relevant CTI may not be feasible any more. The business need is an adaptive connection strategy that can adapt to the dynamically changing business contexts, which has not been fully satisfied. Xu et. al. [3] prompts a connection between CTI and organisations in a dynamic way.

2.4. Computational Intelligence Models

Humans have the ability to reason and act on uncertain information to make decisions, but machines can not. Machines require data and the operations on the data to solve a specific task. However, the data involved in some sophisticated areas is not always crisp, and the information in real-world databases is not always precise and complete. Fuzzy set [32] is a useful mathematical tool to describe vague information. The elements of fuzzy sets have membership degrees to better represent and analyse concepts. Based on fuzzy set theory, fuzzy logic [33] is an extension to classic logical systems that aims to model humans' imprecise reasoning. Fuzzy logic provides a means of handling the vagueness inherent in natural language. Rough set theory [34] is another powerful tool to deal with uncertain information. It has been widely used in machine learning, knowledge discovery, data mining, decision support and analysis, etc. Through the operations, it is quite easy to discover the knowledge and form the rules. A survey of rough set technique is presented in [35]

Grey number and grey set [36, 37] are relatively new model techniques. They refer to partially known information of systems. Grey numbers refer to partially known numbers. They are numbers with clear boundaries, but the position in the boundaries is unknown. Generalised grey number extends grey number to include all possible situations. Grey sets are the sets containing the elements whose characteristic values can be expressed with grey numbers.

Many researchers also focus on the use of rough set under dynamic environments.

The classification of dynamics in rough set has two aspects: synchronic dynamics, i.e., knowledge evolves in time, and diachronic dynamics, i.e., changes from one point of view to another [38]. The dynamics in rough set can also be divided into another two aspects: objects vary over time when attributes keep constant, and attributes vary over time when objects keep constant. Incremental approaches based on rough set aim at developing a more effective computation under dynamic environments. When the information system changes over time, the computations for the changes are based on the previous equivalence classes and approximations, which does not need to re-compute the equivalence classes and approximations again.

As for the objects varying over time, the incremental updating approaches focus on insertion of objects into information systems and deletion of objects from information systems. Chen et. al. [39] propose a rough set based incremental approach for dynamically updating the approximations based on variable precision rough set model (VPRS) [40]. VPRS is an extension of rough set the heart of which is the definition of a majority inclusion function. The approach [39] investigates incremental operations for updating approximations when inserting an object and deleting an object. Liu et. al. [41] propose an incremental approach that uses support matrix, accuracy matrix and coverage matrix describing and tracking the changes of the system. The approach is extended and applied to incomplete information systems [42]. As for the attributes varying over time, the incremental approaches focus on updating attribute values in information systems. The article [43] proposes an approach for coarsening and refining attribute values. Other research on rough set based incremental updating approach can be found in [44] [45] [46].

2.5. Development Representations: A View of Software Development and Object-oriented Approach

2.5.1. Software Development Lifecycle

Any software development project can be said to have four phases in its development lifecycle:

- Phase 1: Inception
  - Make a business case for a new product (or upgrade) and create a plan
- Phase 2: Elaboration
  - Analyse the problem
  - Design a solution
- Phase 3: Construction
  - Physically implement the product
    - Write the code
    - Compile executables
    - Test the code
- Phase 4: Transition
  - Hand over and install completed product
  - provide support

After the problem is identified in the phase 1 and before the final product is developed in the phase 3 and 4, the phase 2 concerns two aspects *analysis* and *design*. *Analysis* aims to identify what the proposed system must do, and *design* aims to identify how a new system will meet the requirements of coding, compiling and testing.

Modern applications have many characteristics, which are summarised as follows.

- Their business environments are volatile and subject to constant changes. Therefore, they need rapid information system developments when the changes happen.
- Their types are complex, such as computer aided design (CAD) / computer aided manufacture (CAM) , geographical information systems (GIS) and e-commerce.

- They increasingly use complex data types, such as text documents, video, sound, graphics and spatial data.
- They mix technologies of software and hardware.
- They have sophisticated user interfaces (GUIs).
- They are always distributed systems in client-server environments.
- They have tendency for larger systems with complex and varied interrelationships among software components.

2.5.2. Object-oriented Approach

To carry out the development tasks of modern applications, object-oriented (OO) approach is a suitable choice. It is based on the development of a set of software objects representing concept that work together in order to provide system functionality. It supports complex application domains, such as CAD/CAM, GIS and e-commerce. It is an approach of rapid application development (RAD), focusing less on planning and more on adaptive development processes. It support complex application domains e.g. CAD/CAM, GIS, e-commerce etc. It can build components, i.e., object libraries, analysis and design patterns, and reuse them. Traditionally, it is defined that an object is a single thing, or concept, that can be represented as an encapsulation of

- state: the values of the attributes
- behaviour: what is done by the operations
- identity: object ID a.k.a. object reference, such as a memory address

An object is a member, or called instance, of class.

In OO approach, there are another two important concepts called *model* and *abstraction*. Abstractions represent something else, they are logical, rather than physical, and they are less detailed than what they represent. The relationship between abstractions and what they represent is:

- One abstraction can represent many things.
- One thing can be abstracted in many ways.

After that, a model can be defined as: *"A model is a logical, small scale representation of something that abstracts the important aspects and hides non-relevant information."*

2.5.3. Model-driven Development and the Modelling Tools

OO is a model-driven approach. In an OO process, multiple OO models will be developed. Generally, there can be three types of models of model-driven development, which are divided based on the aforementioned software development lifecycle. They are listed as follows:

- Problem Space Models a. k. a. Computationally Independent Models: e.g., Proposal business model
- Specification Models a. k. a. Platform Independent Models: e.g., Use Case Model, Analysis Model and Design Model
- Solution Space Models a. k. a. Platform Specific Models: e.g., Implementation Model, Component Model and Deployment Model

UML is a useful modelling tool that is a general-purpose, developmental, modelling language in the field of software engineering that is intended to provide a standard way to visualise the design of a system [47]. Fig. 2.2 listed all the UML diagrams. The books [48] and [47] have enough materials related to system design and analysis using UML.

2.6. Summary

In summary, CTI is increasingly available but it is still insufficiently used in organisations. To use CTI in organisations, human and machines are working together.

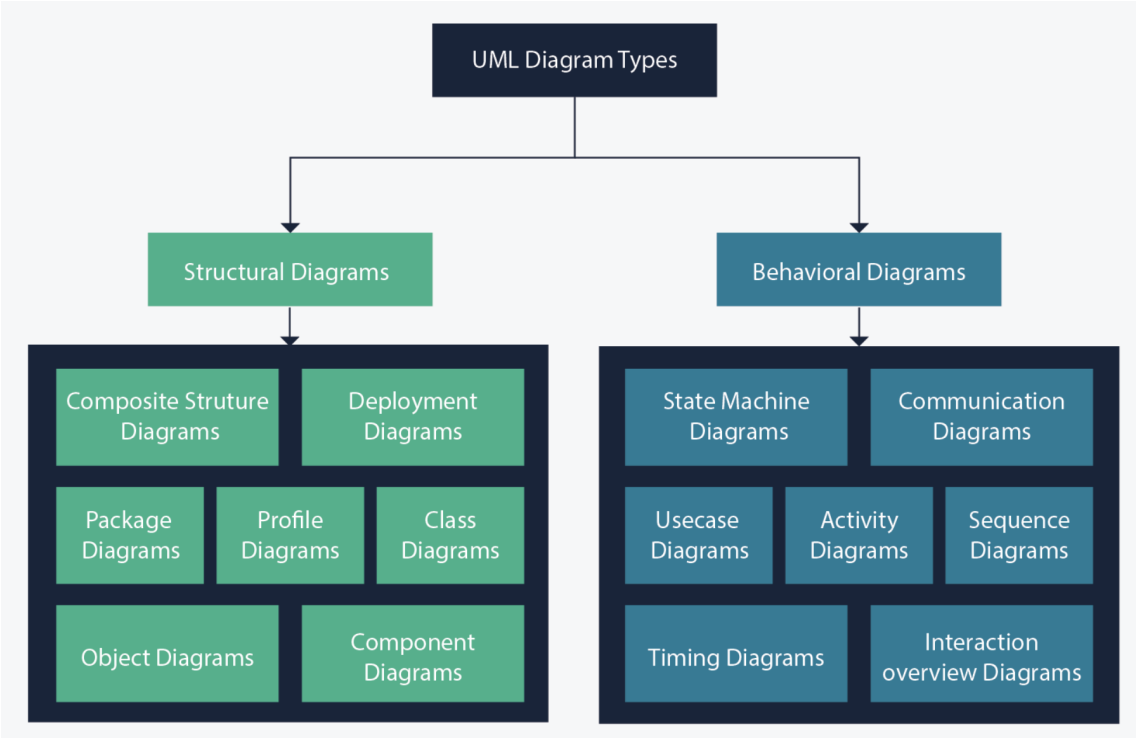

Figure 2.2: The UML Diagrams

Human can help make security decisions using CTI, but not all businesses have enough domain experts. Researchers, therefore, are facilitating the machine methods to identify relevant CTI for business IT context based on the relevance computation. The existing representative systems, which are most relevant to the research domain, are [3], [4], [5].

However, business contexts are varying over time. Few existing methods can meet the requirement of a dynamic connection between CTI and businesses. A dynamic view is proposed in [3] that a connection state can be represented by the states over time. Computational intelligence models are useful for developing an intelligent system. Among these models, rough set theory can be used to develop a rule based system, which might be able to help businesses find the CTI they need. Based on rough set theory, many researchers have already investigated how to effectively apply rough set based system in a dynamic environment, the contributions of which can potentially inspire the system development. To develop the system, UML tools are useful in the design and development process.

In the next chapter, the system architecture will be proposed and some initial

result will be presented.

Chapter 3 Methodology: A System Architecture of Business IT Context Oriented Dynamic Cyber Threat Intelligence

3.1. Introduction

Based on the investigation results of the literature review conducted in Chapter 2, this chapter proposes a system architecture for business IT context oriented dynamic cyber threat intelligence (CTI) (BDC system). BDC system will establish a dynamic connection between CTI and organisations that would like to use CTI to enhance the information security strength of their businesses. The system aims to help these businesses identify CTI feeds relevant to the business IT contexts and aims to have a dynamic ability to adapt to the business IT contextual changes. This chapter introduces the fundamental understanding of how to develop BDC system and presents some initial results.

3.2. Understanding of the Problem Domain

3.2.1. The Elements Identified from the Problem Domain

This section discusses a deconstruction and understanding of the problem domain, i.e.,

how to identify CTI relevant to varying business IT contexts

and the architecture of BDC system will then be introduced based on the under-

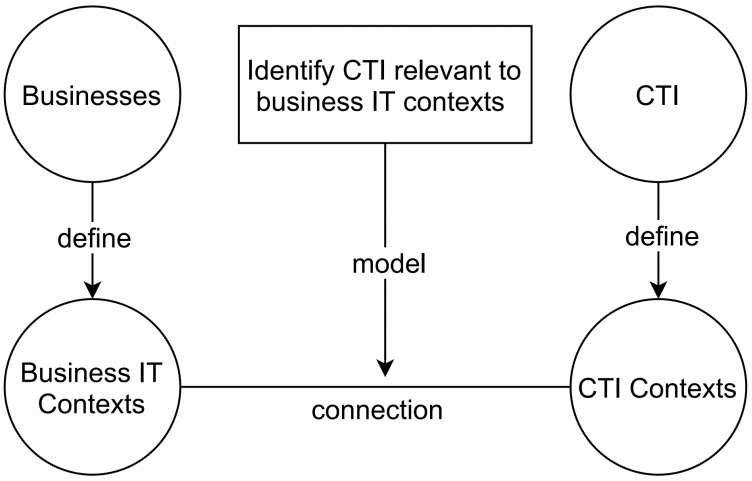

Figure 3.1: Initial Steps

standing.

Four elements are identified from the problem domain, which are listed as follows.

- Element 1: CTI feed
- Element 2: business IT context
- Element 3: a connection between an individual business IT context and multiple CTI feeds
- Element 4: how to apply the connection strategy to a business IT context

Fig. 3.1 shows the initial steps of developing the system. The system needs to understand the data of CTI feeds and businesses, so the system will define dedicated "CTI Contexts" and "Business IT Contexts", which is also referred to as "Business Contexts", to make the data processable for the internal computations. A CTI context describes a piece of CTI data, i.e., a CTI feed, and contains the information in the CTI feed. A business IT context describes and contains the IT information inside the business. The system then models a "Connection" between the contexts to achieve the Element 3. The connection is a strategy that can be used to identify CTI contexts relevant to business IT contexts. The Element 4 is to help business use the system.

3.2.2. A Case Study: A Dynamic Business IT Context and the Corresponding Dynamic Threats

Business environments are varying, and correspondingly, the cyber threats are dynamic. This section presents a case study about how a relevant CTI feed helps a business identify a cyber threat and how the threat changes while the business IT context changes. The case study explains why a dynamic connection between CTI and businesses is important.

Let c_1 denote a CTI context that is a context of a CTI feed, and let b_1 denote a business IT context. b_1 has some devices that are Tp-link and NETGEAR wifi extenders. c_1 has the information of a vulnerability of CVE-2018-12694 [49] that describes a threat about a wifi extender device. Based on the information in c_1 and b_1, the system firstly generates the attributes used for the later analyses. Let P_1 denote the set of attributes generated by the system, and P_1 is

$$P_1 = \{\texttt{wifi extender}, \texttt{tplink}, \texttt{tl we850 hardware 5}, \texttt{denial of service}, \texttt{netgear}, \texttt{ex2700}, \texttt{cti}\}$$

where `tl wa850re hardware 5` is the version of `tplink` device, `ex2700` is the version of `netgear`, `wifi extender` is the device type of the two devices, `denial of service` is the threat and `cti` is the type of the context. Another important attribute is the relevance between c_1 and b_1, and similarity can be used to represent the relevance.

After the system obtains the attributes, the system then needs to obtain the corresponding attribute values. The system can use a term frequency method for the calculation of the attribute values, and the system can use Eq. 3.6 with all the weights being 1 to compute the similarity value between c_1 and b_1. According to the elements in P_1, c_1 has 2 `wifi extender`, 1 `tplink`, 1 `tl we850 hardware 5` and 1 `denial of service`. b_1 has 2 `wifi extender`, 1 `netgear` and 1 `ex2700`. c_1 is a CTI feed and b_1 needs CTI, so each of c_1 and b_1 has 1 `cti`.

Business IT contexts are changing over time. Let t_1, t_2 and t_3 denote three time

moments when the state of the business IT context changes. For example, $b_1(t_2)$ denote the business state at t_2. The system starts at t_2. Table 3.1 presents the IT component frequencies and their normalised values for the two contexts at t_2. c_1 and b_1 can be two fuzzy sets defined on the universe P_1, and their membership

Table 3.1: Similarity Computation Using P_1 at t_2

Attributes in P_1	$c_1(t_2)$		$b_1(t_2)$	
	fre[a]	nor[b]	fre	nor
`wifi extender`	2	0.707	2	0.756
`tplink`	1	0.354	0	0
`tl wa850re hardware 5`	1	0.354	0	0
`denial of service`	1	0.354	0	0
`netgear`	0	0	1	0.378
`ex2700`	0	0	1	0.378
Type: `cti`	1	0.354	1	0.378

[a] Frequency
[b] Normalised Value

function is to obtain the normalised frequencies, using Eq. 3.5, as their attribute values. Using the normalised frequencies, the similarity between the two contexts at t_2 is obtained by $S(c_1(t_2), b_1(t_2))$, using Eq. 3.6, with all the attribute weights being 1. The similarity is 0.692.

At t_3, $b_1(t_2)$ changes to $b_1(t_3)$ and $c_1(t_2)$ keeps unchanged. b_1 removes the NETGEAR device and adds a TP-LINK device at t_3. Table 3.2 shows the frequencies and normalised frequencies. With the updated attribute values, the new similarity between them is $S(c_1(t_3), b_1(t_3)) = 0.771$.

At t_3, both c_1 and b_1 have no `netgear` and `ex2700`. Thus the attribute set P_1 needs to be updated to P_2 without the two attributes.

$$P_2 = \{\texttt{wifi extender}, \texttt{tplink}, \texttt{tl we850 hardware 5}, \texttt{denial of service}, \texttt{cti}\}$$

Table 3.3 shows the new attributes, the frequencies and the normalised frequencies at t_3. With the new attributes in P_2, the new similarity is $S(c_1(t_3), b_1(t_3)) = 0.729$.

Table 3.2: Similarity Computation Using P_1 at t_3

Attributes in P_1	$c_1(t_3)$		$b_1(t_3)$	
	fre	nor	fre	nor
`wifi extender`	2	0.707	1	0.577
`tplink`	1	0.354	1	0.577
`tl wa850re hardware 5`	1	0.354	0	0
`denial of service`	1	0.354	0	0
`netgear`	0	0	0	0
`ex2700`	0	0	0	0
Type: `cti`	1	0.354	1	0.577

Table 3.3: Similarity Computation Using P_2 at t_3

Attributes in P_1	$c_1(t_3)$		$b_1(t_3)$	
	fre	nor	fre	nor
`wifi extender`	2	0.707	1	0.577
`tplink`	1	0.354	1	0.577
`tl wa850re hardware 5`	1	0.354	0	0
`denial of service`	1	0.354	0	0
Type: `cti`	1	0.354	1	0.577

Table 3.4 summaries the changed items during the process. The similarity between the two contexts at t_2 is 0.692, and the similarity at t_3 is 0.771. These values mean that the possibility of the cyber threat `denial of service`, described in c_1, happening in b_1 is 69.2% at t_2 and the value increases at t_3 to 77.1%. From t_2 to t_3, the business becomes more vulnerable to `denial of service`.

3.2.3. The Refined Elements from the Problem Domain

As suggested by the case study in the previous section, Fig. 3.2 shows the changed elements the system needs to consider. As business environments change, the system should consequently modify the corresponding business IT contexts. The system will not modify the CTI contexts because CTI feeds tend to be unchanged after the system collects and stores them into a database. Every time a business IT context changes, the system might need to modify the connection strategy at the same time

Table 3.4: Similarity Computation at t_2

Changing Items	t_1	t_2	t_3
b_1	$b_1(t_1)$	$b_1(t_2)$	$b_1(t_3)$
Attribute used for the similarity computation	`null`	P_1	P_2
Attribute numbers	`null`	7	5
Frequency of `tplink`	`null`	0	1
Frequency of `wifi_extender`	`null`	2	1
Similarity between c_1 and b_1 using P_1	`null`	0.692	0.771
Similarity between c_1 and b_1 using P_2	`null`	`null`	0.729

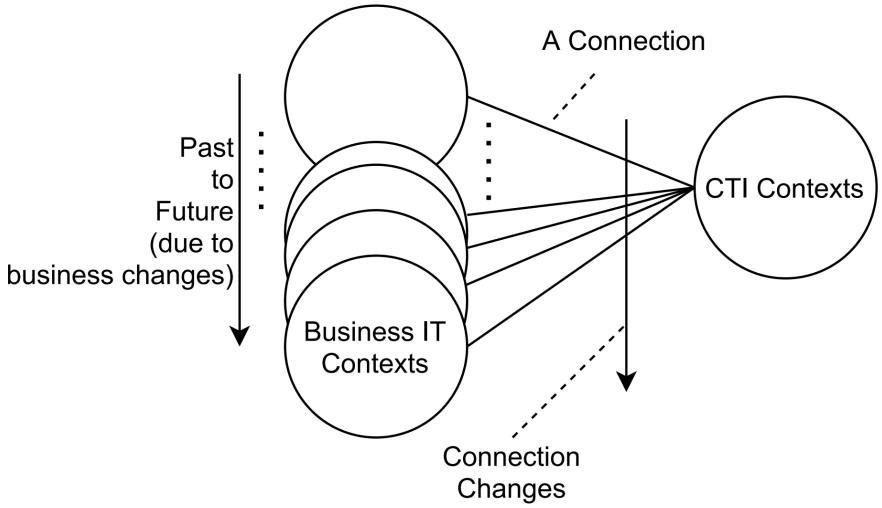

Figure 3.2: What the Changes Are

and change the connected CTI contexts. Therefore, the four elements, identified in Section 3.2.1, are refined to be the following elements.

- Element 1.1: CTI feed: an individual piece of external CTI data, such as a vulnerability
- Element 1.2: CTI context: internal data structure for storing the external CTI data
- Element 2.1: business context: one external business environment
- Element 2.2: business IT context: internal data structure for storing the IT information of an external business environment

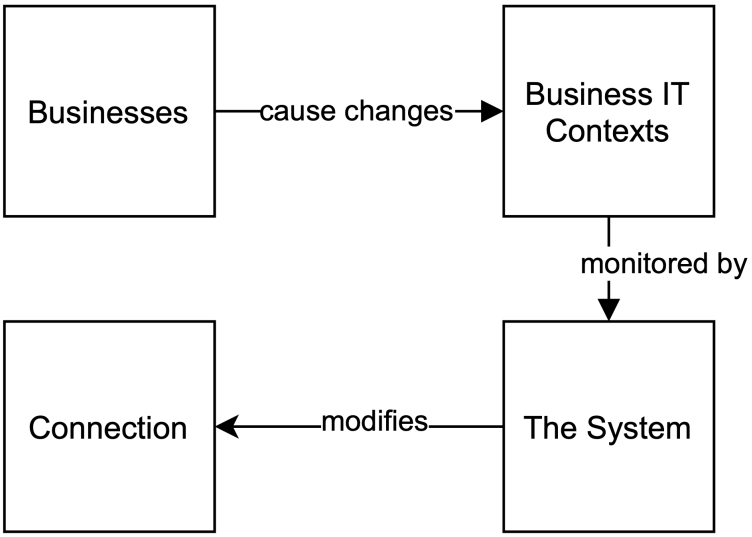

Figure 3.3: The Flow of the Changes

- Element 3: a connection: a strategy used to identify CTI contexts relevant to a business IT context at a specific time moment
- Element 4: how to apply the connection strategy to a business IT context

Element 3 is refined as a connection strategy at a specific time moment. When time elapses, the system might change Element 2.2 and Element 3 when the system captures the changes from the Element 2.1. The connection is refined as a dynamic connection that has additional mechanisms used to modify the connection between the time moments. Fig 3.3 shows the flow of the changes. As "Businesses" change, "Business IT Contexts" change. "The System" continuously monitor the changes of "Business IT Contexts", and due to the changes of "Business IT Contexts", "The System" might modify the "Connection" strategy.

3.2.4. Proposed High-level BDC System Architecture

This section introduces the high-level architecture of BDC system. Specifically, the section introduces the internal modules with their functions and the representation structure for the data in the system.

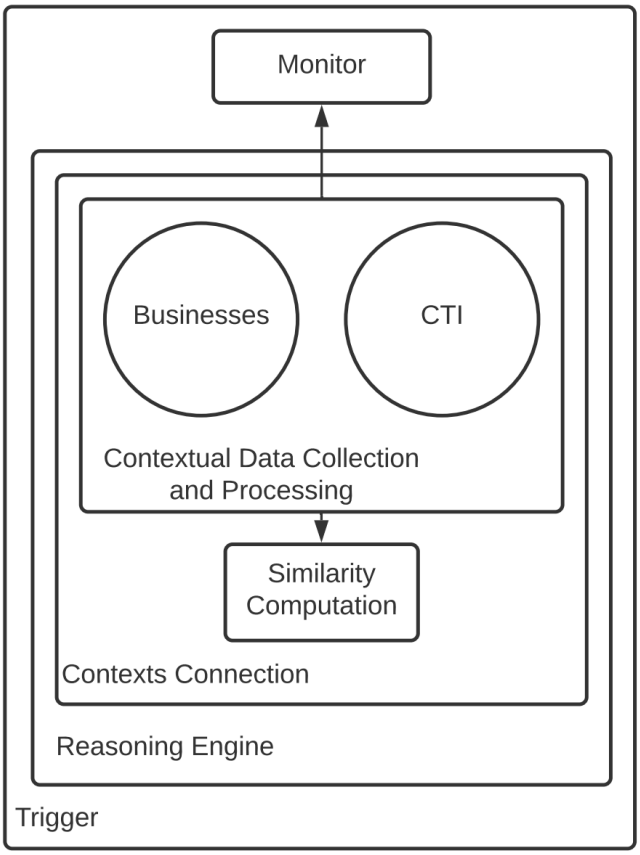

Figure 3.4: Proposed Components of BDC System

3.2.4.1. Proposed System Components

Fig. 3.4 shows the proposed system components with their functions.

Contextual Data Collection Processing Module

"Contextual Data Collection and Processing" is the module to collect external business IT and CTI data, and to process the data for the computations inside the system.

Data is necessary for the system to enable all the computations. In the beginning, it is important to define what these contexts are.

- Context: One individual context is a domain in an IT environment, and the domain contains a set of IT components. For example, a domain can be an individual computer, a set of computers, a folder in an operating system, a

JSON or XML file or even a piece of software. An IT component can be any single element inside an IT environment. For example, an IT component can be a file or a piece of software.

Both "Businesses" and "CTI" are contexts. In the system, the IT components of the contexts are computed in the form of words. For example, an IT component is a piece of software "Powerpoint" and the system uses the word "Powerpoint" for the computations.

- Business IT Contexts: Business IT contexts are constructed by the module using the data from "Businesses" side. To collect the data, the module is deployed with a hybrid approach that comprises automatic and manual methods.
  - Automatic:
    - Operating system level: The module is deployed with a set of scripts such as shell scripts, batch files, etc.
    - Programming language level: The business logics for the module use the functions of the APIs of the programming languages used by the module, such as `os` interface of Python language.
  - Manual: The users can input the words of the IT components into the system.
- CTI Contexts: CTI contexts are constructed by the module using the data from "CTI" side. CTI data is available in various CTI sources, and the mainstream formats of the data currently are JSON and XML files. To collect the data, the module has a program with RESTful APIs developed to retrieve the data from the CTI sources that are based on REST software architecture. Some CTI sources allow their users to download the CTI data in the form of JSON or XML file, so the data can also be manually downloaded from these sources. To retrieve the IT components from the files, the module is deployed with JSON and XML parsers to access the elements in these JSON and XML files.

Contexts Connection Module

"Contexts Connection" is the module to perform a similarity computation to obtain the similarity attribute value between two contexts. The similarity value can represent the relevance between the two contexts. The case study in Section 3.2.2 has provided a feasible example of how to perform the similarity computation. In the case study, the system uses all non-repetitive IT components, in the form of words, in the two contexts to construct the attributes. After that, the module calculates the quantities of these non-repetitive words and normalises the quantities, and the module uses these normalised quantities to be the attribute values. Using these attribute values, the module then uses a set distance-based method, i.e., Eq. 3.6, to obtain the numerical similarity value.

The module uses the following steps to perform the similarity computation, and these steps are based on multiple CTI contexts and one business IT context.

- Step 1: Generate attributes.
  - Step 1.1: Generate attributes for the CTI contexts and the business IT context. For example, the non-repetitive IT components in the business IT context can be the attributes, and specifically, "Powerpoint" can be one attribute for the contexts.
  - Step 1.2: Generate attributes for all pairs of a CTI context and the business IT context. For example, the similarities on the non-repetitive IT components can be the attributes, and specifically, "Similarity on Powerpoint" can be one attribute for the pairs.
- Step 2: Generate attribute values.
  - Step 2.1: Generate attribute values for the contexts, such as the numerical normalised quantities of the non-repetitive IT components.
  - Step 2.2: Generate attribute values for all pairs of a CTI context and the business IT context, such as the numerical numerical similarity values on the non-repetitive IT components.

- Step 3: Generate numerical similarity values for all pairs of a CTI context and the business IT context, using a set distance-based method with the numerical attribute values of the contexts.

Reasoning Engine Module

"Reasoning Engine" is the module to reason the semantic relevance between a CTI context and a business IT context. The decision is made by the module according to a connection knowledge base that is generated based on a set of CTI contexts and one business IT context. The workflow for generating the connection knowledge is presented as follows.

- Step 1: Process all attribute values generated by "Contexts Connection" module, i.e., the attribute values of the contexts, the attribute values of the pairs of a CTI context and the business IT context ($attrPairs$) and the numerical similarity values for all pairs of a CTI context and the business IT context ($attrSim$), into discrete values, using fuzzy set theory.
- Step 2: Assemble connection knowledge
  - Step 2.1: Based on $attrPairs$, $attrSim$ and their corresponding attributes, the module generates a connection rule for one pair of the contexts. The rule generated for the first pair is in the following form:

$$\begin{aligned} &\texttt{if } attr1 = attrPairs_1, attr_2 = attrPairs_2, attr_3 = attrPairs_3, \cdots \\ &\quad attr_n = attrPairs_n, \text{ then } attr_{sim} = attrSim_1 \end{aligned} \tag{3.1}$$

    where $attr$ denotes one attribute name, n denotes the quantity of the non-repetitive attributes obtained based on the business IT context, $attr_{sim}$ denotes the similarity attribute between the two contexts. The module repeats the rule generation for all pairs.
  - Step 2.2: After the module generates all the connection rules, the module uses rough set operations to extract the applicable connection rules.

If there is a new CTI context, the module can compare the discrete attribute values of the pair of the new context and the business IT context with the conditions of the applicable connection rules to reason the semantic relevance.

Monitor and Trigger

Business IT contexts are varying over time, and the system can get new CTI data or remove existing CTI contexts. When the business states change or the system gets new CTI data or the system removes existing CTI contexts, the system needs to adjust the corresponding internal business contexts and then adjust the connection strategy. Therefore, "Monitor" is the module to monitor the changes of the attributes of the business IT contexts. It compares the attributes of the business IT contexts at different time moments and determines whether or not the contextual attributes are changed. Every time the module captures a change, the module will inform "Trigger" module the change. "Trigger" module will then invoke its functions to trigger the changes of the system.

Consequent Analyses

"Consequent Analyses" is a module, which is not presented in Fig. 3.4, to provide the information that can support business IT security decisions. The module will not generate any risk analysis results, and all the results come from the CTI. From the relevant CTI contexts, the module locates and retrieves the risk analysis results. For example, the analyses can include the overall risk level and possible business IT financial losses.

3.2.4.2. Proposed System Representation Layers

Abstraction is a widely-used concept in system design and analysis. Abstractions generalise real-world concepts and developers can concentrate on the development and ignore the detailed and various specific cases of the concepts. Inspired by the view, a hybrid representation is developed for the system and the representation

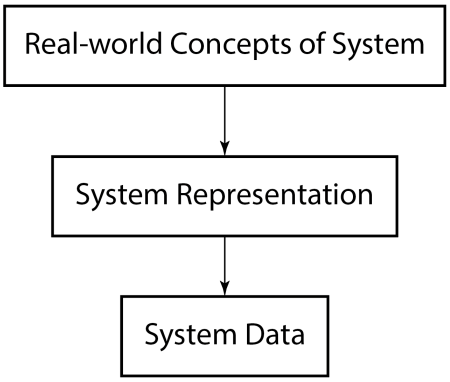

Figure 3.5: Representation Layers

can interpret the data in the system. The representation has multiple layers shown as in Fig. 3.5. "Real-world Concepts of System" includes those elements identified from the problem domain, which are listed in Section 3.2. "System Representation" concerns the representations of the real-world concepts and they can interpret the specific "System Data".

The hybrid representation uses some selected computational intelligence models and software development representations. It reorganises the elements identified in the problem domain into several layers, and it models the system from the problem domain to a mature final product. The computational intelligence models aim to represent the internal computations and the data used for these computations, and software development models aim to represent the Element 4.

3.3. A Hybrid Representation Combining Computational Intelligence Models and Software Development Models

Fig. 3.6 exhibits the high-level structure of the representation. Layer 1 contains the Element 1, Element 2 and Element 3 identified in the problem domain. Layer 2 has two sub layers that are Layer 2.1 and Layer 2.2. Layer 2.1 represents the system mathematically, using the selected computational intelligence models, including grey numbers, fuzzy sets and rough set theory. Layer 2.2 represents the Element 4 using the selected Unified Modelling Language (UML) models, including activity diagrams, use case diagrams and class diagrams.

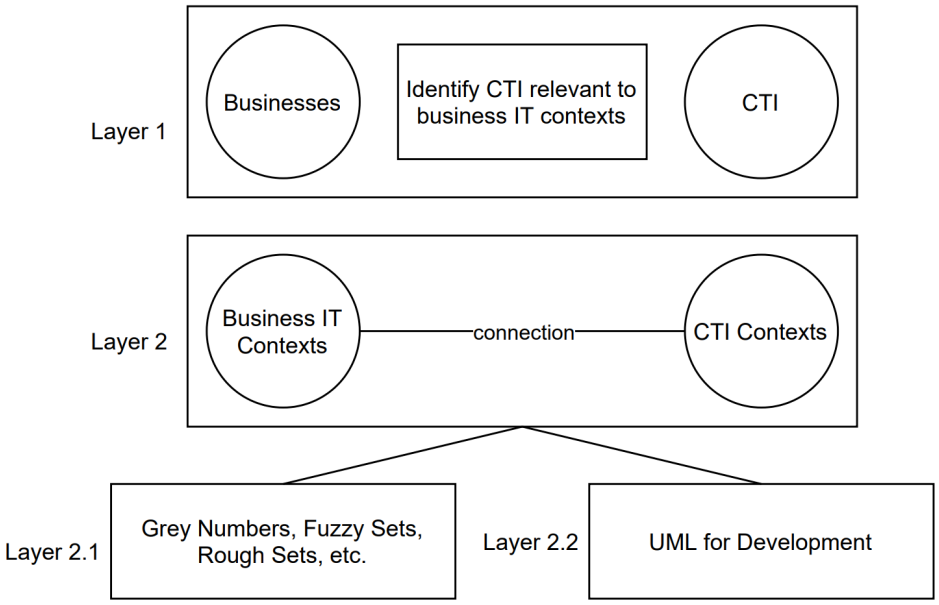

Figure 3.6: The High-level Representation Structure

3.3.1. Layer 2.1: Mathematical Representation

Layer 2.1 representation has different levels to interpret the data. A CTI context has an affected business context, so the representation considers all the contexts, CTI or business, mathematically similar. The Element 1, 2 and 3, therefore, are reduced to the following two elements:

- Element A: Contexts
- Element B: Connection between Contexts

Layer 2.1 then represents a context in contextual attribute level and context object level, and represents a connection in connection attribute level and connection object level.

3.3.1.1. Contextual Attribute Level of Element A

Grey numbers are used by the representation for the attributes of context objects. Whether or not two contexts are relevant can be interpreted based on whether or not some attributes of the contexts are mapped onto each other. A simple connection strategy then can be deployed and expressed as:

- Strategy expression 1: If two events have at least one identical attribute, the two events are mapped onto each other. The two events are then relevant.

The connection strategy has another mathematical expression:

- Strategy expression 2: Let U be a set of finite attributes for the events, and $U = \{x_1, x_2, x_3, \cdots, x_n\}$. All the attributes are term quantities. Let e_1, e_2 denote two events, and they have attribute values of the elements in U. If $\exists x \in U$, $x(e_1) \geq 1$ and $x(e_2) \geq 1$, the two events are relevant.

The remaining content in the section introduces why the representation uses grey numbers. Table 3.5 presents two example events with some attributes. Let e_1

Table 3.5: Two Events

Event 1	Event 2
Windows10	Powerpoint
Microsoft	Windows7
Word	Keynote
MacOS	
Excel	

denote Event 1 and e_2 denote Event 2, and let $x_1 \in U$ be quantity of Microsoft. For Event 1, $x_1(e_1)$ can be 4, because four terms in Event 1 are products of Microsoft. For the same reason, $x_1(e_2)$ can be 2. Based on the expression 2, Event 1 and Event 2 can be mapped onto each other because $x_1(e_1) \geq 1$ and $x_1(e_2) \geq 1$.

However, potential attribute mappings might exist. If the strategy is informed with more information that x_1 only considers the exact term Microsoft, the attribute values become different. In this case, $x_1(e_1)$ becomes 1 because there is only one same term. $x_1(e_2)$ becomes 0 because no terms are available in Event 2. Based on the new values, Event 1 and Event 2 cannot be mapped onto each other because $x_1(e_2) \ngeq 1$. If the strategy is not informed with the information, $x_1(e_1)$ can be a value of 4 or 1 and $x_2(e_2)$ can be a value of 2 or 0. The mapping between Event 1 and Event 2 can be interpreted to be both true and false.

Another example is recycle bin in Windows OS. If an OS user deletes a file, the OS will move the file to recycle bin. The IT component is then removed from the business IT context. However, it is possible for the user to recover the file. If the user recovers the file, the OS will move the file to its original location, and if the user does not want to recover the file, the file will be kept in recycle bin. The file, therefore, exists in the IT context and does not exist in the IT context at the same time, the file existence is determined by the user's possible recovering choice. If a business IT context has such a file, there might be multiple relevances to a CTI context at the same time.

Grey numbers naturally suit this case because they have the ability to represent multiple values at the same time. According to [50] [51] [36], grey numbers are good at representing concepts with unknown value in a certain scope. For the information in Table 3.5, it is easy to know the quantity of all IT components but it is difficult to know the certain value because of the potential term mappings. Therefore, it is more suitable to use grey numbers to represent attribute values of x_1. Let two discrete grey numbers $x_1(e_1)^{\pm}$ and $x_1(e_2)^{\pm}$ represent $x_1(e_1)$ and $x_1(e_2)$ respectively. $x_1(e_1)^{\pm} \in \{1, 4\}$ and $x_1(e_2)^{\pm} \in \{0, 2\}$.

Following the idea, the representation therefore uses grey numbers to represent the attribute values of Element A.

3.3.1.2. Context Object Level of Element A

Fuzzy sets are used for the representation of CTI and business contexts. Multiple fuzzy sets with different purposes are defined for the contexts. Let U_1 be a universe of discourse with finite elements and $U_1 = \{x_1, x_2, x_3, \cdots, x_{n_1}\}$ where $x_1, x_2, x_3, \cdots, x_{n_1}$ denote the attributes describing the contexts. The following groups of fuzzy sets are defined.

- Group 1: A fuzzy set is defined on U_1 to pre-process the attributes. Each context has a fuzzy set representing it. The membership function is to obtain the initial numerical attribute values between $[0, 1]$.

- Group 2: The numerical attribute values of the contexts are interpreted by multiple linguistic densities. A set of fuzzy sets interpreting the linguistic densities of the contextual attribute values are defined on U_1. Each fuzzy set corresponds to one linguistic density.

3.3.1.3. Connection Object Attribute Level of Element B

A connection object is generated based on a pair of context objects.

Similarity Attribute

A connection object has a special attribute named similarity that is used to interpret the relevance degrees of the pairs of the contexts. For one context object, let U_2 be a universe of all pairs of the context objects.

- Group 3: The numerical similarities are interpreted by distances of the pairs of two Group A fuzzy sets representing two context objects. One Group 3 fuzzy set is defined on U_2. The membership function is to obtain the distances.

Similarities can be obtained based on distances of points in a space. Suppose there is a n-dimensional real coordinate space $\mathbb{R}^n$, and $A = \{a_1, a_2, a_3, \cdots, a_n\}$, $B = \{b_1, b_2, b_3, \cdots, b_n\}$ are two points in the space, $A, B \in \mathbb{R}^n$. The Euclidean distance between A and B, $d(A, B)$, is obtained by

$$d(A, B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2} \tag{3.2}$$

Euclidean distance can be treated as a special case of Minkowski distance. The Minkowski distance of the contexts is defined in Definition 3.1.

Definition 3.1. *Let U be a universe with finite elements $U = \{x_1, x_2, x_3, \cdots, x_n\}$. $\mathcal{F}(U)$ is the power set of all the fuzzy sets on U. For two fuzzy sets A, B and $A, B \in \mathcal{F}(U)$, the Minkowski distance between A and B is $d_p(A, B)$.*

$$d_p(A, B) = \left(\frac{1}{n} \sum_{1}^{n} |A(x_i) - B(x_i)|^p \right)^{\frac{1}{p}} \tag{3.3}$$

When $p = 2$, $d_2(A, B)$ is the Euclidean distance between A and B.

The higher the distance, the less the similarity. Some researchers investigate the nearness between fuzzy sets. The higher the nearness, the higher the similarity. Definition 3.2 provides the generalised form of the nearness between two fuzzy sets.

Definition 3.2. *Let U be a universe with finite elements $U = \{x_1, x_2, x_3, \cdots, x_n\}$. $\mathcal{F}(U)$ is the power set of all the fuzzy sets on U. If a mapping N*

$$N : \mathcal{F}(U) \times \mathcal{F}(U) \longrightarrow [0, 1] \tag{3.4}$$

satisfies the following conditions:

$$N(A, B) = N(B, A)$$

$$N(A, A) = 1, N(\emptyset, X) = 0$$

$$A \subseteq B \subseteq C \Longrightarrow N(A, C) \leq N(A, B) \leq N(B, C)$$

N is a nearness function on $\mathcal{F}(U)$, and $N(A, B)$ is the nearness between A and B.

The primary aim of the system is to help a business identify relevant CTI. If there is a fuzzy set representing a business IT context, the aim is to find which fuzzy sets representing CTI contexts are closer to this fuzzy set. During the process, different attributes of the contexts can have different importance values. Thus a weighted normalised Euclidean nearness between the contexts is defined in Definition 3.3.

Definition 3.3. *Let U be a universe with finite elements $U = \{x_1, x_2, x_3, \cdots, x_n\}$. $\mathcal{F}(U)$ is the power set of all the fuzzy sets on U. Let X be a fuzzy set and $X \in \mathcal{F}(\mathcal{U})$. The normalised value of the i-th element of X, $i \in \{1, 2, 3, \cdots, n\}$, is obtained by $NOR(X(x_i))$ that is to compress the value to be a value in $[0, 1]$.*

$$NOR(X(x_i)) \coloneqq \sqrt{\frac{X(x_i)^2}{\sum_{j=1}^{n} X(x_j)^2}} \tag{3.5}$$

The weighted normalised Euclidean nearness of two fuzzy sets $A, B \in \mathcal{F}(U)$ is then

obtained by $S(A, B)$.

$$
\begin{aligned}
S(A,B) &:= 1 - \left(\frac{1}{n} \sum_{i=1}^{n} w(x_i) \left| NOR(A(x_i)) - NOR(B(x_i)) \right|^2 \right)^{\frac{1}{2}} \\
&= 1 - \left(\frac{1}{n} \sum_{i=1}^{n} w(x_i) \left| \sqrt{\frac{A(x_i)^2}{\sum_{j=1}^{n} A(x_j)^2}} - \sqrt{\frac{B(x_i)^2}{\sum_{j=1}^{n} B(x_j)^2}} \right|^2 \right)^{\frac{1}{2}}
\end{aligned}
\tag{3.6}
$$

where $w(x_i) \in [0, 1]$ *is the weight of* x_i.

Other Attributes

As for a connection rule, the similarity attribute is the decision attribute. In addition to the similarity attribute, a connection object has other attributes that are condition attributes. For now, these condition attribute values are selected from the discrete attribute values obtained by "Contexts Connection" module and "Reasoning Engine" module.

3.3.1.4. Connection Object Level of Element B

A connection object models the knowledge used to identify CTI contexts relevant to a business IT context. As for one business context, each pair of a CTI context and this business IT context generates a connection object. One connection object has a discrete similarity density, and Group 4 fuzzy sets are defined on U_2, i.e., the universe of all pairs of the context objects.

- Group 4: A similarity has multiple linguistic concepts interpreting the similarity density. A set of fuzzy sets interpreting the linguistic densities of the similarities are defined on U_2. Each fuzzy set corresponds to one linguistic concept.

3.3.1.5. Connection Knowledge Level

All the connection objects generate the connection knowledge in the form of if-then rules, as in Eq. 3.1. Let U denote the universe of all the connection objects and o denote a connection object in U. To obtain useful connection objects, rough set operations are used to obtain the approximations of U. The values of the objects in lower approximations with respect to the similarity will be used to assemble the connection rules. More applicable connection objects can also be used if the conflicts of the objects in the boundary region can be resolved.

3.3.2. Layer 2.2: Software Development Representation

The system adopts UML models to represent the software development. The following UML models are selected to represent the system product development, i.e., Element 4.

- Use case diagrams: They represent the users' interactions with the system.
- Class diagram: A set of system classes with their attributes and operations are identified and created. The diagrams represent the system structure by showing the classes and the relationships between the classes. Developers can rapidly develop the functions of the system by translating the diagrams using specific programming languages.
- Activity diagrams: They are graphical representations of workflows of stepwise activities and actions of the system. A set of activity diagrams are created to represent the main business processes of the system.

Some initial results of the software development models are developed.

BDC System Activity Diagram

The initial activity diagram of BDC system is shown in Fig. 3.7. The activities are explained as follows.

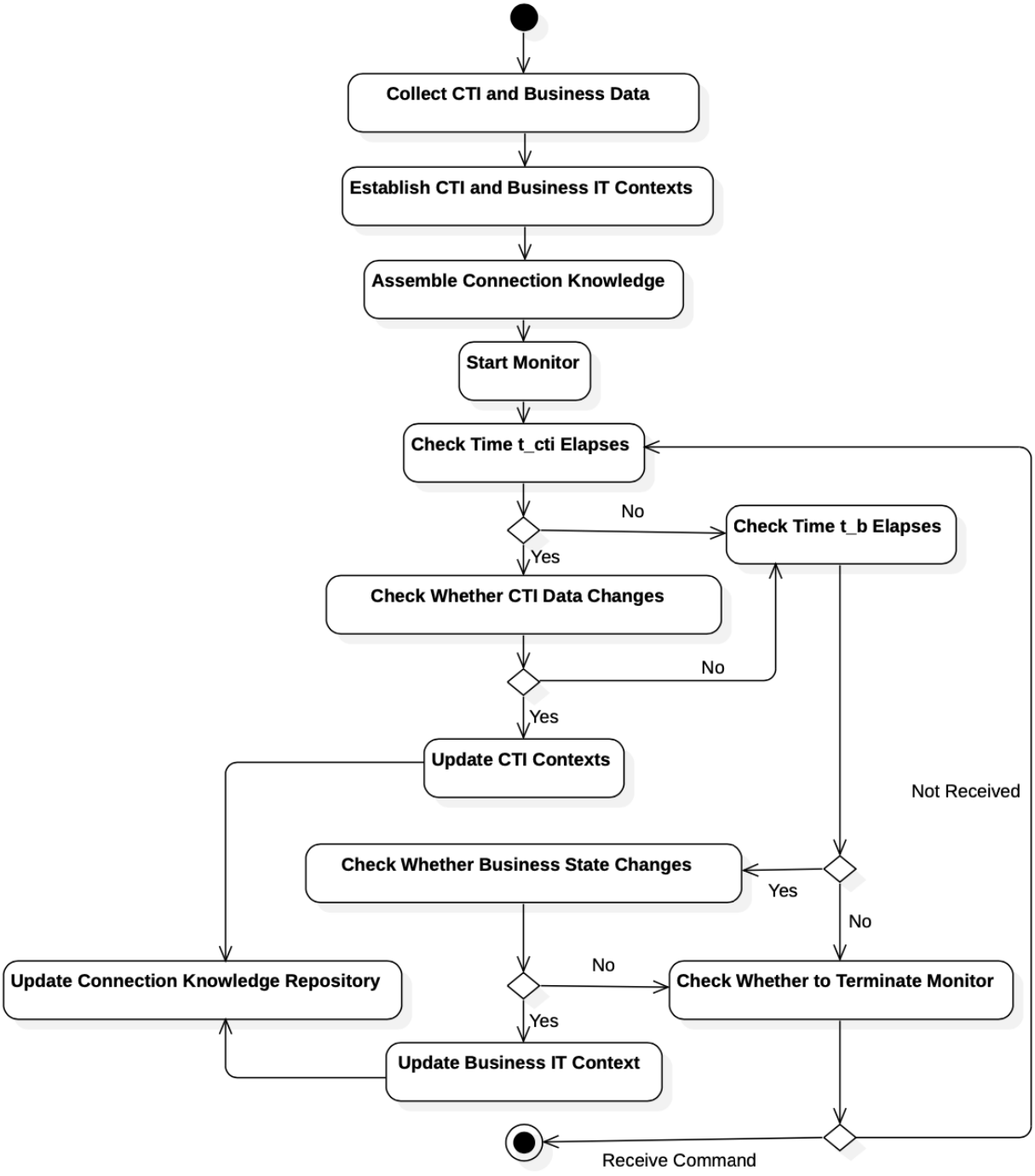

Figure 3.7: Activity Diagram of BDC System

- "Collect CTI and Business Data": The system collects CTI and business information from their sources.
- "Establish CTI and Business IT Contexts": Based on the CTI and business information collected, the system establishes the dedicated CTI and business IT contexts.
- "Assemble Connection Knowledge": The system assembles the connection rules from the contexts.
- "Start Monitor": The system starts the monitor used to periodically monitor the changes of the CTI contexts and the changes of the business IT context.
- "Check Time t_cti Elapses": The system checks whether time t_cti elapses, where t_cti is the time interval for checking the CTI changes. If yes, go to "Check Whether CTI Data Changes". If no, go to "Check Time t_b Elapses".
- "Check Whether CTI Data Changes": The system checks whether the system gets or removes CTI data. If yes, go to "Update CTI Contexts". If no, go to "Check Time t_b Elapses".
- "Update CTI Contexts": The system updates the CTI contexts based on the CTI changes from the sources.
- "Update Connection Knowledge Repository": Based on the changes of the contexts, the system updates the connection knowledge to ensure it is up to date.
- "Check Time t_b Elapses": The system checks whether t_b elapses, where t_b is the time interval for checking the business changes. If yes, go to "Check Whether Business Information Changes". If no, go to "Check Whether to Terminate Monitor".
- "Check Whether Business State Changes": The system checks whether business information collected has changed. If yes, go to "Update Business Contexts". If no, go to "Check Whether to Terminate Monitor".

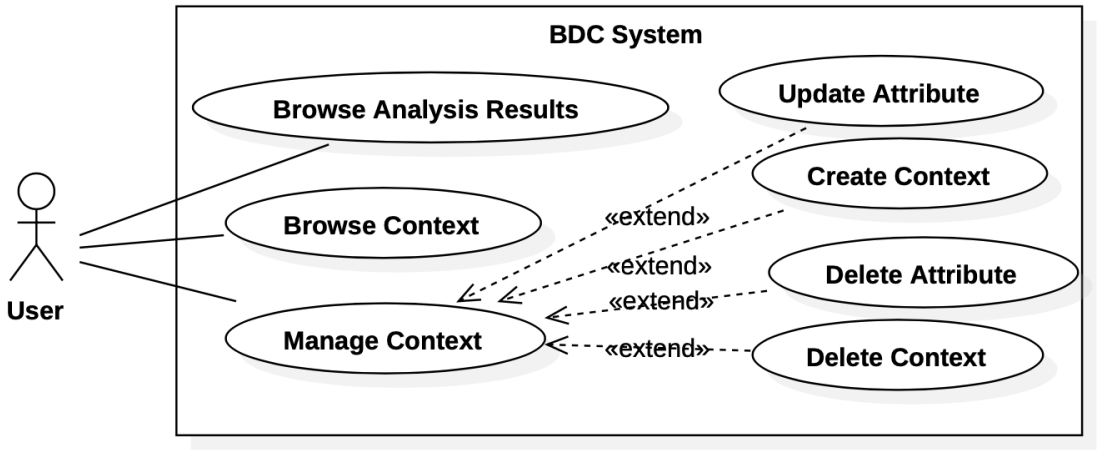

Figure 3.8: Class Diagram of BDC System

- "Update Business Context": The system updates the business context based on the changes of the business information.
- "Check Whether to Terminate Monitor": The system checks whether it receives the command to terminate the monitor. If it receives the command, go to the end point. If not, go to "Check Time t_cti Elapses".

BDC System Use Case Diagram

A use case diagram is exhibited in Fig. 3.8. A user of BDC system can "Browse Context" information, "Manage Context" and "Browse Analysis Results". When a user manages context, the user can "Update Attribute" of contexts, "Delete Attribute" of contexts, "Create Context" and "Delete Context".

BDC System Class Diagram

A class diagram of BDC system is presented in Fig. 3.9. The classes in the diagram are explained as follows.

- "Context": This class generalises the classes "BusinessContext" and "CTIContext". The classes "ConnectionObject", "BusinessContext", "CTIContext", and "ConnectionObject" extend this class.

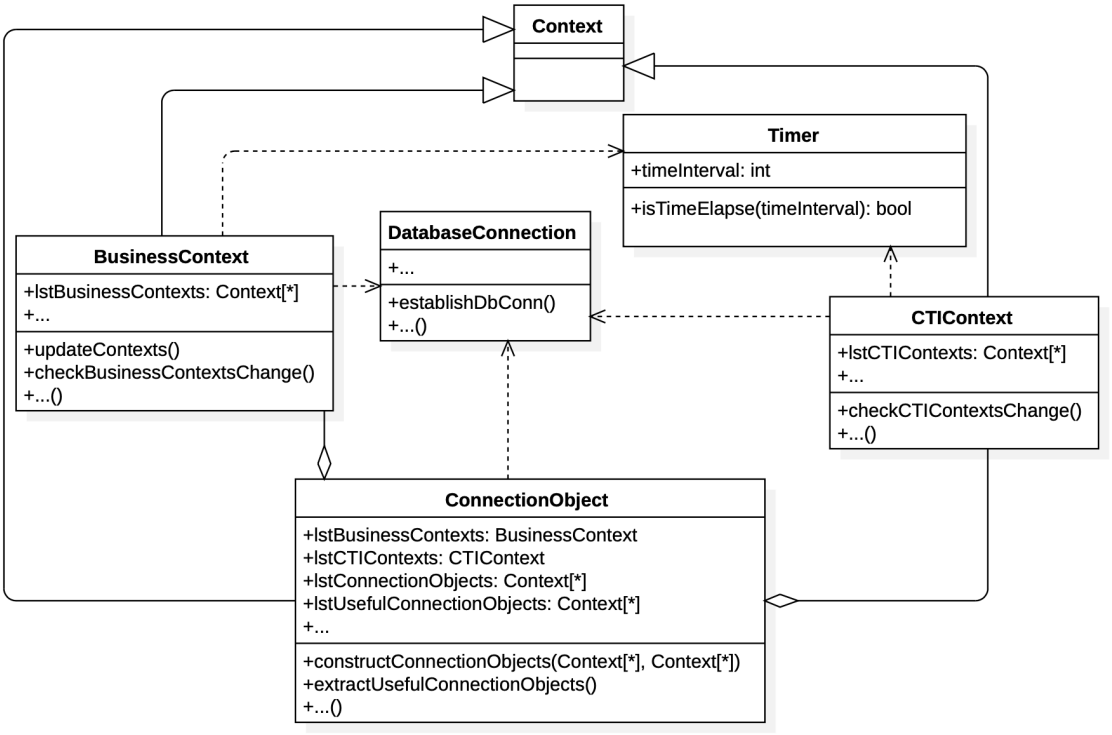

Figure 3.9: Class Diagram of BDC System

- "DatabaseConnection": This class has a function `establishDbConn()` to connect a database and other functions to perform CRUD (Create, Read, Update and Delete) operations. The classes "BusinessContext", "CTIContext", and "ConnectionObject" depend on this class. When the classes "BusinessContext", "CTIContext", and "ConnectionObject" are instantiated, this class will also be instantiated in their constructors.

- "Timer": This class models a timer to perform periodical checks every `timeInverval` using function `isTimeElapse()`. The classes "BusinessContext" and "CTIContext" depend on this class. When they are instantiated, this class will also be instantiated in their constructors.

- "BusinessContext": This class inherits "Context" class. It has an attribute `lstBusinessContexts` that is a list of contexts. It has a function `updateContexts()` to update the contexts and a function `checkBusinessContextsChange()` to check whether the states of the business IT contexts change.

- "CTIContext": This class inherits "Context" class. It has an attribute `lstCTIContexts` that is a list of contexts. It has a function `checkCTIContextsChange()` to check whether CTI contexts are created or deleted.

- "ConnectionObject": This class inherits "Context" class. In addition to the attributes `lstBusinessContexts` and `lstCTIContexts`, it has another two attributes `lstConnectionObjects` and `lstUsefulConnectionObjects` that are a list of all constructed connection objects and a list of useful connection objects, respectively, extracted from all connection objects. It has a function `constructConnectionObjects()` to construct all connection objects based on the CTI and business IT contexts, and a function `extractUsefulConnectionObjects()` to extract those connection objects with useful and applicable connection knowledge.

3.3.3. Summary

In summary, the chapter introduces the system architecture. The system aims to propose a hybrid representation to abstract and represent the four elements identified from the problem domain, i.e., CTI feed, business IT context, the connection between them and how to apply the connection strategy to a business IT context. The representation uses computational intelligence models and software development models for the representation. As for the computational intelligence models, grey numbers, fuzzy sets and rough set theory are selected. As for the software development models, use case diagrams, class diagrams and activity diagrams are selected. Finally, the initial results of the representation are presented.

The next chapter introduces the formal definitions of the static part of the representation using the selected computational intelligence models and the static part of the representation is initialised to be a rule based system. A case study then explains how the initialised rule based system obtains useful connection knowledge for CTI onto a business IT context.

Chapter 4 A Connection Knowledge Assembly System for Cyber Threat Intelligence onto Business IT Context

4.1. Introduction

The previous chapter introduces the fundamental architecture of BDC system. This chapter details and formalises the static part of the mathematical representation of the system, which is a system state at an individual time moment, using grey number theory, fuzzy set theory and rough set theory. The representation is then instantiated to be a connection rule assembly system. For the system, algorithms used to construct the connection objects and extract useful connection objects are developed. Finally, a case study demonstrates how to use the system. In the case study, the system is applied to assembling useful connection objects from NVD database [7] for a business IT context. These connection objects contain the connection knowledge that can be used to identify further relevant CTI.

4.2. The Symbol Space

This section introduces the mathematical notations used to represent the static part of the system.

Initial Notations

The following initial notation are introduced.

- n: a function used to denote and obtain the cardinality of a set. Let X be a set. $n(X)$ then denotes the cardinality of the set X.
- g: a context object that is a CTI context or a business IT context
- G: a finite non-empty set of context objects that can be described by a set of attributes, and $G = \{g_1, g_2, g_3, \cdots, g_{n(G)}\}$
- A: a finite set of attributes possessed by the objects in G, and $A = \{a_1, a_2, a_3, \cdots, a_{n(A)}\}$.
- $\mathbb{R}$: the set of real numbers
- L: a set of finite non-empty linguistic variables
- a: an attribute that is a function, $a \in A$, used to obtain and denote its value of a context object. For example, $a_1(g_1) = high$ means that the value of the attribute a_1 of the context object g_1 is the linguistic value $high$.

4.2.1. Contextual Attributes and Contextual Attribute Values

As discussed in Section 3.3.1.1, there are potential term mappings for the attributes. It is easier to know the quantity of all the IT components in a context than the quantity of one component. Thus contextual attribute values are represented by discrete grey numbers.

Definition 4.1 (Contextual Attribute Values as Grey Numbers)**.** *Let D be a subset of $\Re$ representing the value range of possible attribute values of a context object g, $g \in G$. An attribute value in D, $a(g)$, $a \in A$, is represented by a discrete grey number. The discrete grey number $a(g)^{\pm}$ is expressed as*

$$a(g)^{\pm} \in [a(g)^{-}, a(g)^{+}] = \{a(g)^{-} \leq a(g) \leq a(g)^{+}\} \tag{4.1}$$

An attribute value of a context object, represented by a grey number, needs a whitenisation to reduce its uncertainty and make it processable for the computations in the system.

Definition 4.2 (Whitenisation of Contextual Attribute Values)**.** *Let D, $D \subset \Re$, be the attribute value range of possible attribute values of a context object g, $g \in G$. Let $a(g)^{\pm}$ be a grey number representing $a(g)$, $a \in A$, in D. W is a function defined on D. If $W(a(g)^{\pm})^{\circ} < a(g)^{\circ}$, W is called a whitenisation function. $a(g)^{\circ}$ is the degree of greyness of $a(g)^{\pm}$.*

Definition 4.3 (Degree of Greyness of An Attribute Value)**.** *Let D be a subset of $\Re$ representing the attribute value range of a context g, $g \in G$. Let d_{min} and d_{max}, $d_{min}, d_{max} \in \Re$, be the minimum and maximum values of D respectively. The degree of greyness of a grey number $a(g)^{\pm}$, $a \in A$, $a(g)^{\pm} \in D$ is defined as*

$$a(g)^{\circ} = \frac{|a(g)^{+} - a(g)^{-}|}{|d_{max} - d_{min}|} \tag{4.2}$$

Obviously, $a(g)^{\circ} = 0$ iff $a(g)^{+} = a(g)^{-}$.

4.2.2. Similar Context Objects

4.2.2.1. Pre-processing of Contextual Attribute Values

According to the proposed system architecture in Section 3.3, there are different groups of fuzzy sets defined on the contexts.

One Group A fuzzy set is defined on one context object to pre-process the attribute values. The membership function is to process the initial attribute values into a value in $[0, 1]$.

Definition 4.4 (One Group A Fuzzy Set on One Context)**.** *Let U_{Ag} be the universe set of the initial attribute values of a context g, $g \in G$, and U_{Ag} is*

$$U_{Ag} = \{a_1(g), a_2(g), a_3(g), \cdots, a_{n(A)}(g)\} \tag{4.3}$$

A fuzzy set $Ag(U_{Ag})$ describing U_{Ag} is defined as a set with fuzzy numbers

$$Ag(U_{Ag}) \coloneqq \{< x, \mu_{Ag}(x) > | x \in U_{Ag}\} \tag{4.4}$$

where $\mu_{Ag} : U_{Ag} \longrightarrow [0, 1]$ *is the membership function and* $\mu_{Ag}(x)$ *is the belongingness degree of* x *into* $Ag(U_{Ag})$.

For example, the pre-processing can be the normalisation of the whitenised attribute values. Let b be a context object. It has the whitenised attribute values and $U_{Ab} = \{1, 2, 3\}$. With the normalisation, $A_b(U_{Ab}) = \{0.267, 0.535, 0.802\}$. An instantiation of Group A fuzzy sets is introduced in Section 4.3.1.2.

4.2.2.2. Similarities between Context Objects

One Group B fuzzy set is defined on all the pairs of the context objects.

Definition 4.5 (One Group B Fuzzy Set on Pairs of Contexts)**.** *Let* U_{Bs} *be the universe set of all pairs of the context objects in* G.

$$U_{Bs} = G \times G \tag{4.5}$$

A fuzzy set $Bs(U_{Bs})$ *describing* U_{Bs} *is defined as a set with fuzzy numbers*

$$Bs(U_{Bs}) \coloneqq \{< x, \mu_{Bs}(x) > | x \in U_{Bs}\} \tag{4.6}$$

where $\mu_{Bs} : U_{Bs} \longrightarrow [0, 1]$ *is the membership function and* $\mu_{Bs}(x)$ *is the belongingness degree of* x *into* $Bs(U_{Bs})$.

An instantiation of Group B fuzzy set is introduced in Section 4.3.2.1.

4.2.2.3. Linguistic Attribute Value Densities

Definition 4.6 (Multiple Group C Fuzzy Sets on One Context)**.** *Let* U_{Cg} *be the universe set of the attribute values of a context* g, $g \in G$, *after the pre-processing. The set is*

$$U_{Cg} = \{\mu_{Ag}(a_1(g)), \mu_{Ag}(a_2(g)), \mu_{Ag}(a_3(g)), \cdots, \mu_{Ag}(a_{n(A)}(g))\} \tag{4.7}$$

Let L be a set containing the linguistic values used to describe the densities of the attribute values. Multiple fuzzy sets describing U_{Cg} are defined as sets with fuzzy numbers. Each fuzzy set is corresponding to one linguistic density in L. Let i be an object in $\{1, 2, 3, \cdots, n(L)\}$. The i-th fuzzy set is

$$Cg(U_{Cg})_i \coloneqq \{< x, \mu_{Cg}i(x) > | x \in U_{Cg}\} \tag{4.8}$$

where $\mu_{Cg}i : U_{Cg} \longrightarrow [0, 1]$ is the membership function and $\mu_{Cg}i(x)$ is the belongingness degree of x into $Cg(U_{Cg})_i$.

For example, the membership functions are used to get the compatibilities to the discrete value densities. Let $L = \{low, high\}$. Because two elements are inside L, two Group C fuzzy sets then are defined on the attribute values after the preprocessing. The fuzzy set corresponding to *high* with the compatibility values to *high* is $C_b(U_{C_b})_2 = \{1, 1, 1\}$. An instantiation of Group C fuzzy sets for one context object is introduced in Section 4.3.1.3.

Definition 4.7 (Multiple Group C Fuzzy Sets on All Similarities)**.** *Let the membership function of Group B fuzzy set be a function obtaining the similarities between the context objects. Let U_{Cs} be the universe set of all the similarities. It is*

$$U_{Cs} = \{\mu_{Bs}(x_1), \mu_{Bs}(x_2), \mu_{Bs}(x_3), \cdots, \mu_{Bs}(x_{n(U_{Bs})})\} \tag{4.9}$$

Let L be a set containing the linguistic values used to describe the densities of the attribute values. Multiple fuzzy sets describing U_{Cs} are defined as sets with fuzzy numbers. Each fuzzy set is corresponding to one linguistic density in L. Let i be an object in $\{1, 2, 3, \cdots, n(L)\}$. The i-th fuzzy set is

$$Cs(U_{Cs})_i \coloneqq \{< x, \mu_{Cs}i(x) > | x \in U_{Cs}\} \tag{4.10}$$

where $\mu_{Cs}i : U_{Cs} \longrightarrow [0, 1]$ is the membership function and $\mu_{Cs}i(x)$ is the belongingness degree of x into $Cs(U_{Cs})_i$.

An instantiation of Group C fuzzy sets for the similarities is introduced in Section

4.3.2.2.

4.2.2.4. Linguistic Densities Selections

Multiple fuzzy sets have been defined on each context object and the set of all similarities, and each fuzzy set refers to a linguistic concept. However, the later computations, which assemble connection rules, in the system need suitable densities. Fuzzy set operations of Group C fuzzy sets are used for selections of the suitable densities. Definition 4.8 defines the generalised form of one density selection.

Definition 4.8 (Density Selection)**.** *Let i be an object in $\{1, 2, 3, \cdots, n(L)\}$, and let*

$$U = \sum_{i=1}^{n(L)} (C_s(U_{Cs})_i \bigcup \sum_{i=1}^{n(L)} \sum_{j=1}^{n(G)} C_{g_j}(U_{Cg_j})_i) \tag{4.11}$$

Let U' contain all sets of the membership degrees. Density selection needs to be performed for each object in U', and there is no need to perform a density selection for a set with only one membership degree. Thus $U' = 2^U - \{U\}$. Obviously, U' at least has two elements. A density section method u is then defined as

$$u : U' \longrightarrow U \tag{4.12}$$

An instantiation of u is introduced in Section 4.3.3.

4.2.3. Connection Knowledge Space

This section introduces the symbols and definitions for the connection knowledge space. Some initial symbols are introduced as follows:

- o: one connection object which is generated based on a pair of context objects
- O: a finite non-empty set of all the connection objects
- A_O: a finite non-empty set of all the attributes for the connection objects

Definition 4.9 (Information System). *Let O be the universe set of all the finite connection objects. An information system is defined as $Conn$.*

$$Conn \coloneqq (O, A_O) \tag{4.13}$$

Let V be a set containing all the attribute values of the connection objects, and

$$V = \bigcup_{a \in A_O} V_a \tag{4.14}$$

A_O is a non-empty set finite set of attributes such that $a : O \longrightarrow V_a$ for every $a \in A_O$. The last attribute of A_O is the decision attribute, and other attributes in A_O are condition attributes.

For example, $a_2(o_2) = low$ means that the value of the attribute a_2 of the connection object o_2 is the linguistic value *high*.

Definition 4.10 (Indiscernibility and Equivalence Classes). *Let $Conn$ be an information system and Q be a subset of the attributes, $Q \subseteq A_O$. An indiscernibility relation $\sim (Q)$ is defined as*

$$\sim (Q) \coloneqq \{(h, o) \in O \times O \mid \forall a \in Q,\ a(h) = a(o)\} \tag{4.15}$$

$\forall o \in O$, let

$$[o]_{\sim(Q)} = \{h \in O \mid o \sim (Q)h\} \tag{4.16}$$

$[o]_{\sim(Q)}$ is called the equivalence class of o with respect to $\sim (Q)$, which is simply written as $[o]$. All equivalence classes of O are represented by the quotient set $O/\sim (Q)$:

$$O/\sim (Q) = \{[o]_{\sim(Q)} \mid o \in O\} = \{E_1, E_2, E_3, \cdots, E_m\} \tag{4.17}$$

It can be simply written as O/Q.

Definition 4.11 (Approximations and Regions). *Let Q be a subset of the attributes, $Q \subseteq A_O$, and (O, Q) is regarded as a knowledge base. For a subset $X \subseteq O$, the Q-*

lower approximation of X is

$$Q_*(X) \coloneqq \bigcup \{E \in O/Q | E \subseteq X\} \tag{4.18}$$

The Q-lower approximation of X is also the Q-positive region of X, denoted by $POS_Q(X)$. The Q-upper approximation of X is

$$Q^*(X) \coloneqq \bigcup \{E \in O/Q | E \cap X \neq \emptyset\} \tag{4.19}$$

The Q-negative region of X is $O - Q^*(X)$, denoted by $NEG_Q(X)$. The Q-boundary region of X is $Q^*(X) - Q_*(X)$, denoted by $BN_Q(X)$.

4.2.3.1. Connection Rules

Definition 4.12 (Connection Rules)**.** *Let Ω_2 be the set containing all logical connectives of arity 2, and $\{\wedge, \Longrightarrow\} \subseteq \Omega_2$. The elements of $\{\wedge, \Longrightarrow\}$ are used for the connection rules, where $\wedge$ means logical conjunction and $\Longrightarrow$ means implication. Let f denote a fuzzy linguistic concept in L. Let p denote an atomic formula, and it is defined as*

$$p : a(o) \implies f \tag{4.20}$$

where $a(o)$ denotes an attribute value of o, $o \in O$. p means the value of $a(o)$ is f. Let Π be a set containing all the atomic formulas for the connection rules, so there are $n(A_O) \times n(L)$ formulas in Π. One connection rule is defined as

$$p_1 \wedge p_2 \wedge p_3 \wedge \cdots \wedge p_{n(x)} \implies p_{n(A_O)} \tag{4.21}$$

where $p_1, p_2, p_3, \cdots, p_{n(A_O)} \in \Pi$ and $n(x) \leq n(A_O) - 1$.

Example 4.1 (A Connection Rule Interpreting the Similarity)**.** *Let o_3 be a connection object generated by the system based on the CTI context c_3 and a business IT context b_1. o_3 has four attributes that are a_1, a_2, a_3 and similarity, where a_1, a_2, and a_3 are condition attributes and similarity is decision attribute. After the processings by the system, the attribute values of a_1, a_2, a_3 and similarity are low,*

low, high and high respectively. Thus the connection rule is

$$(a_1(o_3) \implies low) \wedge (a_2(o_3) \implies low) \wedge (a_3(o_3) \implies high) \implies (similarity \implies high) \tag{4.22}$$

The system can use the connection rule to interpret the similarity between b_1 and further CTI. After the system gets a new CTI context c_4, the system generates a new connection object o_4 based on c_4 and b_1. If the attribute values of a_1, a_2 and a_3 of o_4 are low, low and high respectively, then the system can interpret the similarity between c_4 and b_1 is high.

4.3. A Symbol Space Instantiation for the Connection Knowledge Assembly System

The representation proposed in the previous sections of the chapter defines a generalised form of the concepts involved by the connection objects assembly system. Before how the system works is discussed, the representation needs to be instantiated as dedicated and suitable notations for a specific business IT context and specific CTI feeds. The instantiation in this section can be regarded as a specific example explaining how to use the symbols and definitions in the previous sections of the chapter.

4.3.1. An Instantiation for a Business IT Context and CTI Contexts

A context g is firstly instantiated for each specific CTI feed and a specific business IT context, and g is instantiated to be b and multiple c.

- b: a specific business IT context
- c: a specific CTI context. All the CTI feeds, $c_1, c_2, c_3, \cdots, c_{n(C)}$, are contained by C.

The attributes describing the contexts are instantiated to be quantities of the IT

components of the contexts, contained by A_G.

- A_G: a set of attributes that are quantities, $A_G = \{a_1, a_2, a_3, \cdots, a_{n(A_G)}\}$. Each a, $a \in A_G$, represents the frequency of an IT component. Let $a(g)$ denote an IT component quantity, and for every $a \in A_G$

$$a(g) \in \mathbb{N}^+ \cup \{0\} \tag{4.23}$$

  where $g \in C \cup \{b\}$ and $\mathbb{N}^+$ is the set of positive natural numbers.

4.3.1.1. An Initiation for Contextual Attributes

Value Ranges and Contextual Attribute Values

The attribute value range D is instantiated as:

- D: for a context g, $g \in C \cup \{b\}$, D is a set with the integers,

$$D = \{0, 1, 2, 3, \cdots, d_{max}\} \tag{4.24}$$

  where d_{max} is the total quantity of IT components in the context g.

Afterwards, a discrete grey number $a(g)^{\pm}$, $a \in A_G$, is instantiated to represent a contextual attribute, the values of which belong to the D, for a context g, $g \in C \cup \{b\}$.

$$a(g)^{\pm} = \{a(g)^{-}, a(g)_2, a(g)_3, \cdots, a(g)_{n(a(g)^{\pm})-1}, a(g)^{+}\} \in \{0, 1, 2, \cdots, d_{max}\} \tag{4.25}$$

This means that, even if a contextual attribute might have multiple possible values at the same time, the value will not be less than 0 and larger than the quantity of the total IT components in the context.

The whitensation function W is instantiated to obtain the average value of all possible values of a contextual attribute $a(g)^{\pm}$, $a \in A_G$, $g \in C \cup \{b\}$. The function is

$$W(a(g)^{\pm}) = \frac{a(g)^{-} + a(g)^{+} + \sum_{i=2}^{n(a(g)^{\pm})-1} a(g)_i}{n(a(g)^{\pm})} \tag{4.26}$$

Obviously,

$$W(a(g)^{\pm})^{\circ} = 0 \tag{4.27}$$

4.3.1.2. An Instantiation for Group A Fuzzy Sets for Contexts

Group A fuzzy sets are instantiated for b and all $c \in C$, the membership functions of which are used to pre-process the attribute values. Let g be a context, $g \in C \cup \{b\}$. The universe set U_{Ag} is

$$U_{Ag} = \{a_1(g), a_2(g), a_3(g), \cdots, a_{n(A_G)}(g)\} \tag{4.28}$$

A Group A fuzzy set is instantiated for g. Let x_i be the i-th element in U_{Ag}. The membership function u_{Ag} is instantiated as a normalisation function. The function is

$$\mu_{Ag}(x_i) = w(x_i)\sqrt{\frac{W(x_i^{\pm})^2}{\sum_{j=1}^{n(A_G)} W(x_j^{\pm})^2}} \tag{4.29}$$

where $w(x_i) \in [0, 1]$ is the weight of x_i.

4.3.1.3. An Instantiation for Group C Fuzzy Sets for Contexts

Group C fuzzy sets are used for the linguistic densities for the attributes. The set L containing all linguistic variables is instantiated with three elements. The L is

$$L = \{high, medium, low\} \tag{4.30}$$

Group C fuzzy sets are instantiated for b and all c in C for the discrete attribute densities. Let g be a context, $g \in C \cup \{b\}$. The universe set U_{Cg} is

$$U_{Cg} = \{\mu_{Ag}(a_1(g)), \mu_{Ag}(a_2(g)), \mu_{Ag}(a_3(g)), \cdots, \mu_{Ag}(a_{n(A_G)}(g))\} \tag{4.31}$$

Three fuzzy sets, corresponding to the three linguistic variables in L, are instantiated. Their membership functions are $\mu_{Cg}1$, $\mu_{Cg}2$ and $\mu_{Cg}3$ that are used to obtain the corresponding compatibilities of an element $x \in U_{Cg}$ onto the three discrete linguistic variables in L. $\mu_{Cg}1$, $\mu_{Cg}2$ and $\mu_{Cg}3$ are corresponding to the variables *high*, *medium* and *low* respectively. The three membership functions are instantiated as three commonly used functions. Specifically, $\mu_{Cg}1$, used for *high*, is

$$\mu_{Cg}1(x) = \begin{cases} 0, x < x1 \\ \frac{x-x1}{x2-x1}, x1 \leq x \leq x2 \\ 1, x2 < x \end{cases} \tag{4.32}$$

$\mu_{Cg}2$, used for *medium*, is

$$\mu_{Cg}2(x) = \begin{cases} \frac{x-x3}{x4-x3}, x1 \leq x < x2 \\ 1, x4 \leq x < x5 \\ \frac{x6-x}{x6-x5}, x5 \leq x < x6 \\ 0, x < x3 \text{ or } x > x6 \end{cases} \tag{4.33}$$

μ_{Cg3}, used for *low*, is

$$\mu_{Cg}3(x) = \begin{cases} 1, x < x7 \\ \frac{x8-x}{x8-x7}, x7 \leq x \leq x8 \\ 0, x8 < x \end{cases} \tag{4.34}$$

In the three functions, $x1$, $x2$, $x3$, $x4$, $x5$, $x6$, $x7$ and $x8$ are all real numbers.

4.3.2. An Instantiation for Similarities

4.3.2.1. An Instantiation for Group B Fuzzy Sets for Numerical Similarities

One Group B fuzzy set is instantiated for all the pairs of c, for all $c \in C$, and b. The universe set U_{Bs} then is

$$U_{Bs} = \{\{c_1, b\}, \{c_2, b\}, \{c_3, b\}, \cdots \{c_{n(C)}, b\}\} \tag{4.35}$$

The membership function of the fuzzy set, μ_{Bs}, is instantiated to obtain the similarity between a pair $\{x, y\}$ in U_{Bs}. Eq. 3.6 is applied to the function. The function is

$$\mu_{Bs}(\{x, y\}) = 1 - \sqrt{\frac{\sum_{i=1}^{n(A_G)} w(a_i) \left(\mu_{Ag}(a_i(x)) - \mu_{Ag}(a_i(y))\right)^2}{n(A_G)}} \tag{4.36}$$

where $w(a_i)$, $w(a_i) \in [0, 1]$, is the weight.

4.3.2.2. An Instantiation for Group C Fuzzy Sets for Discrete Similarity Densities

Group C fuzzy sets are instantiated for all the similarities for the discrete similarity densities. The universe set U_{Cs} is

$$U_{Cs} = \{\mu_{Bs}(c_1, b), \mu_{Bs}(c_2, b), \mu_{Bs}(c_3, b), \cdots, \mu_{Bs}(c_{n(C)}, b)\} \tag{4.37}$$

Three fuzzy sets, corresponding to the three linguistic variables in L, are initialised. Their membership functions are $\mu_{Cs}1$, $\mu_{Cs}2$ and $\mu_{Cs}3$ that are used to obtain the corresponding compatibilities of an element $x \in U_{Cs}$ onto the three linguistic variables in L. The forms of the membership functions are similar to $\mu_{Cg}1$, $\mu_{Cg}2$ and $\mu_{Cg}3$, but different values of $x1$, $x2$, $x3$, $x4$, $x5$, $x6$, $x7$ and $x8$ are used.

4.3.3. An Instantiation for Discrete Density Selection

A density selection function is instantiated to select suitable densities. According to Definition 4.8, let a set U'' be an object in the set U', $U'' = \{\mu_1, \mu_2, \mu_3, \cdots, \mu_{n(U'')}\}$.

The density selection function u is instantiated as u_1 that is used to select only one value from U''. The function is

$$u_1(U'') = max(U'') = max\{\mu_1, \mu_2, \mu_3, \cdots, \mu_{n(U'')}\} \tag{4.38}$$

The function is to select the maximum values from U''. Based on the selected value, the corresponding discrete density in L is then selected by the assembly system.

4.3.4. An Instantiation for Connection Objects

Let o denote one connection object, let O contain all the connection objects and let A_O be the set of the attributes used for all the connection objects. The attribute value of one a for one o, $a \in A_O$ and $o \in O$, is $a(o)$. The value is one element in L, $L = \{high, medium, low\}$. The decision attribute is similarity, denoted by S, that is the last attribute in A_O, and the condition attributes are the other attributes in A_O.

4.3.4.1. Two Methods of Constructing Connection Objects

The system proposes two methods of assembling the connection objects. Fig. 4.1 and Fig. 4.2 exhibit how the two methods work. Both methods use the similarity between a CTI context and a business context to be the decision attribute. The two methods differ in how to construct the condition attributes.

Method 1

Method 1, as in Fig. 4.1, uses all the discrete densities based on Group C fuzzy sets without any further processing to be the condition attributes. The decision attribute is similarity.

To get the decision attribute value of a connection object, if *sim* denotes the

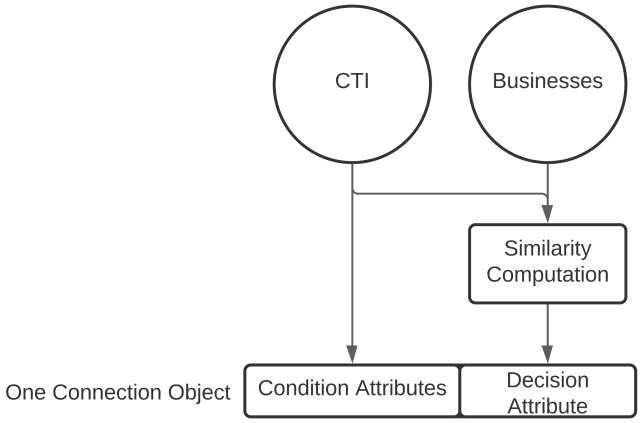

Figure 4.1: Connection Object Construction Method 1

similarity, the value of $sim(o)$, where o is generated based on a pair of c and b, is

$$sim(o) = \begin{cases} high, \text{ if } \mu_1(\{\mu_{Cs1}(x), \mu_{Cs2}(x), \mu_{Cs3}(x)\}) = \mu_{Cs1}(x) \\ medium, \text{ if } \mu_1(\{\mu_{Cs1}(x), \mu_{Cs2}(x), \mu_{Cs3}(x)\}) = \mu_{Cs2}(x) \\ low, \text{ if } \mu_1(\{\mu_{Cs1}(x), \mu_{Cs2}(x), \mu_{Cs3}(x)\}) = \mu_{Cs3}(x) \end{cases} \tag{4.39}$$

where $x = \mu_{Bs}(b, c)$, based on the instantiated density selection function.

To get the condition attribute values of the connection object, if a denotes an non-similarity attribute, the value of $a(o)$, where o is generated based on a pair of c and b, is

$$a(o) = \begin{cases} high, \text{ if } \mu_1(\{\mu_{Cg1}(x), \mu_{Cg2}(x), \mu_{Cg3}(x)\} = \mu_{Cg1}(x) \\ medium, \text{ if } \mu_1(\{\mu_{Cg1}(x), \mu_{Cg2}(x), \mu_{Cg3}(x)\} = \mu_{Cg2}(x) \\ low, \text{ if } \mu_1(\{\mu_{Cg1}(x), \mu_{Cg2}(x), \mu_{Cg3}(x)\} = \mu_{Cg3}(x) \end{cases} \tag{4.40}$$

where $x = \mu_{Ag}(W(a(c)))$, based on the instantiated density selection function.

Method 2

Method 2, as in Fig. 4.2, uses discrete relevances between all attributes of the contexts to be the condition attributes. S' is defined to obtain each discrete relevance between two attribute values of one attribute in A_G for two contexts.

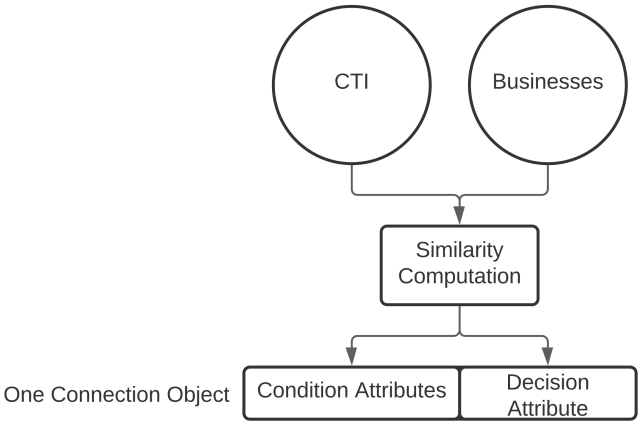

Figure 4.2: Connection Object Construction Method 2

Example 4.2 (An Example of S')**.** *Based on Group C fuzzy sets for contexts, each attribute of a context g, $g \in C \cup \{b\}$, has a discrete density value which is a value in $L = \{high, medium, low\}$. As for a pair of attribute values, $a(c)$ and $a(b)$, S' is defined as*

$$S'(a(c), a(b)) = \begin{cases} high, d = 0 \\ medium, d = \frac{1}{3} \\ low, d = \frac{2}{3} \end{cases} \tag{4.41}$$

where

$$d = \frac{|f(a(c)) - f(a(b))|}{n(L)} \tag{4.42}$$

where $f : L \longrightarrow \{1, 2, 3\}$ and $f(high) = 3$, $f(medium) = 2$, $f(low) = 1$. Additionally, it is arbitrarily defined that the result of S' is low when both of the attribute values are low.

4.3.4.2. Different Abilities to Reflect Contextual Changes

The two methods have different abilities to reflect the business IT contextual changes. When business IT contexts changes, in Method 1, only the decision attribute values, i.e., the similarity, might change, and in Method 2, all the attribute values might change.

4.4. The Connection Knowledge Assembly System for CTI Contexts onto Business IT Context

Using the instantiated concepts in the previous section, this section introduces the connection knowledge assembly system. This section introduces the workflow of the system, the algorithms developed for the connection knowledge extraction and finally a case demonstrating how the system works.

4.4.1. The Connection Knowledge Assembly Workflow of the System

The workflow of the connection knowledge assembly is separated into the following steps.

- Step 1: Retrieve CTI data from CTI data sources. The system prepares CTI feeds that can be gathered through multiple ways divided into "Automatic" and "Manual". The CTI data collected is in the format of JSON or XML. The system has a parser that is used to process the JSON or XML files and to retrieve the elements inside the files. The collection methods are
  - Automatic CTI data collection: The system has dedicated RESTful APIs used to collect the CTI data in the format of JSON or XML from the sources that are developed based on REST software architecture.
  - Manual CTI data collection: The system users manually download the CTI in the format of JSON or XML from the sources that allow their users to download.
- Step 2: Gather business-wide IT information. The system collects the IT components from the business IT environment. The collection ways are divided into "Automatic" and "Manual". They are
  - Automatic collection of business IT components: The system has scripts developed to collect the IT components from the business IT environment. For example, in Unix-like OS, a bash shell is used to gather the

information.

  - Manual: The system users can input the words of the IT components into the system.

- Step 3: Assemble the connection objects. The system firstly constructs the context objects and then assembles the connection objects.

  - Step 3.1: Construct the context objects. The system uses a quantity-based method to construct the attributes of the context objects. With a set of terms, such as "Windows 10", a CTI context and a business IT context have the frequencies of the terms. The system firstly whitenises all possible values for all quantity-based attributes of all the context objects. The system then uses the membership functions of the Group A fuzzy sets to normalise the whitenised values. To get the discrete attribute values, the system uses the membership functions of the Group C fuzzy sets to get the compatibility values to the linguistic densities. After that, the system selects the linguistic densities with the highest compatibilities for the attribute values of the terms of the context objects.

  - Step 3.2: Construct all the connection objects. The system uses the membership function of the Group B fuzzy set to obtain the similarities between the context objects. For the similarities, the system uses membership functions of the Group C fuzzy sets to get the compatibility values to the linguistic densities. The system then selects the densities with the highest compatibilities for the similarities. After that, the system uses the Method 1, introduced in Section 4.3.4.1, for the attribute values of the connection objects. The condition attributes of the connection objects are the discrete attribute values of the CTI contexts, and the decision attribute is similarity.

  - Step 3.3: Extract the useful connection objects. The system uses the rough set operations to get the connection objects from the lower approximation. To get more useful and applicable connection objects, the system then calculates the quantities of the connection objects in the

boundary region and selects the connection objects with higher quantities.

With the attributes and densities in the useful and applicable connection objects obtained, the system can reason the relevances between this business IT environment and further CTI data.

A Parser for Processing CTI Data

The remaining part of this section introduces a parser used to process the CTI data in the format of JSON. Algorithm 1 describes the parser. Because a JSON file has an unknown depth with nested dictionaries and lists, the algorithm is a recursive method used to get the JSON values. In Algorithm 1, *ctiFeed* denotes a CTI feed in

Algorithm 1: Recursive CTI Feed Parsing

```
1  Function ParseCTIFeed(dic):
       Input     : ctiFeed
       Initialise: dic ← ctiFeed
2      foreach dicK, dicV in cti.Items() do
3          if GetType(dicV) is dictionary then
4              ParseCTIFeed(dicV)
5          else if GetType(dicV) is list then
6              if Length(dicV) ≠ 0 then
7                  for i ← 0 to Length(dicV) − 1 do
8                      if dicV[i] is dictionary then
9                          ParseCTIFeed(dicV[i])
10                     else
11                         AddToCTITermList(dicV[i])
12         else
13             AddToCTITermList(dicV)
```

JSON type, which is a CTI context. "`.Items()`" is to get all the pairs of the JSON keys and JSON values. For example, in Python, it could be "`for dicK, dicV in cti.Items():`". "`GetType()`" is to get the type of either a dictionary or a list. For example, in Python, to check whether an element is a list, "`isinstance(dicv, list)`" returns a bool value. "`Length()`" is to get the number of the elements of

a list. "$dicV[i]$" denotes the i-th element of $dicV$. The algorithm adds the JSON values to a list outside the algorithm through "`AddToCTITermList()`". With this algorithm, each CTI feeds in JSON format can be parsed to be a list containing the JSON values.

4.4.2. The Algorithm for the Construction of the Connection Objects

An algorithm for the construction of the connection objects is developed, as in Algorithm 2. The algorithm aims to construct the connection objects based on the business IT context and the available CTI contexts. The symbols used in the algorithm are according to the instantiation in Section 4.3. Step 1 is to perform a whitenisation by averaging all the possible quantity-based attribute values for all the context objects. Step 2 is to pre-process the whitenised values, by normalising the values, using the membership functions of the Group A fuzzy sets for all the context objects. Step 3 is to compute the numerical similarities for the contexts. Step 4 and 5 are to compute the compatibilities for the attribute values to the linguistic densities for all the context objects. Step 6 is to compute the compatibilities for the similarities to the linguistic densities. Step 7 is to select the highest compatibilities for the attribute values to the discrete densities. Step 8 is to use the Method 1, as in Section 4.3.4.1, to construct the connection objects.

4.4.3. The Algorithm for the Extraction of Applicable Connection Objects

An algorithm for the extraction of applicable connection objects with useful connection knowledge is developed, as in Algorithm 3. The symbols used in the algorithm are according to the instantiation in Section 4.3. In the algorithm, S is the similarity attribute. The algorithm aims to get the applicable connection objects from the approximations. To get more connection objects, the quantities of the uncertain connection objects, whose condition attribute values are all the same but decision attribute values are different, are calculated. After that, the connection objects with

Algorithm 2: The Construction of Connection Objects

Input : b: the business IT context
C: all available CTI contexts
A_G: the attributes describing the contexts, i.e., the terms of the IT components

Output : O: all constructed connection objects with discrete attribute values generated based on b and C

1 Perform a whitenisation on the grey numbers for each attribute a, $a \in A_G$, of b and all $c \in C$, using $W(a(g)^{\pm}) = \frac{a(g)^{-} + a(g)^{+} + \sum_{i=2}^{n(a(g)^{\pm})-1} a_i}{n(a(g)^{\pm})}$, i.e., Eq. 4.3.1.1, $g \in C \cup \{b\}$.

2 Normalise the whitenised quantities using $\mu_{Ag}(x) = \frac{W(x^{\pm})^2}{\sum_{i=1}^{n(A_G)} W(x_i^{\pm})^2}$ for all $x \in U_{A_g}$ for all $g \in C \cup \{b\}$.

3 Compute the numerical similarity scores using $\mu_{Bs}(\{g, b\}) = 1 - \left(\frac{1}{n(A_G)} \sum_{i=1}^{n(A_G)} \left(\mu_{Ag}(a_i(g)) - \mu_{Ag}(a_i(b))\right)^2\right)^{\frac{1}{2}}$, i.e., Eq. 4.3.2.1, for all $g \in C$ using the normalised whitenised quantities.

4 Compute the membership degree of $\mu_{A_b}(x)$ into $C_b(U_{C_b})_1$, $C_b(U_{C_b})_2$ and $C_b(U_{C_b})_3$, i.e., $\mu_{C_g}1(\mu_{A_b}(x))$, $\mu_{C_g}2(\mu_{A_b}(x))$ and $\mu_{C_g}3(\mu_{A_b}(x))$ for all $x \in U_{A_b}$.

5 Compute the membership degree of $\mu_{A_g}(x)$ into $C_g(U_{C_g})_1$, $C_g(U_{C_g})_2$ and $C_g(U_{C_g})_3$ for all $x \in U_{A_g}$ for all $g \in C$.

6 Compute the membership degree of y into $C_s(U_{C_s})_1$, $C_s(U_{C_s})_2$ and $C_s(U_{C_s})_3$ for all $y \in U_{B_s}$.

7 Perform density selection for U'' using the instantiated density selection function $u_1(U'')$ for all

$$
\begin{aligned}
U'' \in \{&\{\mu_{C_b}1(\mu_{A_b}(a_1(b))), \mu_{C_b}2(\mu_{A_b}(a_1(b))), \mu_{C_b}3(\mu_{A_b}(a_1(b)))\},\\
&\dots\\
&\{\mu_{C_b}1(\mu_{A_b}(a_{n(A_G)}(b))), \mu_{C_b}2(\mu_{A_b}(a_{n(A_G)}(b))), \mu_{C_b}3(\mu_{A_b}(a_{n(A_G)}(b)))\},\\
&\{\mu_{C_{c_1}}1(\mu_{A_{c_1}}(a_1(c_1))), \mu_{C_{c_1}}2(\mu_{A_{c_1}}(a_1(c_1))), \mu_{C_{c_1}}3(\mu_{A_{c_1}}(a_1(c_1)))\},\\
&\dots\\
&\{\mu_{C_{c_1}}1(\mu_{A_{c_1}}(a_{n(A_G)}(c_1))), \mu_{C_{c_1}}2(\mu_{A_{c_1}}(a_{n(A_G)}(c_1))),\\
&\mu_{C_{c_1}}3(\mu_{A_{c_1}}(a_{n(A_G)}(c_1)))\},\\
&\dots\\
&\{\mu_{C_{c_{n(C)}}}1(\mu_{A_{c_{n(C)}}}(a_{n(A_G)}(c_{n(C)}))), \mu_{C_{c_{n(C)}}}2(\mu_{A_{c_{n(C)}}}(a_{n(A_G)}(c_{n(C)}))),\\
&\mu_{C_{c_{n(C)}}}3(\mu_{A_{c_{n(C)}}}(a_{n(A_G)}(c_{n(C)})))\},\\
&\{\mu_{C_s}1(\mu_{B_s}(c_1, b), \mu_{C_s}2(\mu_{B_s}(c_1, b), \mu_{C_s}3(\mu_{B_s}(c_1, b)\},\\
&\{\mu_{C_s}1(\mu_{B_s}(c_2, b), \mu_{C_s}2(\mu_{B_s}(c_2, b), \mu_{C_s}3(\mu_{B_s}(c_2, b)\},\\
&\dots\\
&\{\mu_{C_s}1(\mu_{B_s}(c_{n(C)}, b), \mu_{C_s}2(\mu_{B_s}(c_{n(C)}, b), \mu_{C_s}3(\mu_{B_s}(c_{n(C)}, b)\}\}
\end{aligned}
$$

8 Construct a connection object o based on c and b for all $c \in C$ according to the Method 1 and add o to O.

9 Return O.

Algorithm 3: The Extraction of Applicable Connection Objects

Input : O: all the constructed connection objects
A_O: attributes for the connection objects

Output : K: the set containing the applicable connection objects with useful connection knowledge
Un_K: the set containing the uncertain connection objects

Initialise: $U\_1 \leftarrow U$, $Q \leftarrow A_O - \{S\}$

1 Remove the repetitive connection objects from U: $U \leftarrow$`RemoveRepeats(`U`)`, and remove the repetitive similarity densities from V_S, i.e., $V_S \leftarrow$`RemoveRepeats(`V_S`)`.

2 For each density value v in V_S: $G \leftarrow U$.`SelectTargetObjects(`S`)`, K.`Add(`$Q_*(G)$`)`.

3 For each density value v in V_S: $G \leftarrow U$.`SelectTargetObjects(`S`)`, $Un\_K \leftarrow Q^*(G) - Q_*(G)$.

4 For each object in Un_K: count the frequency in U_1.

5 Use the frequencies counted in step 4 to determine the certain objects in Un_K with higher frequencies, and remove the certain objects from Un_K and add them to K.

6 Return K, Un_K.

higher quantities are also considered as useful.

4.4.4. A Case Study

This case study demonstrates how the connection object assembly system works. A prototype of the system is developed to carry out the case study. In this case study, the system assembles the useful connection objects from all the vulnerabilities of NVD database [7], and until 11 March 2020, the database has 140959 vulnerabilities. After the system extracts the useful connection objects for a business IT context, the knowledge described in the connection objects can be used to identify further CTI.

4.4.4.1. Prepare Data for Context Objects

The system needs to prepare data used for the computations.

The CTI Data

The vulnerabilities are manually downloaded in JSON format, and only the JSON values, rather than the JSON keys, are used for this case study. The parser, as in Algorithm 1, is used to process the downloaded JSON files for the system to get the JSON values.

The Business IT Context Data

The business context is with the following items and the frequencies that are placed into a dictionary format: {"windows": 6, "macos": 6, "tp-link": 5, "linux": 6, "freeBSD": 6, "microsoft": 12, "word": 10}.

To test the model of the attributes represented by grey numbers, a synonym dictionary is attached to the system that informs more information. The dictionary tells:

- "windows10" and "windows 10" are synonyms for "windows".
- "unix", "unix-like" are synonyms for "linux".

All the possible values for one attribute will be averaged by the corresponding algorithm steps. For example, "unix" frequency is 3, "unix-like" is 3, and "linux" frequency is 4, so the final frequency is 3.3333.

4.4.4.2. Process the Data for the Context Objects

The system constructs a CTI context object for each vulnerability and a business IT context object for the business IT environment.

The Toolkit Used to Process the Data

The prototype uses Python as the programming language. It has a large number of easy-to-use, powerful and comprehensive libraries for the computation tasks. One of them is pandas [52] that provides out-of-the-box features to manipulate a

dataframe that is a data structure of pandas. Pandas also provides useful functions that work on the dataframe. One important function of them is `groupby`. It can be used to get the equivalence classes of a dataframe.

Construct the Attributes for the Context Objects

The attributes of these context objects are the frequencies of the IT components. With the business IT contextual information, the attributes for the similarity computation are established for the system that are "frequency of windows", "frequency of macos", "frequency of tp-link", "frequency of linux", "frequency of freeBSD", "frequency of microsoft", and "frequency of word". These attributes are used for both the context types and the similarity computation.

Assign Numerical Values to the Attributes of the Context Objects

The system whitenises all the initial attribute values by averaging all the possible quantities of the attributes. Table 4.1 shows some connection objects with the numerical values. Before the system processes these values into discrete densities, one Group A fuzzy set is defined on the quantity-based attributes of each context object and the membership function is to normalise the values.

Assign Discrete Density Values to the Attributes of the Context Objects

To obtain the discrete attribute values, the system initialises three linguistic values that are *high*, *medium* and *low* representing the value densities. For example, an attribute value with *high* means that attribute density is high. Three Group C fuzzy sets, corresponding to the three linguistic variables, are then defined on the normalised attribute values for each context object. Their membership functions are used to obtain the compatibilities to the linguistic densities. Each attribute of a context object has three compatibility values corresponding to three densities. To select appropriate densities, u_1 is used to select the maximum value among the three compatibility values. If two compatibilities are equal, the system chooses the density

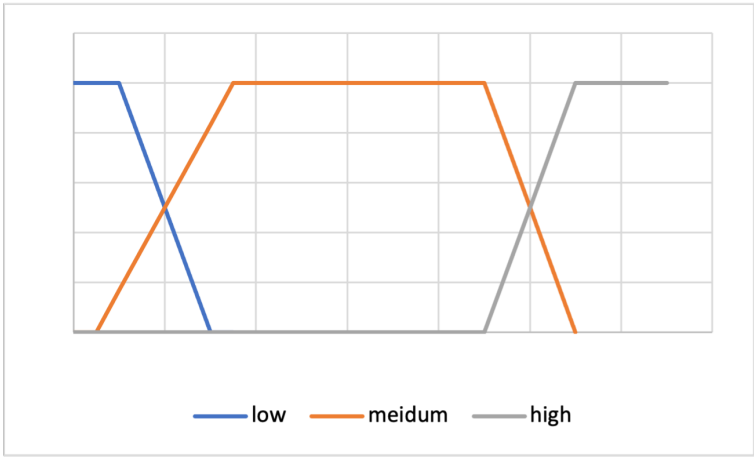

Figure 4.3: Density Membership Function Graph 1

with *high* and *low* rather than *medium*. An example of the three membership functions used to get the compatibilities to attribute value densities is shown in Fig. 4.3. Table 4.2 shows some connection objects with the numerical values.

4.4.4.3. Process the Data for the Connection Objects

The system processes one CTI context object and the business IT context object into one connection object.

Construct the Attributes for the Connection Objects

The system uses the Method 1, as in Section 4.3.4.1, to construct the attributes for the connection objects. The system constructs the condition attributes of the connection objects using the discrete densities of the attribute values of CTI context objects without any further processing, and the system constructs the decision attributes for the connection objects using the similarity densities between the context objects.

To get the numerical similarities, one Group B fuzzy set is defined on all the pairs of CTI context and the business IT context. The membership function is to obtain the similarity. To get the discrete similarities, three Group C fuzzy sets, corresponding to three linguistic variables *high*, *medium* and *low* are defined on all the pairs of CTI context and the business IT context. The membership functions

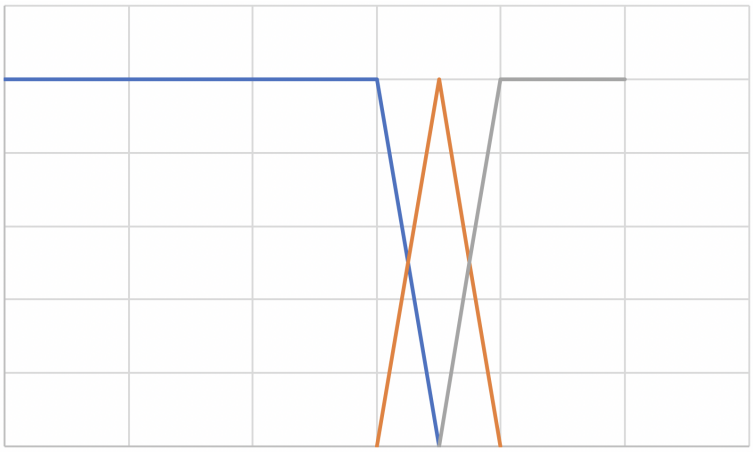

Figure 4.4: Density Membership Function Graph 2

are to obtain the compatibilities to the linguistic densities. Each similarity has three compatibility values. u_1 is used to select the maximum value among the three values. An example of the three membership functions used to get the compatibilities to similarity densities is shown in Fig. 4.4. Table 4.2 shows some objects with the density values.

4.4.4.4. Extract the Applicable Connection Objects

With Algorithm 3, the system firstly removes the repetitive connection objects. There are only 62 connection objects after the removal. The system then extracts 48 certain connection objects and 14 uncertain connection objects. All the uncertain connection objects are listed in the Table 4.3. After counting the frequencies of these uncertain objects, the system determines 6 more useful objects, but the system can not determine which one to choose between the connection objects with ID 7 and 8 because they have the same frequency.

4.5. Discussion

The comparison of the effectiveness of the system is discussed in this section.

Similar to the the existing systems [14] [3] [5], the system uses relevance computation method to be the first step for the later analysis using CTI. All of them aim

to help businesses identify CTI relevant to their IT context.

The system has differences comparing with the existing representative systems. While [5] generates a numerical value representing the relevance, the system generates a discrete relevance value instead. The discrete value is a linguistic explanation on the relevance. While MISP [14] maps the contexts onto each other when the two contexts have an identify IT component, the system generates a semantic relevance based on all possibly identical IT components. While them [5] [14] consider a specific instance of the connection between CTI and business, the system proposes a representation that builds a generalised form of all cases.

The system is regarded as a specific state at a time moment according to the dynamic view presented in [3]. The system provides a novel view of considering the IT components used for the relevance computation. It is easy to know the quantity of the IT components inside a context, but it is difficult to know the certain quantity of one IT component because potential term mappings might exist in the CTI contexts and business IT context. Due to the advance of grey numbers [36] [50] [51], they are good at the representation of an unknown value within a known scope. Therefore, the system novelly uses grey numbers to represent the IT component quantity based attributes.

4.6. Summary

In this chapter, a novel representation using computational intelligence models for business IT context oriented CTI is proposed. Discrete grey numbers are used for representing the attributes of the CTI and business contexts. Different groups of fuzzy sets are defined and used for representing the CTI and business IT contexts, and fuzzy set operations can be used to select appropriate densities for the attributes. Afterwards, for one business context, each connection object is constructed based on a CTI context. For the construction of the connection objects, two methods are identified to construct the attributes of the connection objects. One method is to use the discrete attribute values of the CTI contexts directly to be the condition attribute values of the connection objects, and another method is to compare the

values of each attribute of both context objects to get the condition attribute values. After all the connection objects are constructed, rough set operations are used to extract the useful connection objects.

The representation then can be instantiated to be a connection knowledge assembly system. The workflow of the instantiated system is also introduced in the chapter. To test the feasibility of the representation and the instantiated system, a prototype of the system is developed, and a case study is carried out to demonstrate how the system works. In the case study, the prototype is applied to assemble the useful connection objects from NVD database [7] for a business context. IT component quantities are used to be the attributes of the contexts, and the similarities between the contexts are computed based on these attributes. These numerical attribute values of the contexts and numerical similarities are then reasoned to be discrete density values, and the connection objects are constructed using the Method 1, i.e., using all the discrete CTI attribute values to be the condition attribute values and the discrete similarity densities to be the decision attribute values. The system then extracts applicable connection objects from the lower approximations. To get more useful connection objects, the quantities of the connection objects in the boundary region are calculated, and the connection objects with higher quantities are also considered as useful ones. Finally, the system proves the feasibility of the representation, and the system successfully obtains 48 useful connection objects with the useful connection knowledge. The attributes in the applicable connection objects can be used as the conditions to identify further CTI.

This chapter establishes the static part of the representation using computational intelligence models. In the next chapter, the dynamic part of the the mathematical representation will be introduced, and an incremental updating strategy is developed to update the approximations when the information system changes.

Table 4.1: Some Constructed Connection Objects

conn_obj_id[a]	**fre**[b]**_windows**	**fre_macos**	**fre_tp_link**	**fre_linux**	**fre_freebsd**	**fre_microsoft**	**fre_word**	**sim**[c]
o_1	0	0	0	0	40	0	0	0.8744
o_2	0	0	0	0.3333	0	0	0	0.3782
o_3	0	0	0	2	0	0	0	0.7456
o_4	0.6667	0	0	0	0	1	0	0.6557
o_5	0	3	0	0	0	1	0	0.787
o_6	1.3333	0	0	0	0	2	0	0.7658
o_7	0.3333	0	0	0	0	1	0	0.6479
...								

[a] A CTI context is a vulnerability of NVD database [7].
[b] fre stands for frequency.
[c] similarity

Table 4.2: Some Connection Objects after the Value Discretisation

conn_obj_id	des[a]_windows	des_macos	des_tp_link	des_linux	des_freebsd	des_microsoft	des_word	des_sim
o_8	medium	low	low	low	low	medium	low	high
o_9	medium	low	low	low	low	high	low	high
o_{10}	high	low	low	low	low	high	low	high
o_{11}	low	low	low	medium	low	low	low	high
o_{12}	low	low	low	medium	low	low	low	high
o_{13}	low	low	low	low	low	low	low	low
o_{14}	low	low	low	low	low	low	low	medium
				...				

[a]des stands for density.

Table 4.3: All Uncertain Connection Objects

ID	des_windows	des_macos	des_tp_link	des_linux	des_freebsd	des_microsoft	des_word	des_sim	Num	Coverage
1	low	low	low	low	medium	low	low	high	322	0.228%
2	low	low	low	low	medium	low	low	medium	64	0.045%
3	low	low	low	medium	low	low	low	medium	980	0.695%
4	low	low	low	medium	low	low	low	high	1370	0.971%
5	low	low	medium	low	low	low	low	high	81	0.057%
6	low	low	medium	low	low	low	low	medium	23	0.016%
7	low	medium	low	low	low	low	low	medium	17	0.012%
8	low	medium	low	low	low	low	low	high	17	0.012%
9	medium	low	low	low	low	low	low	medium	113	0.080%
10	medium	low	low	low	low	low	low	high	186	0.132%
11	medium	low	low	medium	low	low	low	high	12	0.0085%
12	medium	low	low	medium	low	low	low	medium	13	0.0092%
13	low	low	low	low	low	low	low	low	114329	81.111%
14	low	low	low	low	low	low	low	medium	11295	8.013%

Chapter 5 An Incremental Updating Strategy for Cyber Threat Intelligence in Dynamic Business IT Context

5.1. Introduction

In this chapter, a dedicated incremental updating strategy for CTI in dynamic business IT context is proposed. Based on the approach introduced in the previous chapter, the system generates one connection object for a business IT context using one CTI context. Rough set operations can be used to extract applicable connection objects from all the constructed connection objects.

However, the generated connection objects are not always static because the business IT context keeps changing over time due to its internal business operations. These changes will lead to the changes of the attribute values of the connection objects. Regarding the CTI contexts, the CTI feeds collected from CTI sources are not always unchanged. In the working environment after a deployment, the system will absorb more new CTI feeds or delete the obsolete CTI feeds. These changes will result in the changes of the quantity of the constructed connection objects. After these changes, the system will need to optimise the extracted applicable connection objects used for generating the useful connection knowledge. Because the extraction process is based on the computation using the equivalence classes, lower and upper approximations of the information system, the system will correspondingly re-compute these structures every time the system knows the information system has been changed. However, due to the high volume of CTI feeds, the re-computation

usually takes long time. Therefore, a rough set based incremental updating strategy is proposed. Using the strategy, the system does not need to re-compute every time it knows the changes, and the system will incrementally update the internal structures, i.e., the equivalence classes, lower and upper approximations, based on the previous computation result.

The strategy considers the incremental updates of the equivalence classes, lower and upper approximations when one or multiple connection objects are inserted into the information system, one or more connection objects are deleted from the information system and multiple attribute values of the connection objects are changed in the information system. Corresponding propositions of the incremental updating operations are proposed in this chapter, and their computation complexities and efficiencies are analysed towards the end of the chapter.

5.2. Dynamic Context Objects

The changes of the context objects might cause the changes of the connection knowledge space.

Definition 5.1 (Moments When the Changes Take place)**.** *Let t denote the time moment at which the system state changes. A countable and infinite set T is defined to contain all the time moments. Let $t + t$ denote the next time moment from t for all $t \in T$. Any element e with respective to the system has a state at t denoted by $e(t)$.*

The changes of the context objects then can be defined.

Definition 5.2 (The Changes of the Context Objects)**.** *Let T be the set of the time moments at which the changes of the context objects take place, let b denote a business IT context, let C be the set of all the CTI contexts, let A_G be the set of attributes describing the context objects. The changes that might lead to the changes of the connection knowledge space are defined as*

- *Insertion or deletion of the attributes describing the context objects: $n(A_G(t+1)) \neq n(A_G(t))$, $t \in T$*

- *Update of the attributes describing the context objects:* $\exists a \in A_G,\ a(t+1) \neq a(t),\ t \in T$
- *Insertion or deletion of CTI context objects:* $n(C)(t+1) \neq n(C)(t),\ t \in T$
- *Update of the attribute values of the business context object:* $\exists a \in A_G,\ a(b)(t+1) \neq a(b)(t),\ t \in T$

n is used to denote the cardinality.

5.3. The Monitor and Trigger Principles

This section introduces what to be monitored by the monitor mechanism and what operations the trigger mechanism performs, guiding the further design and development of the two mechanisms.

5.3.1. Identification of Incremental Updating Operations

Fig. 5.1 describes the principles of the monitor and trigger mechanisms. The monitor mechanism monitors the changes on the connection objects because of the changes of the business IT contexts and CTI contexts. According to Definition 5.2, attribute values of the business IT context might change; new CTI contexts can be created because the system gets new CTI feeds; existing CTI contexts can be deleted because they might be outdated and no longer be needed. These change will lead to the changes of the information system, which is listed as follows.

- One connection object is added because a new CTI context is created.
- One connection object is deleted because a new CTI context is deleted.
- Multiple connection objects are added because multiple CTI contexts are created.
- Multiple connection objects are deleted because multiple CTI contexts are deleted.

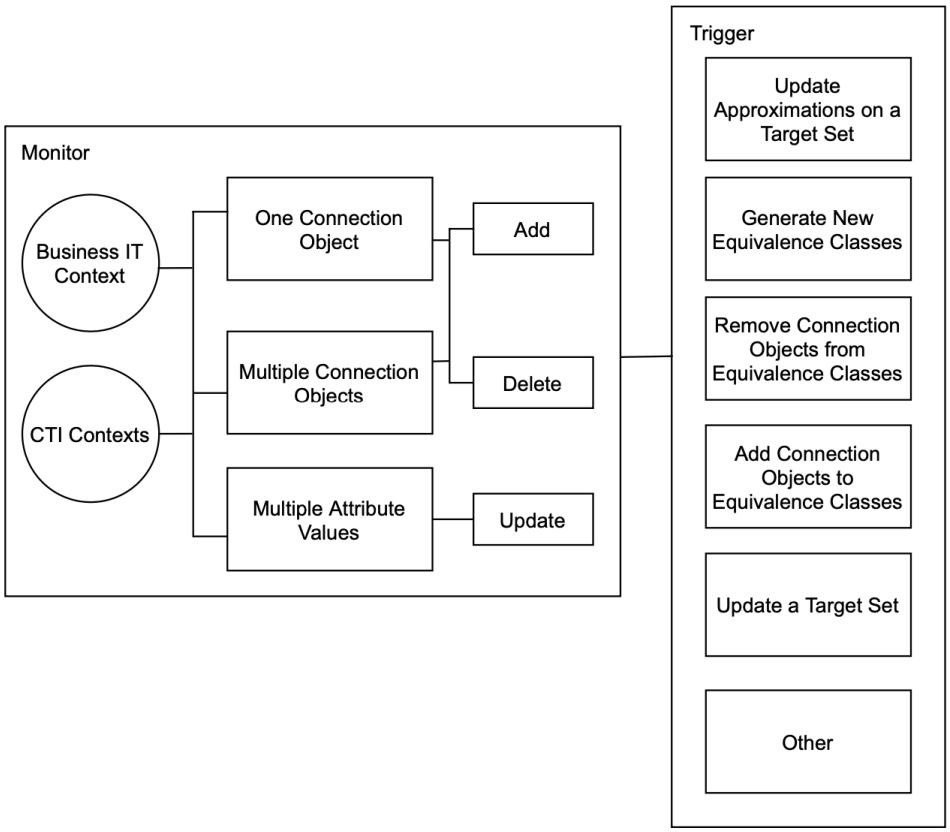

Figure 5.1: The Principles of the Monitor Mechanism and the Trigger Mechanism

- Multiple attribute values of the connection objects are updated because of the changes of the attribute values of the business IT context object.

After the monitor mechanism captures the changes of the context objects, it informs the trigger mechanism to trigger the changes listed as follows.

- U1. Generate new equivalence classes
- U2. Remove connection objects from existing equivalence classes
- U3. Add connection objects to existing equivalence classes
- U4. Update approximations with respect to a target set
- U5. Update a target set
- U6. "Other" includes other updates related to rough set theory, including the updates of cores, etc.

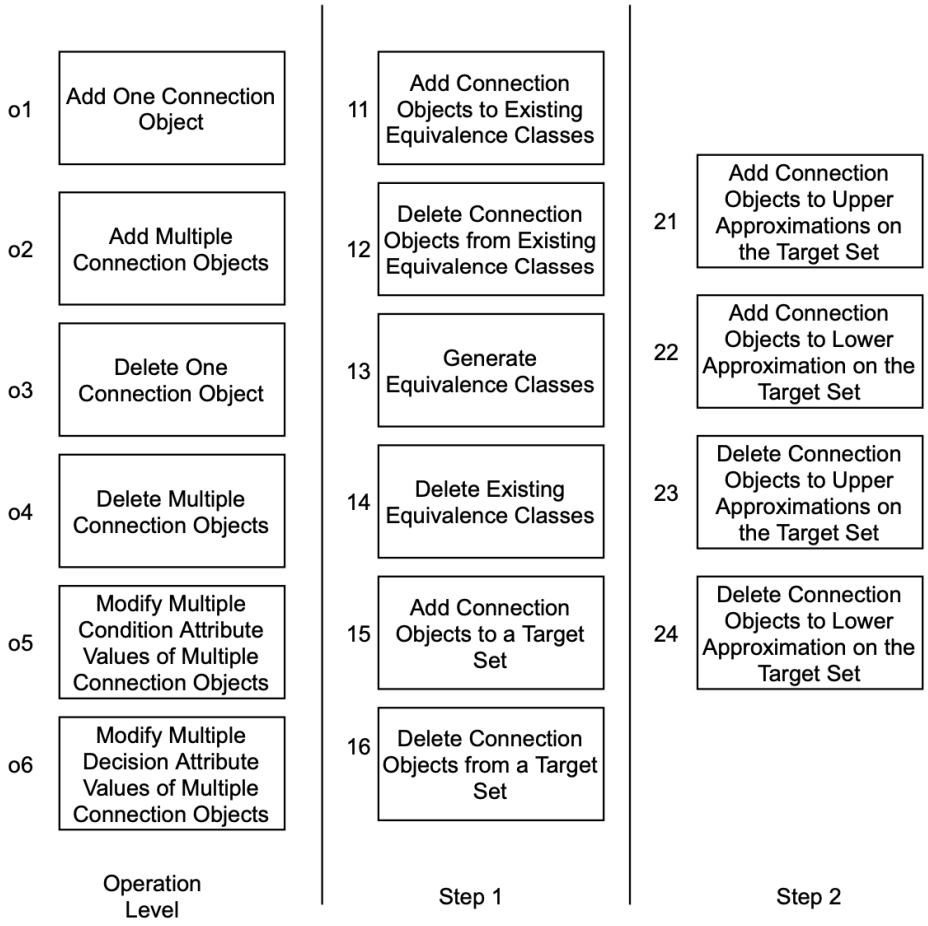

Figure 5.2: Identification of the Changes of the Internal Structures

How to carry out U1 to U5 will be developed and analysed in the remaining part of this chapter.

5.3.2. Chains of the Connection Knowledge Space Changes

After the incremental updating operations are identified, the chain for each operation, showing the sequence of the changes, is then identified.

The changes of the internal structures are identified in Fig. 5.2. "Step 1" and "Step 2" indicate the sequence of the changes, i.e., from step 1 to 2. The elements in the figure are labelled as o1 to o6, 11 to 16 and 21 to 24.

For the purpose of clearer demonstration, a format of a change chain is defined. An example is o1–11,12–21,22 that means because of o1 operation, 11 and 12 firstly

take place and then 21 and 22 take place.

Sequence of the Changes of the Internal Structures of Adding One Connection Object Operation

The chain is: o1–11,13,15–21,22

Sequence of the Changes of the Internal Structures of Deleting One Connection Object Operation

The chain is: o2–12,14,16–23,24

Sequence of the Changes of the Internal Structures of Adding Multiple Connection Objects Operation

The chain is: o3–11,13,15–21,22

Sequence of the Changes of the Internal Structures of Deleting Multiple Connection Objects Operation

The chain is: o4–12,14,16–23,24

Sequences of the Changes of the Internal Structures of Modifying Multiple Attribute Values of the Connection Objects Operations

In this case, the modification is divided into two types. One is modifying multiple conditional attribute values and another one is modifying multiple decision attribute values. Afterwards, the chain of o5 is: o5–11,12,13,14–21,22,23,24. The chain of o6 is: o6–15,16–21,22,23,24.

5.4. Dynamic Connection Knowledge Space

After the fundamental analyses of the monitor and trigger mechanisms, the static connection knowledge space needs to be extended to be a dynamic one.

Definition 5.3 (Dynamic Information System)**.** *Let T be the set of the time moments at which the changes of the connection knowledge space take place. At t moment, $t \in T$, let $O(t)$ contain all the constructed connection objects. A dynamic information system is then defined as $Conn(t)$.*

$$Conn(t) \coloneqq (O(t), A_O(t)) \tag{5.1}$$

Let $V(t)$ be a set contains the attribute values, and

$$V(t) = \bigcup_{a \in A_O(t)} V_a(t) \tag{5.2}$$

$O(t)$ is called the dynamic universe that is a non-empty finite set of the connection objects. A_O is a non-empty set finite set of the attributes, describing the connection objects, such that $a : O(t) \longrightarrow V_a(t)$ for every $a \in A_O(t)$.

Definition 5.4 (Indiscernibility and Dynamic Equivalence Classes)**.** *Let T be the set of the time moments at which the changes of the connection knowledge space take place. Let $Conn(t)$, $t \in T$, be an information system at t, let $A_O(t)$ be the attribute set and let $Q \subseteq A_O(t)$. An indiscernibility relation $\sim (Q)$ is defined as*

$$\sim (Q) \coloneqq \{(h, o) \in O(t)^2 | \forall a \in Q, a(h) = a(o)\} \tag{5.3}$$

$\forall o \in O(t)$, let

$$[o]_{\sim(Q)} = \{h \in O(t) | o \sim (Q) h\} \tag{5.4}$$

$[o]_{\sim(Q)}$ is called the dynamic equivalence class of o with respect to $\sim (Q)$, which is simply written as $[o]$. All equivalence classes of $O(t)$ are represented by the quotient

set $O(t)/\sim(Q)$

$$O(t)/\sim(Q) \coloneqq \{[o]_{\sim(Q)} | o \in O(t)\} = \{E_1(t), E_2(t), E_3(t), \cdots, E_m(t)\} \tag{5.5}$$

It can be simply written as $O(t)/Q$.

Definition 5.5 (Dynamic Approximations and Regions)**.** *Let* T *be the set of the time moments at which the changes of the connection knowledge space take place. Let* $A_O(t)$ *be the attribute set and let* $Q \subseteq A_O(t)$. $(O(t), Q)$, $t \in T$, *is regarded as a dynamic knowledge base. For a subset* $X(t) \subseteq O(t)$, *the* Q*-lower approximation of* $X(t)$ *is*

$$Q_*(X(t)) \coloneqq \bigcup \{E(t) \in O(t)/Q | E(t) \subseteq X(t)\} \tag{5.6}$$

The Q*-lower approximation of* $X(t)$ *is also the* Q*-positive region of* $X(t)$, *denoted by* $POS_Q(X(t))$. *The* Q*-upper approximation of* $X(t)$ *is*

$$Q^*(X(t)) \coloneqq \bigcup \{E(t) \in O(t)/Q | E(t) \cap X(t) \neq \emptyset\} \tag{5.7}$$

The Q*-negative region of* $X(t)$ *is* $O(t) - Q^*(X(t))$, *denoted by* $NEG_Q(X(t))$. *The* Q*-boundary region of* $X(t)$ *is* $Q^*(X(t)) - Q_*(X(t))$, *denoted by* $BN_Q(X(t))$.

5.5. Incremental Updates for Connection Knowledge Space Changes

Based on the aforementioned analysis, the rough set theory based incremental updating approach for CTI in a dynamic business IT environment is proposed in this section.

5.5.1. Preparation

Some notations are initialised in this section for the incremental updates.

Let T be the set of the time moments at which the changes of the connection knowledge space take place. Let $A_O(t)$ be the attribute set at t. Because the updates when attributes change have not been developed for now, A_O is used to denote the

attribute set and it remains unchanged. Let $Conn(t)$ be an information system at moment t, $t \in T$. Let P and Q be two subsets of A_O, $P \in A_O$, $Q \in A_O$. At moment t, the following parts of the system will be updated to other states at the next moment $t+1$:

- The target set: $X(t)$, $X(t) \in O(t)/\sim(P)$, and in the scope of the system, $P = S$ where S denotes the similarity attribute
- The equivalence classes: $O(t)/\sim(Q) = \{E_1(t), E_2(t), E_3(t), \cdots, E_m(t)\}$
- The Q-lower approximation of $X(t)$: $Q_*(X(t)) = \bigcup\{E(t) \in O(t)/Q | E(t) \subseteq X(t)\}$
- The Q-upper approximation of $X(t)$: $Q^*(X(t)) = \bigcup\{E(t) \in O(t)/Q | E(t) \cap X(t) \neq \emptyset\}$

5.5.2. Update Target Set

Some steps of the incremental operations need the information of $X(t+1)$. Algorithm 4 is proposed and used to update the target set. After the algorithm, the target set at $t+1$, $X(t+1)$, is obtained.

5.5.3. Incremental Updates When One Connection Object Is Added or Deleted

5.5.3.1. Add One New Connection Object

At moment $t+1$, $t \in T$, a new connection object o' is added into the information system. Thus $O(t)$ changes to $O(t+1)$ by $O(t+1) = O(t) \cup \{o'\}$, and $X(t)$ changes to $X(t+1)$.

Two cases then exist:

- Case 1: At moment $t+1$, the new connection object added, o', belongs to an existing equivalence class at moment t. It enlarges that equivalence class.

Algorithm 4: Update Target Set

Input : $X(t)$: the target set at moment t
P: a subset of A_O
Y: a set of inputed connection objects

Output : $X(t+1)$: the target set at moment $t+1$

1 If one connection object is added, go to Step 2. If one connection object is deleted, go to Step 6. If multiple connection objects are added, go to Step 4. If multiple connection object are deleted, go to Step 8. If multiple similarity values are updated, go to Step 10.

2 $Y = \{o'\}$ where o' is the connection object added. If $\exists o \in X(t)$, $\forall a \in P$, $a(o') = a(o)$, $X(t+1) = X(t) \cup \{o'\}$. Otherwise, $X(t+1) = X(t)$.

3 Go to Step 11.

4 Y contains the connection object added. For each $o' \in Y$, if $\exists o \in X(t)$, $\forall a \in P$, $a(o') = a(o)$, $X(t+1) = X(t) \cup \{o'\}$, and $X(t+1) = X(t)$ otherwise.

5 Go to Step 11.

6 $Y = \{o'\}$ where o' is the connection object deleted. If $\exists o \in X(t)$, $\forall a \in P$, $a(o') = a(o)$, $X(t+1) = X(t) - \{o'\}$. Otherwise, $X(t+1) = X(t)$

7 Go to Step 11.

8 Y contains the connection object deleted. For each $o' \in Y$, if $\exists o \in X(t)$, $\forall a \in P$, $a(o') = a(o)$, $X(t+1) = X(t) - \{o'\}$, and $X(t+1) = X(t)$ otherwise.

9 Go to Step 11.

10 In this case, $P = \{S\}$ where S denotes similarity. Y contains the connection objects whose similarity values change. For each $o' \in Y$, if $\exists o \in X(t)$, $\forall a \in P$, $a(o') = a(o)$, $X(t+1) = X(t) \cup \{o'\}$. For each $o' \in Y$, if $\exists o \in X(t)$, $\forall a \in P$, $a(o'(t)) = a(o)$, $X(t+1) = X(t) - \{o'\}$. Otherwise, $X(t+1) = X(t)$.

11 Return $X(t+1)$.

- Case 2: At moment $t+1$, the new connection object added, o', does not belong to any existing equivalence classes at moment t, so the new connection object generates a new equivalence class at $t+1$.

The updates for the two cases will be discussed one by one.

Case 1: The New Connection Object Belongs to an Existing Equivalence Class

In this case, the new connection object enlarges the equivalence class where the object belongs to. Proposition 5.1 is developed and used to update the equivalence classes.

Proposition 5.1. *$\exists i \in \{1, 2, 3, \cdots, m\}$, $\exists o \in E_i(t)$, $\forall a \in Q$, $a(o') = a(o)$, so o' belongs to the i-th equivalence class. The i-th equivalence class expands with the connection object o', and the other equivalence classes keep unchanged. The updates of all the equivalence classes are*

$$E_j(t+1) = \begin{cases} E_j(t) \cup \{o'\}, if\ j = i \\ E_j(t), \forall j \in \{1, 2, 3, \cdots, m\} - \{i\} \end{cases} \tag{5.8}$$

The updates of the approximations are based on whether the object added, o', will be added into $X(t)$. Two cases then exist:

- Case 1.1: The new connection object added at moment $t+1$, o', does not belong to $X(t+1)$.
- Case 1.2: The new connection object added at moment $t+1$, o', belongs to $X(t+1)$.

The updates for the two sub-cases will be discussed one by one. The equivalence class where the newly added connection object belongs to is already found, and j is used to store the index of the equivalence class.

Case 1.1: The New Connection Object Does Not Belong to $X(t+1)$

In this case, Proposition 5.2 is developed and used to update the approximations.

Proposition 5.2. *$\forall o \in X(t)$, $\exists a \in Q$, $a(o') \neq a(o)$, so $o' \notin X(t+1)$. The updates of Q-upper and Q-lower approximations are*

$$Q_*(X(t+1)) = \begin{cases} Q_*(X(t)) - E_j(t),\ if\ E_j(t) \subseteq X(t) \\ Q_*(X(t)),\ if\ E_j(t) \not\subseteq X(t) \end{cases} \tag{5.9}$$

$$Q^*(X(t+1)) = \begin{cases} Q^*(X(t)),\ if\ E_j(t) \cap X(t) = \emptyset \\ Q^*(X(t)) \cup \{o'\},\ if\ E_j(t) \cap X(t) \neq \emptyset \end{cases} \tag{5.10}$$

Case 1.2: The New Connection Object Belongs to $X(t+1)$

In this case, Proposition 5.3 is developed and used to update the approximations.

Proposition 5.3. *$\exists o \in X(t)$, $\forall a \in Q$, $a(o') = a(o)$, so $o' \in X(t+1)$ which means o' is added into $X(t)$. The updates of the Q-upper approximation and Q-lower approximations are*

$$Q_*(X(t+1)) = \begin{cases} Q_*(X(t)) \cup \{o'\},\ if\ E_j(t) \subseteq X(t) \\ Q_*(X(t)),\ if\ E_j(t) \not\subseteq X(t) \end{cases} \tag{5.11}$$

$$Q^*(X(t+1)) = \begin{cases} Q^*(X(t)) \cup \{o'\}, if\ E_j(t) \cap X(t) \neq \emptyset \\ Q^*(X(t)) \cup E_j(t+1),\ if\ E_j(t) \cap X(t) = \emptyset \end{cases} \tag{5.12}$$

Case 2: The New Connection Object Generates a New Equivalence Class

In this case, Proposition 5.4 is developed and used to update the equivalence classes.

Proposition 5.4 (Update Equivalence Classes When the New Connection Object Generates a New Equivalence Class)**.** *$\forall i \in \{1, 2, 3, \cdots, m\}$, $\forall o \in E_i(t)$, $\exists a \in Q$, $a(o') \neq a(o)$, so a new equivalence class is generated, and the other equivalence classes keep unchanged. The updates of all the equivalence classes are*

$$E_j(t+1) = \begin{cases} E_j(t), \forall j \in \{1, 2, 3, \cdots, m\} \\ \{o'\}, j = m+1 \end{cases} \tag{5.13}$$

The updates of the approximations are based on whether o' belongs to $X(t+1)$. Two cases then exist.

- Case 2.1: The new connection object added at moment $t+1$, o', does not belong to $X(t+1)$.

- Case 2.2: The new connection object added at moment $t+1$, o', belongs to $X(t+1)$.

Case 2.1: The New Connection Object Does Not Belong to $X(t+1)$

Proposition 5.5. *$\forall o \in X(t)$, $\exists a \in Q$, $a(o') \neq a(o)$, so $o' \notin X(t+1)$. The Q-upper and Q-lower approximations keep unchanged.*

$$Q_*(X(t+1)) = Q_*(X(t)) \tag{5.14}$$

$$Q^*(X(t+1)) = Q^*(X(t)) \tag{5.15}$$

Case 2.2: The New Connection Object Belongs to $X(t+1)$

In this case, Proposition 5.6 is developed and used to update the approximations. According to proposition 5.4, a new equivalence is generated with the index $m+1$.

Proposition 5.6. *$\exists o \in X(t)$, $\forall a \in Q$, $a(o') = a(o)$, so $o' \in X(t+1)$. The updates of Q-upper and Q-lower approximations are*

$$Q_*(X(t+1)) = Q_*(X(t)) \cup \{o'\} \tag{5.16}$$

$$Q^*(X(t+1)) = Q^*(X(t)) \cup \{o'\} \tag{5.17}$$

The Algorithm to Update Approximations When One Connection Object Is Added

Algorithm 5 is developed and used to update the Q-upper and Q-lower approximations when a new connection object is added into the information system at moment $t+1$.

Algorithm 5: Incremental Updates of Approximations When One Connection Object Is Added

Function `InsertAConnObj()`:
  Input : $Q \subseteq A_O - \{S\}$,
    $O(t)/Q = \{E_1(t), E_2(t), E_3(t), \cdots, E_m(t)\}$,$Q_*(X(t))$,
    $Q^*(X(t))$,$X(t)$, o'
  Output : $Q_*(X(t+1))$, $Q^*(X(t+1))$
  Get $X(t+1)$
  if $\exists i \in \{1, 2, 3, \cdots, m\}$, $\exists o \in E_i(t)$, $\forall a \in Q$, $a(o') = a(o)$ **then**
    $j \leftarrow i$
    $E_j(t+1) \leftarrow E_j(t) \cup \{o'\}$
    if $o' \notin X(t+1)$ **then**
      if $E_j(t) \subseteq X(t)$ **then**
        $Q_*(X(t+1)) \leftarrow Q_*(X(t)) - E_j(t)$
      else
        $Q_*(X(t+1)) \leftarrow Q_*(X(t))$
      if $E_j(t) \cap X(t) = \emptyset$ **then**
        $Q^*(X(t+1)) \leftarrow Q^*(X(t))$
      else
        $Q^*(X(t+1)) \leftarrow Q^*(X(t)) \cup \{o'\}$
    else
      if $E_j(t) \subseteq X(t)$ **then**
        $Q_*(X(t+1)) \leftarrow Q_*(X(t)) \cup \{o'\}$
      else
        $Q_*(X(t+1)) \leftarrow Q_*(X(t))$
      if $E_j(t) \cap X(t) \neq \emptyset$ **then**
        $Q^*(X(t+1)) \leftarrow Q^*(X(t)) \cup \{o'\}$
      else
        $Q^*(X(t+1)) \leftarrow Q^*(X(t)) \cup E_j(t+1)$
  else if $\forall i \in \{1, 2, 3, \cdots, m\}$, $\forall o \in E_i(t)$, $\exists a \in Q$, $a(o') \neq a(o)$ **then**
    $E_{m+1}(t+1) \leftarrow \{o'\}$
    if $o' \notin X(t+1)$ **then**
      $Q_*(X(t+1)) \leftarrow Q_*(X(t))$
      $Q^*(X(t+1)) \leftarrow Q^*(X(t))$
    else
      $Q_*(X(t+1)) \leftarrow Q_*(X(t)) \cup \{o'\}$
      $Q^*(X(t+1)) \leftarrow Q^*(X(t)) \cup \{o'\}$
  `Return` $Q_*(X(t+1))$, $Q^*(X(t+1))$

5.5.3.2. Delete One Existing Connection Object

At moment $t+1$, $t \in T$, an existing connection object o' is deleted from the information system. $O(t)$ changes to $O(t+1) = O(t) - \{o'\}$, and $X(t)$ changes to $X(t+1)$.

After the deletion, an existing equivalence class also deletes the connection object. Proposition 5.7 is developed and used to update the equivalence classes.

Proposition 5.7. *$\exists i \in \{1, 2, 3, \cdots, m\}$, $\exists o \in E_i(t)$, $\forall a \in Q$, $a(o') = a(o)$, so the connection object o' is deleted from the i-th equivalence class, and the other equivalence classes keep unchanged. The updates of all the equivalence classes are*

$$E_j(t+1) = \begin{cases} E_j(t) - \{o'\}, j = i \\ E_j(t), \forall j \in \{1, 2, 3, \cdots, m\} - \{i\} \end{cases} \tag{5.18}$$

The updates of the approximations are based on whether o' belongs to $X(t)$. Two cases then exist.

- Case 1: The connection object deleted at moment $t+1$, o', does not belong to $X(t)$.
- Case 2: The connection object deleted at moment $t+1$, o', belongs to $X(t)$.

Case 1: The Deleted Connection Object Does Not Belong to $X(t)$

In this case, Proposition 5.8 is developed and used to update the approximations. According to Proposition 5.8, the connection object is deleted from the i-th equivalence class. Let j be the index i.

Proposition 5.8. *$\forall o \in X(t)$, $\exists a \in Q$, $a(o') \neq a(o)$, so $o' \notin X(t)$. The updates of Q-upper and Q-lower approximations are*

$$Q_*(X(t+1)) = Q_*(X(t)) \tag{5.19}$$

$$Q^*(X(t+1)) = \begin{cases} Q^*(X(t)) - \{o'\},\ if\ E_j(t) \cap X(t) \neq \emptyset \\ Q^*(X(t)),\ if\ E_j(t) \cap X(t) = \emptyset \end{cases} \tag{5.20}$$

Case 2: The Deleted Connection Object Belongs to $X(t)$

In this case, Proposition 5.9 is developed and used to update the approximations.

Proposition 5.9. *$\exists o \in X(t)$, $\forall a \in Q$, $a(o') = a(o)$, so $o' \in X(t)$. The updates of the approximations are*

$$Q_*(X(t+1)) = \begin{cases} Q_*(X(t)) - \{o'\},\ if\ E_j(t) \subseteq X(t) \\ Q_*(X(t)),\ if\ E_j(t) \not\subseteq X(t) \end{cases} \tag{5.21}$$

$$Q^*(X(t+1)) = Q^*(X(t)) - \{o'\} \tag{5.22}$$

The Algorithm to Update Approximations When One Connection Object Is Deleted

Algorithm 6 is developed and used to update the Q-upper and Q-lower approximations when a connection object is deleted from the information system.

5.5.3.3. An Example of How to Use the Updating Operations When One Connection Object is Added or Deleted

Suppose there is an information system $Conn(t) = (O(t), A_O)$ at moment t, as in Table 5.1, which is generated for one business IT context. The 9 connection objects of the information system are generated based on the business IT context and 9 CTI feeds. A_O contains 5 attributes, a_1, a_2, a_3, a_4, S. The attributes of $A_O - \{S\}$ are the relevance densities of 4 IT components, and the attribute S is the similarity densities of a pair of the business context and a CTI context. The densities have three levels that are *high* denoting high density, *medium* denoting medium density and *low* denoting low density.

Algorithm 6: Incremental Updates of Approximations When One Connection Object Is Deleted

Function `DelAConnObj()`:
  Input : $Q \subseteq A_O - \{S\}$,
    $O(t)/Q = \{E_1(t), E_2(t), E_3(t), \cdots, E_m(t)\}$,$Q_*(X(t))$,
    $Q^*(X(t))$, $X(t)$,o'
  Output : $Q_*(X(t+1))$, $Q^*(X(t+1))$
  if $\exists i \in \{1, 2, 3, \cdots, m\}$, $\exists o \in E_i(t)$, $\forall a \in Q$, $a(o') = a(o)$ **then**
    $j \leftarrow i$
    $E_j(t+1) \leftarrow E_j(t) - \{o'\}$
    if $o' \notin X(t)$ **then**
      $Q_*(X(t+1)) \leftarrow Q_*(X(t))$
      if $E_j(t) \cap X(t) \neq \emptyset$ **then**
        $Q^*(X(t+1)) \leftarrow Q^*(X(t)) - \{o'\}$
      else
        $Q^*(X(t+1)) \leftarrow Q^*(X(t))$
    else
      if $E_j(t) \subseteq X(t)$ **then**
        $Q_*(X(t+1)) \leftarrow Q_*(X(t)) - \{o'\}$
      else
        $Q_*(X(t+1)) \leftarrow Q_*(X(t))$
      $Q^*(X(t+1)) \leftarrow Q^*(X(t)) - \{o'\}$
  `Return` $Q_*(X(t+1))$, $Q^*(X(t+1))$

Table 5.1: An Example Information System

$O(t)$	a_1	a_2	a_3	a_4	S
o_1	low	low	medium	high	high
o_2	low	low	medium	low	medium
o_3	low	low	medium	high	high
o_4	medium	medium	low	low	medium
o_5	medium	medium	low	low	high
o_6	low	low	high	low	medium
o_7	low	low	high	low	medium
o_8	low	low	medium	low	medium
o_9	low	low	high	low	medium

Let

$$Q = \{a_1, a_2, a_3, a_4\}$$

$$X(t) = \{o_2, o_4, o_6, o_7, o_8, o_9\}$$

where $X(t) \in O(t)/\{S\}$ for S is medium. Firstly, after the computation of the equivalence classes, the equivalence classes $O(t)/Q$ are

$$E_1(t) = \{o_1, o_3\}$$

$$E_2(t) = \{o_2, o_8\}$$

$$E_3(t) = \{o_4, o_5\}$$

$$E_4(t) = \{o_6, o_7, o_9\}$$

The Q-lower approximation at t is

$$Q_*(X(t)) = \{o_2, o_6, o_7, o_8, o_9\}$$

and the Q-upper approximation at t is

$$Q^*(X(t)) = \{o_2, o_4, o_5, o_6, o_7, o_8, o_9\}$$

.

At moment $t + 1$, a new connection object o_{10} is added into the information system. Its attribute values are

$$o_{10} = \{low, high, high, high, high\}$$

According to Proposition 5.4, it generates a new equivalence class $E_5(t + 1)$,

$$E_5(t + 1) = \{o_{10}\}$$

The other equivalence classes keep unchanged. According to Proposition 5.5, the

Q-lower and Q-upper approximations keep unchanged,

$$Q_*(X(t+1)) = Q_*(X(t)) = \{o_2, o_6, o_7, o_8, o_9\}$$

$$Q^*(X(t+1)) = Q^*(X(t)) = \{o_2, o_4, o_5, o_6, o_7, o_8, o_9\}$$

At moment $t+2$, a new connection object o_{11} is added into the information system. It is

$$o_{11} = \{medium, medium, low, low, medium\}$$

According to Proposition 5.1, o_{11} expands $E_3(t+1)$, so

$$E_3(t+2) = E_3(t+1) \cup \{o_{11}\} = \{o_4, o_5, o_{11}\}$$

According to Proposition 5.2, the Q-lower approximation keeps unchanged,

$$Q_*(X(t+2)) = Q_*(X(t+1)) = \{o_2, o_6, o_7, o_8, o_9\}$$

and Q-upper approximation expands with o_{11},

$$Q^*(X(t+2)) = Q^*(X(t+1)) \cup \{o_{11}\} = \{o_2, o_4, o_5, o_6, o_7, o_8, o_9, o_{11}\}$$

At moment $t+3$, the connection object o_2 is deleted from the information system. According to Proposition 5.7, $E_2(t+2)$ is updated to $E_2(t+3)$ with the deletion of the connection object,

$$E_2(t+3) = E_2(t+2) - \{o_2\} = \{o_8\}$$

According to Proposition 5.9, the Q-lower approximation is updated

$$Q_*(X(t+3)) = Q_*(X(t+2)) - \{o_2\} = \{o_6, o_7, o_8, o_9\}$$

and the Q-upper approximation is updated

$$Q^*(X(t+3)) = Q^*(X(t+2)) - \{o_2\} = \{o_4, o_5, o_6, o_7, o_8, o_9, o_{11}\}$$

5.5.4. Incremental Updates When Multiple Connection Objects Are Added or Deleted

This section talks about the incremental updates based on rough set theory when multiple connection objects are added or deleted.

More notations are introduced as follows.

- Y_1: a set of the connection objects added or deleted at moment $t+1$
- Y_1' and Y_1'': $Y_1 = Y_1' \cup Y_1''$. When multiple connection objects are added, Y_1' contains the connection objects that belong to the existing equivalence classes and Y_1'' contains the connection objects that generate new equivalence classes. When multiple connection objects are deleted, $Y_1' = Y_1$ and $Y_1'' = \emptyset$.
- M_1: the set of the indices of the equivalence classes whose elements change at moment $t+1$. In other words, the connection object in Y_1' belongs to the equivalence classes with these indices.
- M_2: the set of the indices of the equivalence classes generated at moment $t+1$
- Y_2: the set of the new equivalence classes generated at moment $t+1$
- Y_{1j}: a set of the connection objects added into or removed from the j-th equivalence class, $j \in \{1, 2, 3, \cdots, m, m+1, m+2, m+3, \cdots, n(Y_2)\}$, at moment $t+1$

5.5.4.1. Prerequisite Constructions

Construct M_1

At moment $t+1$, the information system needs to track which equivalence classes the connection objects of Y_1' belong to, so M_1 is used to store the indices. To find these indices, if $\forall i \in \{1, 2, 3, \cdots, m\}$, $\exists E_i(t)$, $\forall o \in E_i(t)$, $\exists o' \in Y_1'$, $\forall a \in Q$, $a(o') = a(o)$, then record the i into M_1. The other connection objects in Y_1 belong to Y_1'', and they will generate new equivalence classes.

Construct Y_2 and M_2

Y_1'' contains the connection objects that generate new equivalence classes generated at moment $t+1$. Let a set Y_2 contain the new equivalence classes generated. $Y_2 = Y_1''/Q = \{E_{m+1}, E_{m+2}, E_{m+3}, \cdots, E_{m+n(Y_2)}\}$, and $M_2 = \{m+1, m+2, m+3, \cdots, n(Y_2)\}$.

Construct $Y_{1j}s$

While M_1 and M_2 are being constructed, the corresponding Y_{1j}s can be constructed.

5.5.4.2. Add Multiple Connection Objects

Three cases then exist.

- Case 1: $Y_1' \neq \emptyset$, $Y_1'' \neq \emptyset$. Some of the connection objects added belong to the existing equivalence classes, and some of the connection objects added generate new equivalence classes.
- Case 2: $Y_1' \neq \emptyset$, $Y_1'' = \emptyset$. All the connection objects added belong to existing equivalence classes.
- Case 3: $Y_1' = \emptyset$, $Y_1'' \neq \emptyset$. All the connection objects added generate new equivalence classes.

Case 1: $Y_1' \neq \emptyset$, $Y_1'' \neq \emptyset$

In this case, Proposition 5.10 is developed and used to update the equivalence classes.

Proposition 5.10. *Add some of the newly added connection objects to the existing equivalence classes, generate new equivalence classes and keep the other equivalence classes unchanged.*

$$E_j(t+1) = \begin{cases} E_j(t) \cup Y_{1j}, \forall j \in M_1 \\ E_j(t), \forall j \in \{1, 2, 3, \cdots, m\} - M_1 \\ Y_{1j}, \forall j \in M_2 \end{cases} \tag{5.23}$$

Proposition 5.11 presents how to update the approximations in Case 1.

Proposition 5.11. *To update the approximations, for each* $j \in M_1 \cup M_2$, *perform*

$$Q_*(X(t+1)) = \begin{cases} Q_*(X(t)) \cup Y_{1j}, \, if \; j \in M_1, E_j(t) \subseteq X(t), Y_{1j} \subseteq X(t+1) \\ \qquad\qquad or \; j \in M_2, Y_{1j} \subseteq X(t+1) \\ Q_*(X(t)) - E_j(t+1), \; if \; j \in M_1, E_j(t) \subseteq X(t), Y_{1j} \not\subseteq X(t+1) \\ Q_*(X(t)), \; if \; j \in M_1, E_j(t) \not\subseteq X(t) \; or \; j \in M_2, Y_{1j} \not\subseteq X(t+1) \end{cases} \tag{5.24}$$

$$Q^*(X(t+1)) = \begin{cases} Q^*(X(t)) \cup Y_{1j}, \, if \; j \in M_1, E_j(t) \cap X(t) \neq \emptyset \\ \qquad\qquad or \; j \in M_2, Y_{1j} \cap X(t+1) \neq \emptyset \\ Q^*(X(t)) \cup E_j(t+1), \; if \; j \in M_1, E_j(t) \cap X(t) = \emptyset, Y_{1j} \cap X(t) \neq \emptyset \\ Q^*(X(t)), if \; j \in M_1, E_j(t) \cap X(t) = \emptyset, Y_{1j} \cap X(t+1) = \emptyset \\ \qquad\qquad or \; j \in M_2, Y_{1j} \cap X(t+1) = \emptyset \end{cases} \tag{5.25}$$

Case 2: $Y_1' \neq \emptyset$, $Y_1'' = \emptyset$

Proposition 5.10 is used to update the equivalence classes in Case 2, but $M_1 = \emptyset$.

Proposition 5.11 is used to update the approximations in Case 2, but $M_1 = \emptyset$.

Case 3: $Y_1' = \emptyset$, $Y_1'' \neq \emptyset$

Proposition 5.10 is used to update the equivalence classes in Case 3, but $M_2 = \emptyset$

Proposition 5.11 is used to update the approximations in Case 3, but $M_2 = \emptyset$.

The Algorithm to Update Approximations When Multiple Connection Objects Are Added

Algorithm 7 is developed and used to update the approximations when multiple connection objects are added. In the algorithm, Step 3 is to perform the prerequisite constructions according to Chapter 5.5.4.1. Step 4 to Step 12 are to update the equivalence classes. Step 13 to Step 41 are to update the approximations.

5.5.4.3. Delete Multiple Connection Objects

When multiple existing connection objects are deleted from the information system at moment $t+1$, the updates do not need to construct M_2 and Y_2.

Proposition 5.12 presents how to update the equivalence classes.

Proposition 5.12. *When multiple connection objects are deleted from the information system, the equivalence classes that have the deleted objects remove these connection objects and the other equivalence classes keep unchanged.*

$$E_j(t+1) = \begin{cases} E_j(t) - Y_{1j}, \forall j \in M_1 \\ E_j(t), \forall j \in \{1, 2, 3, \cdots, m\} - M_1 \end{cases} \tag{5.26}$$

Proposition 5.13 presents how to update the approximations.

Algorithm 7: Incremental Updates of Approximations When Multiple Connection Objects Are Added

1 **Begin**
  Input : $Q \subseteq A_O - \{S\}$,
    $O(t)/Q = \{E_1(t), E_2(t), E_3(t), \cdots, E_m(t)\}$,$Q_*(X(t))$,
    $Q^*(X(t))$,$X(t)$, Y
  Output : $Q_*(X(t+1))$, $Q^*(X(t+1))$
2 Get $X(t+1)$
3 Construct M_1, M_2 and all the Y_{1j}
4 **foreach** $j \in \{1, 2, 3, \cdots, m\} \cup M_2$ **do**
5 **if** $j \in M_1$ **then**
6 $E_j(t+1) = E_j(t) \cup Y_{1j}$
7 **else if** $j \in \{1, 2, 3, \cdots, m\} - M_1$ **then**
8 $E_j(t+1) = E_j(t)$
9 **else if** $j \in M_2$ **then**
10 $E_j(t+1) = Y_{1j}$
11 **end if**
12 **end foreach**
13 **foreach** $j \in M_1 \cup M_2$ **do**
14 **if** $j \in M_1$ **then**
15 **if** $E_j(t) \subseteq X(t), Y_{1j} \subseteq X(t+1)$ **then**
16 $Q_*(X(t+1)) = Q_*(X(t)) \cup Y_{1j}$
17 **else if** $E_j(t) \subseteq X(t), Y_{1j} \not\subseteq X(t+1)$ **then**
18 $Q_*(X(t+1)) = Q_*(X(t)) - E_j(t+1)$
19 **else if** $E_j(t) \not\subseteq X(t)$ **then**
20 $Q_*(X(t+1)) = Q_*(X(t))$
21 **end if**
22 **if** $E_j(t) \cap X(t) \neq \emptyset$ **then**
23 $Q^*(X(t+1)) = Q^*(X(t)) \cup Y_{1j}$
24 **else if** $E_j(t) \cap X(t) = \emptyset, Y_{1j} \cap X(t+1) \neq \emptyset$ **then**
25 $Q^*(X(t+1)) = Q^*(X(t)) \cup E_j(t+1)$
26 **else if** $E_j(t) \cap X(t) = \emptyset, Y_{1j} \cap X(t+1) = \emptyset$ **then**
27 $Q^*(X(t+1)) = Q^*(X(t))$
28 **end if**
29 **end if**
30 **if** $j \in M_2$ **then**
31 **if** $Y_{1j} \subseteq X(t)$ **then**
32 $Q_*(X(t+1)) = Q_*(X(t)) \cup Y_{1j}$
33 **end if**
34 **if** $Y_{1j} \cap X(t+1) \neq \emptyset$ **then**
35 $Q^*(X(t+1)) = Q^*(X(t)) \cup Y_{1j}$
36 **else if** $Y_{1j} \cap X(t+1) = \emptyset$ **then**
37 $Q^*(X(t+1)) = Q^*(X(t))$
38 **end if**
39 **end if**
40 **end foreach**
41 Return $Q_*(X(t+1))$, $Q^*(X(t+1))$

Proposition 5.13. *To update the approximations, for each $j \in M_1$, perform*

$$Q_*(X(t+1)) = \begin{cases} Q_*(X(t)) - Y_{1j},\ if\ E_j(t) \subseteq X(t) \\ Q_*(X(t)) \cup E_j(t+1),\ if\ E_j(t) \not\subseteq X(t), \\ \qquad\qquad\qquad\qquad E_j(t) - Y_{1j} \subseteq X(t) \\ Q_*(X(t)),\ if\ E_j(t) \not\subseteq X(t) \end{cases} \tag{5.27}$$

$$Q^*(X(t+1)) = \begin{cases} Q^*(X(t)) - Y_{1j},\ if\ E_j(t) \cap X(t) \neq \emptyset \\ Q^*(X(t)),\ if\ E_j(t) \cap X(t) = \emptyset \end{cases} \tag{5.28}$$

The Algorithm to Update Approximations When Multiple Connection Objects Are Deleted

Algorithm 8 is developed and used to update the approximations when multiple connection objects are deleted. Step 3 is to perform the prerequisite constructions according to Chapter 5.5.4.1. Step 4 to Step 10 are to update the equivalence classes. Step 11 to Step 25 are to update the approximations.

5.5.5. Incremental Updates When Attribute Values Are Changed

This section talks about the incremental updates based on rough set theory when attribute values of connection objects change. In this case, both condition attribute values and decision values might be changed simultaneously. The update process then has two steps. The first step is to update the corresponding structures only considering the changes of the condition attribute values and considering the decision attribute values are unchanged. After that, the second step is to update the corresponding structures considering the changes of the decision attribute values.

Some notations are re-used for different purposes and some new notations are introduced, which are listed as follows.

- Y_1: the set of the connection objects whose condition attribute values change at moment $t+1$

Algorithm 8: Incremental Updates of Approximations When Multiple Connection Objects Are Deleted

Begin
  Input : $Q \subseteq A_O - \{S\}$,
    $O(t)/Q = \{E_1(t), E_2(t), E_3(t), \cdots, E_m(t)\}$,$Q_*(X(t))$,
    $Q^*(X(t))$,$X(t)$, Y
  Output : $Q_*(X(t+1))$, $Q^*(X(t+1))$
  Get $X(t+1)$
  Construct M_1 and all the Y_{1j}
  foreach $j \in \{1, 2, 3, \cdots, m\}$ **do**
    if $j \in M_1$ **then**
      $E_j(t+1) = E_j(t) - Y_{1j}$
    else
      $E_j(t+1) = E_j(t)$
    end if
  end foreach
  foreach $j \in M_1$ **do**
    if $E_j(t) \subseteq X(t)$ **then**
      $Q_*(X(t+1)) = Q_*(X(t)) - Y_{1j}$
    else if $E_j(t) \not\subseteq X(t), E_j(t) - Y_{1j} \subseteq X(t+1)$ **then**
      $Q_*(X(t+1)) = Q_*(X(t)) \cup E_j(t+1)$
    else if $E_j(t) \not\subseteq X(t)$ **then**
      $Q_*(X(t+1)) = Q_*(X(t))$
    end if
    if $E_j(t) \cap X(t) \neq \emptyset$ **then**
      $Q^*(X(t+1)) = Q^*(X(t)) - Y_{1j}$
    else if $E_j(t) \cap X(t) = \emptyset$ **then**
      $Q^*(X(t+1)) = Q^*(X(t))$
    end if
  end foreach
  Return $Q_*(X(t+1))$, $Q^*(X(t+1))$
end

- Y_1' and Y_1'': $Y_1 = Y_1' \cup Y_1''$. Y_1' contains the connection objects in Y_1 which belong to the existing equivalence classes, Y_1'' contains the connection objects in Y_1 which does not belong to any existing equivalence class.

- M_1: the set of the indices of the equivalence classes whose elements' condition attribute values change at moment $t+1$

- M_2: the set of the indices of the equivalence classes generated at moment $t+1$

- M_3: the set of the new indices of the equivalence classes where the connection objects, whose condition attribute values change at moment $t+1$, belong to

- Y_2: the set of the new equivalence classes generated at moment $t+1$

- Y_{1j}: a set of the connection objects added into the i-th equivalence class, $j \in \{1, 2, 3, \cdots, m, m+1, m+2, m+3, \cdots, n(Y_2)\}$, at moment $t+1$ after the changes

- $E_j(t)'$: When the connection objects' condition attribute values change, each corresponding equivalence class, having the objects, is divided into two parts. $E_j(t)'$ contains the connection objects whose condition attribute values change in $E_j(t)$, $j \in \{1, 2, 3, \cdots, m\}$.

- Y_3: the set of the connection objects whose decision attribute values change at moment $t+1$

- M_4: the set of the indices of the equivalence classes where the connection objects of Y_3 belong to

- M_5: the set of the indices of the equivalence classes which are in $Q_*(X(t))$

- M_6: the set of the indices of the equivalence classes which are in $Q^*(X(t))$

5.5.5.1. Prerequisite Constructions

Construct M_1 and $E_j(t)'$

At moment $t+1$, the information system needs to track which equivalence classes' connection objects' condition attribute values change, so M_1 is used to store the indices of the equivalence classes. $\forall j \in M_1$, divide $E_j(t)$ into two parts. One part is $E_j(t)'$, where $E_j(t)' \cap Y_1 \neq \emptyset$.

Construct Y_2, M_2

Y_1'' contains the connection objects that generate new equivalence classes generated at moment $t+1$. Let a set Y_2 contain the new equivalence classes generated. $Y_2 = Y_1''/Q = \{E_{m+1}, E_{m+2}, E_{m+3}, \cdots, E_{m+n(Y_2)}\}$, and $M_2 = \{m+1, m+2, m+3, \cdots, n(Y_2)\}$.

Construct Y_{1j} and M_3

While M_1 and M_2 are being constructed, the corresponding Y_{1j}s can be constructed. The M_3 records the indices of these Y_{1j}s.

Construct Y_3, M_4, M_5, M_6

Y_3 contains the connection objects whose decision attribute values change at moment $t+1$. M_4, M_5 and M_6 can be retrieved from the structures in the previous moment.

5.5.5.2. Condition Attribute Values Are Changed

Three cases then exist.

- Case 1: At moment $t+1$, the connection objects in Y_1'' do not belong to any existing equivalence class, so these connection objects in Y_1'' generate some

new equivalence classes. Meanwhile, the connection objects in Y_1' belong to some existing equivalence classes, so these connection objects in Y_1' are added to those existing equivalence classes, which enlarges those equivalence classes.

- Case 2: At moment $t+1$, all the connection objects in Y_1 do not belong to any existing equivalence class, so these connection objects in Y_1 generate some new equivalence classes. In this case, $Y_1'' = Y_1$.

- Case 3: At moment $t+1$, all the connection objects in Y_1 belong to some existing equivalence classes, so these connection objects in Y_1 are added to those existing equivalence classes. In this case, $Y_1' = Y_1$.

After the new notations are introduced, the propositions of how to update the equivalence classes and approximations of the three cases can be then discussed.

Case 1

$Y_1' \neq \emptyset$, $Y_1'' \neq \emptyset$. Proposition 5.14 presents how to update the equivalence classes in Case 1.

Proposition 5.14. *To update the equivalences when connection objects' attributes change, remove the connection objects whose attribute values change at moment* $t+1$ *from the equivalence classes, add the removed connection objects to new equivalence classes, add new equivalence classes generated at moment* $t+1$ *and keep the other equivalence classes unchanged.*

$$E_j(t+1) = \begin{cases} E_j(t) - E_j(t)', \forall j \in M_1, j \notin M_3 \\ E_j(t) - E_j(t)' + Y_{1j}, \forall j \in M_1, j \in M_3 \\ E_j(t) + Y_{1j}, \forall j \notin M_1, j \in M_3 \\ Y_{1j}, \forall j \in M_2 \\ E_j(t), \forall j \in \{1, 2, 3, \cdots, m\} - M_1 - M_3 \end{cases} \tag{5.29}$$

Proposition 5.15 presents how to update the approximations in Case 1.

Proposition 5.15. *To update the approximations, for each $j \in M_1 \cup M_3 \cup M_2$, perform*

$$Q_*(X(t+1)) = \begin{cases} Q_*(X(t)) \cup Y_{1j}, \, if \; j \in M_1, j \in M_3, E_j(t) \not\subseteq X(t), Y_{1j} \subseteq X(t+1), \\ \qquad\qquad E_j(t)' = E_j(t), \\ \qquad\qquad or \; E_j(t) \subseteq X(t), Y_{1j} \subseteq X(t+1), \\ \qquad\qquad or \; j \in M_2, Y_{1j} \subseteq X(t+1) \\ Q_*(X(t)) - E_j(t)' \cup Y_{1j}, \, if \; j \in M_1, j \in M_3, E_j(t) \subseteq X(t), \\ \qquad\qquad Y_{1j} \subseteq X(t+1) \\ Q_*(X(t)) - E_j(t)', \; if \; j \in M_1, j \notin M_3, E_j(t) \subseteq X(t) \\ Q_*(X(t)) - E_j(t), \; if \; j \in M_1, j \in M_3, E_j(t) \subseteq X(t), Y_{1j} \not\subseteq X(t+1) \\ Q_*(X(t)) \cup E_j(t+1), \, if \; j \in M_1, j \notin M_3, E_j(t) \not\subseteq X(t), \\ \qquad\qquad E_j(t+1) \subseteq X(t+1) \\ Q_*(X(t)) - E_j(t+1), \; if \; j \notin M_1, j \in M_3, E_j(t) \subseteq X(t), Y_{1j} \not\subseteq X(t+1) \\ Q_*(X(t)), \, if \; j \in M_1, j \in M_3, E_j(t) \not\subseteq X(t), Y_{1j} \not\subseteq X(t+1), \\ \qquad\qquad or \; j \in M_1, j \in M_3, E_j(t) \not\subseteq X(t), Y_{1j} \subseteq X(t+1), E_j(t)' \neq E_j(t), \\ \qquad\qquad or \; j \in M_1, j \notin M_3, E_j(t) \not\subseteq X(t), E_j(t+1) \not\subseteq X(t+1), \\ \qquad\qquad or \; j \notin M_1, j \in M_3, E_j(t) \not\subseteq X(t) \end{cases} \tag{5.30}$$

$$
Q^*(X(t+1)) = \begin{cases}
Q^*(X(t)) \cup Y_{1j}, \, if \; Y_{1j} \cap X(t+1) \neq \emptyset, \\
\qquad or \; j \notin M_1, j \in M_3, E_j(t) \cap X(t) \neq \emptyset \\
Q^*(X(t)) - E_j(t)' \cup Y_{1j}, \, if \; j \in M_1, j \in M_3, E_j(t) \cap X(t) = \emptyset, \\
\qquad E_j(t+1) \cap X(t+1) \neq \emptyset \\
Q^*(X(t)) - E_j(t)', \, if \; j \in M_1, j \notin M_3, E_j(t) \cap X(t) \neq \emptyset, \\
\qquad or \; j \in M_1, j \in M_3, E_j(t) \cap X(t) \neq \emptyset, \\
\qquad E_j(t+1) \cap X(t+1) \neq \emptyset \\
Q^*(X(t)) \cup E_j(t+1), \, if \; j \notin M_1, j \in M_3, E_j(t) \cap X(t) = \emptyset, \\
\qquad Y_{1j} \cap X(t+1) \neq \emptyset, \\
\qquad or \; j \in M_1, j \notin M_3, E_j(t) \cap X(t) = \emptyset, \\
\qquad E_j(t+1) \cap X(t+1) \neq \emptyset \\
Q^*(X(t)) - E_j(t+1), \, if \; j \in M_1, j \notin M_3, E_j(t) \cap X(t) \neq \emptyset, \\
\qquad E_j(t+1) \cap X(t+1) = \emptyset, \\
\qquad or \; j \in M_1, j \in M_3, E_j(t) \cap X(t) \neq \emptyset, \\
\qquad E_j(t+1) \cap X(t+1) = \emptyset \\
Q^*(X(t)), \, if \; j \in M_1, j \notin M_3, E_j(t) \cap X(t) = \emptyset, \\
\qquad E_j(t+1) \cap X(t+1) = \emptyset, \\
\qquad or \; j \notin M_1, j \in M_3, E_j(t) \cap X(t) = \emptyset, \\
\qquad Y_{1j} \cap X(t+1) = \emptyset, \\
\qquad or \; j \in M_1, j \in M_3, E_j(t) \cap X(t) = \emptyset, \\
\qquad E_j(t+1) \cap X(t+1) = \emptyset
\end{cases} \tag{5.31}
$$

Case 2

$Y_1' \neq \emptyset$, $Y_1'' = \emptyset$, so $M_2 = \emptyset$. Proposition 5.14 is also used to update the equivalence classes. Proposition 5.15 is also used to update the approximations in Case 2.

Case 3

$Y_1' = \emptyset$, $Y_1'' \neq \emptyset$. Proposition 5.14 is also used to update the equivalence classes. Proposition 5.15 is also used to update the approximations in Case 3.

The Algorithm to Update Approximations When Attribute Values Change

Algorithm 9 is developed and used to incrementally update the equivalence classes when attribute values change.

Algorithm 9: Incremental Updates of Equivalence Classes When Attribute Values Change

```
1  Begin
       Input  : O(t)/Q = {E_1(t), E_2(t), E_3(t), ··· , E_m(t)}, Y
       Output : O(t+1)/Q
2      Construct M_1, M_2, M_3, all the Y_1j and all the E_j(t)'
3      foreach j ∈ {1, 2, 3, ··· , m} ∪ M_2 do
4          if j ∈ M_1, j ∉ M_3 then
5              E_j(t+1) = E_j(t) − E_j(t)'
6          else if j ∈ M_1, j ∈ M_3 then
7              E_j(t+1) = E_j(t) − E_j(t)' ∪ Y_1j
8          else if j ∉ M_1, j ∈ M_3 then
9              E_j(t+1) = E_j(t) ∪ Y_1j
10         else if j ∈ M_2 then
11             E_j(t+1) = Y_1j
12         else if j ∈ {1, 2, 3, ··· , m} − M_1 − M_3 then
13             E_j(t+1) = E_j(t)
14         end if
15     end foreach
16     Return O(t+1)/Q
```

Algorithm 10 is developed and used to update the Q-lower approximation when connection objects' attribute values change.

Algorithm 11 is developed and used to update the Q-upper approximation when connection objects' attribute values change.

Algorithm 10: Incrementally Update Lower Approximation When Attribute Values Change

Begin

  Input : $O(t)/Q = \{E_1(t), E_2(t), E_3(t), \cdots, E_m(t)\}$, $O(t+1)/Q$, $Q_*(X(t))$, $X(t)$, M_1, M_2, M_3, $\forall i \in M_1, E_i(t)'$, $\forall i \in M_2 \cup M_3, Y_{1i}$

  Output : $Q_*(X(t+1))$

  Get $X(t+1)$

  foreach $j \in M_1 \cup M_3 \cup M_2$ **do**

    if $j \in M_2$ **then**

      if $Y_{1j} \subseteq X(t+1)$ **then** $Q_*(X(t+1)) = Q_*(X(t)) \cup Y_{1j}$

    end if

    if $j \in M_1, j \in M_3$ **then**

      if $E_j(t) \subseteq X(t), Y_{1j} \not\subseteq X(t+1)$ **then** $Q_*(X(t+1)) = Q_*(X(t)) - E_j(t)$

      else if $E_j(t) \subseteq X(t), Y_{1j} \subseteq X(t+1)$ **then** $Q_*(X(t+1)) = Q_*(X(t)) - E_j(t)' \cup Y_{1j}$

      else if $E_j(t) \not\subseteq X(t), Y_{1j} \not\subseteq X(t+1)$ **then** $Q_*(X(t+1)) = Q_*(X(t))$

      else if $E_j(t) \not\subseteq X(t), Y_{1j} \subseteq X(t+1)$ **then**

        if $E_j(t)' = E_j(t)$ **then** $Q_*(X(t+1)) = Q_*(X(t)) \cup Y_{1j}$

        else if $E_j(t)' \neq E_j(t)$ **then** $Q_*(X(t+1)) = Q_*(X(t))$

      end if

    end if

    if $j \in M_1, j \notin M_3$ **then**

      if $E_j(t) \subseteq X(t)$ **then** $Q_*(X(t+1)) = Q_*(X(t)) - E_j(t)'$

      else if $E_j(t) \not\subseteq X(t), E_j(t+1) \subseteq X(t+1)$ **then** $Q_*(X(t+1)) = Q_*(X(t)) \cup E_j(t+1)$

      else if $E_j(t) \not\subseteq X(t), E_j(t+1) \not\subseteq X(t+1)$ **then** $Q_*(X(t+1)) = Q_*(X(t))$

    end if

    if $j \notin M_1, j \in M_3$ **then**

      if $E_j(t) \not\subseteq X(t)$ **then** $Q_*(X(t+1)) = Q_*(X(t))$

      else if $E_j(t) \subseteq X(t), Y_{1j} \subseteq X(t+1)$ **then** $Q_*(X(t+1)) = Q_*(X(t)) \cup Y_{1j}$

      else if $E_j(t) \subseteq X(t), Y_{1j} \not\subseteq X(t+1)$ **then** $Q_*(X(t+1)) = Q_*(X(t)) - E_j(t+1)$

    end if

  end foreach

  Return $Q_*(X(t+1))$

end

Algorithm 11: Incrementally Update Upper Approximation When Attribute Values Change

Begin
  Input : $O(t)/Q = \{E_1(t), E_2(t), E_3(t), \cdots, E_m(t)\}$, $O(t+1)/Q$, $Q_*(X(t))$, $X(t)$, M_1, M_2, M_3, $\forall i \in M_1, E_i(t)'$, $\forall i \in M_2 \cup M_3, Y_{1i}$
  Output : $Q^*(X(t+1))$
  Get $X(t+1)$
  foreach $j \in M_1 \cup M_3 \cup M_2$ **do**
    if $j \in M_2$ **then**
      if $Y_{1j} \cap X(t+1) \neq \emptyset$ **then** $Q^*(X(t+1)) = Q^*(X(t)) \cup Y_{1j}$
    end if
    if $j \in M1, j \in M_3$ **then**
      if $E_j(t) \cap X(t) \neq \emptyset, E_j(t+1) \cap X(t+1) \neq \emptyset$ **then** $Q^*(X(t+1)) = Q^*(X(t)) - E_j(t)'$
      else if $E_j(t) \cap X(t) \neq \emptyset, E_j(t+1) \cap X(t+1) = \emptyset$ **then** $Q^*(X(t+1)) = Q^*(X(t)) - E_j(t+1)$
      else if $E_j(t) \cap X(t) = \emptyset, E_j(t+1) \cap X(t+1) = \emptyset$ **then** $Q^*(X(t+1)) = Q^*(X(t))$
      else if $E_j(t) \cap X(t) = \emptyset, E_j(t+1) \cap X(t+1) \neq \emptyset$ **then**
        $Q^*(X(t+1)) = Q^*(X(t)) - E_j(t)' \cup Y_{1j}$
      end if
    end if
    if $j \in M_1, j \notin M_3$ **then**
      if $E_j(t) \cap X(t) \neq \emptyset, E_j(t+1) \cap X(t+1) = \emptyset$ **then** $Q^*(X(t+1)) = Q^*(X(t)) - E_j(t+1)$
      else if $E_j(t) \cap X(t) \neq \emptyset, E_j(t+1) \cap X(t+1) \neq \emptyset$ **then** $Q^*(X(t+1)) = Q^*(X(t)) - E_j(t)'$
      else if $E_j(t) \cap X(t) = \emptyset, E_j(t+1) \cap X(t+1) \neq \emptyset$ **then** $Q^*(X(t+1)) = Q^*(X(t)) \cup E_j(t+1)$
      else if $E_j(t) \cap X(t) = \emptyset, E_j(t+1) \cap X(t+1) = \emptyset$ **then** $Q^*(X(t+1)) = Q^*(X(t))$
    end if
    if $j \notin M_1, j \in M_3$ **then**
      if $E_j(t) \cap X(t) \neq \emptyset$ **then** $Q^*(X(t+1)) = Q^*(X(t)) \cup Y_{1j}$
      else if $E_j(t) \cap X(t) = \emptyset, Y_{1j} \cap X(t+1) = \emptyset$ **then** $Q^*(X(t+1)) = Q^*(X(t))$
      else if $E_j(t) \cap X(t) = \emptyset, Y_{1j} \cap X(t+1) \neq \emptyset$ **then** $Q^*(X(t+1)) = Q^*(X(t)) \cup E_j(t+1)$
    end if
  end foreach
  Return $Q^*(X(t+1))$
end

5.5.5.3. An Example of How to Use the Operations When Condition Attribute Values Change

Suppose there is an information system $Conn(t) = (O(t), A_O)$, as in Table 5.1. At moment t, the set Q, the set $X(t)$, all the equivalence classes, the Q-lower approximation and the Q-upper approximations are identical to those in Section 5.5.3.3.

$$X(t) = \{o_2, o_4, o_6, o_7, o_8, o_9\}$$

$$E_1(t) = \{o_1, o_3\}$$

$$E_2(t) = \{o_2, o_8\}$$

$$E_3(t) = \{o_4, o_5\}$$

$$E_4(t) = \{o_6, o_7, o_9\}$$

The Q-lower approximation at t is

$$Q_*(X(t)) = \{o_2, o_6, o_7, o_8, o_9\}$$

and the Q-upper approximation at t is

$$Q^*(X(t)) = \{o_2, o_4, o_5, o_6, o_7, o_8, o_9\}$$

.

At moment $t+1$, $o_9(t)$ changes to

$$o_9(t+1) = \{low, low, low.low, medium\}$$

so

$$Y_1 = \{o_9(t)\}$$

Because $o_9(t) \in E_4(t)$,

$$M_1 = \{4\}$$

and

$$E_4(t)' = \{o_9(t)\}$$

A new equivalence class is then generated, so

$$Y_2 = \{o_9(t+1)\}$$

$$Y_2 = Y_1'/Q = \{E_{4+1}(t+1)\} = \{o_9(t+1)\}$$

The Y_{1j} for Y_1 is

$$Y_{15} = \{o_9(t+1)\}$$

Use Proposition 5.14 to update the equivalence classes,

$$E_4(t+1) = E_4(t) - E_4(t)' = \{o_6(t+1), o_7(t+1)\}$$

$\forall j \in \{1,2,3,4\} - \{4\}, E_j(t+1) = E_j(t)$, so

$$E_1(t+1) = E_1(t) = \{o_1(t+1), o_3(t+1)\}$$

$$E_2(t+1) = E_2(t) = \{o_2(t+1), o_8(t+1)\}$$

$$E_3(t+1) = E_3(t) = \{o_4(t+1), o_5(t+1)\}$$

For $j \in M_2$

$$E_5(t+1) = Y_{1j} = \{o_9(t+1)\}$$

According to proposition 5.15, for each $j \in M_1 \cup M_2$, the update of the Q-lower approximation is

$$Q_*(X(t+1)) = Q_*(X(t)) - E_4(t)' = \{o_2(t+1), o_6(t+1), o_7(t+1), o_8(t+1)\} - \{o_9(t)\}$$

when j is 4, and

$$Q_*(X(t+1)) = Q_*(X(t)) \cup Y_{14} = \{o_2(t+1), o_6(t+1), o_7(t+1), o_8(t+1), o_9(t+1)\}$$

The update of the upper approximation is

$$Q^*(X(t)) = Q^*(X(t)) - E_4(t)' \cup Y_{14} =$$

$$\{o_2(t+1), o_4(t+1), o_5(t+1), o_6(t+1), o_7(t+1), o_8(t+1), o_9(t+1)\}$$

5.5.5.4. Decision Attribute Values Are Changed

When the decision attribute values of the connection objects are changed, all the equivalence classes will not be updated but the approximations might be changed. Proposition 5.16 shows how to update the approximations when decision attribute values are changed.

Proposition 5.16. $\forall j \in M_4$,

$$Q_*(X(t+1)) = \begin{cases} Q_*(X(t)) - E_j(t), \ if \ E_j(t) \not\subseteq X(t+1), j \in M_5 \\ Q_*(X(t)) \cup E_j(t), \ if \ E_j(t) \subseteq X(t+1), j \notin M_5 \\ Q_*(X(t)), \ if \ E_j(t) \not\subseteq X(t+1), j \notin M_5 \end{cases} \tag{5.32}$$

$$Q^*(X(t+1)) = \begin{cases} Q^*(X(t)), if \ E_j(t) \cap X(t+1) = \emptyset, i \notin M_6, \\ \qquad or \ E_j(t) \cap X(t+1) \neq \emptyset, i \in M_6 \\ Q^*(X(t)) - E_j(t), E_j(t) \cap X(t+1) = \emptyset, i \in M_6 \\ Q^*(X(t)) \cup E_j(t), E_j(t) \cap X(t+1) \neq \emptyset, j \notin M_6 \end{cases} \tag{5.33}$$

The algorithm used to update the approximations is to traverse all the indices in M_4 to perform the updates when the corresponding conditions are met.

5.6. Complexities of the Incremental Updating Operations

Table 5.2 shows the time complexities of the non-incremental and incremental updating operations for updating lower and upper approximations. An operation of updating lower and upper approximations has the same complexity. In the table,

Table 5.2: Complexities of the Non-Incremental and Incremental Updating Operations

Operation	Complexity of Non-incremental Method	Complexities of Incremental Methods
Add One Connection Object	$r(\|Q\|\|O(t)\|^2 + X(t)\|)$	$r(\|O(t)/Q\|\|Q\| + \bar{E})$
Add Multiple Connection Objects		$r(\|O(t)/Q\|\|Q\|\|Y_1\| + X(t) + \bar{E}\|M_2\|)$
Delete One Connection Object		$r(\|O(t)/Q\|\|Q\| + \bar{E})$
Delete Multiple Connection Objects		$r(\|O(t)/Q\|\|Q\|\|Y_1\| + \|M_1\|\bar{E}$
Update Conditional Attribute Values		$r(\|O(t)/Q\|\|Q\|\|Y_1\| + (\|M_1 \cup M_3\| + \|M_2\|)\bar{E}$
Update Decision Attribute Values		$r(\|O(t)/Q\|\|Q\|\|Y_3\| + \|M_4\|\bar{E})$

r is used to represent the time complexity and $\bar{E}$ denotes the average length of the equivalence classes. Normally, O is used to represent time complexity, but the symbol has been used to represent an information system.

From the table, the main advantage of the incremental methods is reflected in changing the business logics of traversing all the connection objects to traversing all the equivalence classes.

5.7. Efficiencies of the Incremental Updating Operations

These incremental operations are then compared with the non-incremental operations in computation efficiencies. For a business IT context with 10 attributes, the system firstly constructs the connection objects for all the vulnerabilities in NVD database using the Method 2 in Section 4.3.4.1. Some cases are tested which are shown in the first column in Table 5.3. The programming language used is Python, and the CPU is a 3.1 GHz Intel Core i5 CPU. Table 5.3 shows the computation times of both the non-incremental and incremental updating methods, and the unit of the numbers is second. The results show the incremental methods has the advantage in

Table 5.3: Computation Time in Some Cases

Operation	Non-incremental Updating Method	Incremental Updating Method
Add 1 Connection Object	183.1	19.4
Add 10 Connection Objects		23.2
Delete 1 Connection Object		21.2
Delete 10 Connection Objects		20.4
Update Attribute Values of 40% Connection Objects		93.2

computation time.

The work in [3] is an existing system using the non-incremental updating method to obtain useful connection objects. To my best knowledge, there is no more connection strategy using rough set based incremental updating approach in this problem domain. However, from the test in this section, It is obvious that these incremental updating operations can increase the computation efficiency. This also means that, if any other connection strategy using rough set approach exists, the incremental updating operations can still be more advanced in computation time.

5.8. Summary

In this chapter, a dedicated incremental updating strategy for CTI in dynamic business IT context is developed. The incremental updating strategy includes the operations to update the equivalence classes, lower and upper approximations when one or more connection objects are added into the information system, when one or more connection objects are deleted from the information system and when multiple attribute values are updated in the information system. These operations are corresponding to when one or more CTI feeds are added into the system, when one or more CTI feeds are removed from the system and when the attribute values of

business IT context objects change. These proposed methods show the advantage in computation time comparing with the non-incremental methods.

So far, the static part and the dynamic part of the mathematical representation have been completed. In the next chapter, an application of BDC system is developed, and the design and development of the system will also presented.

Chapter 6 An Application of Business IT Context Oriented Dynamic Cyber Threat Intelligence

6.1. Introduction

This chapter introduces an application of business IT context oriented dynamic CTI (BDC system).

The chapter firstly introduces a design of the application. Some software development models are created to represent the development process. The application has two environments that are test environment and deployed working environment for different groups of users. The test environment is deployed with the functionality for expert users. The deployed working environment is deployed with the functionality for expert users and normal users. Use case diagrams are created to describe the human interactions with the system. Class diagrams are created to show the internal classes and their associations, and observer pattern is used for the development of the monitor-trigger mechanism. Activity diagrams are created to show the primary business processes of the application.

The chapter then introduces an implementation of the working environment of the system. The working environment uses Client-Server architecture and the communication between the clients and the server is based on RESTful APIs. The clients use a script based method to periodically collect the business IT information and then post it to the server in a JSON file. With the initial business IT information, the server assembles applicable connection objects for these clients using the

CTI data in the database. With new business IT information and the further CTI data, the server periodically updates the knowledge bases and periodically sends the analysis results to the clients. The server does not conduct any analysis for business security decisions. All the analysis results come from the CTI data, and the server just needs to retrieve them from the appropriate locations in the CTI data.

6.2. A Design of the Application

This section uses software development models to represent the development process of application.

6.2.1. The Use Cases

6.2.1.1. The Use Cases for the Test Environment

Fig. 6.1 presents the use cases for the expert users in test environment.

Adjust CTI Sources

"Adjust CTI sources" is to adjust the sources of the CTI feeds used for BDC system. Fig. 6.2 shows the relationship between a CTI source, business IT contexts and the generated connection knowledge bases. A set of CTI feeds from a specific CTI source and a specific business IT context can generate a connection knowledge base. Thus a system response of a path of the use case is to generate new connection knowledge bases for all the business IT contexts after the existing CTI source is changed to another one. Moreover, a new set of CTI feeds can be used to improve the existing connection knowledge bases. When new CTI feeds is obtained from new sources, a response of a path is to generate new connection objects and insert them into the information system.

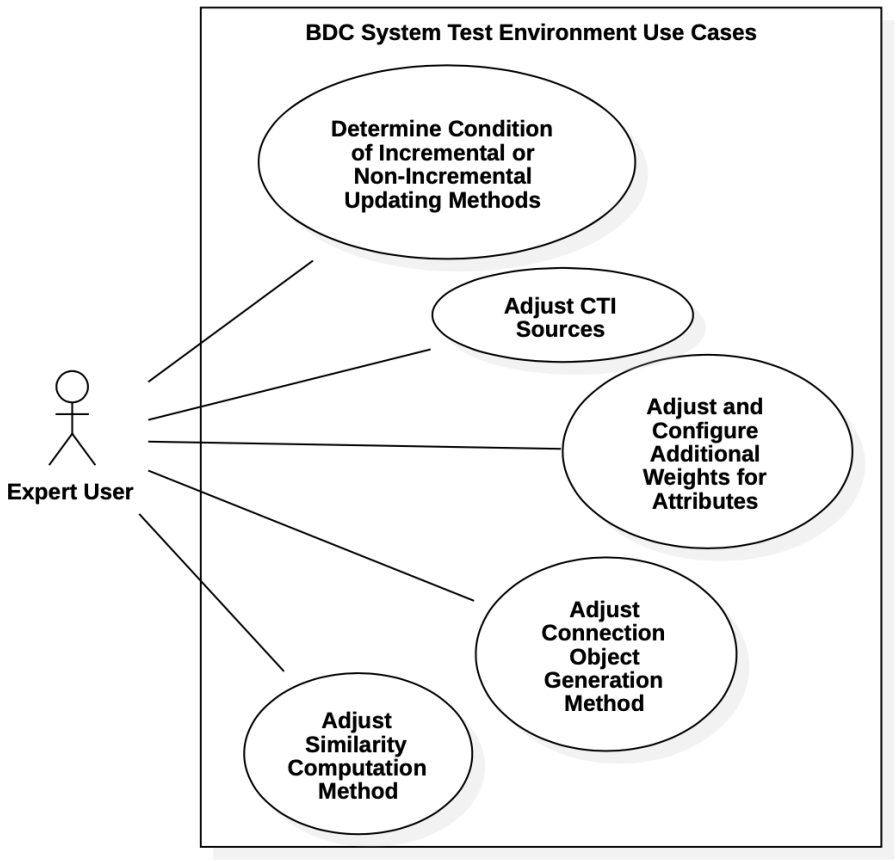

Figure 6.1: Use Cases in Test Environment

Determine Conditions of Incremental or Non-incremental Updating Methods

The proposed incremental methods have their advantage, but the system does not use the incremental methods all the time. When some conditions are met, it might be more efficient to use non-incremental methods. For example, when an extremely large set of CTI contexts are added into the information system, a re-computation of the model might be more suitable. Therefore, in test environment, the expert users can determine the specific conditions for when to use the incremental methods and when to use non-incremental methods.

Adjust and Configure Additional Weights for Attributes

For the similarity computation in BDC system, additional weights can be integrated on the attributes used. For example, the expert users can discuss and

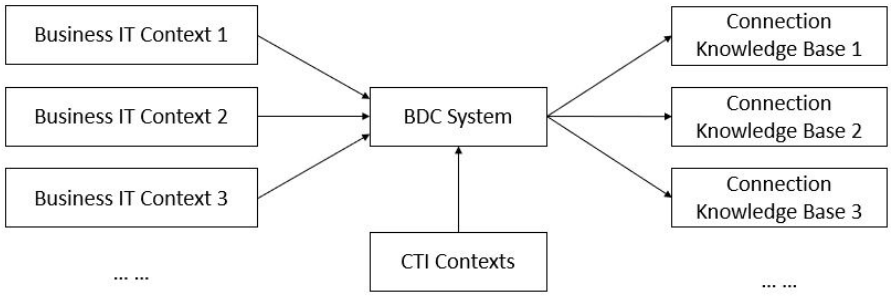

Figure 6.2: Relationship Between a CTI Source, Business IT Contexts and Connection Knowledge Bases

determine the order of the weights according to the importances of the attributes, and then the order can be processed into values between 0 to 1 as the weights.

Adjust Connection Object Generation Method

The expert users can adjust the method used to generate the connection objects. BDC system for now develops two methods, as in Chapter 4.3.4.1. One method is using discrete CTI attribute values as the corresponding condition attribute values of the connection object without any further processing, and the other method is comparing both the CTI context attribute values and the business context attribute values to get one value as the condition attribute values. Both of the methods use discrete similarity values as the decision values. The expert users can switch between the two methods.

Adjust Similarity Computation Method

BDC system uses a set distance based method to get the similarities between a business IT context and CTI contexts. The expert users can adjust the similarity computation method, as long as it does not conflict with the corresponding definitions in the system.

6.2.1.2. The Use Case for the Working Environment

Fig. 6.3 presents the human interactions with BDC system in the deployed working environment. Some of the use cases have been introduced in the methodology chap-

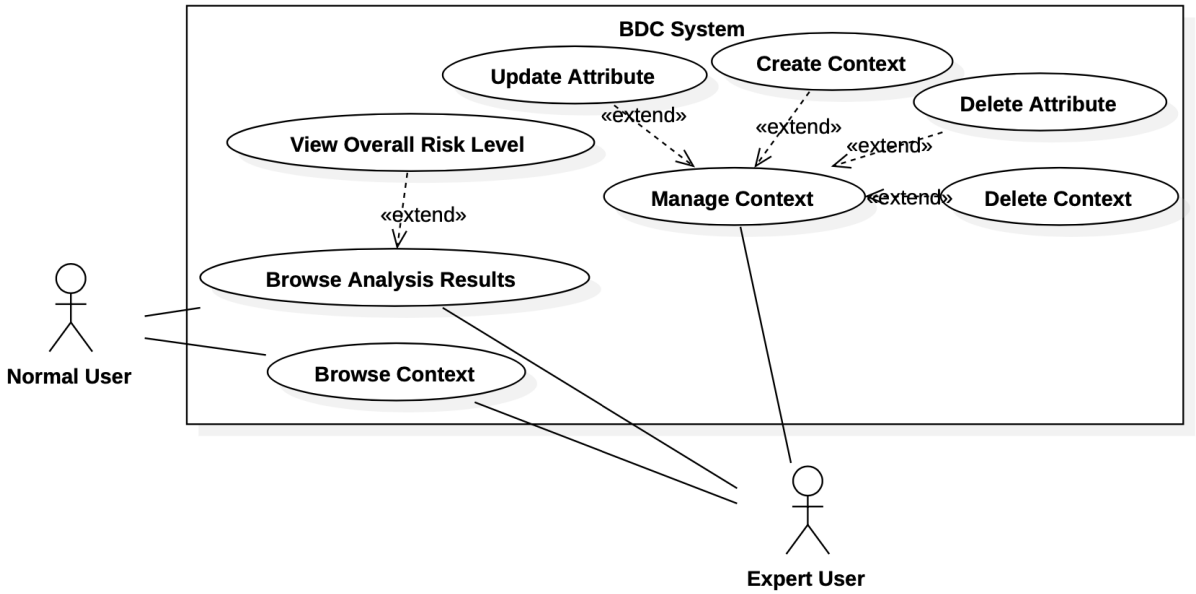

Figure 6.3: Use Cases in Deployed Working Environment

ter. The figure details the use cases. Normal users will only use the system in BDC client environment, and expert users can use the system in both client and server environments. This means "Manage Context" and the extended use cases are for server side, and the other use cases are for client side. The added use case is an extension to "Browse Context" that is "View Overall Risk Level". The risk level can be retrieved based on the information from the CTI feeds identified as relevant. The users clicks a button and the response of the system is to send the retrieved analysis results.

6.2.2. The Class Diagrams

6.2.2.1. Observer Pattern

In the deployed working environment, observer pattern is used for the monitor and trigger mechanisms. Fig. 6.4 shows the relationship and the operations for a subject

and the observers. Observer pattern allows the subscribed observers to observe a

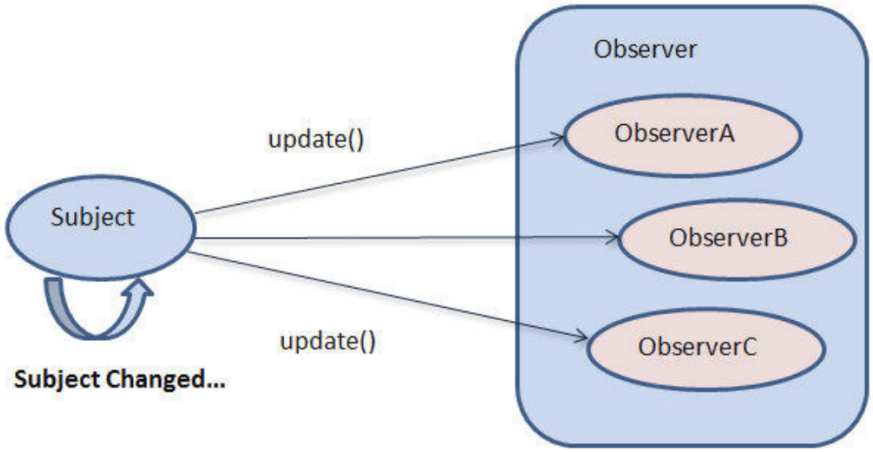

Figure 6.4: Observer Pattern

subject, and when the attributes of the subject are changed, the observers can then be correspondingly updated. It establishes a trigger mechanism between a subject and the observers. As for the monitor, a `Timer` class is used by the corresponding classes to periodically invoke the operations to achieve the fully monitor-trigger function. More details are shown in the class diagram in Fig. 6.5.

6.2.2.2. The Class Diagrams for the Working Environment

Fig. 6.5 shows the class diagram of the application in the working environment. Key and important classes with their attributes and operations are listed in the diagram and explained in this section.

- `BDCClientDemo`: This class models the client environment of BDC system with the functions of the designed business processes. The detailed business processes are shown in Fig. 6.7.
- `BDCServerDemo`: This class models the server environment of BDC system with the function of the designed business processes. The detailed business processes are shown in Fig. 6.6.
- `ISubject`: This is an interface describing an abstracted target observed by

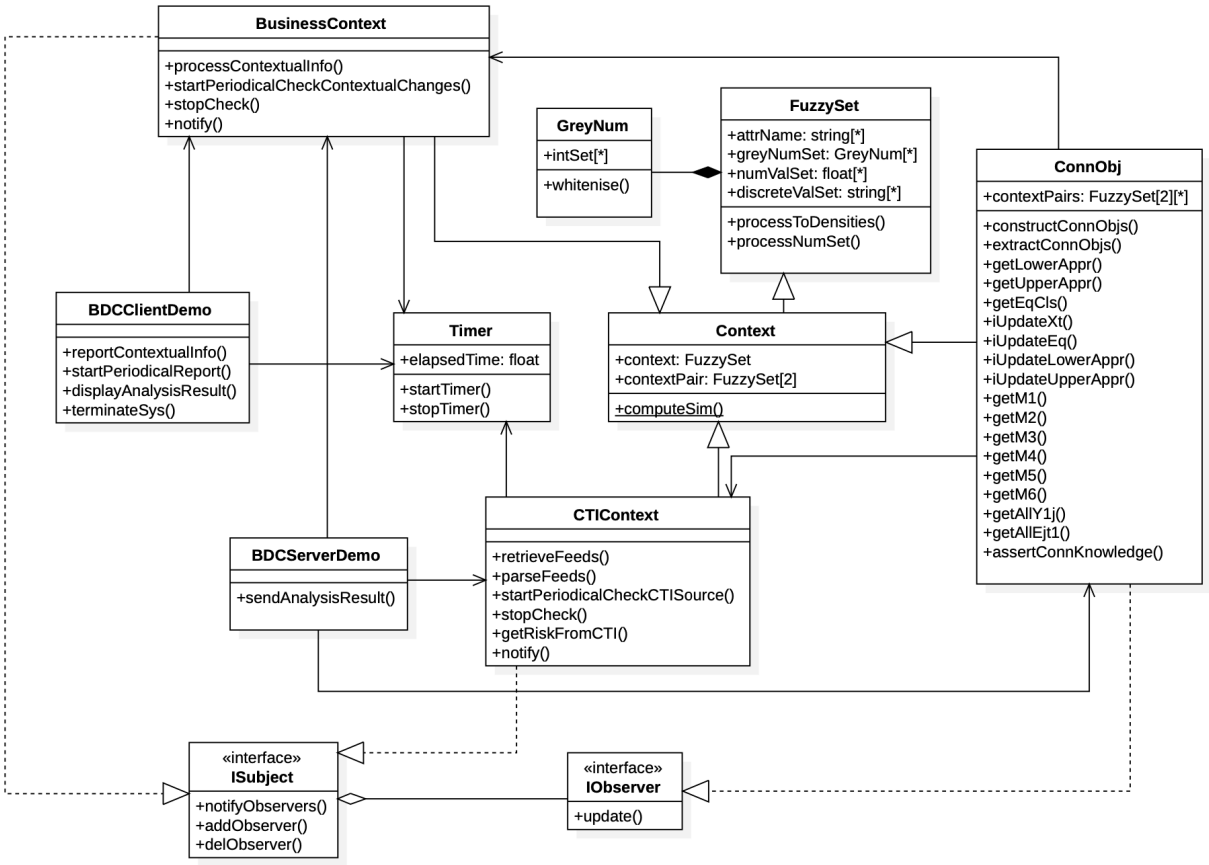

Figure 6.5: Class Diagram for BDC System

the observers. It has three functions that can be inherited. `addObserver()` and `delObserver()` are the two operations to register and remove observers. `notifyObservers()` is used to invoke the observers' corresponding functions after the subject's attributes change.

- `IObserver`: Working with `ISubject`, this is an interface describing an abstracted observer. The interface has a function `update()` that can be inherited by concrete observer classes.

- `GreyNum`: BDC system uses discrete grey numbers to represent the context objects' attributes. This class describing a grey number. `intSet` contains a set of integers representing those possible attribute values. The minimal value is 0 and the maximum value is the quantity of all the IT components of a context object. This class has a function `whitenise()` that is used to whitenise the grey number. In BDC system, the function is to average all the

possible values.

- `FuzzySet`: BDC system uses fuzzy set to represent a context object. `greyNumSet` attribute is a set of `GreyNum` classes. This class has `attrName` describing and containing the names of the attributes, `numValSet` containing the numerical attribute values and `discreteValSet` containing the discrete density values. This class have the corresponding operations `processToDensities()` based on the defined fuzzy set operations and `processNumSet()` to process the grey number instances to numerical values between [0,1].

- `Context`: This class inherits `FuzzySet` and it has another attribute `contextPairs` that is an attribute of pairs of fuzzy sets representing pairs of contexts. It has an operation `computeSim()` to compute the similarity between the pairs of fuzzy sets.

- `BusinessContext`: This class describing a business context, and this class is a concrete subject implementing `ISubject` interface. `processContextInfo()` is used to translate the business contextual information to JSON object. `notify()` is the operation to invoke its observers' functions. This class has other attributes and operations to achieve the functions of the main business processes of `BDCClientDemo` and `BDCServerDemo`.

- `CTIContext`: This class describing a CTI context. This class is a concrete subject implementing `ISubject` interface. `notify()` is the operation to invoke its observers' functions. This class has other attributes and operations to achieve the functions of the main business processes of `BDCClientDemo` and `BDCServerDemo`.

- `ConnObj`: This class describing a rough set based information system. This class is a concrete observer observing two classes `BusinessContext` and `CTIContext` that implement `IObserver` interface. When the defined attributes of the two concrete subject classes change, the operations in this case will then be correspondingly invoked. It has the operations to assemble connection objects, and it has the operations achieving the incremental updates. `assertConnKnowledge()`

is the operation used to assert whether a CTI feed meeting the conditions of the connection knowledge.

- `Timer`: This class is used to periodically invoke the operations of the classes using this class.

6.2.3. The Activity Diagrams

6.2.3.1. Business Activities of BDC Server Environment

Fig. 6.6 shows the business process of BDC server environment. It models the main activities of an instance of `BDCServerDemo` class. The server will firstly "Construct CTI context and the business IT context", and then it "Assemble useful connection objects". After that, it "Assert CTI feeds and contexts meeting the conditions of useful connection objects". With those CTI feeds asserted `True`. The server then "Locate the information for creating analysis report in the CTI feeds and the CTI contexts asserted as True". For example, the server will locate the risk level information in CTI feed files. After the information is located, the server "Retrieve the located information and process it into the pre-defined JSON file", and then the server "Send the JSON file to BDC client" through the RESTful APIs. After that, the timer starts to begin monitoring. The server uses the timer to periodically "Check whether CTI contexts change" and "Check whether business context changes". If the two contexts are changed, the server will "Incrementally updating useful connection objects" or go to non-incrementally "Assemble useful connection objects".

6.2.3.2. Business Activities of BDC Client Environment

Fig. 6.7 shows the business process of BDC server environment. It models the main activities of an instance of `BDCClientDemo` class. The client firstly "Run the pre-defined scripts to collect the corresponding contextual IT information". The client then "Process the collected contextual IT information into the pre-defined JSON file". After the processing, the client "Post the JSON file to BDC server through

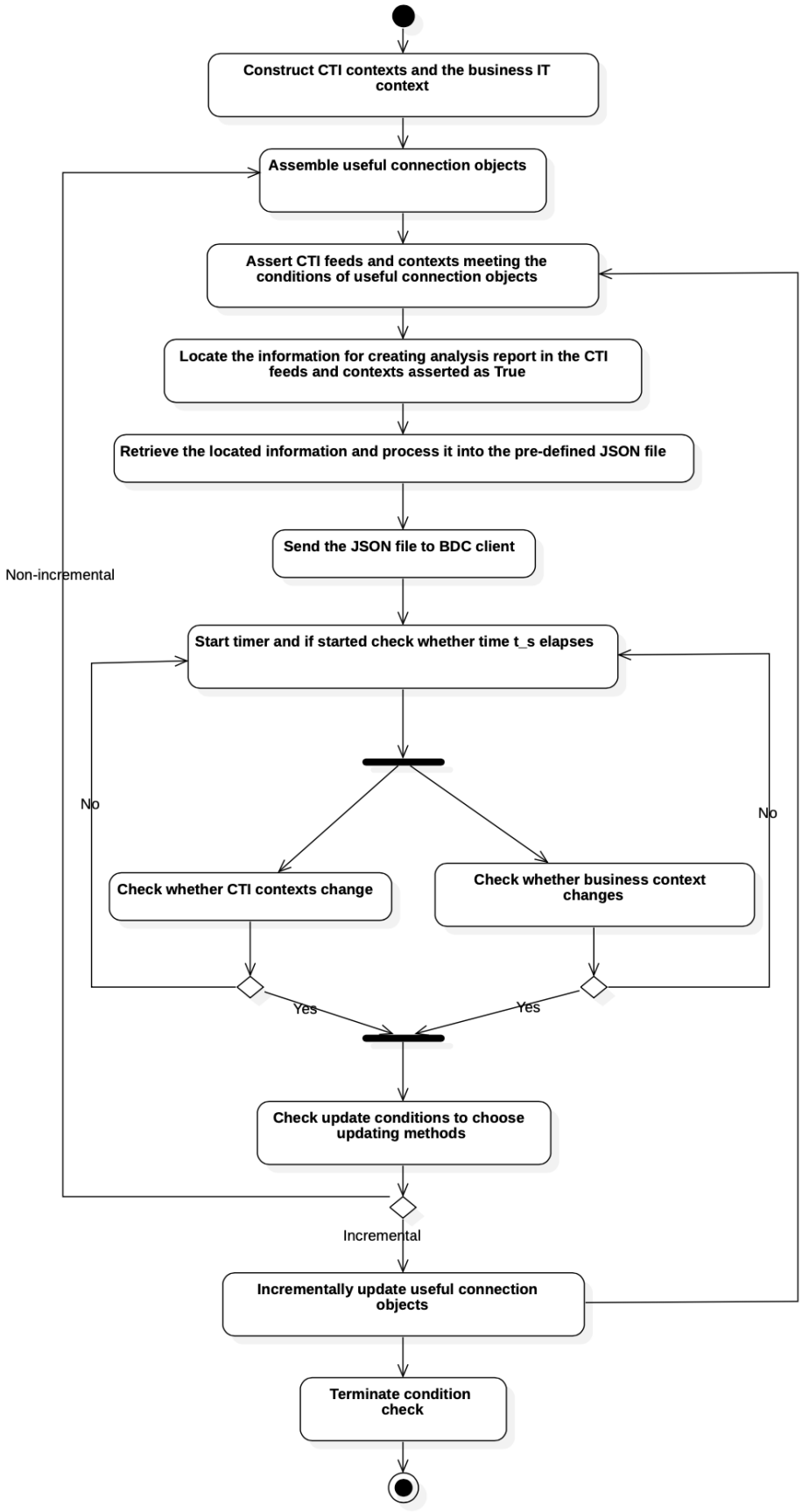

Figure 6.6: Business Activities of BDC Server Environment

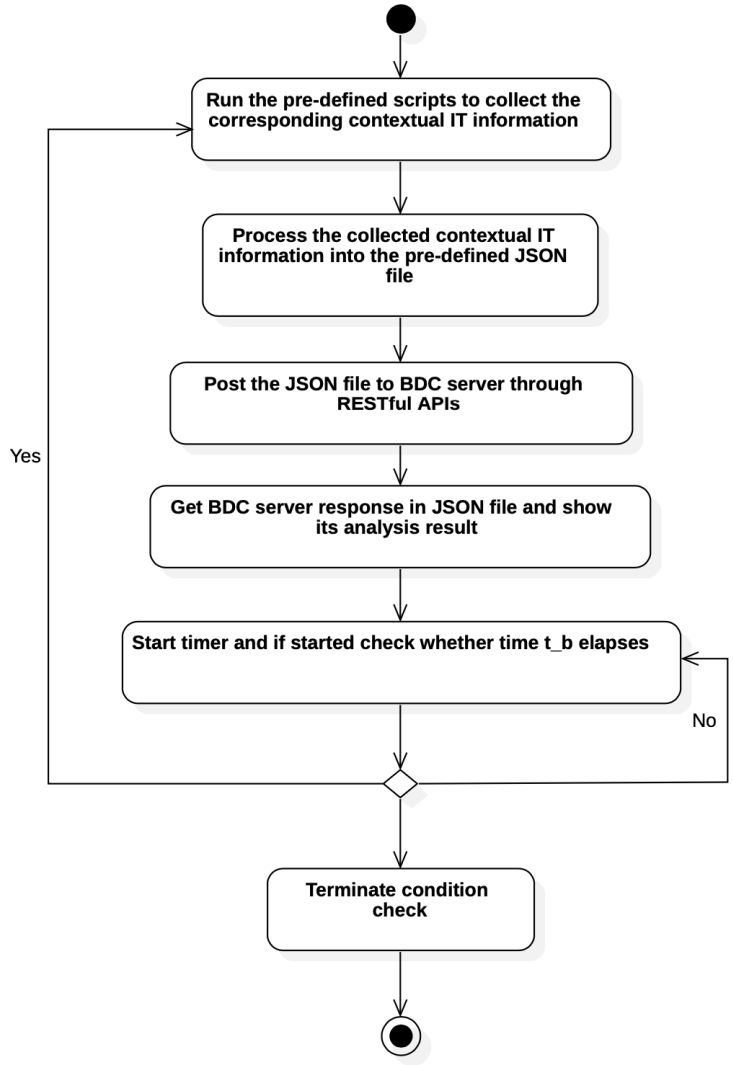

Figure 6.7: Business Activities of BDC Client Environment

RESTful APIs". After that, the client "Get BDC server response in JSON file and show its analysis result" to normal users. The timer then starts and the system can periodically perform the activities.

6.3. An Implementation of the Working Environment

6.3.1. Scripts Based Business IT Contextual Information Collection Method

A set of scripts are deployed in the working environment of BDC system. After an investigation, BDC system defines and deploys the following types of collection methods of collecting business IT contextual information, listed as follows.

- Operating system based scripts: Operating systems have their own programs to collect environmental information. For example, the following bash is used to collect cpu information of MacOS: `sysctl -a | grep machdep.cpu`.
- Programming language based scripts: Almost all OS expose APIs to be invoked by programming languages to obtain system environmental information using the packages of these languages. For example, `wmi` is the package used by Python to collect Windows system information.
- Other methods: Some "clever" methods can also be used to collect the IT contextual information. For example, if BDC system would like to know what applications are installed on a BDC client, BDC system just needs to know the file names in the folder containing the applications. Additionally, the users can manually input the contextual information.

With the designed scripts based method, the system can collect almost all IT contextual information as long as corresponding permissions are in place.

6.3.2. RESTful Architecture Between BDC Clients and BDC System Server

BDC system applies RESTful architecture to communicating business IT contextual information and analysis results between BDC clients and BDC servers. Fig. 6.8 shows the communication architecture.

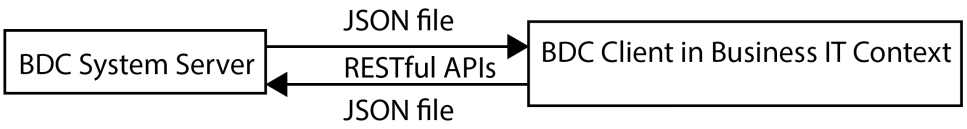

Figure 6.8: The Communication Between BDC Clients and BDC System Server

BDC client in business IT context uses the script based method to collect the IT environmental information, and the client processes the contextual information into a JSON file. An example of the JSON file for storing the business IT contextual information is presented in Listing 6.1.

Listing 6.1: An Example Business IT Contextual Information JSON File

```
{
    "client_id": 1,
    "communication_type": "contextual_info",
    "href": "example href",
    "business_context": {
        "software": [
            {"software1name": "software1value",
             "software2name": "software2value",
             "software3name": "software3value"
             }
        ],
        "hardware": [
            {"hardware1name": "hardware1value",
             "hardware2name": "hardware2value",
             "hardware3name": "hardware3value"
            }
        ]
    }
}
```

The JSON file serialises the information to make it processable for BDC system server. `client_id` indicates the business IT context id, and `communication_type`

indicates the communication type is for business IT context. Two types of information are collected and `software` and `hardware` show the type names. In each type, there are three IT components collected by the scripts. `software1name: software1value` is a key-value pairs for one IT components. The JSON can have more items regarding the contextual IT components.

Another JSON file is for BDC system server to convey the analysis result for a specific business IT context. Listing 6.2 shows an example JSON file for storing the information retrieved from CTI data. `client_id` indicates the response is for the client with the id, and the communication type is `analysis_result`. The analysis includes the information of an analysis of `overall_risk` is `high` and `asset_loss` is GBP1000 calculated for the business IT context. The JSON serialises the analysis result to make it processable for BDC client. The JSON file can have more items regarding analysis information.

Listing 6.2: An Example Response JSON File

```
{
    "client_id": 1,
    "communication_type": "analysis_result",
    "href": "example href",
    "analysis": {
        "overall_risk": "high"
        "asset_loss": "GBP1000"
    }
}
```

6.3.3. How the System Uses CTI Data With the Applicable Connection Objects

This section introduces how the system uses CTI data with the generated applicable connection objects.

Using the existing CTI feeds, the system can construct the connection objects

for a business IT context, and the system can then extract the useful connection objects. After that, the system can determine whether a CTI feed is applicable with the useful connection objects using an assertion.

Definition 6.1. *Let CTI denote all the CTI feeds. The function of Assert is defined as*

$$Assert : CTI \longrightarrow \{True, False\} \tag{6.1}$$

$True$ means the CTI feeds are applicable, and $False$ means the CTI feeds are inapplicable. For example, an $Assert$ can be instantiated as: If the decision attribute is $high$ and $medium$, the system asserts the corresponding CTI feeds are $True$, and otherwise, $False$. Those useful connection objects have name-value pairs of the condition attributes and the decision attribute. These name-value pairs of the attributes of all useful objects are the conditions used for determining whether or not a CTI context is applicable. After the CTI context, corresponding to a CTI feed, is determined as applicable, the system can locate the pre-defined useful information the system needs in the CTI feed. For example, the risk level information is valuable and the system can locate the information in the JSON files if the CTI feeds obtained from a source have such information.

6.3.4. A Case Study

This section demonstrates a case study of how the implemented system provides the users with useful information.

Regarding the development techniques used, Python is the programming language used for the implementation and Flask [53] is used to develop the RESTful software architecture. The JSON objects used for the communication are shown in List 6.3 and List 6.4. List 6.3 has the business IT contextual information and the IT components are input into "IT Components". List 6.4 has the information of the response of the system with the risk information retrieved from the applicable CTI feeds.

Listing 6.3: Business Contextual Information

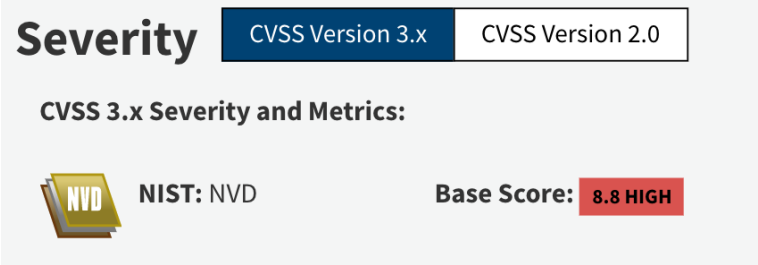

Figure 6.9: CVSS on the Webpage

```
{
    "client_id": 1,
    "communication_type": "analysis_result",
    "href": "api.bdcsystem.example",
    "analysis": {
        "it_components": [(will be retrieved from the CTI data)]
    }
}
```

Listing 6.4: The Response

```
{
    "client_id": 1,
    "communication_type": "analysis_result",
    "href": "api.bdcsystem.example",
    "analysis": {
        "risk": {will be retrieved from the CTI data)}
    }
}
```

After that, what analysis results in CTI are identified. Fig. 6.9 shows a CVSS [54] score on NVD website. CVSS is a scoring system for scoring the vulnerability severity. The information can also been found in the vulnerabilities, in JSON, downloaded from [7]. Fig. 6.10 shows the location in a JSON file. Therefore, the CVSS base score is selected as the analysis result and the score will be sent to the

```
"cvssV3" : {
  "version" : "3.1",
  "vectorString" : "CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H",
  "attackVector" : "NETWORK",
  "attackComplexity" : "LOW",
  "privilegesRequired" : "NONE",
  "userInteraction" : "NONE",
  "scope" : "UNCHANGED",
  "confidentialityImpact" : "NONE",
  "integrityImpact" : "NONE",
  "availabilityImpact" : "HIGH",
  "baseScore" : 7.5,
  "baseSeverity" : "HIGH"
},
```


Figure 6.10: CVSS in a JSON Object

clients.

Following that, the system assembles the connection objects for a business IT contexts. To test the feasibility of the implementation, some software names are input into the database in the client as the business IT contextual information. These items are input into the value of "it_component" in List 6.3. The client sends the JSON object with the contextual information to the server through the RESTful APIs. Using these software names, the system then assembles 32 connection objects using the vulnerabilities from NVD. With these applicable connection objects, the system asserts the CTI feeds using: If the decision attribute is *high* and *medium*, the system asserts the corresponding CTI feeds are $True$, and otherwise, $False$. The system then locates and retrieves the CVSS scores from those CTI feeds asserted as $True$, and inputs the highest score into the value of "risk" in Listing 6.4. In this step, the system gets 4.3 that is the highest score. The system sends the JSON object with this score to the client.

Afterwards, the client periodically sends the contextual information to the server and the server periodically sends back the risk to the client. If the server needs to update the connection objects, the system invokes the incremental updating operations to update the applicable connection objects and sends a new highest score to the client.

6.4. Summary

In this chapter, an application of business IT context oriented dynamic CTI is developed. The application of BDC system has two environment, test environment and deployed working environment. The use cases instructing the human interactions with the system in this environment are created. In the working environment, the application has system server users and the client users in business environments. RESTful software architecture is used for the communication between the server and the clients. The server can assemble connection knowledge and provide analysis using CTI for the clients. The clients are deployed with a set of scripts to obtain various business IT contextual information. To achieve the functions of the monitor and trigger mechanisms, the application uses observer design pattern and a timer to periodically invoke the corresponding functions. Business context and CTI context are defined as the subjects of the pattern, and the information system is defined as the observer of the pattern. With the pattern, every time the attributes of the subjects change, the corresponding observers' functions will be invoked. Moreover, the use cases of the users' interactions with the system in working environment, the activity diagrams describing the server's business processes and clients' business processes and the class diagram instructing how to develop the system are created and presented.

Chapter 7 Discussions and Conclusions

This chapter summaries the research, the contributions to knowledge and the future work.

7.1. Summary of Research

The research presented in this thesis has made many achievements, and this section summarises the work. In the research project's heart, different system modelling and computation approaches to the problem domain of "how to dynamically connect CTI onto business IT context" are employed, including grey number theory, fuzzy set theory, rough set theory, rough set based incremental updating approach related to computational intelligence models and use case diagram, class diagram and activity diagram related to software computation models.

In summary, this thesis has describes the following works.

- A mathematical representation is developed for interpreting the data of CTI contexts and business IT contexts. Discrete grey numbers are used to represent contextual attributes. Different groups of fuzzy sets with different purposes are used to represent the contexts and the similarities between the contexts. Rough set theory is used to represent the connection knowledge space.

- A software development representation is developed for instructing the application of the system. Use case diagrams are created and used to represent human interactions with the system to instruct the users' interactions with the system. Class diagrams are created and used to represent the internal classes and their associations for developers. Activity diagrams are created

and used to describe the business processes of the system.

- Using an instantiation of the mathematical representation, a connection knowledge assembly system is developed that can be used to identify CTI relevant to business IT contexts.

- An incremental updating approach to incrementally updating the approximations is developed for a more efficient computation of useful connection knowledge.

- Instructed by the software development representation, a business IT context oriented dynamic CTI system is implemented. A RESTful APIs communication method is developed for the communication between the system server and the clients. A script based contextual information collection method is developed in the application. A method of how the system uses CTI feeds is developed, in which the system will locate the information that can inform security decisions from the CTI feeds the contexts of which are identified as relevant using the assembled applicable connection knowledge.

7.2. Summary of Contributions to Knowledge

This thesis aims to address the research gap that is "how to dynamically connect CTI onto business IT context" by:

- A. developing a system analysis and modelling approach that can interpret the data and the computations for the aim, and then using the interpretations to develop a static connection rule based system for identifying relevant CTI for a business contexts

- B. developing an incremental updating strategy to updating the approximations used to extract useful connection knowledge with better computation efficiency

- C. developing an application using advanced and mainstream software development methods to applying the system to a business IT context. The appli-

cation is deployed with a monitor-trigger mechanism to inform the system the environmental changes to trigger the updates of the connection knowledge to achieve the dynamic aspect of the system.

This section summarises the reflections on the contributions that this thesis has made to knowledge.

A novel hybrid representation with selected computational intelligence models, including grey numbers, fuzzy set theory and rough set theory, and software development models, use case diagram, class diagram and activity diagram, is designed and developed. This is a bold attempt in the research context and the feasibility is tested as successful in this research project. The generalised representation can interpret the data, and the work such as [4] is a specific case of the generalisation. A novel view of representing the potential term mappings between the IT components in CTI and business contexts is identified, and grey numbers are used to represent the contextual attribute values to reflect the possible mappings between the IT components. This is the first attempt in the research context. This extends the method such as [14] that determines a connection with one identical IT component. Using the representation, a connection rule assembly system is developed. It proves that it is feasible and effective to apply the computation intelligence models to the research context. In the system, the discretisation for the attribute values provides a further semantic interpretation on the attribute values, such as similarity, in the work such as [3] [5]. An incremental updating strategy with multiple operations based on rough set theory used to incrementally updating the approximations is developed. This shows and proves the advantages of the system in computation efficiency. This increases the performance of the work such as [4] in computation time. Using the representation, an application of the system is successfully developed for informing the business IT context their security postures using the retrieved analysis results from the applicable CTI data. This proves the value of CTI and the value of this research project.

7.3. Future Work

In future, as for the connection rule assembly system, a detailed attributes used for the similarity computation and the construction of the connection objects can be developed, and feature selection methods can be developed to select the relatively more effective attributes. In the incremental updating strategy, more cases to testing the computation times can be conducted to better distinguish the conditions of when to use incremental updating methods and when to use non-incremental updating methods in the research context. More rough set related structures can be updated incrementally, such as the reducts and cores. More incremental updating operations can be developed, such as the incremental operations used to updating the approximations when attributes, not attribute values, are added or deleted. Regarding the application, a detailed list of scripts used to collect business IT contextual information can be developed.

7.4. Conclusion

CTI is increasingly available, and there is a need of instructing how to use CTI in a dynamic context because the CTI might become inappropriate and inapplicable when the context changes. Therefore, a business IT oriented dynamic CTI system is proposed and developed in this thesis. The system has a hybrid representation using computational intelligence models, including grey numbers, fuzzy set theory and rough set theory, and software development models, including use case diagram, class diagram and activity diagram. Using the representation, a connection rule based system is developed. To efficiently update the approximations with useful connection knowledge, a rough set based incremental updating strategy with multiple operations is developed. An application of the system is developed using mainstream software development techniques. A monitor-trigger mechanism is deployed in the application to monitor the business IT contextual changes and trigger the updates of the connection knowledge. The system can provide the business IT context with timely support and useful information that can inform security

decisions.

Bibliography

[1] "Cyber security breaches survey 2021." [Online]. Available: https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2021/cyber-security-breaches-survey-2021

[2] "STIX and TAXII." [Online]. Available: https://oasis-open.github.io/cti-documentation/

[3] Y. Xu, Y. Yang, and Y. He, "A business process oriented dynamic cyber threat intelligence model," in *Proceedings of the 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, 2019, pp. 648–653.

[4] ——, "A representation of business oriented cyber threat intelligence and the objects assembly," in *Proceedings of 10th International Conference on Information Science and Technology*, 2020.

[5] S. Qamar, Z. Anwar, M. A. Rahman, E. Al-Shaer, and B.-T. Chu, "Data-driven analytics for cyber-threat intelligence and information sharing," *Computers & Security*, vol. 67, pp. 35 – 58, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404817300287

[6] Ponemon Institute LLC, "The value of threat intelligence: Annual study of north american & united kingdom companies."

[7] "National Vulnerability Database." [Online]. Available: https://nvd.nist.gove

[8] D. Chrismon and M. Ruks, "Threat intelligence: Collecting, analysing, evaluating," 2015.

[9] S. Barnum, "Standardizing cyber threat intelligence information with the structured threat information expression (stix™)," Feb 2014. [Online]. Available: http://stixproject.github.io/getting-started/whitepaper/

[10] E. W. Burger, M. D. Goodman, P. Kampanakis, and K. A. Zhu, "Taxonomy model for cyber threat intelligence information exchange technologies," in *Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security*, ser. WISCS '14. New York, NY, USA: Association for Computing Machinery, 2014, pp. 51–60. [Online]. Available: https://doi.org/10.1145/2663876.2663883

[11] "HAIL A TAXII." [Online]. Available: http://hailataxii.com/

[12] "Automated Indicator Sharing." [Online]. Available: https://www.us-cert.gov/ncas

[13] "ThreatConnect." [Online]. Available: https://threatconnect.com/

[14] "MISP." [Online]. Available: https://www.misp-project.org/index.html

[15] "National Cyber Awareness System." [Online]. Available: https://www.us-cert.gov/ncas

[16] "Dynamic business process mangement." [Online]. Available: https://www.gartner.com/en/information-technology/glossary/dynamic-business-process-management-bpm

[17] P. Y. . T. C. Osterwalder, A., "Clarifying business models: Origins, present, and future of the concept," *Communications of the Association for Information Systems*, vol. 31, 2005.

[18] L. Aldin and S. de Cesare, "A literature review on business process modelling: new frontiers of reusability," *Enterprise Information Systems*,

vol. 5, no. 3, pp. 359–383, 2011. [Online]. Available: https://doi.org/10.1080/17517575.2011.557443

[19] F. M. M. Thomas, O., "Semantic process modeling – design and implementation of an ontology-based representation of business processes," *Bus. Inf. Syst. Eng.*

[20] "Bpmn." [Online]. Available: https://www.bpmn.org

[21] H. Smith and F. Peter, "Business process management: The third wave," 2003.

[22] E. Hutchins, M. Cloppert, and R. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research*, vol. 1, 01 2011.

[23] A. Ahmad, J. Hadgkiss, and A. Ruighaver, "Incident response teams – challenges in supporting the organisational security function," *Computers & Security*, vol. 31, p. 643–652, 07 2012.

[24] M. Vielberth, F. Böhm, I. Fichtinger, and G. Pernul, "Security operations center: A systematic study and open challenges," *IEEE Access*, vol. 8, pp. 227 756–227 779, 2020.

[25] A. Torres, "Building a world-class security operations center: A roadmap," 2015.

[26] EY, "Security operations centers against cybercrime," 2013.

[27] T. B. T. P. Gorka Sadowski, Avivah Litan, "Market guide for user and entity behavior analytics," 2018.

[28] A. Chuvakin and A. Barros, "Preparing your security operations for orchestration and automation tools," 2018.

[29] Y. Merah and T. Kenaza, "Ontology-based cyber risk monitoring using cyber threat intelligence," 2021, cited By 0. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85113233715&doi=10.1145%2f3465481.3470024&partnerID=40&md5=e925876ff2b8be16d33f28814e57f190

[30] A. Gylling, M. Ekstedt, Z. Afzal, and P. Eliasson, "Mapping cyber threat intelligence to probabilistic attack graphs," 2021, pp. 304–311, cited By 0. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85115727510&doi=10.1109%2fCSR51186.2021.9527970&partnerID=40&md5=f0a6fc19c2b279a8df5eea25defd99f4

[31] Y. Qin, Y. Peng, K. Huang, C. Zhou, and Y.-C. Tian, "Association analysis-based cybersecurity risk assessment for industrial control systems," *IEEE Systems Journal*, vol. 15, no. 1, pp. 1423–1432, 2021, cited By 0. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85102730038&doi=10.1109%2fJSYST.2020.3010977&partnerID=40&md5=492c3b0136f98d9c70e4531c30f03dfb

[32] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338 – 353, 1965.

[33] L. A. Zadeh, "Fuzzy logic," *Computer*, vol. 21, no. 4, pp. 83–93, April 1988.

[34] Z. Pawlak, "Rough sets," *International Journal of Computer & Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982. [Online]. Available: https://doi.org/10.1007/BF01001956

[35] Q. Zhang, Q. Xie, and G. Wang, "A survey on rough set theory and its applications," *CAAI Transactions on Intelligence Technology*, vol. 1, no. 4, pp. 323–333, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468232216300786

[36] Y. Yang and R. John, "Grey sets and greyness," *Information Sciences*, vol. 185, no. 1, pp. 249 – 264, 2012.

[37] L. Sifeng, F. Zhigeng, Y. Yingjie, and F. Jeffrey, "General grey numbers and their operations," vol. 2, no. 3, pp. 341–349, 2020/03/13 2012. [Online]. Available: https://doi.org/10.1108/20439371211273230

[38] G. D., "Classification of dynamics in rough sets," in *Szczuka M., Kryszkiewicz M., Ramanna S., Jensen R., Hu Q. (eds) Rough Sets and Current Trends*

in Computing. RSCTC 2010. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2010.

[39] H. Chen, T. Li, D. Ruan, J. Lin, and C. Hu, "A rough-set-based incremental approach for updating approximations under dynamic maintenance environments," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 2, pp. 274–284, 2013.

[40] W. Ziarko, "Variable precision rough set model," *Journal of Computer and System Sciences*, vol. 46, no. 1, pp. 39 – 59, 1993.

[41] D. Liu, T. Li, D. Ruan, and J. Zhang, "Incremental learning optimization on knowledge discovery in dynamic business intelligent systems," *Journal of Global Optimization*, vol. 51, no. 2, pp. 325–5344, 2011.

[42] D. Liu, T. Li, and J. Zhang, "A rough set-based incremental approach for learning knowledge in dynamic incomplete information systems," *International Journal of Approximate Reasoning*, vol. 55, no. 8, pp. 1764 – 1786, 2014.

[43] H. Chen, T. Li, S. Qiao, and D. Ruan, "A rough set based dynamic maintenance approach for approximations in coarsening and refining attribute values," *International Journal of Intelligent Systems*, vol. 25, no. 10, pp. 1005–1026, 2010.

[44] Y. Guo, E. C. Tsang, M. Hu, X. Lin, D. Chen, W. Xu, and B. Sang, "Incremental updating approximations for double-quantitative decision-theoretic rough sets with the variation of objects," *Knowledge-Based Systems*, vol. 189, p. 105082, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705119304642

[45] Y. Yang, D. Chen, and H. Wang, "Active sample selection based incremental algorithm for attribute reduction with rough sets," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 4, pp. 825–838, 2017.

[46] Y. Yang, D. Chen, H. Wang, and X. Wang, "Incremental perspective for feature selection based on fuzzy rough sets," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 3, pp. 1257–1273, 2018.

[47] M. Fowler, *UML Distilled*, 2003.

[48] R. F. Simon Bennett, Steve McRobb, *Object-Oriented Systems Analysis and Design Using UML*, 2010.

[49] "Cve-2018-12694." [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2018-12694

[50] A. S. Khuman, "The similarities and divergences between grey and fuzzy theory," *Expert Systems with Applications*, vol. 186, p. 115812, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421011805

[51] A. S. Khuman, Y. Yang, and R. John, "A commentary on some of the intrinsic differences between grey systems and fuzzy systems," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, pp. 2032–2037.

[52] "pandas." [Online]. Available: https://pandas.pydata.org/

[53] "Flask." [Online]. Available: https://flask.palletsprojects.com/en/2.0.x/

[54] "Cvss." [Online]. Available: https://www.first.org/