VALIDATING BUSINESS PROBLEM HYPOTHESES:

A GOAL-ORIENTED AND MACHINE LEARNING-BASED APPROACH

by

Sung Soo Ahn

APPROVED BY SUPERVISORY COMMITTEE:

Lawrence Chung, Chair

Farokh Bastani

Tien N. Nguyen

Shiyi Wei

This dissertation is dedicated to my mother, who has given endless love to her children.

VALIDATING BUSINESS PROBLEM HYPOTHESES:
A GOAL-ORIENTED AND MACHINE LEARNING-BASED APPROACH

by

SUNG SOO AHN, BS, MS

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

SOFTWARE ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

December 2021

ACKNOWLEDGMENTS

My humble journey into a PhD program started with some hopes and worries at UT Dallas. In the beginning, the road was unknown and not visible in the dark, but fortunately, I had met trustworthy, helpful, and kind people during my PhD program. I could manage to find a road to the destination with the help of those people.

I wish to express my sincere appreciation to my supervisor, Dr. Lawrence Chung, for his fundamental, thought-provoking, and challenging questions about my research work, which helped crystallize research ideas, problems, and solutions. I could understand, develop and refine my research ideas by standing on his shoulders. I will miss the question and answer sessions focusing on categories of essential concepts, relationships between concepts, and constraints on those concepts and relationships, with sensible examples. I would like to extend my sincere thanks to my supervisory committee members: Dr. Farokh B. Bastani, Dr. Tien N. Nguyen, and Dr. Shiyi Wei, for the kind comments and for encouraging my research efforts.

I would like to give my special regards to Dr. Tom Hill for his practical and insightful help and comments on my research work in software engineering. I also would like to pay my special thanks to Dr. Sam Supakkul and Dr. Liping Zhao for the project we performed together and the research discussion. The research communication helped me refine research ideas, write an excellent technical paper, and experiment with new ideas, resulting in good research work.

I thank the PhD students of the RE lab, Ronaldo Gonçalves Junior, Kirthy Kolluri, Anthony Opara, Niranjan Marathe, Aamir Abbas, Justin Punghee Cho, and Julie Rauer, for helping each other and getting along the long PhD journey. I also thank former PhD students Dr. Haan Mo Johng and Dr. Grace Eunjung Park for sharing their positive attitudes in achieving their goals.

I thank the parents of Cub Scout Pack 421, who were good neighbors and friends for my son and as me. I also thank my long-time friend Jeonghurk and his family, who kindly helped my family live well socially and emotionally. I also thank Dr. Sunhwa Hahn, then the president of Korea Institute of Science and Technology Information (KISTI), and my colleagues in KISTI for giving me an excellent opportunity to study my PhD program at UT Dallas.

I wish to acknowledge the support and great love of my mother, Junghee Goh. She kept me going on and this work would not have been possible without her love and sacrifice. I am indebted to my sister and brothers, who prayed for my family's safety and happiness during this journey. Last but not least, I wish to thank my wife, Sookhyun, and my children Jaeyoung and Andrew (Joonyoung) for being great companions for this long but cheerful journey. I hope this journey helps my sons make their bright future.

November 2021

VALIDATING BUSINESS PROBLEM HYPOTHESES:

A GOAL-ORIENTED AND MACHINE LEARNING-BASED APPROACH

Sung Soo Ahn, PhD
The University of Texas at Dallas, 2021

Supervising Professor: Lawrence Chung, Chair

Validating a business problem hindering a business goal is often more important than finding solutions to the problem, specifically during requirements engineering. For example, validating the impact of a client's low account, high transactions, or high loan payment per month for a client's unpaid loan decreasing the loan revenue of one bank would be critical as an information system can be designed for the bank to take some actions to mitigate the loan default. However, many business organizations are struggling to confirm whether some potential problems hidden in Big Data are against a business goal or not. In other words, they face difficulties finding real business problems and then improving business value, although the investment in Big Data and Machine Learning (ML) projects has increased. One challenge might include a lack of understanding about relationships between business problems and data. The other challenges might consist of determining a testable factor associated with a potential business problem, preparing a relevant dataset corresponding to the business problem, analyzing the impact on the business problem to other problems, and reasoning about inter-connected problems. Information systems solving unconfirmed problems frequently tackle an erroneous problem and give incorrect predictions, leading to some dissatisfying systems, consequently not achieving business goals, even redeveloping the systems and taking many business resources. This dissertation presents a *G*oal-*O*riented and

*M*achine learning-based framework using the notion of a *P*roblem *HY*pothesis, *Gomphy*, to help validate potential business problems. We propose five main technical contributions: 1. The domain-independent Gomphy ontology and process are presented explicitly and formally for describing categories of essential concepts and relationships concerning business goals, problems, problem hypotheses, ML, and a dataset. The ontology ensures that business goals and the related business problem hypotheses are traceable to an ML dataset. 2. An entity modeling method of a problem hypothesis is elaborated to help capture business events and determine testable factors. 3. A data preparation method is described to build an ML dataset, mapping a concept of a problem hypothesis to a data feature. 4. A feature evaluation method is presented using ML and ML Explainability library to detect contribution relationships among the business hypotheses and problems. 5. A set of formalized validation rules are described for reasoning about connected problem hypothesis validation in a goal-oriented problem hypothesis model. To see the strength and weaknesses of the Gomphy framework, we have validated potential banking problems about an unpaid loan and customer churn in one retail bank as empirical studies. We feel that at least the proposed framework helps validate business events that negatively contribute to a goal, providing insights about the validated problem.

TABLE OF CONTENTS

ACKNOWLEDGMENTS v
ABSTRACT vii
LIST OF FIGURES xiii
LIST OF TABLES xv
CHAPTER 1 INTRODUCTION 1
1.1 Motivation 1
1.2 Problem 1
1.3 Solution Overview 2
1.4 Validation 3
1.5 Contribution 4
1.6 Dissertation Outline 5
CHAPTER 2 RELATED WORK 6
2.1 Requirements Engineering 6
2.1.1 Problem Analysis in Requirements Engineering 7
2.1.2 Goal Validation in Requirements Engineering 10
2.2 Data Preparation in Big Data and Data Mining 13
2.3 Feature Importance in Machine Learning 15
CHAPTER 3 GOMPHY: A MODELING FRAMEWORK FOR VALIDATING BUSINESS PROBLEM HYPOTHESES 18
3.1 The Gomphy Ontology 19
3.2 The Gomphy Process 20
3.2.1 Step 1: Explore Problem Hypotheses 21
3.2.2 Step 2: Prepare an ML dataset 21
3.2.3 Step 3: Evaluate the Impact of Problem Hypotheses Using ML 21
3.2.4 Step 4: Validate Problem Hypotheses 22
3.3 The Gomphy Methods 22
3.3.1 A Method of Determining a Testable Factor of a Problem Hypothesis 22
3.3.2 The Gomphy Mapping Method 23

3.3.3 The Gomphy Semantic Reasoning Methods . . . 25
CHAPTER 4 VALIDATING BANKING PROBLEMS BEHIND UNPAID LOAN . 27
4.1 Introduction . . . 27
4.2 The Gomphy In Action . . . 29
4.2.1 Step 1: Explore the Case Bank's Problem Hypotheses . . . 29
4.2.2 Step 2: Prepare an ML Dataset . . . 32
4.2.3 Step 3: Evaluate the Impact of Problem Hypotheses Using ML . . . 35
4.2.4 Step 4: Validate Problem Hypotheses . . . 37
4.3 Experimental Results . . . 39
4.3.1 Experiment 1 . . . 39
4.3.2 Experiment 2 . . . 40
4.3.3 Experiment 3 . . . 41
4.4 Discussion and Related Work . . . 44
4.5 Conclusion and Future Work . . . 46
CHAPTER 5 DATA PREPARATION FOR VALIDATING BANKING PROBLEMS BEHIND UNPAID LOAN . . . 47
5.1 Introduction . . . 47
5.2 Related Work . . . 49
5.3 The Dregon Approach . . . 50
5.3.1 The Dregon Ontology . . . 51
5.3.2 The Dregon Process . . . 53
5.4 The Dregon in Action . . . 53
5.4.1 Step 1: Explore Business Goals . . . 54
5.4.2 Step 2: Hypothesize Business Problems Hindering Goals . . . 54
5.4.3 Step 3: Identify Data Features for a Problem Hypothesis . . . 56
5.4.4 Step 4: Extract and Transform an ML Dataset . . . 58
5.5 Experimental Results . . . 61
5.5.1 Experiment 1 . . . 61
5.5.2 Experiment 2 . . . 62

5.5.3 Experiment 3 . . . 63
5.6 Discussion and Observation . . . 65
5.7 Conclusion and Future Work . . . 66
CHAPTER 6 VALIDATING BANKING PROBLEMS BEHIND CUSTOMER CHURN 68
6.1 Introduction . . . 68
6.2 Metis: A Goal-Oriented Problem Hypothesis Validation Method using Machine Learning . . . 69
6.2.1 Domain-Specific Banking Ontology . . . 69
6.2.2 The Metis Ontology . . . 70
6.2.3 The Metis Process . . . 72
6.3 Experiment and Results . . . 77
6.3.1 Dataset Analysis . . . 77
6.3.2 Prediction Models . . . 79
6.3.3 Explainability Model . . . 79
6.3.4 Validating Problem Hypotheses . . . 81
6.4 Related Work and Discussion . . . 81
6.5 Conclusions . . . 83
CHAPTER 7 VALIDATING POTENTIAL PHENOMENA CAUSING THE OCCURRENCE OF WEST NILE VIRUS . . . 84
7.1 Introduction . . . 84
7.2 The Metis Framework . . . 85
7.2.1 The Metis Ontology . . . 85
7.2.2 The Formal Semantics for Validating a Problem Hypothesis and Reasoning Methods . . . 88
7.2.3 The Metis Process . . . 91
7.3 The Metis in Action . . . 93
7.3.1 Step 1: Explore Problem Hypotheses of Increased Mosquitoes Infected by West Nile Virus . . . 93
7.3.2 Step 2: Prepare an ML dataset for Hypotheses Validation . . . 94
7.3.3 Step 3: Evaluate Problem Hypotheses about the Spread of West Nile Virus . . . 95

7.3.4 Step 4: Validate Problem Hypotheses about the Spread of West Nile Virus . . . 97
7.4 Related Work . . . 98
7.5 Discussion and Observation . . . 99
7.5.1 Discussion . . . 99
7.5.2 Observation . . . 100
7.5.3 Limitations . . . 100
7.6 Conclusion and Future Work . . . 100
CHAPTER 8 CONCLUSION . . . 102
8.1 Summary . . . 102
8.2 Contribution . . . 102
8.3 Future Work . . . 103
REFERENCES . . . 105
BIOGRAPHICAL SKETCH . . . 112
CURRICULUM VITAE

LIST OF FIGURES

1.1 Unpaid loan problem in a bank 4
2.1 Three areas related to validating problem hypotheses 7
2.2 An example of a Fishbone diagram 8
2.3 Fault Tree Diagram representing a Boolean algebra 8
2.4 Analysis of ineffective manual ambulance dispatch using PIG 9
2.5 An Ontology of GO-BigBPRML in IRIS 10
2.6 Validating goal models using a questionnaire-based survey 11
2.7 A goal model for the transportation simulator with Bayesian Networks 11
2.8 Goal models with simulation results for architectural decisions on scalability . . 12
2.9 CRISP-DM Process and Data Preparation 13
2.10 The partial metamodel for the Data Preparation View in GR4ML 14
2.11 Coefficients as Feature Importance 15
2.12 Feature Importance for predicting cervical cancer with a random forest 16
2.13 Explaining Individual Predictions in LIME 17
2.14 Explaining John' Loan Application in SHAP 17
3.1 The Gomphy ontology for validating a problem hypothesis 19
3.2 The Gomphy process for validating a problem hypothesis 21
4.1 Unpaid loan problem in a bank 27
4.2 Research questions corresponding to research challenges 28
4.3 The schema of the Financial database 30
4.4 Hypothesizing problems for an unpaid loan 31
4.5 Preparing an ML dataset for validating a problem hypothesis 33
4.6 Mapping attributes between a problem hypothesis entity and a database entity . 34
4.7 Feature importance for one unpaid loaner 36
4.8 Validating a problem hypothesis using feature importance 38
4.9 Top important features in experiment 1 and 2 40
4.10 Deposit and withdrawal classification in Transaction 41
4.11 Important features and contribution type in experiment 3 42

4.12 ML models' performance comparison in experiment 3 43
5.1 Unpaid loan in Case bank (empirical study context) 49
5.2 The data preparation ontology at a high-level 51
5.3 The detailed data preparation ontology for a validating problem hypothesis . . . 52
5.4 The data preparation process and the Financial database schema 53
5.5 Applying the Dregon process for an unpaid loan 55
5.6 Mapping types from a problem hypothesis entity into a database entity 58
5.7 Mapping a problem hypothesis entity into a database entity 59
5.8 Merging partial datasets into an ML dataset 60
5.9 Top important features in experiment 1 62
5.10 Top important features in experiment 2 63
5.11 Deposit and withdrawal classification in Transaction 63
5.12 ML performance using the prepared dataset in experiment 3 64
6.1 Banking domain-specific ontology diagram 70
6.2 Metis domain-independent ontology diagram 71
6.3 The *Metis* process for validating goal-oriented hypotheses of business problems . 72
6.4 Step 1: Model business goals and problems 74
6.5 Step 2, 3, 4 of the *Metis* process 76
6.6 Analysis of customer dissatisfaction (score less than 5 in a 0 to 10 scale) for the features distance from residence, immediate attention, and pleasant ambiance . 78
6.7 Features importance for one churner's responses 80
6.8 Feature importance for all churners' responses 81
6.9 Validated customer churn problems 82
7.1 The types of essential modeling concepts for validating a problem hypothesis . . 86
7.2 The Metis process for validating a problem hypothesis 92
7.3 A portion of a West Nile Virus problem hypothesis model 94
7.4 The XGBoost ROC Curve for validating a WNV problem hypothesis 96
7.5 The XGBoost ROC Curve for validating a WNV problem hypothesis 97
7.6 Validating the problem hypotheses of West Nile Virus cases 98

LIST OF TABLES

2.1 Comparison of RE work for validating problem hypotheses 13
4.1 Experiments comparison for validating problem hypotheses 43
5.1 Experiments comparison of data preparation 65
7.1 A problem hypothesis in a Baclus-Naur form 87
7.2 A problem hypothesis template with an infected mosquitoes 88
8.1 Challenges for validating business problems and the Gomphy solutions 104

CHAPTER 1

INTRODUCTION

1.1 Motivation

The assertion that *"A problem unstated is a problem unsolved"* expresses the importance of eliciting business needs and problems [1]. Understanding and validating a business problem likely to hinder a business goal is often more critical than developing solutions to the problem, especially during requirements engineering. Understanding and validating a business problem helps define system boundaries in the early phase of requirements engineering [2]. For example, validating the impact of a client's low account, high transactions, or high loan payment per month for a client's unpaid loan decreasing the loan revenue of one bank would be critical as an information system can be designed for the bank to take some actions to mitigate the loan default. If the correct problems are validated first, a business can save precious time and cost to deal with erroneous problems [3, 4, 5, 6].

Business organizations invest in Big Data and Machine Learning (ML) projects to obtain business value [7, 8, 9, 10]. Data analytics and ML technologies benefit from a continuous improvement cycle where large amounts of data are constantly created. Although businesses strive to solve actual business problems and then gain business value, many of these projects are likely to fail [11, 12]. A study may have suggested a possible reason: the lack of understanding of how to use data analytics to improve business value [13, 14]. This finding clearly shows that stakeholders have a vague perception of the end-to-end relationship between essential business goals and the emerging Big Data and ML technologies [15, 16, 17].

1.2 Problem

Many business organizations face difficulties confirming whether some potential problems hidden in Big Data are negatively impacting a business goal or not [18, 19]. In other words,

they are struggling to discover real business problems and improve business value. Specifically, some challenges to validate a business problem might include

- a lack of understanding about relationships among business problems, business goals, a dataset, and ML,
- determining a testable factor associated with a potential business problem,
- preparing a relevant dataset corresponding to the business problems,
- analyzing the impact of the business problem to other problems, and
- reasoning about inter-connected potential problems and goals.

Information systems solving unconfirmed problems or using unimportant factors and irrelevant data frequently tackle an erroneous problem and give incorrect predictions, leading to some dissatisfying systems, consequently not achieving business goals, even to redevelop the systems, taking many business resources [20, 21].

1.3 Solution Overview

This dissertation presents a *G*oal-*O*riented and *M*achine learning-based framework, using the notion of *P*roblem *HY*pothesis, *Gomphy* to help validate potential business problems [22, 23, 24, 25]. Gomphy helps explore alternatives of problem hypotheses or potential problems against goals, evaluate the alternatives with trade-off analysis, and select the best one in a goal-oriented manner. Gomphy also utilizes ML and ML Explainability library to get feature importance value which helps to get insights of hidden relationships or patterns among problem hypotheses.

Gomphy consists of a domain independent ontology, a process, an entity modeling method, a data preparation method, semantic reasoning methods, and an assistant tool.

The domain-independent Gomphy ontology consists of categories of essential modeling concepts, relationships among modeling concepts, and constraints on the concepts and relationships for validating problem hypotheses and preparing a relevant dataset while preventing omissions and commissions in modeling essential concepts.

The Gomphy process is intended to help validate a problem hypothesis, providing traceability among a goal, a problem, a dataset, and ML. The process consists of four steps but should be understood as iterative, interleaving, and incremental in ML projects.

Gomphy provides several methods to support validating a problem hypothesis. The entity modeling method of a problem hypothesis helps to determine a testable factor of a problem hypothesis. The data preparation method helps build an ML dataset, mapping a testable factor to a data feature in the database. The semantic reasoning methods formally describe decomposition, selection, and other procedures necessary for validating problem hypotheses.

The evolving assistant tool helps model a problem hypothesis in the context of goals and reason relationships among problem hypotheses and goals.

1.4 Validation

To see the strengths and weaknesses of the Gomphy framework, we apply the proposed Gomphy framework for validating hypothesizedbanking events behind an unpaid loan and a customer churn problem in one retail bank as empirical studies. Fig. 1.1 shows a high-level context diagram for the unpaid loan problem. The PKDD'99 [1] Financial database [26] was used to represent data that the bank may have managed. We feel that at least the proposed Gomphy framework helps validate business events that negatively contribute to a goal, providing insights about the validated problem.

[1]European Conferences on Principles and Practice of Knowledge Discovery in Databases, Third European Conference, 1999

1.5 Contribution

Five technical contributions are made in this dissertation: 1. The domain-independent Gomphy ontology and systematic Gomphy process are presented explicitly and formally for describing categories of essential concepts and relationships of business goals, problems, problem hypotheses, ML, and a dataset. It ensures that business goals and the related business problem hypotheses are traceable to an ML dataset. 2. An entity modeling method of a problem hypothesis is presented to help capture business events and determine testable factors. 3. A data preparation method is described to build an ML dataset, mapping a concept of a problem hypothesis to a domain data feature and extracting a dataset from a source dataset. 4. A feature evaluation method is presented to detect contribution relationships among the business events and a problem, using ML and ML Explainability techniques. 5. A set of formalized validation rules are described for reasoning about connected problem hypothesis validation in a goal-oriented problem hypothesis model.

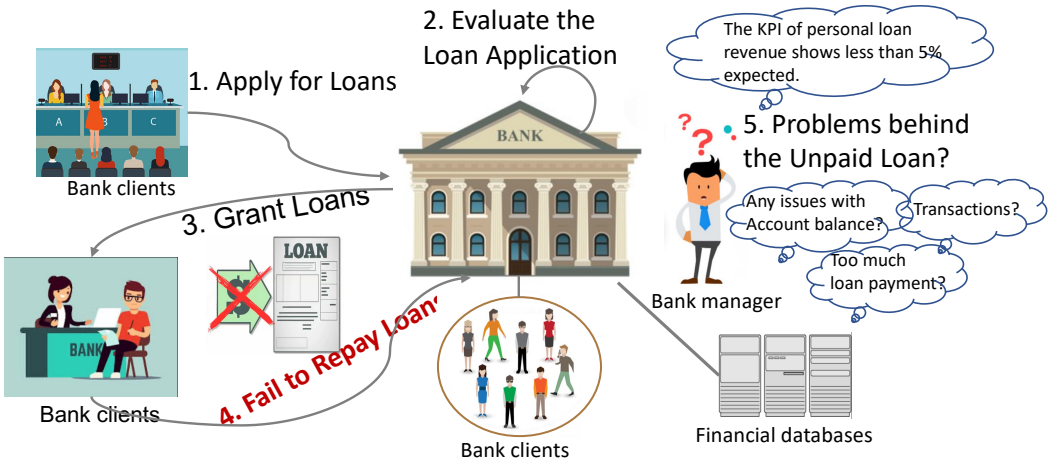


Figure 1.1. Unpaid loan problem in a bank

1.6 Dissertation Outline

The rest of this dissertation is structured as follows. Chapter 2 discusses related work, and Chapter 3 presents the *Gomphy* approach. Next, Chapters 4, 5, 7, and 6 illustrate the Gomphy process in detail with an unpaid loan, a customer churn, and the occurrence of the West Nile Virus problem. Finally, Chapter 8 summarizes the dissertation and future work.

CHAPTER 2

RELATED WORK

The key distinctive of the Gomphy framework is to use a problem hypothesis for validating potential business problems in an iterative, incremental, and interleaving manner using a goal-oriented and Machine Learning-based approach. Gomphy explores alternative causes of a potential business problem against a business goal, adopting a goal-oriented approach. Next, a relevant ML dataset is prepared, and then ML techniques are used to get insights into relationships among business events encoded in the dataset towards business problems and goals. Finally, a problem hypothesis is then validated using Gomphy semantic rules. Gomphy connects categories of essential concepts concerning goals, problems, ML, and an ML dataset during the process. We believe our work is one of the first to propose a framework for validating a business problem hypothesis in an end-to-end, explicit and formal manner that shows traceability from business goals to problems, data, and ML.

Our work lies in the intersection of three evolving areas as shown in Fig. 2.1: goal orientation, problem analysis, and goal validation in requirements engineering (RE), data preparation, and data extraction in Big Data and database, and utilization of classification, feature importance, feature selection, and feature reduction in Machine Learning.

2.1 Requirements Engineering

In the area of requirements engineering, goal-orientation, problem analysis, specifically root-cause analysis, problem validation, and goal validation methods are related [27, 28, 29, 30]. Here, goal-orientation means exploring alternatives, evaluating alternatives with trade-off analysis, and selecting the best one [31, 32]. The goal-oriented approach has been used in other application domains, such as recommendation and risk analysis [33, 34].

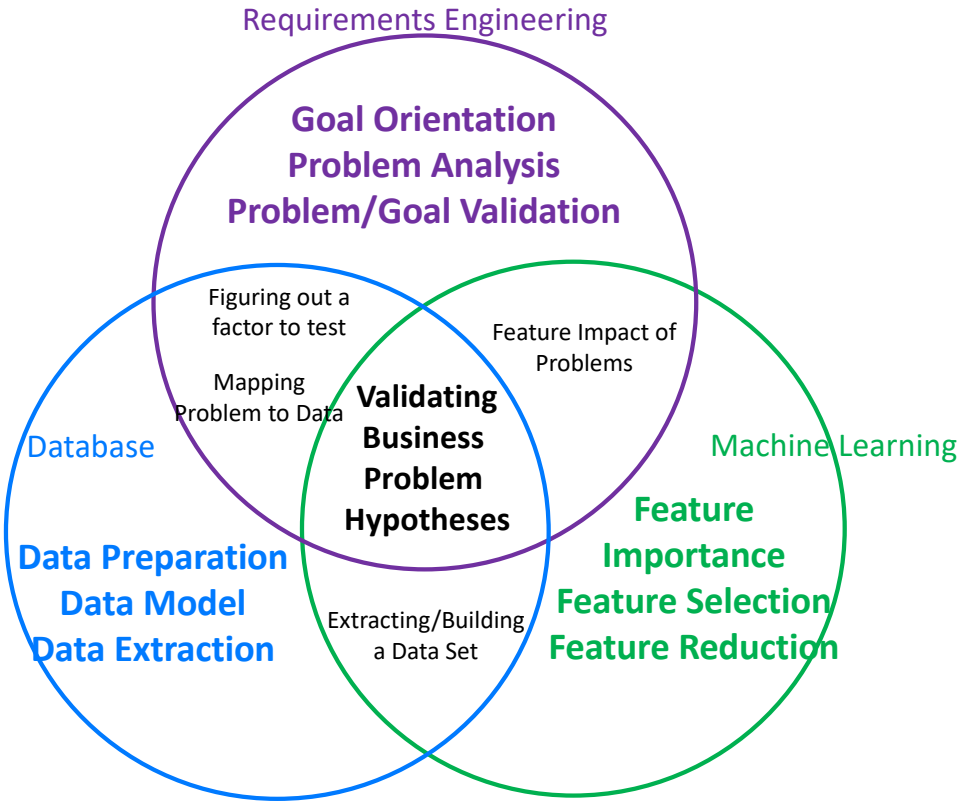


Figure 2.1. Three areas related to validating problem hypotheses

2.1.1 Problem Analysis in Requirements Engineering

In problem analysis, a Fishbone diagram has been used to identify possible causes for a problem or an effect [35]. This technique helps enumerate potential reasons for a problem, usually utilized in a brainstorming session. Fig. 2.2 shows an example of a Fishbone diagram. However, the lack of a clear relationship between a cause and an effect, e.g., logical connectives, such as AND, OR, and other kinds of relationships, makes problem validation difficult. Suppose a Fishbone diagram needs to calculate the degree of relationships among identified root causes. In that case, it should provide some methods for determining a testable factor of a root cause and connecting the identified root causes to a business database.

Fault Tree Analysis (FTA) provides deductive procedures and a logic diagram to help determine failures or errors of software, hardware, and people with a top-down approach [36].

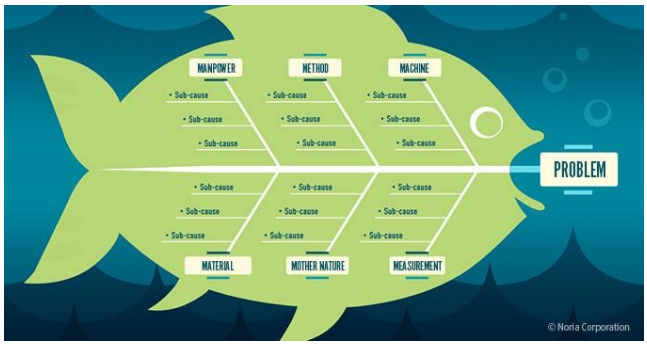


Figure 2.2. An example of a Fishbone diagram

FTA provides Boolean logic operators, which help create a series of True or False statements. Fig. 2.3 shows a Fault Tree diagram representing a Boolean algebra $D = A \cdot (B + C)$. When linked in a chain, these statements form a logic diagram of failure. However, FTA does not provide relationship direction and degrees, such as positive, negative, full, and partial, making it challenging to validate business problems having uncertain relationships.

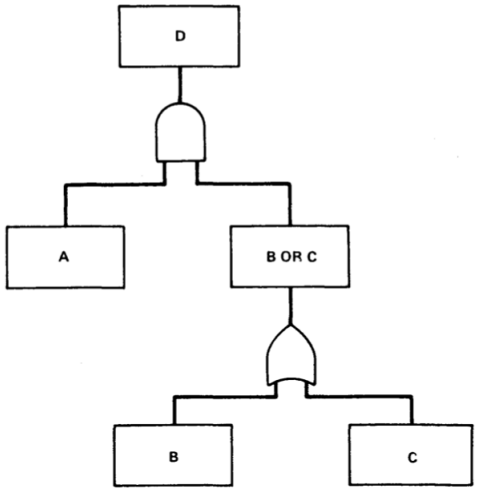


Figure 2.3. Fault Tree Diagram representing a Boolean algebra

(Soft-)Problem Inter-dependency Graph (PIG) uses a (Soft−)problem concept to represent a stakeholder problem against stakeholder goals, where a problem is refined into sub-problems [37], which helps analyze a problem in the context of goals. As shown in Fig. 2.4,

PIG provides contribution relationships, such as Satisficing conjunction/disjunction, partial/full, and positive/negative Contributions. It also includes an evaluation mechanism. However, PIG lacks the means to determine a testable factor of sub-problems and connect the testable factor to data features to test. While the Fishbone diagram, FTA, and PIG provide a sound high-level model, they lack validation mechanisms for confirming the causes behind a business problem.

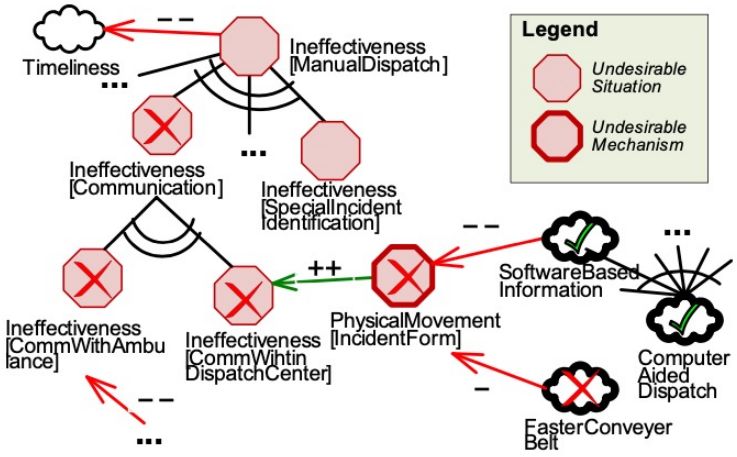


Figure 2.4. Analysis of ineffective manual ambulance dispatch using PIG

IRIS is a modeling framework for reengineering business processes that use big data analytics and a goal-oriented approach [38, 39]. IRIS provides a modeling ontology, as illustrated in Fig. 2.5. IRIS also provides a guiding process and an assistant tool for effective business processes reengineering. IRIS is similar to Gomphy in that it hypothesizes problems in business process goals and validates the problems. However, IRIS implicitly hypothesizes a problem and lacks a mechanism to prepare a dataset to test. Whereas IRIS validates potential problems using Big Analytics Queries, Gomphy explicitly models a problem using a concept of a problem hypothesis and validates problem hypotheses using ML techniques.

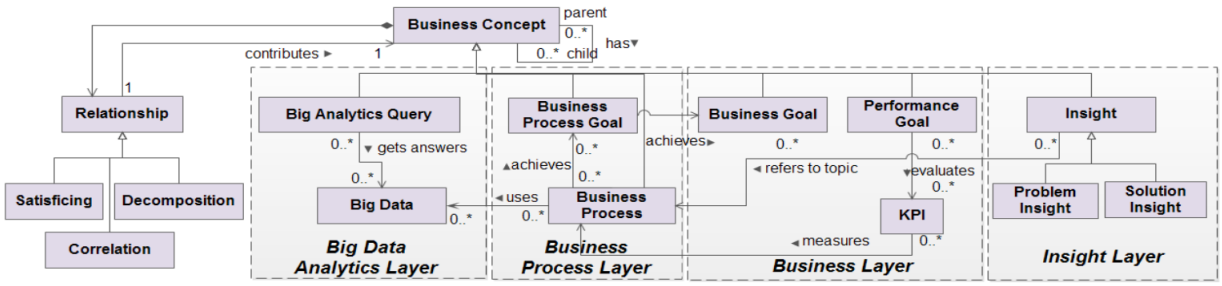


Figure 2.5. An Ontology of GO-BigBPRML in IRIS

2.1.2 Goal Validation in Requirements Engineering

In goal validation, a survey method to validate the GRL (Goal-oriented Requirements Language) goal models was proposed in [40] using questionnaires and the statistical hypothesis testing to validate different goal model elements (e.g., actors, goals, resources) and their relationships (e.g., depends, make, hurt). It utilizes questionnaires constructed from goals models and then gathers survey data as an empirical approach to validate them. Fig. 2.6 shows an essential process for validating goal models using a questionnaire-based survey. However, this approach is not easy to apply as the questionnaire needs to be prepared and data are gathered, which is time-consuming. In addition, some collected data may be incomplete enough to validate. This approach also was not used to validate a problem model.

A goal model, an example of requirements modeling language, is created by requirements engineers with assumptions that goals relate to each other and should be satisfied at some specific time. [41] addresses the automated validation of goal models in requirements monitoring. It uses probabilistic techniques (e.g., Bayesian Network model) to confirm a goal model's assumptions with a quantitative degree, such as fully valid, partially valid, or fully wrong assumption, as shown in Fig. 2.7. A dataset from a traffic simulation is used to map a goal model to a Requirements Bayesian Network. This work is similar to ours in terms of relationship directions and degrees in goal and problems contributions. However, it does not show how to identify data features from a goal model, and some relationships among

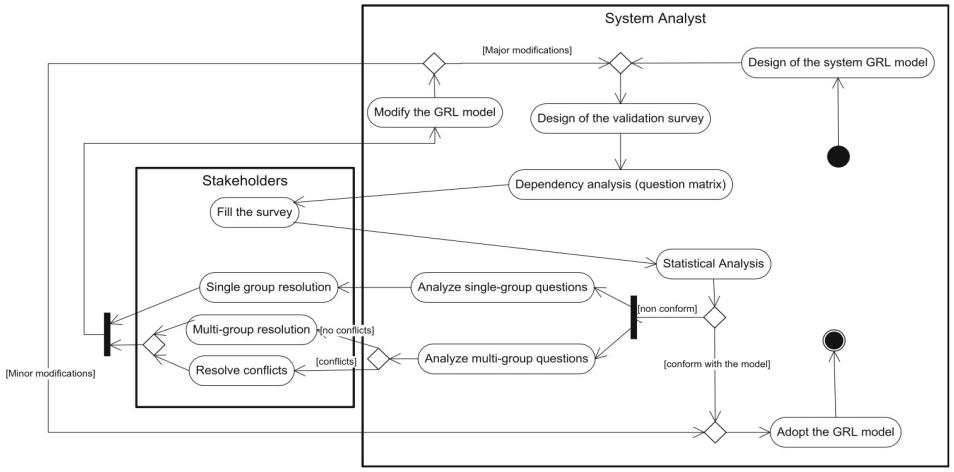


Figure 2.6. Validating goal models using a questionnaire-based survey

goals were not fully simulated. The Bayesian network often requires datasets to be complete, often difficult to use for real-world applications. This approach was not applied to validate a problem hypothesis model.

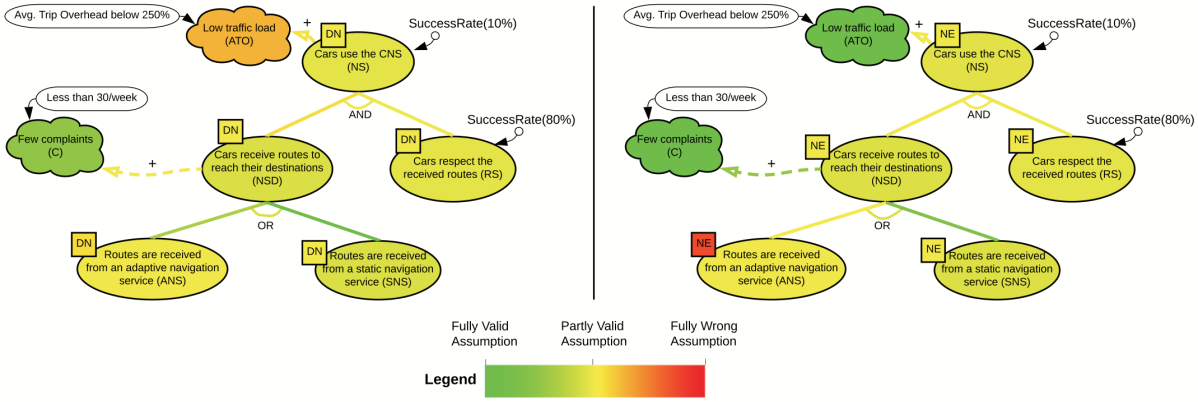


Figure 2.7. A goal model for the transportation simulator with Bayesian Networks

Some non-functional requirements, such as scalability, cost, and reliability, are essential goals in architectural design decisions before time-consuming and costly tasks are performed. [42, 43, 44, 45] combines goal-orientation and simulation to validate and re-validate architectural decisions on scalability and performance, as illustrated in Fig. 2.8. The goal model for an architectural decision is used to develop simulation models. This work is similar to ours

regarding the use of goal orientation for exploring architectural options. Whereas this work uses simulation techniques to reconfirm the goal model, and the outcome depends on the parameters, our work uses ML-based techniques, such as classification and feature importance, to validate problem hypotheses.

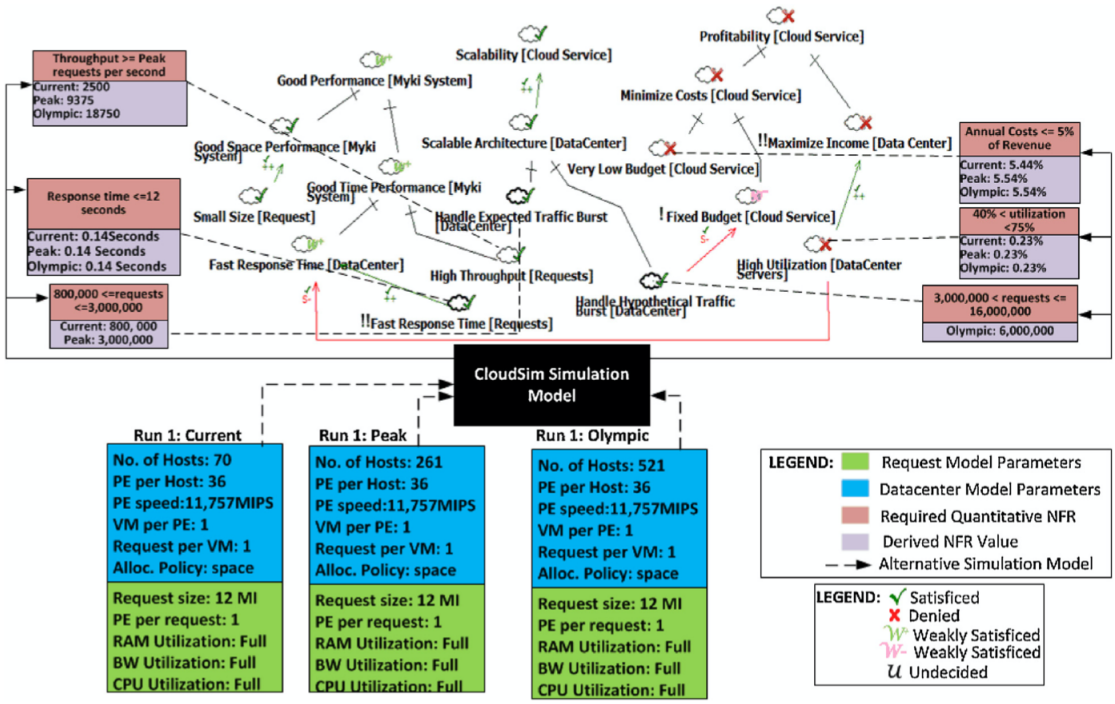


Figure 2.8. Goal models with simulation results for architectural decisions on scalability

Table 2.1 compares the surveyed techniques in Requirements Engineering concerning a problem analysis and goal (problem) validation. We compared Gomphy with the related work regarding ontology, process, problem validation methods while focusing on whether each piece helps explore a business problem using logical connectives, relationship directions, and degrees. We also compared whether each research work provides data preparation methods and utilization of ML. Lastly, we compared the end-to-end traceability and easy-to-use quality of each work.

Table 2.1. Comparison of RE work for validating problem hypotheses

Relative Strength: - < S+ < + < ++

Related work	Ontology for Problem Validation	Problem Validation Process	Problem Validation Method	Relationship Logical Connectives	Relationship Direction and Degree (positive/negative partial/full),	Relationship b/w Problem and Goal	Data Preparation for Problem Validation	Problem Validation using ML / statiscal analysis	End-to-End Traceability	Easy to Use
Fishbone Diagram	S+	S+	S+	-	-	-	-	-	S+	++
Fault Tree Diagram	S+	S+	S+	++	-	-	-	S+	S+	++
Problem Interdependency Diagram	S+	S+	S+	++	++	++	-	-	+	+
A Modeling Framework for BPR	+	S+	+	++	++	++	S+	S+	+	+
Validating GRL Goal Models	-	-	-	++	++	+	-	+	-	-
Validating Goal Models via Bayesian Networks	-	-	-	++	-	-	-	++	-	-
Confirming and Reconfirming Architectural Decisions	-	-	-	++	++	++	-	-	+	+
Gomphy	+	+	++	++	++	++	+	++	+	+

2.2 Data Preparation in Big Data and Data Mining

Some conceptual data preparation frameworks guide data preparation tasks for ML processing in the database or Big Data domain.

CRISP-DM (CRoss Industry Standard Process for Data Mining) model is a de facto standard for the data mining process [46]. It has a comprehensive process model for carrying out data mining projects, as shown in Fig. 2.9. The data preparation process includes *Select Data*, *Clean Data*, *Construct Data*, and *Integrate Data* steps. However, it lacks a systematic mapping method from business problems to data features [47, 5]. It also lacks support for building an ML dataset for problem hypothesis validation.

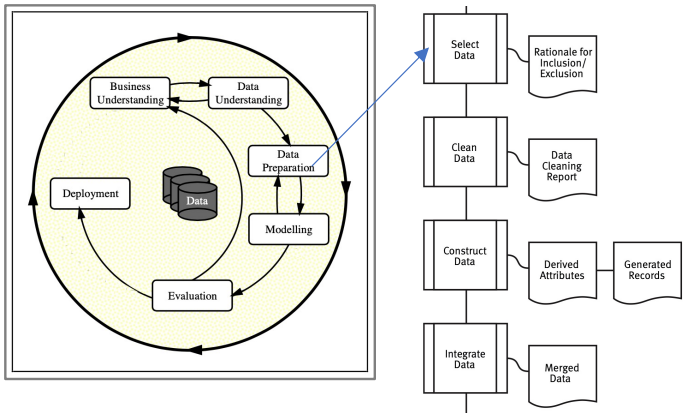


Figure 2.9. CRISP-DM Process and Data Preparation

GR4ML (Goal-Oriented RE for ML) is a modeling framework intending to design data analytics systems based on requirements analysis [48]. The GR4ML framework includes three modeling views: business view, analytics design view, and data preparation view. These views are connected from business strategies to analytics mechanisms and data preparation tasks. It supports data preparation view concerning mechanisms, algorithms, and preparation tasks while linking business view and analytics design view, as shown in Fig. 2.10. However, it is not easy to get traceability from business views to data preparation views. It also does not deal with concepts of business problems and systematic mapping methods from business problems to data features for problem validation.

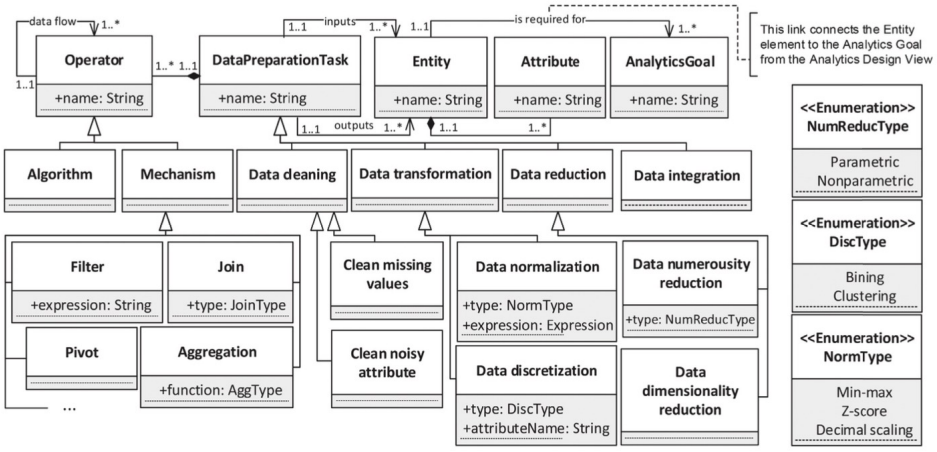


Figure 2.10. The partial metamodel for the Data Preparation View in GR4ML

An ML dataset is prepared in the structure or format that fits each machine learning task. As business databases may include noise, missing values, similar features, or redundant data, some low-quality data should be preprocessed or reduced for good prediction. There can be two kinds of preparation techniques, data preprocessing and data reduction [49].

The data preprocessing techniques may include data cleaning, transformation, integration, normalization, missing data imputation, and noise identification [50, 51]. In data reduction, the amount of data is downsized, while the reduced data still includes the essential

structure of the source data. The data reduction methods cover feature selection, instance selection, discretization, feature extraction, and instance generation [52, 53].

Although the data preprocessing and data reduction techniques in ML are valuable in partly preparing data, these techniques often lack high-level concepts, such as goals and problems and their relationships, such as positive, negative contributions. These techniques are often used to identify low-level problems informally and do not provide traceability to higher-level problems [54]. Our approach prepares an ML dataset to support business problem validation, adopting essential concepts of the goal-oriented and ML-based approach in a complementary manner.

2.3 Feature Importance in Machine Learning

In the area of Machine Learning, some ML algorithms, such as Linear Regression and Decision Trees, provide feature importance value concerning their predictions. When ML models predict a numerical value in the regression model or a target label in the classification, relative feature importance scores are calculated for the features in the dataset [55, 5].

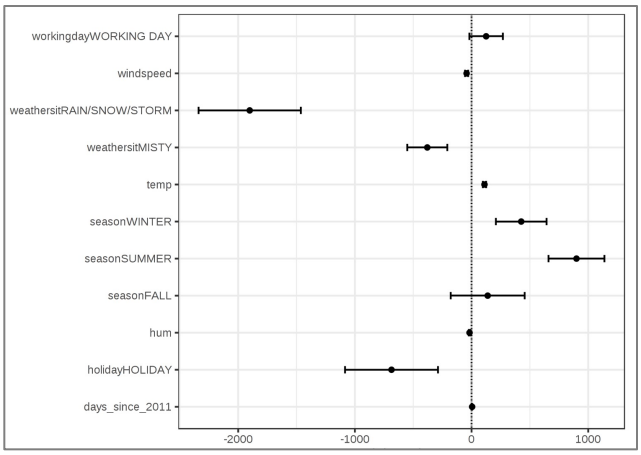


Figure 2.11. Coefficients as Feature Importance

Coefficients can be used as feature importance in the Linear regression model [56]. It is easy to understand the feature importance (i.e., coefficients) in Fig. 2.11. However, co-

efficients may oversimplify the complex reality of business interactions, and it is a little bit simple to use the coefficient as a feature value in validating complex business problems.

Next, permutation feature importance can be used to get the relative importance of each feature in the ML dataset towards classification. Permutation feature importance examines the model's prediction error development after permuting feature values and provides highly condensed, global views or patterns into the model's behavior. However, it is not easy to know whether feature importance positively/negatively contributes to a business problem (a target label). Fig. 2.12 shows the importance of each of the features for predicting cervical cancer with a random forest.

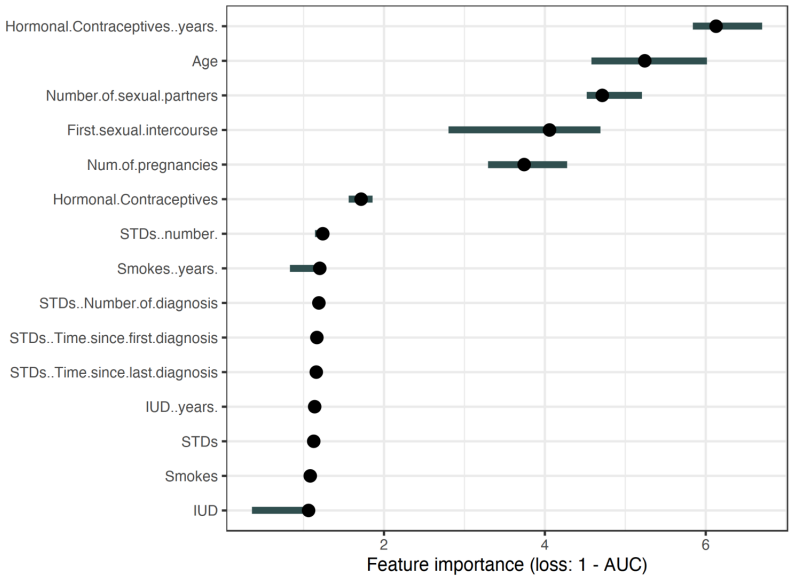


Figure 2.12. Feature Importance for predicting cervical cancer with a random forest

Explainable machine learning models also provide feature importance. LIME(Local Interpretable Model-agnostic Explanations) explains individual predictions [57]. However, there is some instability of the explanations, which may hurt validating business problems, as shown in Fig. 2.13.

SHAP (SHapley Additive exPlanations) is a framework for interpreting predictions, as illustrated in Fig. 2.14. SHAP assigns each feature an importance value for a particular

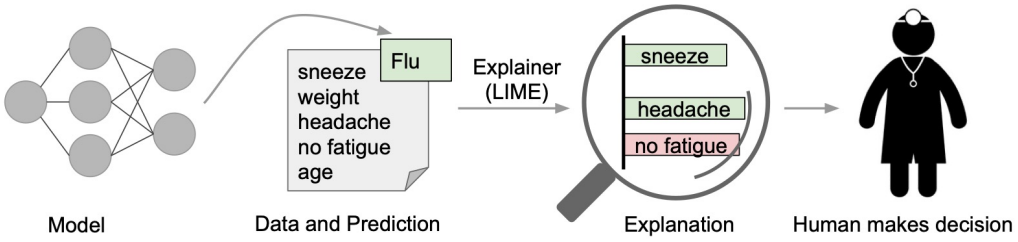


Figure 2.13. Explaining Individual Predictions in LIME

prediction and outputs intuitive feature value that helps to understand and validate business problems. However, SHAP may take a long computational time [58].

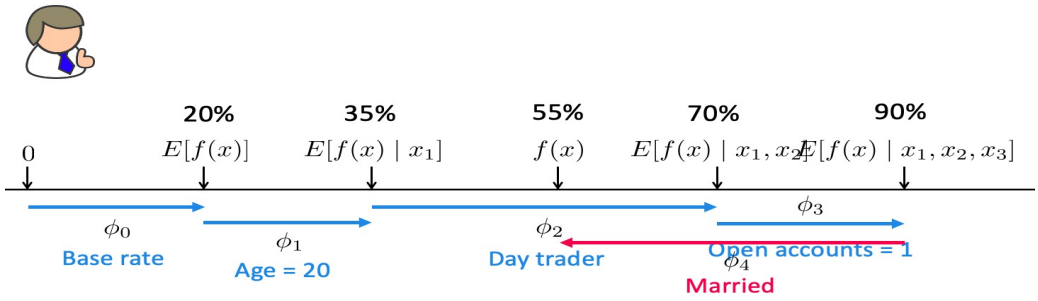


Figure 2.14. Explaining John' Loan Application in SHAP

Feature importance in ML algorithms could be utilized to get insights about features in an ML dataset. However, there are some issues identifying factors to test corresponding to the business events and preparing the dataset, such as mapping business events to data features. The data features are often selected on informal identification of a low-level problem, making it challenging to understand transparent relationships between the low-level problem and high-level business problems in the context of goals [54]. The Gomphy adopts a goal-oriented and ML-based approach, bridging the gaps in a complimentary manner.

CHAPTER 3

GOMPHY: A MODELING FRAMEWORK FOR VALIDATING BUSINESS PROBLEM HYPOTHESES

The Gomphy [1] framework, aiming to help validate business problem hypotheses, consists of a domain-independent ontology, the Gomphy process adopting a goal-oriented and ML-based approach, a method of determining a testable factor of a problem hypothesis, a data preparation method, semantic reasoning methods, and an assistant tool.

The domain-independent Gomphy ontology consists of categories of essential modeling concepts, relationships between modeling concepts, and constraints on the concepts and relationships for validating problem hypotheses and preparing a relevant dataset while preventing omissions and commissions in modeling essential concepts. The Gomphy process is intended to help guide the validation of a problem hypothesis, providing traceability among a goal, a problem, a dataset, and ML. The process consists of four steps but should be understood as iterative, interleaving, and incremental in ML projects.

Gomphy provides several methods to support validating a problem hypothesis. The entity modeling method of a problem hypothesis helps to determine testable factors of a problem hypothesis. The data preparation method helps build an ML dataset, mapping a testable factor to a data feature in the database. The Gomphy reasoning methods formally describe decomposition, evaluation, selection, and other procedures necessary to validate problem hypotheses. The evolving assistant tool helps model a problem hypothesis in the context of goals and reason relationships among problem hypotheses and goals.

---

[1] This chapter contains material previously published as: ©2021 Springer. Reprinted, with permission, from Ahn, R., Supakkul, S., Zhao, L., Kolluri, K., Hill, T., & Chung, L. (2021, December). Validating Business Problem Hypotheses: A Goal-Oriented and Machine Learning-Based Approach. In International Conference on Big Data (In press). Springer, Cham.

3.1 The Gomphy Ontology

The Gomphy ontology consists of categories of essential modeling concepts, relationships among modeling concepts, and constraints on the concepts and relationships for a validating problem hypothesis, as shown in Fig. 3.1, where boxes and arrows represent the concepts and relationships.

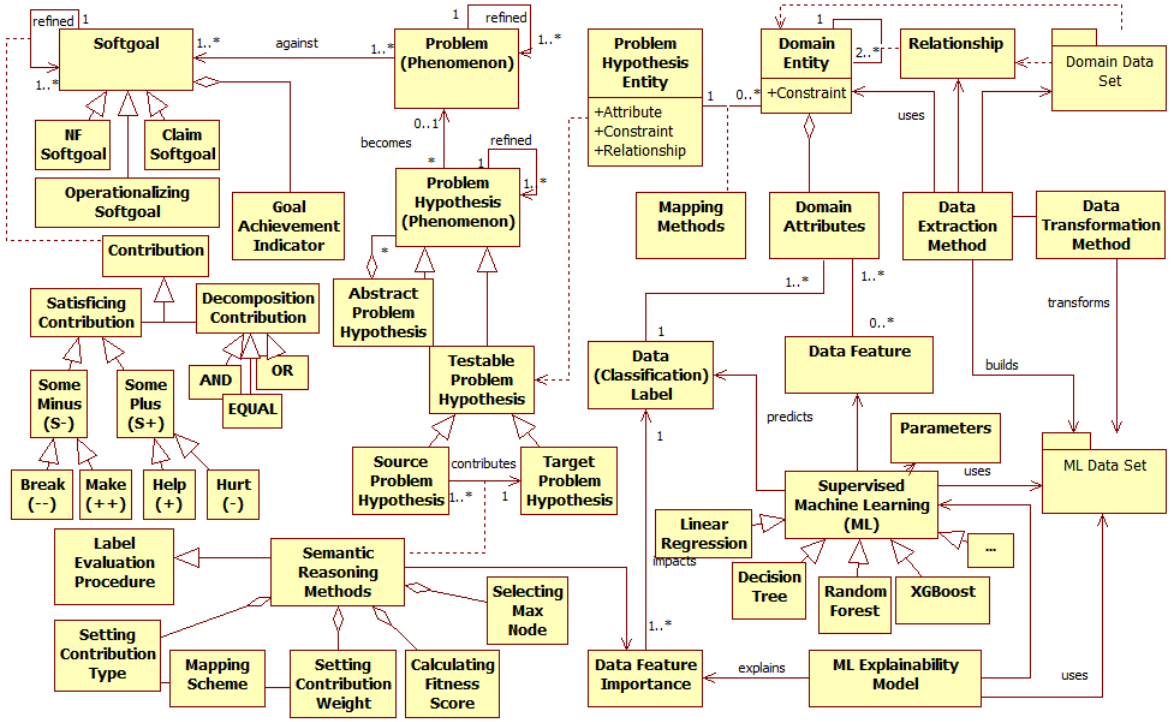


Figure 3.1. The Gomphy ontology for validating a problem hypothesis

Some essential concepts of Gomphy ontology are introduced. A *(Soft-)Goal* is defined as a goal that may not have a clear-cut criterion and can be specialized into a Non-Functional (NF) softgoal, an Operationalizing softgoal, and a Claim softgoal. While a *(Soft-)Problem* is a phenomenon against a softgoal, a *Problem Hypothesis* is a hypothesis that we believe a phenomenon is against a softgoal.

There are two kinds of problem hypotheses, an *Abstract Problem Hypothesis* and a *Testable Problem Hypothesis*. An abstract problem hypothesis is conceptual and not con-

crete enough to test, whereas a testable problem hypothesis is measurable and testable. A Testable Problem Hypothesis may be further refined, forming a testable *Source Problem Hypothesis* and a *Target Problem Hypothesis*. A *Problem Hypothesis Entity*, consisting of *Attributes*, *Constraints*, and *Relationships*, is an entity representing a Testable Problem Hypothesis and may be mapped to a relevant *Database Entity* having *Attributes*, *Constraints*, and *Relationships* in a source data model. The identified database entities are used to extract data from source data using *Data Extraction Method*. The selected attributes of a database entity are used to build an ML dataset consisting of *Data Features* and a *Classification Label* depending on either a source or target problem hypothesis.

The Contribution relationships among Goals, Problems, and Problem Hypotheses are categorized into Decomposition types, such as *AND, OR, EQUAL*, or Satisficing types, such as *Make, Help, Hurt, Break, Some-Plus, Some-Minus, Unknown* adopted from the NFR Framework [59]. The relationships between Problem Hypotheses and Problems are either *Validated* or *Invalidated*.

One crucial constraint about a problem hypothesis includes time-order among a target and a source problem hypothesis, where a source problem hypothesis must have occurred before the target problem hypothesis. Other constraints are a positive contribution from a source problem hypothesis to a target problem hypothesis, and the contribution should be reasonably sensible [60].

3.2 The Gomphy Process

The Gomphy process, shown in Fig. 3.2, is intended to help guide the validation of a problem hypothesis, providing traceability among a goal, a problem, a dataset, and ML. The process consists of four steps but should be understood as iterative, interleaving, and incremental in ML projects. The sub-steps of each step are described in detail in the following chapters with empirical studies.

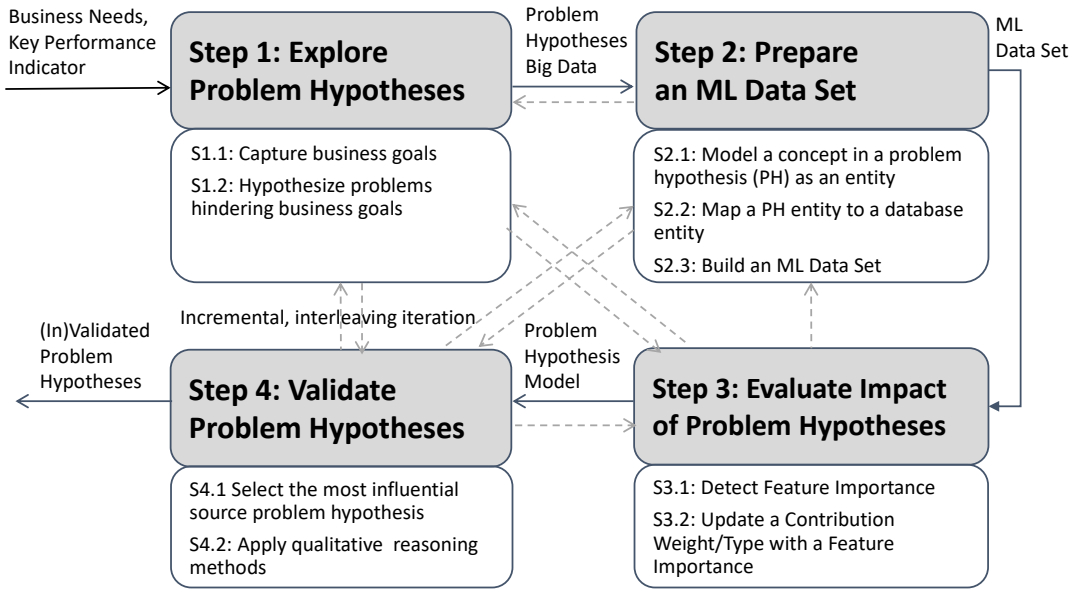


Figure 3.2. The Gomphy process for validating a problem hypothesis

3.2.1 Step 1: Explore Problem Hypotheses

Requirements engineers begin Step 1, understanding the application domain, modeling, and refining stakeholders' goals in business. Potential problems that could hinder the goals are then hypothesized and refined.

3.2.2 Step 2: Prepare an ML dataset

In this step, we model a concept in the testable problem hypothesis as a problem hypothesis entity, map the attributes of a problem hypothesis entity to the database's data attributes, and construct an ML dataset based on the identified data attributes.

3.2.3 Step 3: Evaluate the Impact of Problem Hypotheses Using ML

The impact of problem hypotheses towards a business problem, hidden in the relationships among data features and a target label, is uncovered using Supervised ML models and ML Explainability model, decoding hidden feature patterns in the dataset.

3.2.4 Step 4: Validate Problem Hypotheses

This step selects the most critical problem hypothesis as a validated one among alternative hypotheses and evaluates the impact of the validated problem on other high-level problems.

3.3 The Gomphy Methods

Gomphy provides several methods to support validating a problem hypothesis. The entity modeling method of a problem hypothesis helps to determine testable factors of a problem hypothesis. The data preparation method helps build an ML dataset, mapping a testable factor to a data feature in the database. The semantic reasoning methods formally describe decomposition, selection, and other procedures necessary for validating problem hypotheses.

3.3.1 A Method of Determining a Testable Factor of a Problem Hypothesis

A potential business event against a business goal is hypothesized as an abstract problem hypothesis. The identified abstract problem hypothesis is further decomposed into a testable problem hypothesis that contains a testable factor. A testable factor in a testable problem hypothesis is usually a categorical or numeric type.

The concept in an elicited testable problem hypothesis is modeled as a problem hypothesis entity using the entity-relationship model [61] [62]. A problem hypothesis entity consists of attributes, constraints, and relationships. An *attribute* is a property of an entity having measurable value. A *constraint* is a condition restricting the value or state of a problem hypothesis. A *relationship* shows other entities associated with this entity. Here, an attribute of categorical or numeric type in a problem hypothesis entity may be determined as a testable factor using Algorithm 1. While the categorical type includes nominal and ordinal types, the numeric type includes interval and ration types.

Algorithm 1 Figuring out a Testable Factor of a Testable Problem Hypothesis

```
input: testableProblemHypothesis      ▷ A statement for a testable problem hypothesis
output: testableFactor
procedure DeterminingTestableFactor(testableProblemHypothesis)
    TestableProblemHypothesisEntity tphe =
        EntityModeling(testableProblemHypothesis);
    String testableFactor = null;
  if (tphe.getAttributeType(), 'nominal') then
      testableFactor = tphe.getAttributeName();

  else if (tphe.getAttributeType(), 'oridnal') then
      testableFactor = tphe.getAttributeName();

  else if (tphe.getAttributeType(), 'interval') then
      testableFactor = tphe.getAttributeName();

  else if (tphe.getAttributeType(), 'ratio') then
      testableFactor = tphe.getAttributeName();

  else
      testableFactor = 'unknown';

  return testableFactor;
```

3.3.2 The Gomphy Mapping Method

A mapping rule or method is needed to determine a data feature corresponding to a testable factor of a testable problem hypothesis. The mapping from the attribute of the problem hypothesis entity attributes of the database entity is precisely defined as follows:

$$Mapping : Attributes \rightarrow 2^{Attributes} \tag{3.1}$$

The attribute of the problem hypothesis entity may be mapped to attributes of the database entity, considering the constraints and relationships of the problem hypothesis entity using Algorithm 2.

However, due to ambiguous or abbreviated attributes names of an entity and constraints to enforce in the database, the one-to-one or one-to-many mappings may not be efficiently

Algorithm 2 Mapping a Testable Problem Hypothesis Entity to a Database Entity

```
input: testableProblemHypothesisEntity, databaseEntities
output: mappedEntityList
procedure MappingEntities(testableProblemHypothesisEntity, databaseEntities)
    phName = testableProblemHypothesisEntity.getName();
    phAttName = testableProblemHypothesisEntity.getAttributeName();
    phRelName = testableProblemHypothesisEntity.getRelationshipName();
    phConName = testableProblemHypothesisEntity.getConstraintName();

    DatabaseEntityList mappedEntityList;

  for each dbEntity ∈ databaseEntities do
    DatabaseEntity entity = dbEntity;
    for each dbAtt ∈ dbEntity do
      if dbAtt.getAttributeName() == phAttName then
        entity.addAttribute(dbAtt);
      else if dbAtt.getAttributeName().contains(phAttName) then
        entity.addAttribute(dbAtt);
      else
        entity.addAtribute('not mapped');

    for each dbConstraint ∈ dbEntity do
      if dbConstraint.getConstraintName() == phConName then
        entity.addConstraint(dbConstraint);
      else if dbConstraint.getConstraintName().contains(phConName) then
        entity.addConstraint(dbConstraint);
      else
        entity.addConstraint('not mapped');

    for each dbRelationship ∈ dbEntity do
      if dbRelationship.getRelationshipName() == phRelName then
        entity.addRelationship(dbRelationship);
      else if dbRelationship.getRelationshipName().contains(phRelName) then
        entity.addRelationship(dbRelationship);
      else
        entity.addRelationship('not mapped');

    mappedEntityList.addEntity(entity);

  return mappedEntityList;
```

performed. We also need to consider the relationships of a problem hypothesis entity to determine an accurate data feature corresponding to a testable factor of a testable problem hypothesis. The automatic mapping may not be possible due to the above and other issues [63]. As an alternative, semi-automatic or manual mapping with tool support may be utilized.

3.3.3 The Gomphy Semantic Reasoning Methods

An important aspect of contributions in the hypothesis validation process is the propagation of validations throughout the connected hypotheses since a hypothesis might contribute to multiple other hypotheses. This validation process starts in the lowermost level of the hypotheses and propagates until we validate or invalidate problems.

A formal definition for validating problem hypotheses may be described as follows: Let $validated(P_n)$ be the proposition that the problem hypothesis P_n is validated, for $n \in \mathbb{Z}^+$. For all $i, j \in \mathbb{Z}^+$, let $P_{i+1,j}$ be the jth problem hypothesis directly decomposed from P_i. Assuming this decomposition is of type OR/AND, the validation propagation can be represented by the following:

$$\Big(\bigvee_j validated(P_{i+1,j}) \Big) \rightarrow validated(P_i) \tag{3.2}$$

$$\Big(\bigwedge_j validated(P_{i+1,j}) \Big) \rightarrow validated(P_i) \tag{3.3}$$

Alternatively, hypotheses can also be connected using a *positive (S+)*, *negative (S-)* or *unknown* contribution type.

Gomphy defines *feature importance value* (I) obtained from running ML and ML Explainability models. The feature importance is associated with their respective Contributions, i.e.,

a Contribution from a problem hypothesis P_s (source) to P_t (target) has an importance value $I_{s,t}$, and the following Formulas may determine its weight and Contribution type.

$$weight(P_s, P_t) = I_{s,t} \tag{3.4}$$

$$ctr\_type\big(I_{s,t}\big) = \begin{cases} \text{S+} & \text{if } I_{s,t} \geq 0 \\ \text{S-} & \text{if } I_{s,t} < 0 \end{cases} \tag{3.5}$$

A source hypothesis has a score based on the weight of the targeted hypotheses and their respective contributions. The function $weight(P_t)$ describes the importance weight of a target hypothesis. Hence, the overall score for a source hypothesis P_s can be given by the utility function as follows:

$$score(P_s) = \left(\sum_{t=1}^{\#targets} weight(P_t) \times weight(P_s, P_t) \right) \tag{3.6}$$

After computing the scores for all source hypotheses, the selection process may be carried out in a bottom-up approach [64]. We need to select the maximum value in the lowest source hypothesis set to propagate that validation to the target hypothesis set. In other words, the selection process for a target is represented by choosing the source with the highest score:

$$selection(P_t) = max\Big(score(P_s)\Big)_{s=1}^{\#sources} \tag{3.7}$$

We want to determine which hypothesis in the source set (i.e., hypotheses that originate the contributions) is more relevant to the target set (i.e., hypotheses that receive the contributions) to maximize the validation insights generated by the application of ML models. In this case, validating a hypothesis P_i will now depend on the validation of the selection for P_i. After the lowest source hypothesis set is evaluated, we proceed to the next one until the selection process covers the entire set of hypotheses.

$$validated(selection(P_i)) \rightarrow validated(P_i) \tag{3.8}$$

CHAPTER 4

VALIDATING BANKING PROBLEMS BEHIND UNPAID LOAN

This chapter [1] applies the proposed Gomphy framework to explore hypothesized business events behind an unpaid loan problem in one bank and validate the problem hypotheses as an empirical study. Fig. 4.1 shows a high-level context diagram for the overdue loan problem. We use the PKDD'99 Financial database [26] to represent data that the bank may have managed.

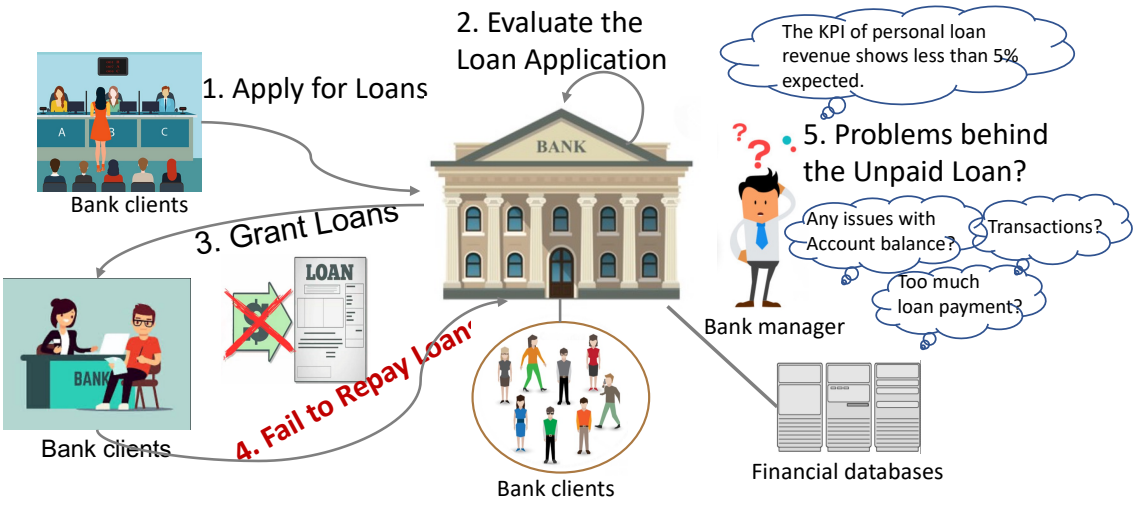


Figure 4.1. Unpaid loan problem in a bank

4.1 Introduction

The assertion that *"A problem unstated is a problem unsolved"* expresses the importance of eliciting business needs and problems [1]. Understanding and validating a business problem likely to hinder a business goal is often more critical than developing solutions. Validating

[1]This chapter contains material previously published as: ©2021 Springer. Reprinted, with permission, from Ahn, R., Supakkul, S., Zhao, L., Kolluri, K., Hill, T., & Chung, L. (2021, December). Validating Business Problem Hypotheses: A Goal-Oriented and Machine Learning-Based Approach. In International Conference on Big Data (In press). Springer, Cham.

a business problem helps define system boundaries in the early phase of requirements engineering [2]. If the correct problems are validated first, a business can save precious time and cost to deal with erroneous problems [5].

However, business organizations face difficulties confirming whether an elicited business event causes or impacts other high-level problems [18, 48]. Specifically, some challenges might be determining testable factors for the elicited problem, constructing a dataset to test, and determining whether the identified problem has some relationships and how many degrees towards the high-level problem and a business goal. The challenges can be illustrated in Fig. 4.2. Developing an information system with unconfirmed problems frequently leads to a system that is not useful enough to achieve business goals, costing valuable business resources [12, 20].

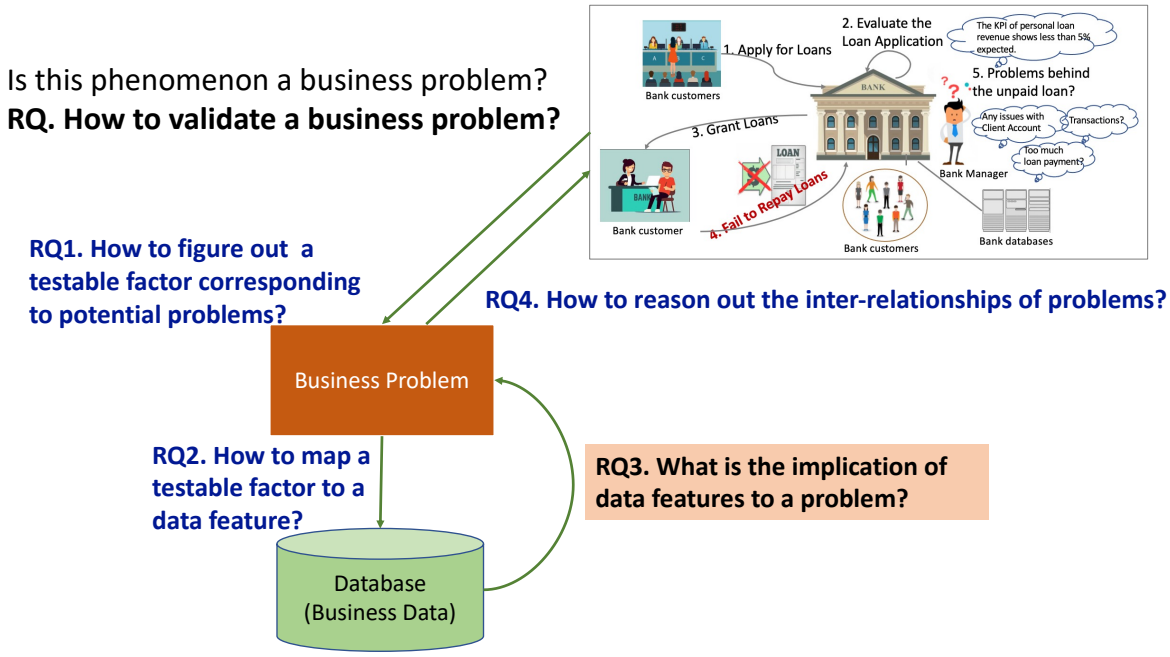


Figure 4.2. Research questions corresponding to research challenges

Drawing on our previous work Metis [22], we present *GOMPHY*, a *G*oal-*O*riented and *M*achine learning-based framework using a *P*roblem *H*Ypothesis, to help validate business problems [29, 65]. This paper proposes four main technical contributions: 1. An ontology

for modeling and validating a business problem hypothesis is described. 2. An entity modeling method for a problem hypothesis is presented to help identify an entity, attributes, constraints, and relationships for a problem hypothesis in the source dataset. 3. A data preparation method is described, mapping a problem hypothesis entity to a database entity and features, extracting and transforming a dataset. 4. An evaluation method is elaborated to detect positive or negative contributions among business problems and goals using Machine Learning (ML) and ML Explainability techniques.

4.2 The Gomphy In Action

We suppose a hypothetical bank, the Case bank providing client services, such as opening accounts, offering loans, and issuing credit cards. The bank has experienced an unpaid loan problem. Some clients failed to make recurring payments when due. However, it did not know what specific clients' banking behaviors were behind this issue. Since this is a hypothetical example, we used the PKDD'99 Financial database to represent data the bank may have managed [26].

PKDD'99 Financial Database: The database contains records about banking services, such as Account (4,500 records), Transaction (1,053,620), Loan (682), Payment Order (6,471), Client (5,369), Credit cards (892), and Demographic (77). Six hundred six loans were paid off within the contract period, and seventy six were not among the loan records. Fig. 4.3 shows the schema of the Financial database.

4.2.1 Step 1: Explore the Case Bank's Problem Hypotheses

Requirements engineers begin Step 1, understanding and modeling the Case bank's goals. Potential problems hindering the goals are then hypothesized.

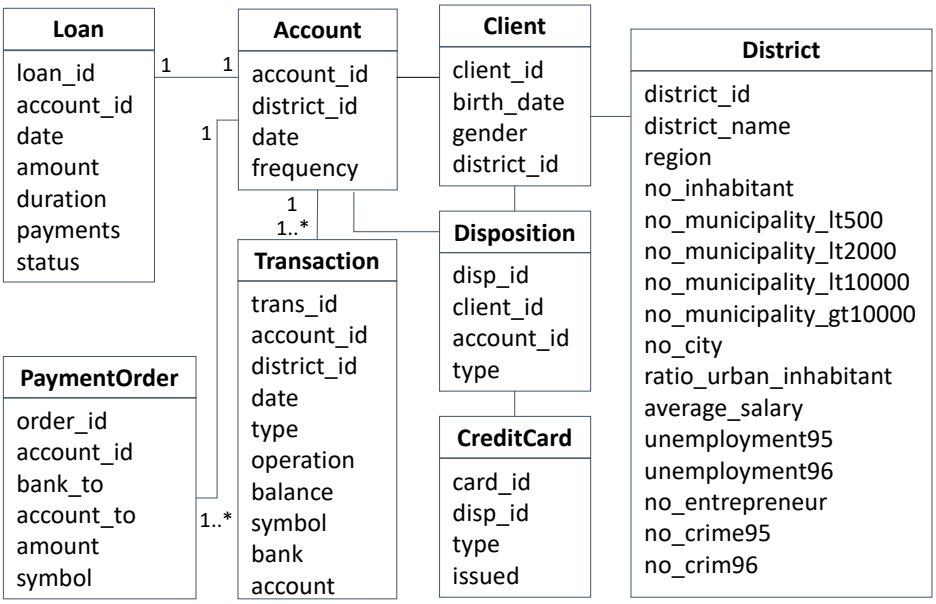


Figure 4.3. The schema of the Financial database

Step 1.1 Capture the Case bank's goals

After understanding the bank domain, one of the bank's goals, *Maximize revenue*$_{NFsoftgoal}$ [2] is modeled as an NF (Non-Functional) softgoal to achieve at the top level, which is AND-decomposed and operationalized by *Increase loan revenue*$_{OPsoftgoal}$ and *Increase fee revenue*$_{OPsoftgoal}$ as operationalizing softgoals, as shown in Fig. 4.4. The former is further AND-decomposed to more specific softgoals of *Increase personal loan revenue*$_{OPsoftgoal}$ and *Increase business loan revenue*$_{OPsoftgoal}$.

During an interview, the bank staff indicated that the personal loan revenue of this quarter is less than 5 percent for the Key Performance Indicator (KPI) they intended to achieve due to some clients' unpaid loans. So, the bank wanted to know which specific banking events of a client contribute to the outstanding loan. However, the bank staff had difficulties with how to do that.

[2]The Gomphy concept is expressed in the notation from [66].

Step 1.2: Hypothesize problems hindering the Case bank's goal

We modeled that a client's *Unpaid loan*$_{OPsoftproblem}$ *Breaks*(−−) the *Increase personal loan revenue*$_{OPsoftgoal}$. After understanding the loan process and analysis of the Financial database, we explored potential clients' banking behaviors against the unpaid loan. We hypothesized that a client's *Loan*$_{AbstractPH}$, *Account Balance*$_{AbstractPH}$, and *Transaction*$_{AbstractPH}$ might somewhat positively contribute to the *Unpaid loan*$_{OPsoftproblem}$, as illustrated in Fig. 4.4.

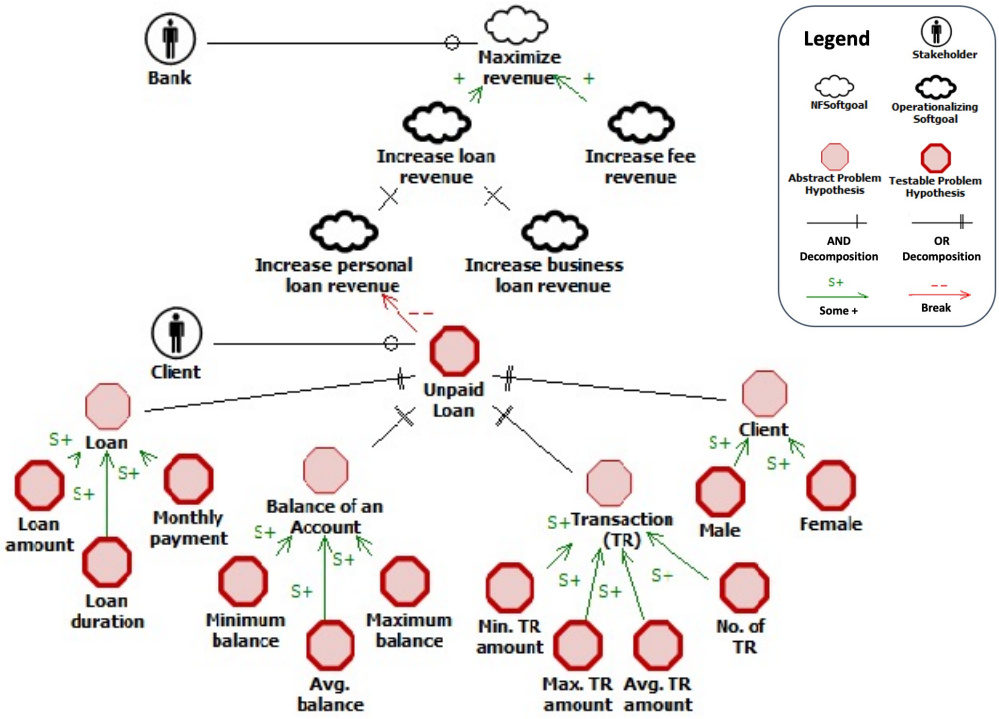


Figure 4.4. Hypothesizing problems for an unpaid loan

An abstract problem hypothesis is further decomposed into a testable problem hypothesis that usually contains nominal, ordinal, interval, or ratio factors. For example, the *Balance of an Account*$_{AbstractPH}$ may be divided into the more specific *Minimum balance of an Account*$_{TestablePH}$, *Average balance of an Account*$_{TestablePH}$, and *Maximum balance of an Account*$_{TestablePH}$ for the client's loan duration using an OR-decomposition method.

Based on the goal and problem hypothesis graph above, we can express one of the problem hypotheses in a conditional statement. Let PH1 be the problem hypothesis *The minimum balance of an Account somewhat positively contributes to an unpaid loan for the loan duration*$_{PH}$. Then, we can consider the *Minimum balance of an Account*$_{SourcePH}$ as a source problem hypothesis (or an independent variable), *Somewhat positively contributes*$_{PHcontribution}$ as a contribution relationship, and an *Unpaid loan for the loan duration*$_{TargetPH}$ as a target problem hypothesis (or a dependent variable).

$$\begin{aligned} &\textit{Minimum balance of an Account}_{SourcePH} \\ &\xrightarrow{Some-plus_{PHcontribution}} \textit{Unpaid loan for the loan duration}_{TargetPH} \end{aligned} \tag{4.1}$$

4.2.2 Step 2: Prepare an ML Dataset

In this step, we model a concept in the testable problem hypothesis as a problem hypothesis entity, maps the attributes of a problem hypothesis entity to the attributes of a database entity in the database or a domain dataset, and constructs an ML dataset based on the identified data attributes.

Step 2.1: Model a concept in a problem hypothesis as an entity

The concept in a testable problem hypothesis is modeled as an entity using the entity-relationship model [67]. The entity has attributes, constraints, and relationships. An *attribute* is a property of an entity having measurable value. A *constraint* is a condition restricting the value or state of a problem hypothesis. A *relationship* shows other entities associated with this entity.

For example, the *Minimum balance of an Account*$_{SourcePH}$ in PH1 is modeled as a problem hypothesis entity of *Account*$_{PHE}$, having an attribute of *balance*$_{PHEattribute}$, a constraint of a *minimum balance*$_{PHEconstraint}$, and a relationship of a *Loan*$_{PHErelationship}$, as shown in

Fig. 4.5. The testable factor, $balance_{PHEattribute}$ of a source problem hypothesis entity, may be determined using Algorithm 1 in Section 3.3.

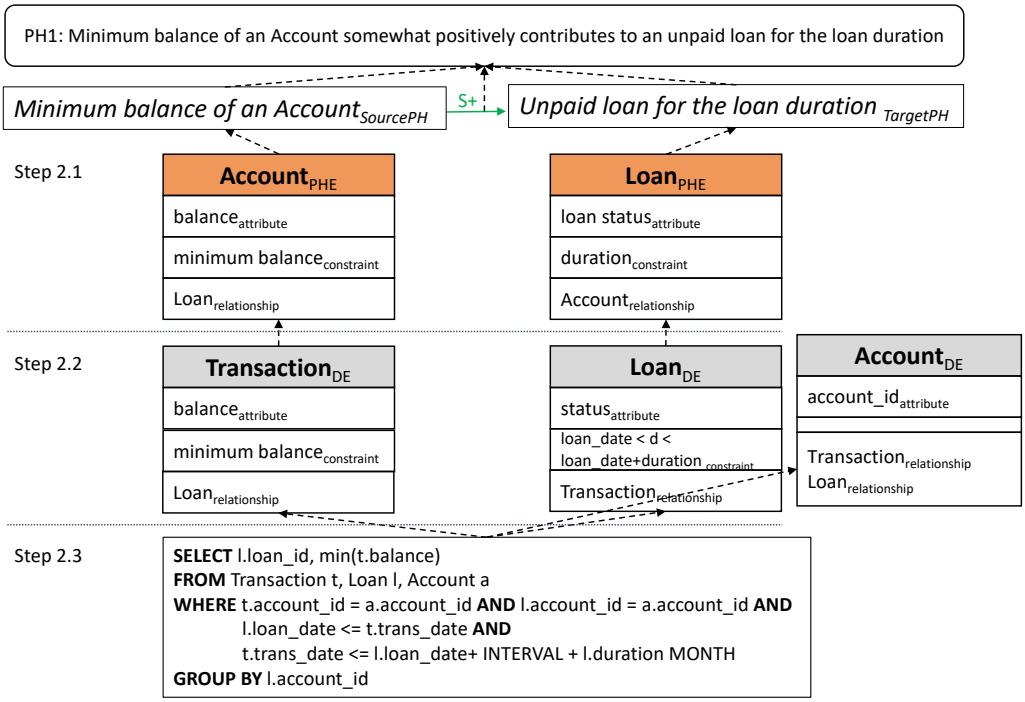


Figure 4.5. Preparing an ML dataset for validating a problem hypothesis

Step 2.2: Map a problem hypothesis entity to a database entity

The attribute of a problem hypothesis entity may be semi-automatically mapped to attributes in the database entity using Algorithm 2 in Section 3.3 while considering the constraints and relationships of a problem hypothesis entity. Another way to map an attribute of a problem hypothesis entity to attributes in the database entity may be possible with tool support in Fig. 4.6. The tool first reads the database schema and shows the concerned entity and attributes. We then select a database entity and check whether attributes in the entity are similar to the attributes of the problem hypothesis entity.

For $balance_{PHEattribute}$ of $Account_{PHE}$, we first select the Account entity and check whether an attribute in the entity semantically matches the $balance_{PHEattribute}$. As we can

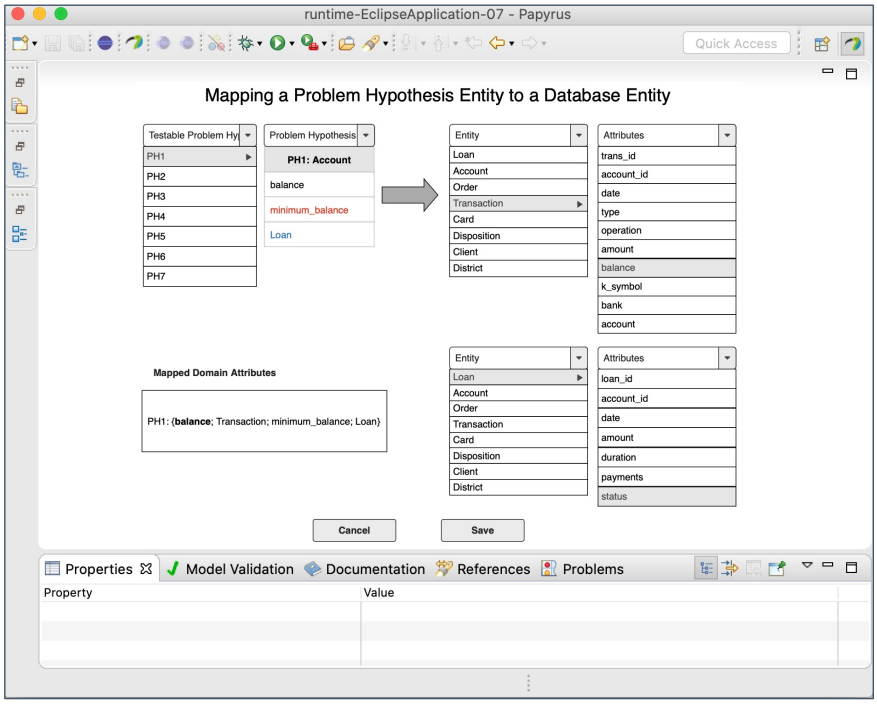


Figure 4.6. Mapping attributes between a problem hypothesis entity and a database entity

not find a relevant attribute of the Account, we check the subsequent entities. While iterating database entities, we can find a *balance* attribute of the Transaction entity, representing a balance after the banking transaction. So, we map $Account_{PHE}$ to $Transaction_{DE}$ and $balance_{PHEattribute}$ to $balance_{DEattribute}$. The constraint and relationships of a problem hypothesis entity are similarly mapped to those of a database entity.

Step 2.3: Build an ML dataset

The identified attributes, constraints, and relationships corresponding to the source and target problem hypothesis entity are used to build a database query and extract a dataset, as shown in Fig. 4.5. Data preprocessing techniques are then applied to the extracted dataset.

For example, the data of the *Minimum balance of an* $Account_{SourcePH}$ can be extracted using the identified $balance_{DEattribute}$, and *minimum* $balance_{DEconstraint}$ in $Transaction_{DE}$. SQL group function, min() may be used to select *minimum* $balance_{DEconstraint}$. Also, to

apply the relationship $Loan_{DErelationship}$, we need to identify a primary key and a foreign key relationship between $Loan_{DE}$ entity and $Transaction_{DE}$, which lead to identifying $Account_{DE}$ entity. The *loan duration*$_{DEconstraint}$ of $Loan_{DE}$ is also applied, as shown in Fig. 4.5.

We then tentatively store the resulting dataset for each testable problem hypothesis and integrate it into an ML dataset. Next, we may need to transform some feature values using preprocessing techniques, including scaling feature value using a normalization method and converting categorical data to a numeric value, such as using a one-hot encoding method, in our example on the transaction type, mode, symbol features. We may also fill in some missing values, replacing the null value with an average value and others [68].

4.2.3 Step 3: Evaluate the Impact of Problem Hypotheses Using ML

The impact of banking events towards the unpaid loan encoded as data features and a target label is uncovered using Supervised ML models and ML Explainability model, decoding hidden feature patterns in the dataset [69, 70].

Step 3.1: Detect feature importance

To decode the relationships among banking events and an unpaid loan, four ML models, such as Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), and eXtreme Gradient Boosting (XGBoost) were built with the dataset. The ML models then predicted the loan instances as *Paid* or *Unpaid Loan*. The accuracy of each ML model was 0.92 (LR), 0.95 (DT), 0.973 (RF), and 0.977 (XGBoost), as shown in Fig. 4.12. The more accurate an ML model, the more confidence we can use feature importance value to get insight for validating a problem hypothesis.

Next, we utilized the SHAP (Shapley Additive exPlanations) model to get an intuitive and consistent feature value [58]. The XGBoost model was given as input to the SHAP model. To analyze the feature importance for prediction results, we first collected predicted

instances of unpaid loans. Fig. 4.7 shows the SHAP value of some important features for one case, where we can notice that the minimum balance, the minimum amount transaction, the average balance, and the household remittance somewhat positively impact the unpaid loan. The wider the width, the higher the impact. After that, we summed up the feature values of all the unpaid loans to detect the feature impact of all unpaid loans.

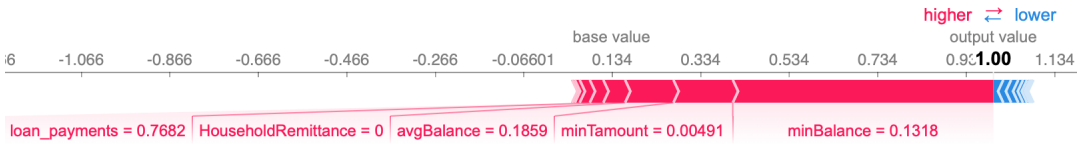


Figure 4.7. Feature importance for one unpaid loaner

Step 3.2: Update a contribution weight and type with feature importance

The collected feature importance value ($I_{source,target}$) can be considered a contribution weight from a source to a target problem hypothesis. The contribution weight and type of each leaf-level problem hypothesis are updated based on the detected feature importance value using Formula 4.2 and 4.3.

$$weight(PH_{source}, PH_{target}) = I_{source,target} \tag{4.2}$$

$$ctr\_type\big(I_{source,target}\big) = \begin{cases} \text{S+} & \text{if } I_{source,target} \geq 0 \\ \text{S-} & \text{if } I_{source,target} < 0 \end{cases} \tag{4.3}$$

For example, the Contribution weight and type of the leaf node, the minimum balance, are updated with the detected value 15.32 and $S+$ in Fig. 4.8. Similarly, the contribution weight and type of other leaf nodes are updated accordingly.

Next, to know the direct and indirect impact of leaf-level problem hypotheses towards a high-level problem in the problem hypothesis model, we first calculate the fitness score of a source problem hypothesis using Formula 4.4.

$$score(PH_s) = \left(\sum_{t=1}^{\#targets} weight(PH_t) \times weight(PH_s, PH_t) \right) \tag{4.4}$$

We assume the weight of each problem hypothesis is 0.2, adopting a weight-based quantitative selection pattern [64]. For example, the fitness score of the *Minimum balance of an* $Account_{SourcePH}$ is calculated as (0.2 * 15.32 =) 3.064.

4.2.4 Step 4: Validate Problem Hypotheses

This step selects the most critical problem hypothesis as a validated one among many alternative hypotheses and evaluates the impact of the validated problem on other high-level problems, as shown in Fig. 4.8.

Step 4.1: Select the most influential source problem hypothesis

Among alternative problem hypotheses contributing to a target problem in the problem hypothesis model, we select a problem hypothesis having the highest fitness score in the leaf nodes. Banking staff may give a qualitative priority for some problem hypotheses, depending on some schemes, such as *normal*, *critical*, or *very critical*. Here, we assume a *normal* priority for all the problem hypotheses.

For example, the *Minimum balance of an* $Account_{SourcePH}$ in the problem hypothesis model was selected by Formula 4.5 as it has the highest fitness score among the leaf problem hypotheses under *Unpaid* $loan_{OPsoftproblem}$.

$$selection(PH_{target}) = max\Big(score(PH_{source})\Big)_{s=1}^{\#sources} \tag{4.5}$$

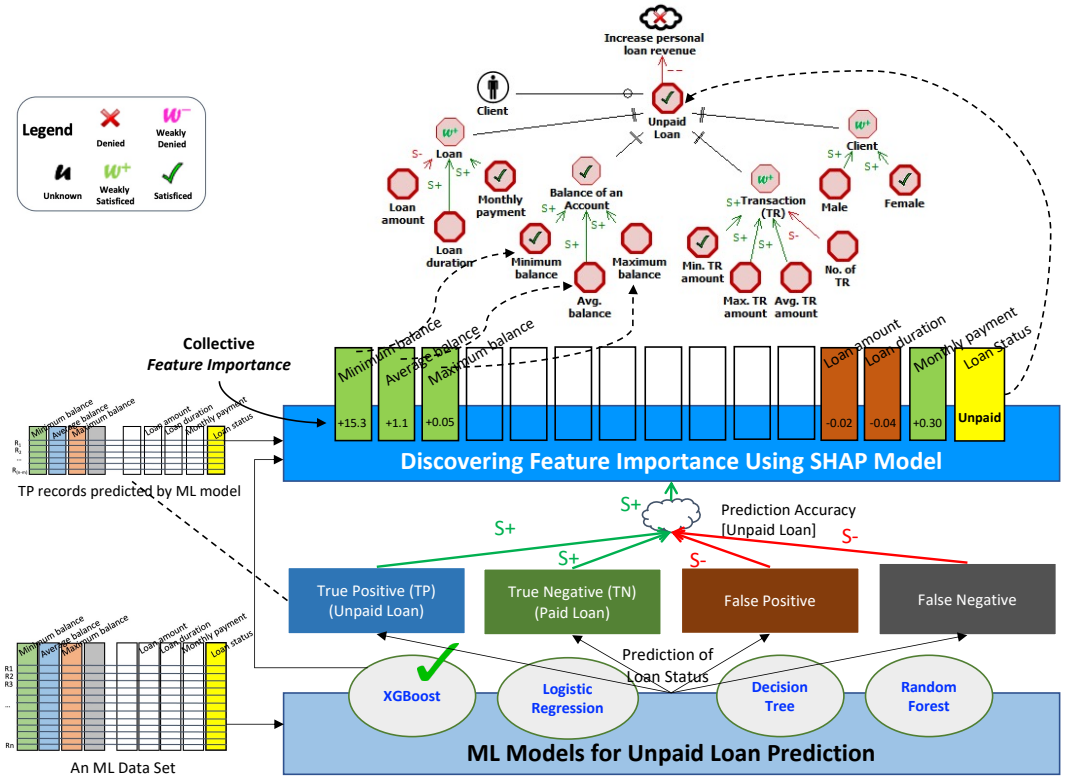


Figure 4.8. Validating a problem hypothesis using feature importance

The chosen problem hypothesis is considered a validated problem hypothesis by Formula 4.6, as it is most likely to be the cause for the target problem hypothesis [59]. It means the *Minimum balance of an Account*$_{SourcePH}$ is likely to be the most important cause of the *Balance of an Account*$_{AbstractPH}$.

$$validated(selection(PH_i)) \rightarrow validated(PH_i) \tag{4.6}$$

Step 4.2: Apply qualitative reasoning methods to reason the validation impact towards a high-level problem

Once the most likely problem hypothesis is validated, as shown by *check mark* in Fig. 4.8, qualitative reasoning, e.g., the label propagation procedure [29], is carried out to determine the validated problem's impact upward a problem.

If the *Minimum balance of an Account*$_{SourcePH}$ and *Somewhat positively contribute to* $_{PHcontribution}$ are satisficed, then the *Balance of an Account*$_{AbstractPH}$ is satisficed. The reasoning propagation shows that the *Balance of an Account*$_{SourcePH}$ somewhat positively contributes to the *Unpaid loan*$_{OPsoftproblem}$, which *Breaks* the *Increase personal loan revenue*$_{OPsoftgoal}$ in Fig. 4.8.

4.3 Experimental Results

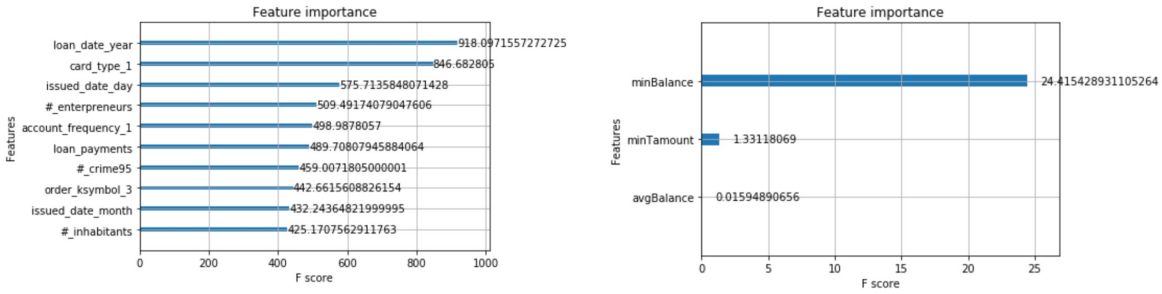
We performed three experiments to see the strength and weakness of Gomphy. While experiments 1 and 2 were performed without Gomphy, experiment 3 was conducted with the Gomphy framework.

4.3.1 Experiment 1

In this experiment, we treated all the features in the Financial database as potential events causing unpaid loans. We prepared the ML dataset by selecting all the data features in the database, except the table identifiers, where the selected features were considered potential problems and loan status as a target label. The prepared ML dataset included 72 features and 449,736 records based on the Transaction id. The large records are due to the *join* operation among Account, Transaction, and Payment Order tables.

As some ML algorithms such as Gradient Boosting Tree provide feature importance, we analyzed whether the provided essential features could be possible banking events leading to the unpaid loan. Fig. 4.9(a) shows some important features predicted by the XGBoost model. However, it was not easy to get some ideas about whether the loan granted year and the credit card type, *classic*, has some relationships towards the unpaid loan.

One critical issue of this approach is that the ML models, e.g., XGBoost, showed different prediction results for the same loan instance. For example, different transaction records,



(a) Top 10 important features in XGBoost Exp. 1

(b) Top 3 important features in XGBoost in Exp.2

Figure 4.9. Top important features in experiment 1 and 2

having the same Loan ID 233, showed different loan prediction results (i.e., paid and unpaid), which confused in identifying a banking event leading to the unpaid loan.

4.3.2 Experiment 2

In this experiment 2, the ML dataset was prepared based on the loan ID, unlike experiment 1, to understand the primary features produced by ML models. For preparing the loan-based dataset, we used SQL group functions, such as Sum, Min, and Avg, to select records for the one-to-many relationships between Account and Transactions. The final dataset contained 682 records, including 72 features. Four ML models were built to predict the loan instances. Fig. 4.9(b) shows some important features for the XGBoost model.

Among the three important features, minimum balance, minimum transaction amount, and average balance, it could be possible that the minimum balance could cause an unpaid loan. However, it was confusing whether the minimum amount of transaction could lead to unpaid loan events. Other banking events related to the minimum amount of Transactions seemed to be needed to get a deep understanding of this issue.

A critical issue of this approach is that the prepared dataset did not consider the boundary condition of the records within the loan duration, which may give incorrect predictions and show a lack of rationale for identifying critical root causes of the unpaid loan. For example, when the loan duration of loan ID 1 is two years from 1993, the dataset included records of

1996 and 1997. The prediction based on the inaccurate dataset would not be reliable and cause other consequences in the bank.

4.3.3 Experiment 3

In this experiment 3, we applied the Gomphy framework to validate clients' banking events towards the unpaid loan. The banking events were hypothesized as four groups: Loan, Account, Transaction, and Client. The hypothesis is further analyzed into testable problem hypotheses, as shown in Fig. 4.8.

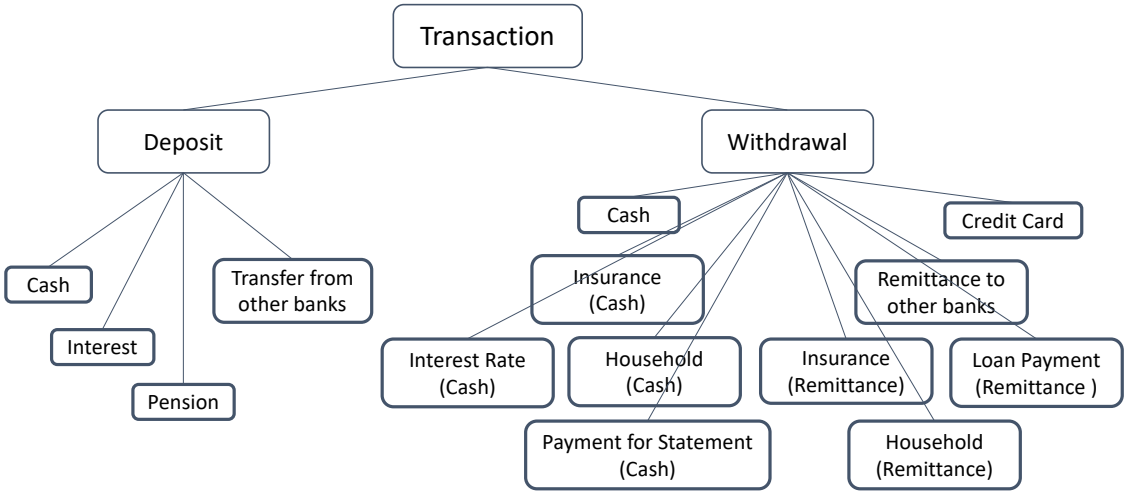
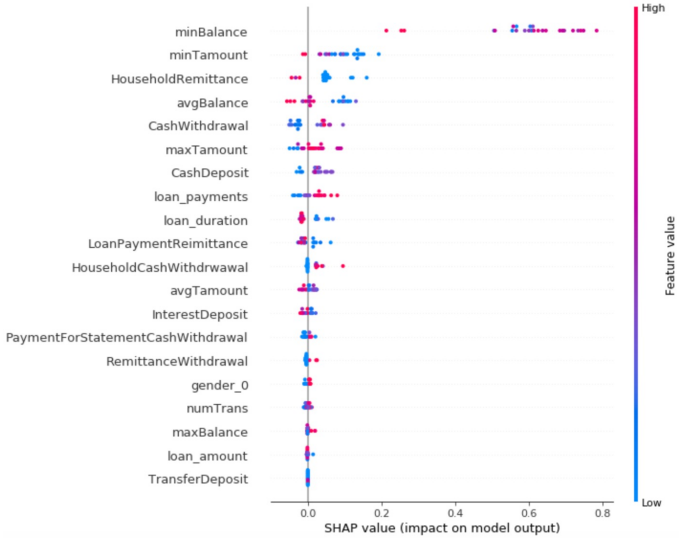


Figure 4.10. Deposit and withdrawal classification in Transaction

While preparing a dataset, we could understand that the *balance* of Transaction depends on transaction type (deposit or withdrawal), operation (mode of a transaction), and symbol (characterization of the transaction) features, as shown in Fig. 4.10. So, we hypothesized events related to the operations and symbols as possible causes to the change of the balance, which were added to the set of a problem hypothesis. Otherwise, the category features would be hot-encoded in a usual ML approach, like in experiments 1 and 2.

Based on the modeled problem hypotheses with banking domain understanding and database analysis, six hundred eighty-two records with 25 features were prepared. Four ML

models were run then to predict whether each loan could be paid off or not. Next, the ML Explainability model was applied with separately collected unpaid loan cases to understand better the impact of the features.



(a) Important features with SHAP in summary graph

Feature	Feature Value (Sum)	Contribution to Unpaid Loan
minBalance	15.318676	S+
minTamount	2.423251	S+
HouseholdRemittance	1.268283	S+
avgBalance	1.065214	S+
CashDeposit	0.589865	S+
maxTamount	0.422186	S+
HouseholdCashWithdrwawal	0.345132	S+
loan_payments	0.304449	S+
avgTamount	0.125425	S+
maxBalance	0.046698	S+
CashWithdrawal	0.046505	S+
loan_duration	0.03666	S+
gender_0	0.00492	S+
gender_1	0	U
TransferDeposit	0	U
InsuranceRemittance	0	U
CreditCardWithdrawal	0	U
InterestCashWithdrawal	0	U
loan_amount	-0.0156	S-
numTrans	-0.04178	S-
InterestDeposit	-0.047154	S-
RemittanceWithdrawal	-0.051432	S-
LoanPaymentReimittance	-0.06082	S-
PaymentForStatementCashWithdrawal	-0.12414	S-

(b) Features Value and Contribution Type

Figure 4.11. Important features and contribution type in experiment 3

Fig. 4.11, produced by the SHAP model, shows some important features for the unpaid loan, including the *minimum balance*, the *minimum transaction amount*, *remittance withdrawal for household cost*, and others. We could also understand that the *minimum transaction amount* is related to the *sanction interest* if the balance of an Account is negative after we performed further analysis.

The accuracy, precision, and F1-Score of the ML models in Experiment 3 are shown in Fig. 4.12. XGBoost showed slightly better accuracy than Random Forest. So, we selected XGBoost as the ML model for getting feature importance.

The trade-off analysis for the experiments is shown in Table 4.1. Experiment 1 is easier to perform validation assuming all the features as problem hypotheses. However, its results are difficult to understand, even giving different predictions for the same loan case. Experiment 2 shows more sensible results than experiment 1 but is still challenging to understand the

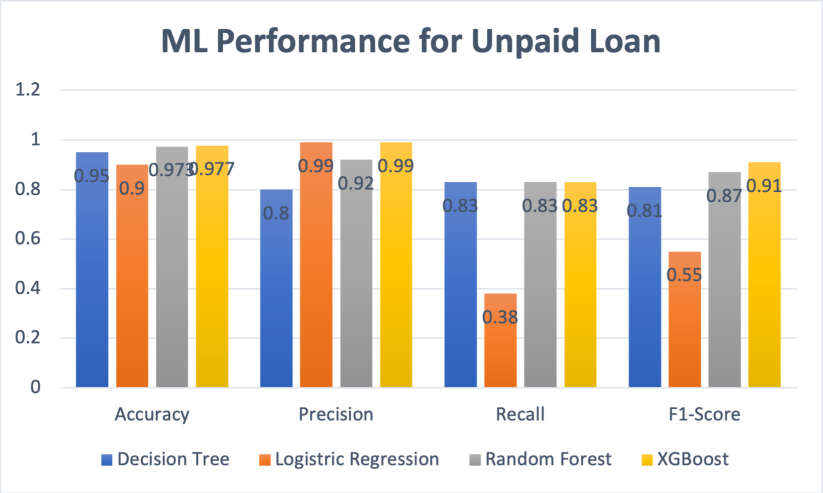


Figure 4.12. ML models' performance comparison in experiment 3

rationale and the relationships among the problem hypotheses and a target label. To validate and explain potential events, it needs to apply some systematic process, data constraints, e.g., data boundary and feature analysis. Experiment 3 provides sensible and understandable relationships between the banking events and an unpaid loan. It takes some time to apply the Gomphy process but helps identify the most critical banking event and provides insights into the hypothesized banking events.

Table 4.1. Experiments comparison for validating problem hypotheses

Relative streangth: - < + < ++

	Easy to Experiment	Understanding of Banking Domain	Loan-Related Feature Selection	Feature Explanation towards Unpaid Loan	Relationship b/w Problem and Goal
Experiment 1	++	-	-	-	-
Experiment 2	+	+	-	+	-
Experiment 3	+	++	+	++	++

4.4 Discussion and Related Work

Problem analysis and validation have been studied to understand real-world problems in two major areas: Requirements Engineering and Machine Learning. The distinctive of our approach is to use a concept of a problem hypothesis to refute or confirm potential business problems using ML.

In Requirements Engineering, a Fishbone diagram has been used to identify possible causes for a problem or an effect [35]. This technique helps enumerate potential causes for a problem. However, the lack of a clear relationship between a cause and an effect, e.g., logical connectives, such as *AND*, or *OR*, makes problem validation difficult.

The Fault Tree Analysis (FTA) provides deductive procedures and a logic diagram to help determine failures or errors of software, hardware, and people with a top-down approach [71]. FTA provides Boolean logic operators. When linked in a chain, these statements form a logic diagram of failure. However, FTA does not provide relationship direction and degrees, such as positive, negative, full, and partial, making it challenging to validate business problems using ML. *(Soft-)Problem Inter-dependency Graph (PIG)* uses a (Soft−)problem concept to represent a stakeholder problem against stakeholder goals, where a problem is refined into sub-problems [37]. However, PIG lacks a mechanism to connect sub-problems to data features to test. While the Fishbone diagram, FTA, and PIG provide a sound high-level model, they need validation mechanisms for confirming the causes behind business problems.

In the area of Machine Learning, some ML algorithms, such as Linear Regression and Decision Trees, provide feature importance value concerning their predictions. When ML models predict a numerical value in the regression model or a target label in the classification, relative feature importance scores are calculated for the features in the dataset [5]. Explainable machine learning models also provide feature importance [56]. LIME(Local Interpretable Model-agnostic Explanations) explains individual predictions, but there is some instability of the explanations, which may hurt validating business problems [57]. SHAP

(SHapley Additive exPlanations) outputs feature value that helps to understand business problems. However, SHAP may take a long computational time [58].

Although we could utilize feature importance in ML algorithms to get insights about business problems, one issue is identifying essential factors to test. Some features or attributes, among many features, in the dataset, might be redundant, irrelevant, or less critical to business problems. Dimensionality reduction techniques, such as Principal Component Analysis (PCA), and Formal Concept Analysis (FCA), are often used to find essential features for a target label [72, 73]. However, the data features selected by the dimensionality reduction techniques often makes it difficult to understand transparent relationships between the features and high-level business problems in the context of goals [74]. The Gomphy framework provides traceability from goals to problems, data features, and ML, bridging the gaps in a complimentary manner.

While some banking events, such as pension deposit at the end of a month, regularly occur, others, such as household payment with cash, randomly. Our work deals with the time-series nature of both banking events, but not in a strict sense. In-depth analysis and validation of the time-series banking events may provide other insights about potential causes of the unpaid loan.

Limitations This paper has some limitations. 1) Correlation among problem hypotheses and goals could be utilized to understand the business events better, but the correlation analysis was not explored yet. 2) The mapping process between a problem hypothesis entity and a database entity is partially supported with a prototype mapping tool, although the tool needs more work to be more effective. 3) After a potential problem, e.g., a minimum balance, is validated, bank staff may take tentative actions. For example, the bank may waive fees on missed loan payments or offer affected clients options to defer loan payments for a finite period. The more long-term and effective solutions need to be explored and validated using a solution hypothesis to mitigate the validated problem.

4.5 Conclusion and Future Work

This paper has presented the Gomphy framework to validate business problems with an empirical study validating clients' banking events behind an unpaid loan. Business organizations may use Gomphy to confirm whether some potential problems hidden in Big Data are against a business goal or not. Gomphy would help find real business problems and improve business value, especially in Big Data and Machine Learning (ML) projects. Four main technical contributions were: 1. A domain-independent Gomphy ontology, helping avoid omissions and commissions in modeling categories of essential concepts and relationships, 2. A method of modeling a concept of a problem hypothesis as a problem hypothesis entity, 3. A data preparation method, supporting to identify relevant features to test in a database and build a dataset; 4. An evaluation method detecting the positive and negative relationships among problem hypotheses and validating the problem hypothesis with feature importance and reasoning scheme.

Future work includes an in-depth study about the positive or negative impact of correlated features and analysis of time-series events on validating problem hypotheses, exploring potential solutions to mitigate the validated problem using ML and a goal hypothesis, and developing a reliable Gomphy assistant tool.

CHAPTER 5

DATA PREPARATION FOR VALIDATING BANKING PROBLEMS BEHIND UNPAID LOAN

Preparing an essential dataset representing business problems is vital in the Machine Learning (ML) project to mine some hidden patterns in data and discover insights leveraging the patterns in alleviating or mitigating business problems. In this chapter [1] , we show how an ML dataset about potential banking problems behind the unpaid loan is constructed from the Case bank's Financial database by applying the Dregon ontology and process.

5.1 Introduction

Preparing an essential dataset representing business problems is vital in the Machine Learning (ML) project to mine some hidden patterns in data and discover insights leveraging the patterns in alleviating or mitigating business problems [3, 49, 5]. For example, preparing a relevant dataset from a banking database for predicting a client's loan default would be critical to the success of the ML project as the bank may take some actions to mitigate the problem with the prediction result.

However, preparing an ML dataset for identifying some events behind a business problem is challenging [13, 18]. Specifically, some challenges might be systematically exploring potential events leading to a business problem, identifying testable factors for the specified events, and mapping the testable factors to data features to extract relevant data from source data. Problem validation with an irrelevant or unimportant dataset may give inaccurate predictions, leading to dissatisfaction systems, consequently not solving business problems and failing to achieve business goals [20, 21].

[1] This chapter contains material previously published as: ©2021 IEEE. Reprinted, with permission, Ahn, R., Supakkul, S., Zhao, L., Kolluri, K., Hill, T., & Chung, L. (2021, October). A Goal-Oriented Approach for Preparing a Machine-Learning Dataset to Support Business Problem Validation. In 2021 IEEE International Conference on Cloud and Big Data Computing (CBDCom). pp. 282-289, IEEE.

Drawing on our previous work, GOMA [75], Metis [22], and Gomphy [24], we present a goal-oriented data preparation framework, *DREGON* (Data pREparation using GOal-orieNtation), to support business problem validation. Four technical contributions are made in this chapter. Firstly, a domain-independent ontology and a process for data preparation are described. Secondly, a method for capturing business events likely causing problems is presented. Thirdly, an entity modeling method determining a testable factor of the captured business event is elaborated. Fourthly, a mapping method for connecting a testable factor to a database entity and features is shown.

This chapter illustrates the proposed *Dregon* approach using a retail banking application and a Financial database. We suppose a hypothetical bank, the Case bank provides client services, such as offering loans and issuing credit cards. The bank has experienced an unpaid loan problem, where some clients failed to pay loan payments when due. However, it was challenging for the bank manager to know what specific clients' banking behaviors were behind this issue. So, the bank consulted a data analytics company to address this issue. The company hypothesized potential events impacting the loan problem against the bank's goals. It then prepared some data from the Financial database, performed an in-depth ML analysis by validating the hypothesized events, and suggested highly likely client's banking behaviors leading to the loan problem to the bank manager. This chapter shows how goals, problems, hypothesized banking events, and some ML concepts, such as data features, a target label, and classification, can be systematically applied to prepare a dataset to validate potential banking events towards the unpaid loan. Our approach could help the bank manager make sound decisions about alternative potential banking events and get confidence in mitigating the problem. Fig. 5.1 shows a high-level context diagram concerning the unpaid loan.

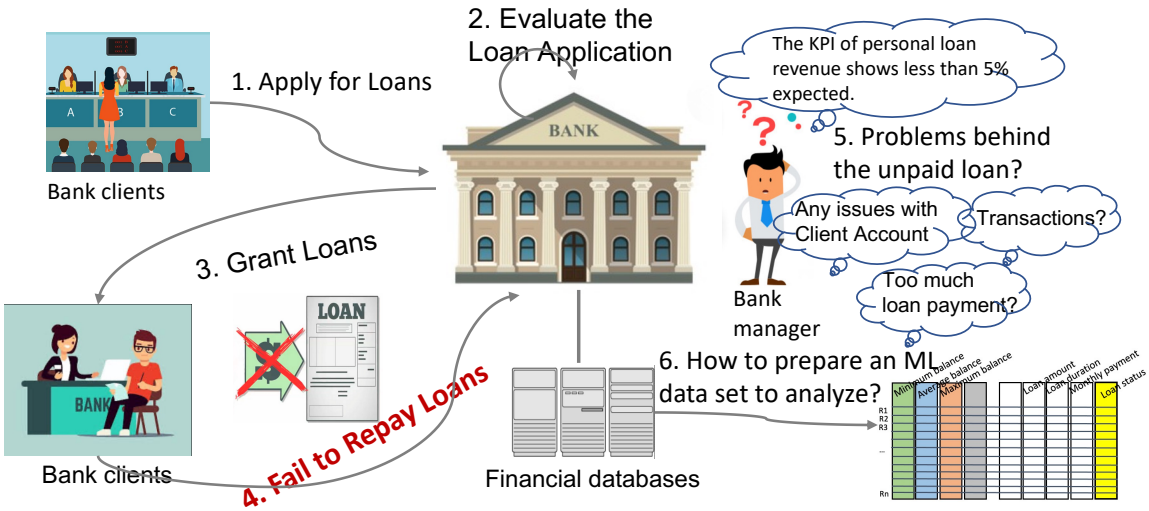


Figure 5.1. Unpaid loan in Case bank (empirical study context)

5.2 Related Work

The distinctive of our data preparation approach is to use a problem hypothesis for exploring alternative causes of a business problem in a goal-oriented manner, map the alternatives to data features of a database entity, and extract relevant data from a source database. The prepared data set is then entered into ML models to support business problem validation.

Problem analysis and data preparation have been studied to understand and solve real-world problems in two major areas: Requirements Engineering and Machine Learning [46, 76]. In Requirements Engineering, a Fishbone diagram [35], Fault Tree Analysis (FTA) [71], Problem Frame [77], and (Soft-)Problem Interdependency Graph (PIG) [37] have been used to analyze root causes behind a problem. A Fishbone diagram supports enumerating potential reasons for a problem and is typically used in a brainstorming session. FTA depicts a failure path and forms a logic diagram of failure. Problem Frame uses concepts including phenomena, shared phenomena, and domain requirements to analyze business problems and develop software solutions. PIG uses a (Soft−)problem concept to represent a stakeholder problem against stakeholder goals and provides refinement methods for a (Soft−)problem.

While these techniques provide a sound, high-level model for analyzing business problems into sub-problems and some relationships, they lack mechanisms for connecting high-level concepts to the data features in a database and validating the identified problems using operational data in business.

In the area of Machine Learning, data is prepared in the structure or format that fits each machine learning task. As business databases may include noise, missing values, similar features, or redundant data, some low-quality data should be preprocessed or reduced for good prediction. There can be two kinds of preparation techniques, data preprocessing and data reduction [49]. The data preprocessing techniques may include data cleaning, transformation, integration, normalization, missing data imputation, and noise identification [50, 51]. In data reduction, the amount of data is downsized, while the reduced data still includes the essential structure of the original data. The data reduction techniques include feature selection, instance selection, discretization, feature extraction and/or instance generation [52, 53]. Although the data preprocessing and data reduction techniques in ML are useful in partly preparing data, these techniques often lack high-level concepts, such as goals and problems and their relationships, such as positive, negative contributions. These techniques are often used to identify low-level problems informally and do not provide traceability to higher-level problems [54]. Our approach prepares an ML data dataset to support business problem validation, adopting essential concepts of the goal-oriented and ML-based approach in a complementary manner.

5.3 The Dregon Approach

The Dregon approach provides a domain-independent ontology and a series of steps, helping prepare a dataset by exploring problems, determining a testable factor and data features, and extracting data.

The Dregon approach provides

1. a domain-independent ontology and a series of steps,

2. helping prepare a dataset by exploring problems,

3. determining a testable factor and data features, and

4. extracting data.

5.3.1 The Dregon Ontology

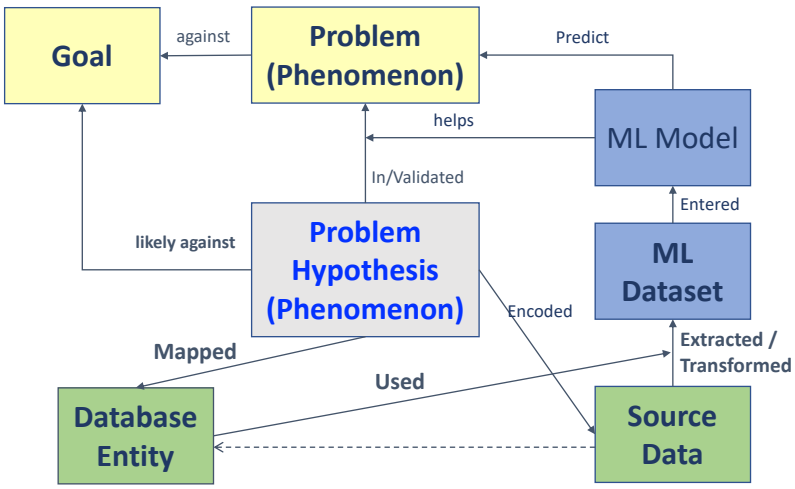


Figure 5.2. The data preparation ontology at a high-level

The Dregon ontology, adopting critical concepts from a goal-oriented [59] and ML-based approach [78], intends to help data preparation for validating a problem hypothesis. The ontology consists of essential categories of modeling concepts, relationships among concepts, and constraints on the concepts and relationships. Fig. 5.2 shows a high-level ontology. The boxes and arrows represent concepts and relationships among concepts.

The more detailed Dregon ontology is shown in Fig. 5.3. A few essential concepts needed for preparing a dataset are described. A *(Soft-)Goal* is defined as a goal that may not have a clear-cut criterion and can be specialized into a Non-Functional (NF) softgoal, an Operationalizing softgoal, and a Claim softgoal. While a *(Soft-)Problem* is a phenomenon

against a softgoal, a *Problem Hypothesis* is a hypothesis that we believe a phenomenon is against a softgoal.

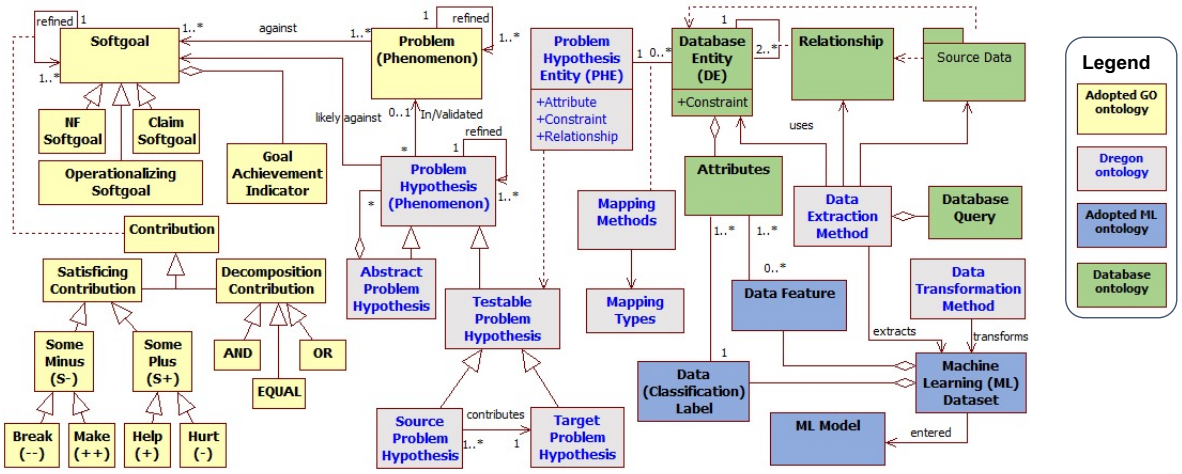


Figure 5.3. The detailed data preparation ontology for a validating problem hypothesis

There are two kinds of problem hypotheses, an *Abstract Problem Hypothesis* and a *Testable Problem Hypothesis*. An abstract problem hypothesis is conceptual and not concrete enough to test, whereas a testable problem hypothesis is measurable and testable. A Testable Problem Hypothesis may be further refined, forming a testable *Source Problem Hypothesis* and a *Target Problem Hypothesis*. A *Problem Hypothesis Entity* is an entity representing a Testable Problem Hypothesis and may be mapped to a relevant *Database Entity* having *Attributes*, *Constraints*, and *Relationships* in a source data model. The identified database entities are used to extract data from source data using *Data Extraction Method*.

The Contribution relationships among goals, problems, and problem hypotheses are categorized into Decomposition types, such as *AND, OR, EQUAL*, or *Satisficing* types, such as *Make, Help, Hurt, Break, Some-Plus, Some-Minus, Unknown* adopted from the NFR Framework [59]. The relationships between problem hypotheses and problems are either *Validated* or *Invalidated*.

One crucial constraint about a problem hypothesis includes time-order among a source and target problem hypothesis, where a source problem hypothesis must have occurred before

the target problem hypothesis. Other constraints are a positive contribution from a source problem hypothesis to a target problem hypothesis, and the contribution relationship should be reasonably sensible [60].

5.3.2 The Dregon Process

The Dregon process, shown in Fig. 5.4(a), consists of four steps, *Step 1: Explore business goals*, *Step 2: Hypothesize business problems*, *Step 3: Identify data features for a problem hypothesis*, and *Step 4: Extract and transform datasets*. The steps are necessary to systematically prepare an ML dataset and should be understood as iterative, interleaving, and incremental in ML projects. The detailed sub-steps are described in the following Section 4.2.

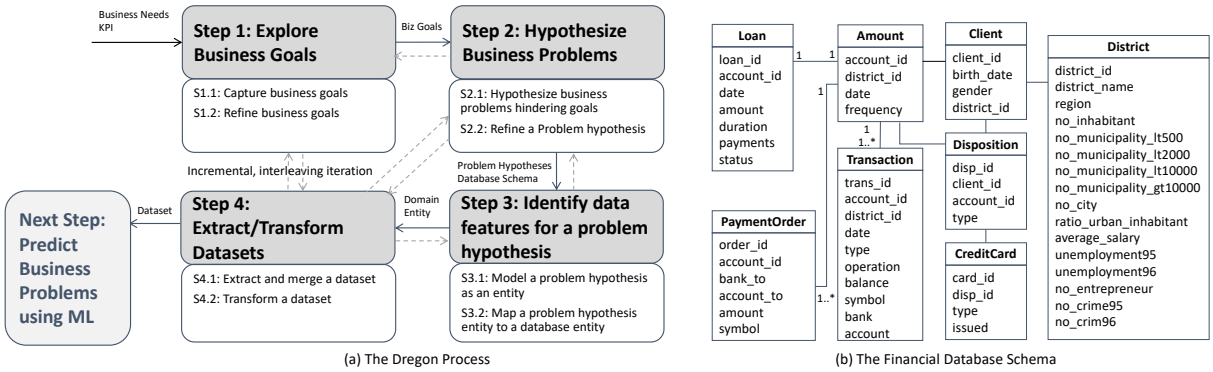


Figure 5.4. The data preparation process and the Financial database schema

5.4 The Dregon in Action

PKDD'99 Financial Database: The database contains records about banking services, such as Account (4,500 records), Transaction (1,053,620), Loan (682), Payment Order (6,471), and Credit cards (892) [26]. Six hundred six (606) loans were paid off within the contract period, and 76 were not among the loan records. The Financial database schema in Fig. 5.4(b) shows the conceptual schema of the Financial database in the UML notation.

5.4.1 Step 1: Explore Business Goals

We begin Step 1, understanding and modeling the Case bank's goals, and then refining high-level goals into concrete and measurable goals.

Step 1.1: Capture the Case bank's goals

To better understand the relationships between the Case bank's goals and problems, we interview the bank manager and staff to understand and capture the Case bank's business goals and process. *Maximize revenue* [2] is captured as one of the bank's high-level goals and then is modeled as an NF softgoal, *Maximize revenue*$_{NFsoftgoal}$ to achieve, as shown in Fig. 5.5(a).

Step 1.2: Refine the Case bank's goal

The modeled NF softgoal is AND-decomposed and operationalized by *Increase loan revenue*$_{OPsoftgoal}$ and *Increase fee revenue*$_{OPsoftgoal}$ as Operationalizing softgoals. The former is further AND-decomposed to more specific Operationalizing softgoals of *Increase personal loan revenue*$_{OPsoftgoal}$ and *Increase business loan revenue*$_{OPsoftgoal}$. During an interview, the bank staff indicated that the personal loan revenue of this quarter is less than 5 percent for the Key Performance Indicator (KPI) [79]. This KPI indicates the unpaid loan is a problem hurting *Increase personal loan revenue*$_{OPsoftgoal}$.

5.4.2 Step 2: Hypothesize Business Problems Hindering Goals

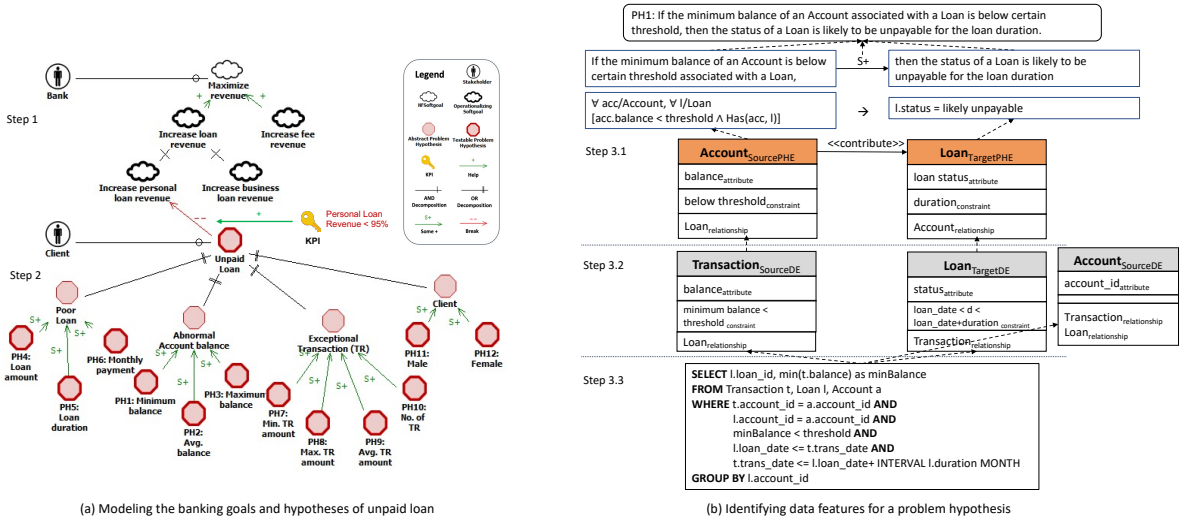
In Step 2, we explore possible banking events leading to the unpaid loan and determine a testable factor of the problem hypothesis to validate.

[2]The Dregon concept is expressed in the notation from [66] to show the modeling concepts in a class and an instance level.

Step 2.1: Hypothesize banking events hindering the Case bank's goal

We first model that a client's *Unpaid loan* $_{OPsoftproblem}$ *Breaks*$(--)$ the *Increase personal loan revenue*$_{OPsoftgoal}$. There could be many banking events related to the *Unpaid loan*$_{OPsoftproblem}$. To narrow the scope of business events to analyze, we then explore potential banking events that could positively contribute to the *Unpaid loan*$_{OPsoftproblem}$ and eventually hurt *Increase personal loan revenue*$_{OPsoftgoal}$. In other words, a goal and a problem are used as the context to search potential banking events.

After more understanding of the loan process and analysis of the Financial database, we hypothesize that a client's *Poor Loan*$_{AbstractPH}$, *Abnormal Account Balance*$_{AbstractPH}$, and *Exceptional Transaction*$_{AbstractPH}$ might positively contribute to the *Unpaid loan*$_{OPsoftproblem}$ at an abstract level, as shown in Fig. 5.5(a).



(a) Modeling the banking goals and hypotheses of unpaid loan

(b) Identifying data features for a problem hypothesis

Figure 5.5. Applying the Dregon process for an unpaid loan

Step 2.2: Refine an abstract problem hypothesis into a testable problem hypothesis

The identified abstract problem hypothesis is further decomposed into a testable problem hypothesis that usually has a value of categorical or numeric type. For example, the *Balance*

of an Account$_{AbstractPH}$ is OR-decomposed into the *Minimum balance of an Account*$_{TestablePH}$, *Average balance of an Account*$_{TestablePH}$, and *Maximum balance of an Account*$_{TestablePH}$ for the client's loan duration, which has a numeric balance.

Based on the goal and problem hypothesis graph, we can express one of the problem hypotheses in a conditional statement. Let PH1 be the problem hypothesis *If the minimum balance of an Account associated with a Loan is below a certain threshold, the status of Loan is likely to be unpayable for the loan duration.* Then, we can consider the *minimum balance of an Account associated with a Loan is below a certain threshold*$_{SourcePH}$ as a source problem hypothesis (or an independent variable), *some positively contributes*$_{PHcontribution}$ as a contribution relationship, and the *status of Loan is likely to be unpayable for the loan duration*$_{TargetPH}$ as a target problem hypothesis (or a dependent variable).

$$\textit{Minimum balance of an account below a threshold}_{SourcePH} \xrightarrow{\textit{Some-plus}_{PHcontribution}} \textit{Status of a loan unpayable for the loan duration}_{TargetPH} \tag{5.1}$$

5.4.3 Step 3: Identify Data Features for a Problem Hypothesis

We model a concept in a testable problem hypothesis as a problem hypothesis entity and map the entity to a database entity.

Step 3.1: Model a concept in a problem hypothesis as an entity

The concept in the elicited testable problem hypothesis is modeled as a problem hypothesis entity using the entity-relationship model [61] [62]. A problem hypothesis entity consists of attributes, constraints, and relationships. An *attribute* is a property of an entity having measurable value. A *constraint* is a condition restricting the value or state of a problem hypothesis. A *relationship* shows other entities associated with this entity.

For example, the *Minimum balance of an Account below a threshold*$_{SourcePH}$ in PH1 is modeled as a *Account*$_{SourcePHE}$, having *balance*$_{PHEattribute}$, *minimum balance, less than threshold*$_{PHEconstraint}$, and a *Loan*$_{PHErelationship}$. Similarly, the *Status of a loan unpayable for the loan duration*$_{TargetPH}$ is modeled as *Loan*$_{TargetPHE}$ having *status*$_{PHEattribute}$, *duration*$_{PHEconstraint}$, and *Account*$_{PHErelationship}$, as shown in Fig. 5.5(b).

Step 3.2: Map a problem hypothesis entity to a database entity

The attribute of the problem hypothesis entity (PHE) may manually be mapped to attributes of the database entity (DE), considering the constraints and relationships of the PHE. To guide systematic mapping, we identified five types of mappings from a PHE to a DE, as shown in Fig. 5.6.

The first type of mapping is from a *target PHE* to a *target DE*. The attribute and constraints of the target PHE are mapped to those of the target DE, where the attribute of the target DE becomes a target or classification label. For example, *loan status*$_{PHEattribute}$ of *Loan*$_{TargetPHE}$ in Fig. 5.5(b) is mapped to the *Loan*$_{DE}$ and *status*$_{DEattribute}$.

The second type is from a *source PHE* to a *target DE*. Here, we can notice that the mapped entity is the same target DE in the first type of mapping, but the attribute of a target DE is not a target label.

The third type is from a *source PHE* to a *source DE*, where the source DE is directly associated with the target DE in the database schema. The attribute and constraints of the source PHE are mapped to those of source DE. The relationship of a source DE is the name of the target DE and vice versa.

The fourth type is from a *source PHE* to a *source DE* similar to the third type, but the source DE is indirectly related to the target DE. In other words, there are other DEs between the source DE and the target DE. For example, for the *balance*$_{PHEattribute}$ of *Account*$_{SourcePHE}$ in Fig. 5.5(b), we first select the Account entity of the database schema and

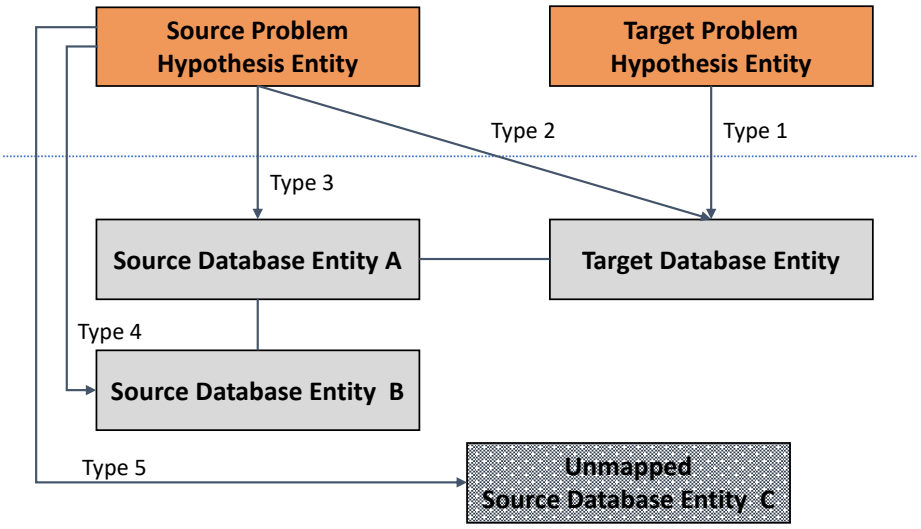


Figure 5.6. Mapping types from a problem hypothesis entity into a database entity

check whether some attributes of the Account semantically match the $balance_{PHEattribute}$. If we could not find a relevant attribute of the Account, then we check the subsequent entities. While iterating domain entities, we could see a *balance* attribute of the Transaction entity, representing a balance after the banking transaction. So, we mapped $Account_{PHE}$ to $Transaction_{DE}$ and $balance_{PHEattribute}$ to $balance_{DEattribute}$. As $Transaction_{DE}$ is not directly related with $Loan_{DE}$, we identify $Account_{DE}$ that is related with both $Loan_{DE}$ and $Transaction_{DE}$.

This mapping may be streamlined with the Dregon prototype tool in Fig. 5.7. The tool first reads the Financial database schema and shows the concerned entity and attributes. Each entity may be selected and checked whether the entity's attributes are similar to that of the problem hypothesis entity.

5.4.4 Step 4: Extract and Transform an ML Dataset

This step extracts a dataset using the identified database entity, data features, and constraints. We then merge each dataset corresponding to the problem hypothesis and transform the integrated dataset for ML processing.

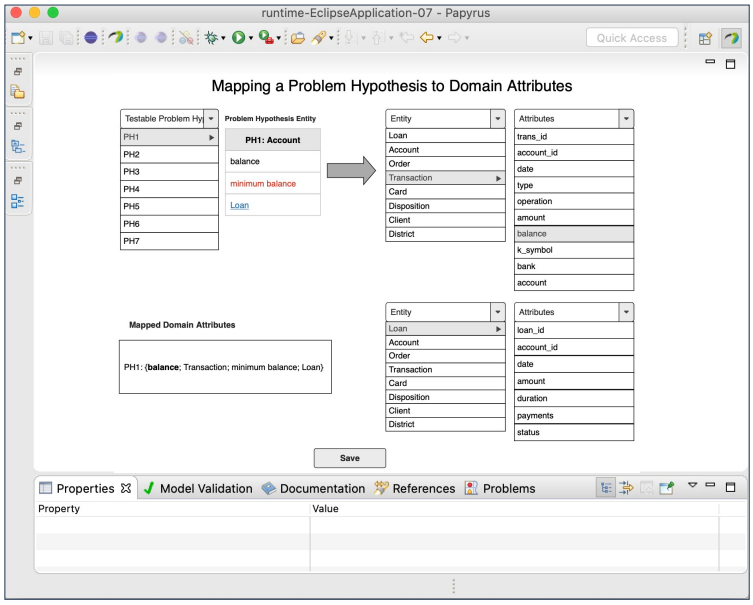


Figure 5.7. Mapping a problem hypothesis entity into a database entity

Step 4.1: Extract and merge an ML dataset

The identified database entities corresponding to the source and target PHE are used to make a database query, as shown in Fig. 5.5(b). For example, the data of the *Minimum balance of an Account below a threshold*$_{SourcePH}$ in PH1 can be extracted using the identified *balance*$_{DEattribute}$, *minimum balance* $<$ *threshold* $_{DEconstraint}$, and *Loan, Account*$_{DErelationship}$ in *Transaction*$_{DE}$. SQL group function, min() may be used to select *minimum balance* $_{DEconstraint}$. Also, to apply for the relationship *Loan, Account*$_{DErelationship}$, we need to identify a primary key and a foreign key relationship between *Loan*$_{DE}$ and *Transaction*$_{DE}$, which leads to identify *Account*$_{DE}$. The *loan duration*$_{DEconstraint}$ of *Loan*$_{DE}$ is also applied, as shown in Fig. 5.5(b).

The data of *Unpaid Loan for the loan duration*$_{TargetPH}$ can be extracted using the following SQL code, which needs to join Loan and Account tables.

```
SELECT l.loan_id, l.status
FROM Loan l, Account a
WHERE l.account_id = a.account_id
```

Each dataset for the hypothesized business events is extracted, tentatively stored in the database, and then integrated into one dataset. Those datasets are then merged into one dataset based on the loan status, as shown in Fig. 5.8.

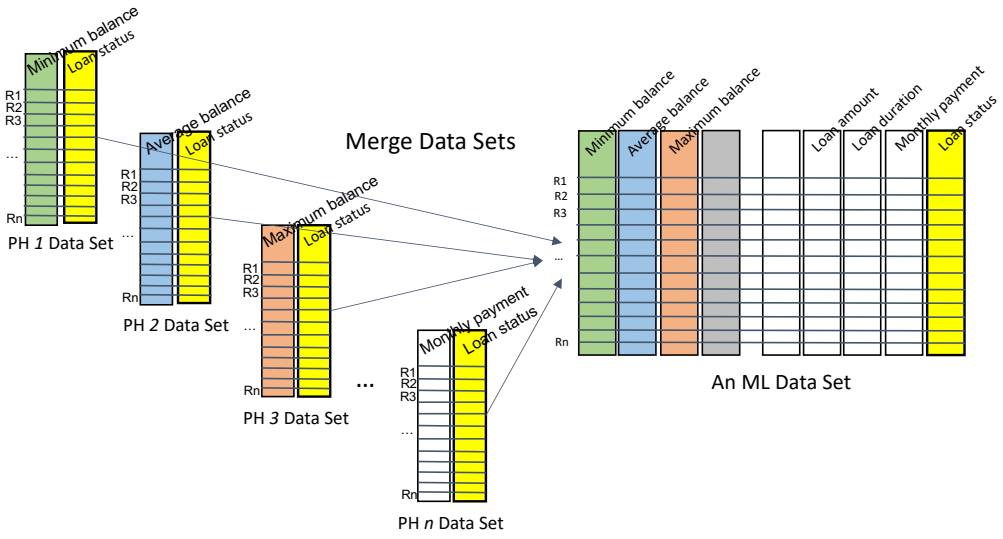


Figure 5.8. Merging partial datasets into an ML dataset

Step 4.2: Transform an ML dataset

The merged dataset may need to be preprocessed for some features, including filling in missing values, scaling feature values, converting categorical data to a numeric value, and others. The clean data are then entered into ML models. For example, we scaled the features of the integrated dataset using the data normalization method. We also used a one-hot encoding on the transaction type, mode, symbol features, and other nominal features.

5.5 Experimental Results

We performed three experiments to see the strength and weakness of the Dregon approach. In experiments 1 and 2, we prepared the ML dataset without the proposed approach, assuming all the features in the Financial database are potential banking events that could cause the unpaid loan. In experiment 3, we prepared the dataset to validate banking events towards the outstanding loan, following the Dregon process.

5.5.1 Experiment 1

For this experiment, we assumed all the attributes, except the table identifiers, of the entities in the Financial database schema as potential events causing unpaid loans without a goal and problem analysis. We selected the loan status as a target feature. The prepared ML dataset included 72 features with some transformation methods, such as hot encoding for the nominal features and 449,736 records based on the transaction id. The significant records are due to the *join* operation among Account, Transaction, and Payment Order tables.

As some ML algorithms, such as Gradient Boosting Tree, provide feature importance [65, 56], we analyzed whether some features could be important factors towards the unpaid loan. Fig. 5.9 shows some crucial features predicted by the XGBoost model. However, it was not easy to get some ideas about whether the loan granted year and the credit card type, e.g., classic, has some relationships towards the unpaid loan.

One critical issue of this approach is that one ML model, e.g., XGBoost, showed different prediction results for the same loan instance. For example, different transaction records, having the same Loan ID 233, showed different loan prediction results (i.e., paid and unpaid), which made the dataset poor in identifying a banking event for the unpaid loan.

Another issue is that this experiment included some unlikely features, such as no. of committed crimes '95. It was not easy to understand whether the no. of committed crimes

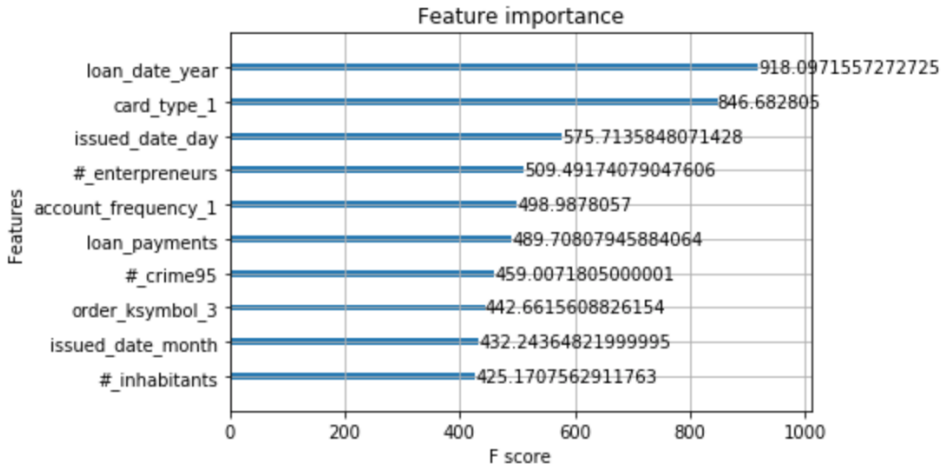


Figure 5.9. Top important features in experiment 1

is related to clients' loan payments as the feature is highly related to the community behavior, not a client's banking behavior.

5.5.2 Experiment 2

In experiment 2, we also assumed all the attributes in the database as potential problems without considering steps 1 and 2 of the Dregon process. However, we prepared the ML dataset centered on the loan ID to prevent duplicate data values of a loan record, unlike experiment 1. We used SQL group functions, such as Sum, Min, and Avg, to select records for the one-to-many relationships, for example, the relationship between Account and Transactions. The final dataset contained 682 records, including 72 features. Fig. 5.10 shows important features the Random Forest model provided, although it was challenging to understand whether they positively contribute to the loan status.

A critical issue of this approach is that the prepared dataset did not consider some boundary constraint of the loan. For example, the loan duration of loan ID 1 is two years from 1993. However, the dataset included records of 1996 and 1997, which could violate the time order constraint between the source and target problem hypothesis and then give incorrect predictions leading to ineffective problem validation. The constraint of time order

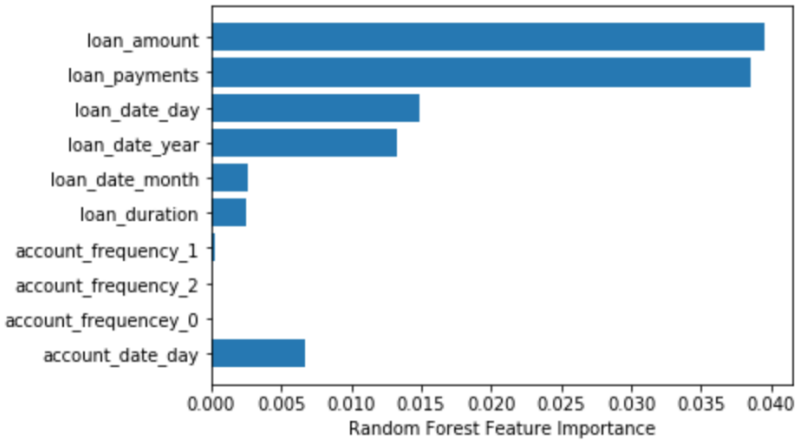


Figure 5.10. Top important features in experiment 2

is essential in identifying a cause and effect relationship between banking events, but difficult to enforce this constraint in this experiment without some mechanisms.

5.5.3 Experiment 3

In this experiment 3, the Dregon approach was applied to prepare an ML dataset to validate business events behind the unpaid loan. The banking events were hypothesized as four groups, including Loan, Account, Transaction, and Client, as shown in Fig. 5.5(a).

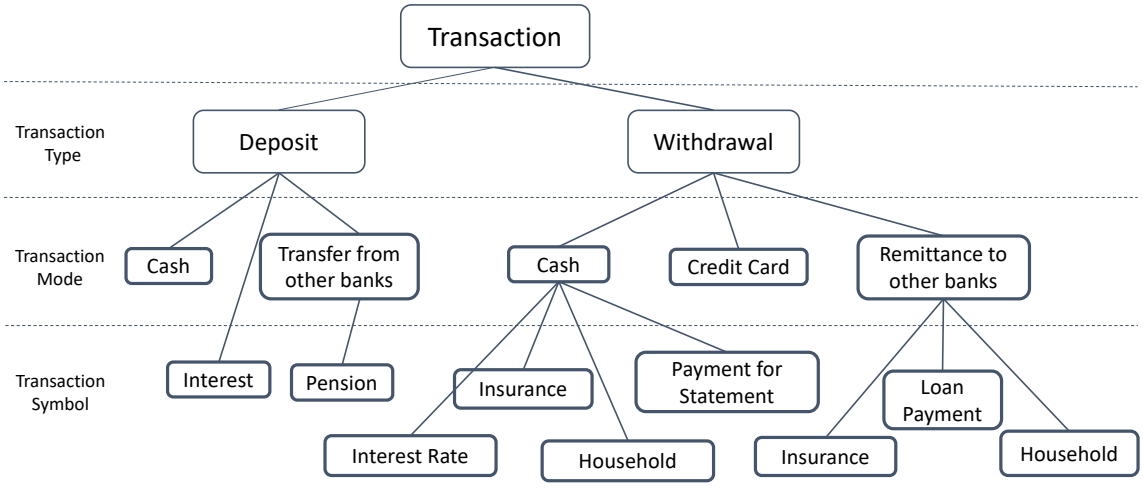


Figure 5.11. Deposit and withdrawal classification in Transaction

While preparing a dataset, we could discover that the balance depends on the transaction *type* (deposit or withdrawal), *operation* (mode of a transaction), and *symbol* (characterization of the transaction) features in the Transaction entity. We could organize the structure of deposit and withdrawal transactions and analyze these features, as shown in Fig. 5.11, to get insights into transaction impact [47]. We then hypothesized deposit and withdrawal of transactions leading to the balance change. These category features would be hot-encoded in a usual ML approach, like in experiments 1 and 2.

Based on the modeled problem hypotheses, six hundred eighty-two (682) loan records with 25 features were prepared. We then ran ML models to predict whether each loan could be paid off or not. Fig. 5.12 shows performance results for some ML models. The accuracy of ML models was overall satisfactory, and XGBoost gave the highest accuracy (0.91). We could also identify significant features regarding the unpaid loan.

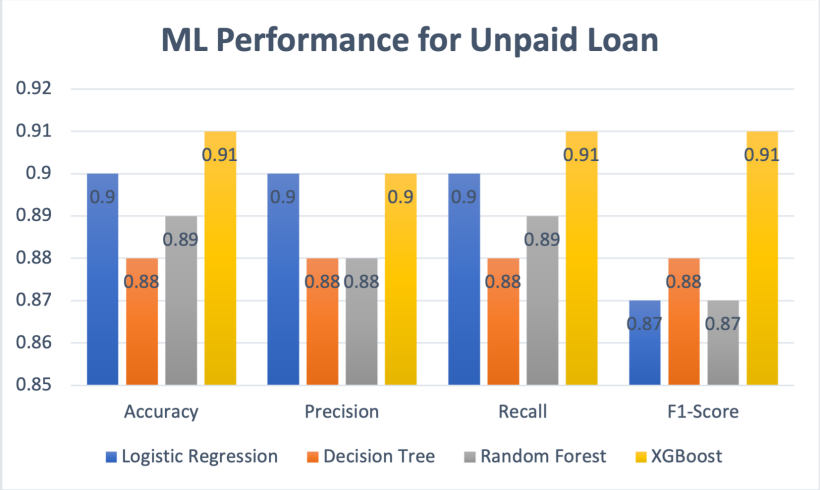


Figure 5.12. ML performance using the prepared dataset in experiment 3

The trade-off analysis for the experiments is shown in Table 5.1. Experiment 1 is easier to prepare an ML dataset assuming all the features as problem hypotheses. However, its results are not easy to understand, even giving different predictions for the same loan case, thus not trustworthy. Experiment 2 shows more sensible results than experiment 1 but still

challenging to understand the relationships among the problem hypotheses and a target label. It needs to apply some systematic process for asserting constraints of time order. Experiment 3 provides more sensible and understandable relationships among the banking events and an unpaid loan with fewer features than experiments 1 and 2. Although experiment 3 may take some time to prepare data, it helps identify potential banking events causing the unpaid loan and get some insights into the hypothesized banking events. In addition, it helps understand some implicit patterns of the data features otherwise overlooked.

Table 5.1. Experiments comparison of data preparation

	Easy to Experiment	Understanding of Banking Domain	Loan-Related Feature Selection	Feature Explanation towards Unpaid Loan	Relationship b/w Problem and Goal
Experiment 1	++	-	-	-	-
Experiment 2	+	+	-	+	-
Experiment 3	+	++	+	++	++

5.6 Discussion and Observation

In constructing a problem hypothesis concerning the unpaid loan, it may not be easy to keep the time constraint between a source and a target problem hypothesis. To prevent the violation of this constraint, we used a date feature and potential banking events together to ensure the time order constraint between a source and a target problem hypothesis.

Some problem hypotheses may not be mapped to data features in the database schema, such as the fifth type mapping in Fig. 5.6, due to unmatched data features or type and cannot be validated. In that case, the data for the problem hypothesis may need to be acquired from external data sources [80].

Our data preparation approach may be applied to identify potential business problems in other business domains, such as logistics, telecommunication, or healthcare. However, as the data preparation in this empirical study is the first attempt and ML performance depends on ML algorithms, their parameters, data characteristics, and others, more empirical studies are needed to show the usefulness of our approach.

Limitations Problem hypotheses are conceived and manually constructed, which tends to be error-prone and ineffective in managing different hypotheses. Some guiding template or tool support may help refine a problem hypothesis into a source and target problem hypothesis and a relationship. A prototype mapping tool partially supports the mapping process between a problem hypothesis entity and a database entity. However, the tool needs more work to automate the presented approach. The process also needs to be fully formalized to define precise semantics.

5.7 Conclusion and Future Work

This chapter has presented a goal-oriented ML data preparation approach to support the validation of a business problem. Starting with modeling business goals, we explored potential business events against goals, modeled the events as a testable problem hypothesis entity, identified data attributes along with constraints and relationships, and built an ML dataset from a source database. Specifically, this chapter presented 1) a domain-independent ontology and a process for guiding the preparation of an ML dataset, 2) a method to capturing potential business events in the context of goals and problems, 3) a modeling method of a problem hypothesis entity to help to determine a testable factor, constraints, and relationships of the captured business events and 4) a mapping method and mapping types from a problem hypothesis entity to a database entity. The experiment, we feel, shows that our approach helps prepare an appropriate ML dataset, enforce time order constraints, and provide traceability from problem hypotheses to data features.

There are several lines of future work. Tool design and support, such as a template, helping to manage a problem hypothesis are needed. Formalization of the mapping process is planned using first-order logic. The development of a fully-fledged tool also would be helpful to automate the mapping between a problem hypothesis entity and a database entity. We also plan to apply the Dregon approach to other domains, such as the public health domain, to see the strength and weaknesses of our work.

CHAPTER 6

VALIDATING BANKING PROBLEMS BEHIND CUSTOMER CHURN

6.1 Introduction

Data Analytics and Machine Learning (ML) technologies benefit from a continuous improvement cycle where large amounts of data are constantly being created. Organizations invest in Big Data and ML projects, but most of these projects are predicted to fail [12, 81]. A study may have suggested a possible reason: the lack of understanding of how to use data analytics to improve business value [13]. This finding clearly shows that stakeholders do not see the end-to-end relationship between important business goals and the emerging Big Data and ML technologies [15][17].

Additionally, some business problems can only be hypothesized as they are difficult to validate using traditional data analysis techniques. For example, applying data analysis on the customer churn dataset [82] during our experiment showed no evidence to suggest that the customers who left the bank had a higher degree of dissatisfaction with many of the service qualities than those loyal customers.

Building on our previous approach, GOMA [75], this chapter [1] proposes *Metis*[2] to support goal-oriented hypotheses and validation of business problems. Three technical contributions are made in this chapter, including 1) an ML-based approach to extracting an actual root cause hidden in the data to validate hypothesized business problems, 2) an ontology that more explicitly and formally describes the relevant modeling concepts related to business goals, problems and ML, and 3) a set of formalized validation rules for reasoning about problem hypothesis validation in a goal-oriented problem model.

[1] This chapter contains material previously published as: ©2021 Springer. Reprinted, with permission, from Supakkul, S., Ahn, R., Junior, R. G., Villarreal, D., Zhao, L., Hill, T., & Chung, L. (2020, September). Validating goal-oriented hypotheses of business problems using machine learning: An exploratory study of customer churn. In International Conference on Big Data (pp. 144-158). Springer, Cham.

[2] A Greek goddess that has been associated with prudence, wisdom, or wise counsel.

The proposed approach is illustrated using a real-world banking customer churn problem, which was adapted from the example used in [75]. In the adopted case study, a retail bank hired a company specializing in data mining to help address the churning problem by using insights from detailed transaction data in a newly installed powerful data warehouse [54, 83]. The company hypothesized potential reasons why the customers were canceling their accounts and validated them with descriptive insights mined using a data classification technique. Since the actual dataset used by the consulting company was not available, we used a publicly available bank customer churn dataset [82] and reversed engineer to update the business problem hypotheses so that they are consistent with the dataset used. We use this example to demonstrate that *Metis* could be used to provide traceability between business problems and an ML solution, which can also reveal insights about the root cause of a problem that may be difficult to discover using data analysis.

6.2 Metis: A Goal-Oriented Problem Hypothesis Validation Method using Machine Learning

To be able to hypothesize business problems and subsequently data features needed for developing an ML model, a good understanding of concepts in the domain in question is required. In this section, we first present an example banking domain-specific ontology that underlies the customer churn problem that we use as a running example. We then describe the *Metis* domain-independent ontology to support the modeling and validation of problem hypotheses in *Metis*; Finally, we describe and illustrate the *Metis* process.

6.2.1 Domain-Specific Banking Ontology

This section describes an example of domain-specific ontology for the banking example, which can vary significantly depending on different organizations and processes. Fig. 6.1 shows a typical set of banking concepts and their relationships. It is worth mentioning that

this domain-specific ontology supports the understanding of the banking domain, and it does not represent a schema or model related to database design.

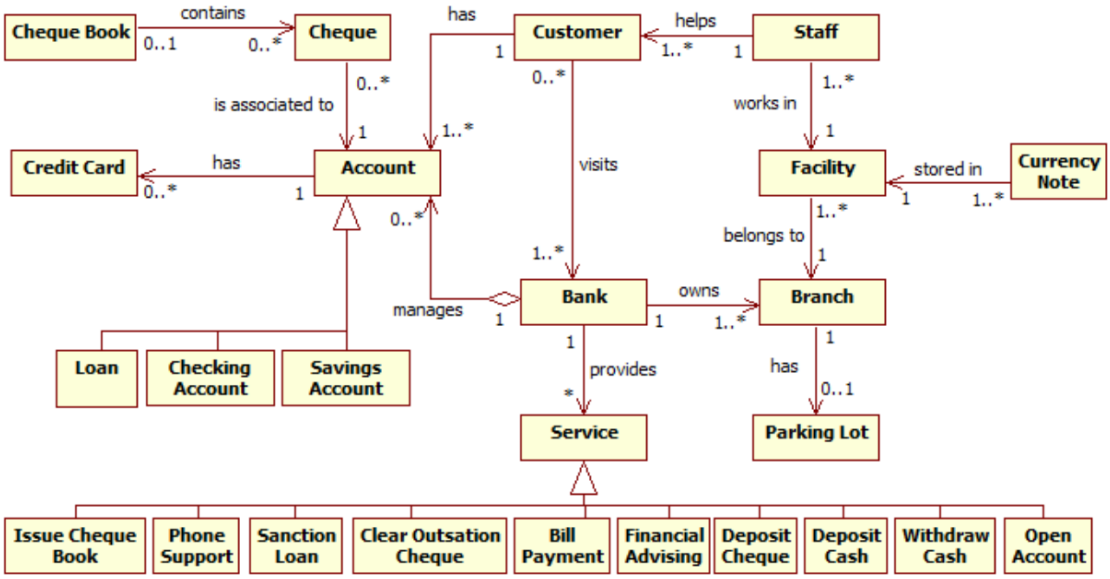


Figure 6.1. Banking domain-specific ontology diagram

Some of the ontological concepts are briefly described here as examples. Banks provide numerous services, such as financial advising and cash withdrawal. It is crucial to study the qualitative aspect of these services in order to have a clear understanding of customer satisfaction. For example, customers may feel that there is not enough parking space, a lack of pleasant ambiance, no comfortable seating arrangements, and lack of immediate attention. For the customer churn problem in the running example in this chapter, the quality aspect of both facility and service-related concepts are essential to generate hypotheses about the customer churn problems.

6.2.2 The Metis Ontology

While modeling the mapping between a goal-oriented ontology and an ML-based ontology, completeness and soundness are two major concerns. To completely and formally address

these concerns for the *Metis* method, the following subsections describe the modeling concepts and the semantic reasoning formalization for the *Metis* ontology.

Modeling Concepts

A complete set of concepts and their relationships can be found in Fig. 6.2. We explicitly represented essential concepts such as Problem, Hypothesis, and Machine Learning Model to avoid omissions while mapping Goal-Orientation and ML. In addition, the ontology also comprehends concepts related to Big Data and Big Queries, and Features are derived from modeling concepts from a domain-dependent ontology. *Metis* is a domain-independent ontology that can be applied to a variety of domains. Section 6.2.1 describes a banking domain-specific ontology example.

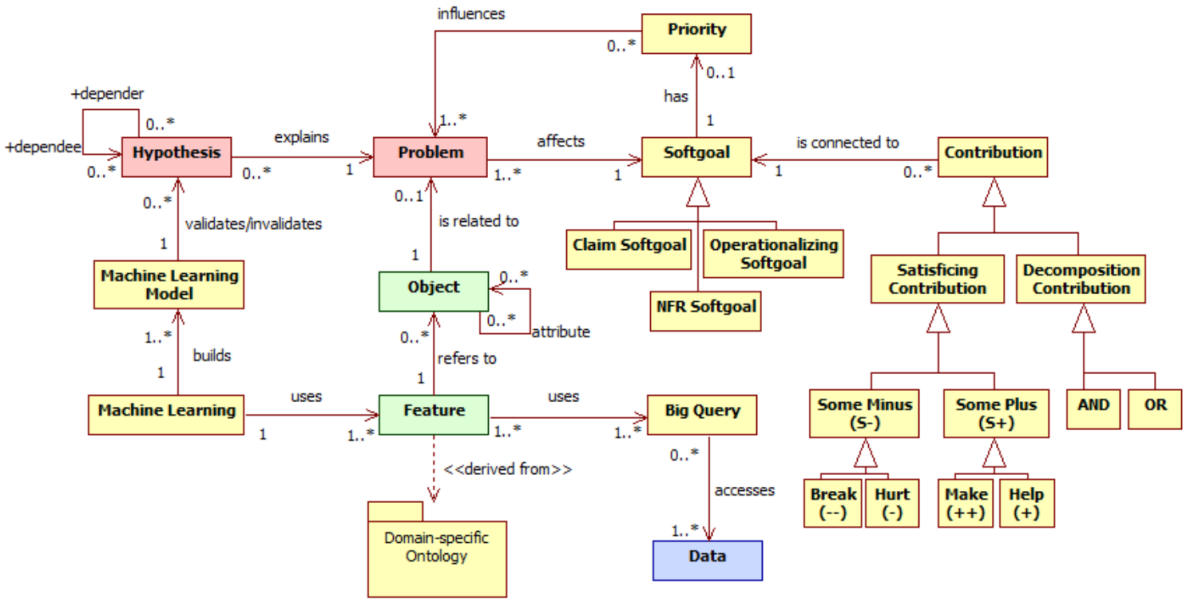


Figure 6.2. Metis domain-independent ontology diagram

An acceptable representation of hypotheses and problems can be generated, but ultimately, we want to determine whether we can validate these hypotheses for the problems in consideration. In this context, ML is used to build models to identify the importance

of features such as *Immediate Attention*. Using the relevant features makes it possible to establish how to validate or invalidate hypotheses. For instance, in Fig. 6.4 we hypothesize that *Lack of immediate attention* has a S+/S- contribution to the problem *Poor Service*, which in turn contributes to *Customer Churn*.

6.2.3 The Metis Process

The *Metis* process consists of four steps: 1) *Model business goals and problems*, 2) *Acquire data*, 3) *Detect feature importance*, and 4) *Validate hypotheses of business problems* as shown in Fig. 6.3.

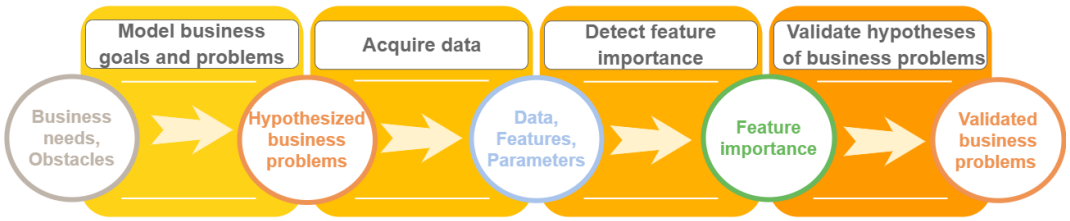


Figure 6.3. The *Metis* process for validating goal-oriented hypotheses of business problems

Step 1: Model business goals and problems explicitly captures stakeholders' needs and obstacles as goals and problems using a goal-oriented modeling approach [29][37], where potential problems are posed as problem hypotheses to be validated. The outputs from this step are problem hypotheses in the context of business goals. *Step 2: Acquire data* derives data features from the business problems hypotheses to acquire the necessary data from external and/or internal sources, for instance, using a customer survey or Big Data Spark SQL if the data are already available online. *Step 3: Detect feature importance* uses ML to learn patterns in the data to identify how problems are collectively associated with the data features. In addition, the output from this step includes Feature Importance that determines the degree of each feature contributing to a problem. The final step, *Step 4: Validate hypotheses of business problems* uses the Feature Importance to validate the problem hypotheses modeled during Step 1.

Step 1: Model business goals and problems

In this step, important business or stakeholders' needs are explicitly captured as Softgoals that can be further refined using AND or OR decomposition [59]. Using Fig. 6.4 as an example, at the highest organizational level, *Increased profitability* is a Softgoal to be achieved, which is refined using an AND decomposition to *Increased revenue* and *Increased profit margin* sub-goals, where the former is to be operationalized by *Increase customer base* strategic level goal. *Increase customer base* is then further AND-decomposed to more specific operationalizing goals of *Retain existing customers* and *Acquire new customers*.

Each lowest level goal is used as the context to identify potential problems that could hinder the goal achievement. The validity of each problem may be unknown at this point. Therefore, each problem is considered a target problem hypothesis to be validated by data. Like the goal refinement, each problem hypothesis may be further refined or realized by more specific problem hypotheses until they are low-level enough to identify the data features needed for data analysis or ML.

In this example, *Customer Churn* is a problem hypothesis that could *BREAK* (--) the *Retain existing customers* goal. *Customer Churn* is further refined using an OR-decomposition to *Poor Facility* or *Poor Service* sub-problem hypotheses, which are used to identify potential causing problem hypotheses. *Poor Facility* is hypothesized to be caused by *Long distance to a residence*, *Lack of pleasant ambiance*, or other causes. Since each potential cause has not been validated whether it is indeed a contributing cause to the problem, the contribution link is labeled as unknown (depicted by a question mark).

Step 2: Acquire data

This step examines the lowest level problem hypotheses to identify data features needed for data analysis. Using Fig. 6.4 as an example, *Long distance to a residence* and *Lack of pleasant ambience* may be used to identify *Distance to a residence* and *Pleasant ambience*

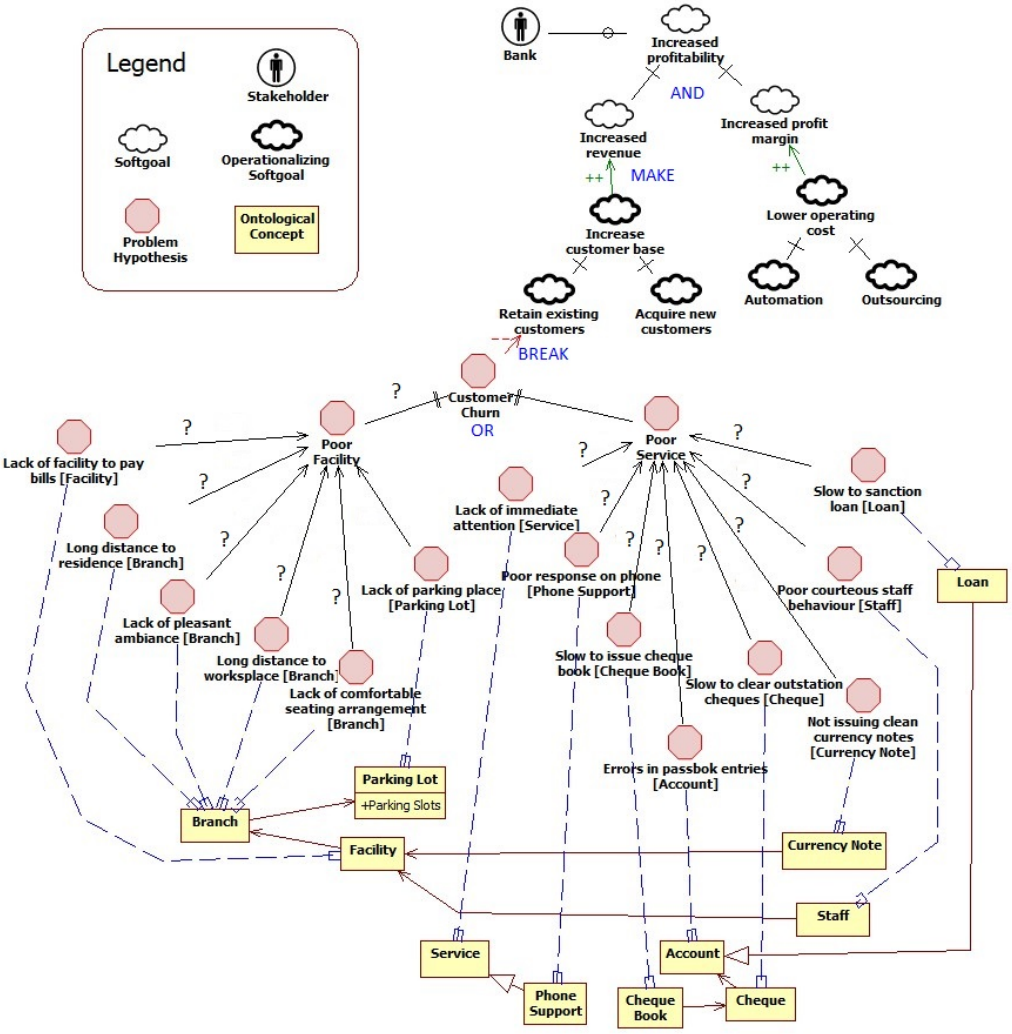


Figure 6.4. Step 1: Model business goals and problems

as the corresponding data features. The identified features are then used to build database queries or Big Data queries if the corresponding data are already available online. Otherwise, the required data features need to be acquired through other means, such as purchasing from a data provider, using a customer survey or generation from online sources [84].

An example of the acquired dataset is given in Fig. 6.5(a), where F_1 - F_5 represent all features and L corresponds to the *Churner* or *Non-churner* indicator associated with the satisfaction scores for F_1 - F_5, provided by individual customers C_1 - C_5. For example, cus-

tomer C_1 expressed dissatisfaction with features F_1 and F_3 with scores of 2 and 3 accordingly. On the other hand, he/she expressed satisfaction with F_2, F_4 ,and F_5 with scores of 8, 7, 8 accordingly. C_1 is noted by label 1 as a *Churner* customer in correlation with the given scores.

Step 3: Detect feature importance

The intuition for using ML is to encode the knowledge about the features hidden in the customer survey data and then decode the knowledge representation to identify which feature is the true cause for the customer churn problem. To encode the feature knowledge, we use a Supervised ML algorithm assuming that an accurate prediction model represents the knowledge about features. To decode the influential features, we use an ML Explainability library [58, 85] that was designed to explain how features contribute to the prediction outcomes.

Referring to Fig. 6.5(b), this step splits the dataset into training and testing datasets. All features F_1 - F_5 and label L are processed by one or more Supervised ML algorithm to obtain the most desirable prediction model M_p. To determine whether M_p has been sufficiently trained to recognize the general patterns in the training dataset, it is measured on how accurately it can predict label L in the testing dataset. The accuracy is represented by an accuracy metric A_1, which is based on the differences between predicted label L' and actual label L, where L' is generated from F_1 - F_5 in the testing dataset.

Once an accurate model M_p is obtained, it is processed by an ML Explainability algorithm to produce an Explainability model M_e, which is in turn used to detect feature importance I_1 - I_5, where I_1 contains two pieces of information: sign and weight of the contribution F_1 makes towards the label L' as predicted M_p. The sign of the value indicates whether the corresponding feature helps or hurts towards the predicted label, while the weight represents the amount of influence the feature has. Similarly, I_2 and I_3 represent the feature importance of F_2 and F_3, respectively. By having the highest value among all feature importance values,

F_1 is considered the most influential feature, followed by F_2 and F_3 in the context of the testing dataset.

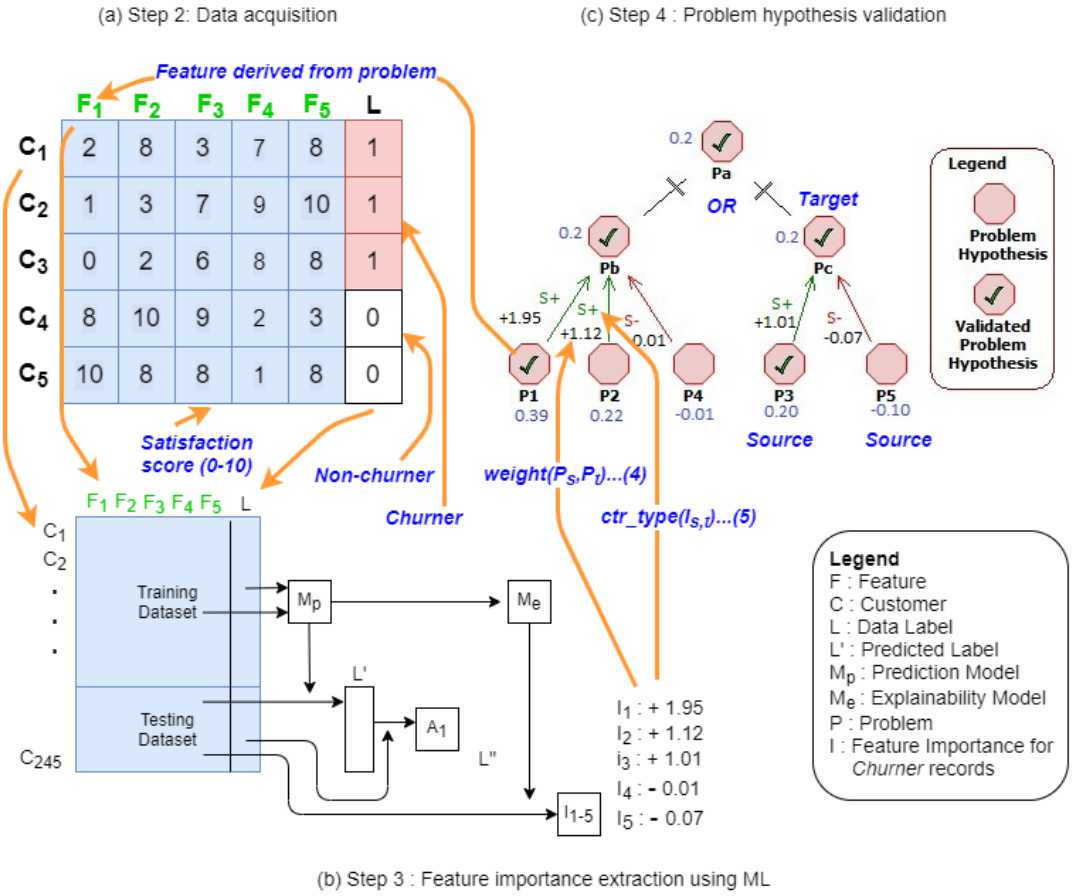


Figure 6.5. Step 2, 3, 4 of the *Metis* process

Step 4: Validate hypotheses of business problems

Referring to Fig. 6.5(c), this step uses the feature importance values produced by the Explainability model M_e to validate problem hypotheses in the goal-problem model created in step 1, one parent-child problem set at a time in a bottom-up approach, using the quantitative and qualitative semantic reasoning formalization, as described in Section 6.2.2.

Using P_b - (P_1, P_2, P_3) parent-child set as an example, the contribution link between each parent-child pair is updated by applying Formula 3.4 and 3.5 against the corresponding

feature importance value where (P_1, P_2, P_3) and Pb are considered *sources* and *target* in the formulas respectively. In this example, the contribution type $ctr\_type(P_1, P_b)$ is assigned with $S+$ by Formula 3.5 with feature importance I_1 with value +1.95 as a function parameter. I_1 is used since the corresponding F_1 was defined based on problem P_1 in step 2. To complete the contribution update, the weight of contribution is assigned with 1.95 by Formula 3.4. Other contribution links with the same parent are updated in a similar fashion. Then, P_1 is selected among P_1, P_2 and P_3 by Formula 3.7 to be a validated problem hypothesis since it is the most influential cause for problem P_b. After P_1 is quantitatively selected based on scores, P_b is qualitatively validated by Formula 3.8. Then, P_a can be qualitatively validated by Formula 3.2.

6.3 Experiment and Results

Analyzing customer feedback information may be beneficial to discerning customer satisfaction for the quality of important services. To this end, a publicly available dataset [82] acquired by Step 2 in the *Metis* process is analyzed in this section. This dataset contains typical customer information such as age and occupation. In addition, the dataset contains feedback information regarding certain banking service-related features (e.g., Immediate Attention) and facility-related features (e.g., Pleasant Ambiance). A customer can score each of these features from 0 to 10 (least to most satisfied). In this context, scores of 4 or less are used to describe some degree of dissatisfaction, an assumption that something might go wrong in a business operation, i.e., problem hypotheses. The next section describes an analysis of these problem hypotheses.

6.3.1 Dataset Analysis

Some examples of problem hypotheses are shown in Fig. 6.4, which includes *Long distance to a residence*, *Lack of immediate attention*, and *Lack of pleasant ambiance*. Fig. 6.6 shows

the customer dissatisfaction for these three features out of the 20 available features. Of the total of customers that believe there is a long distance to their residence, 35% deserted the bank (churner), and 65% remained loyal (non-churner). Assessing this feature by occupation, notice that most unsatisfied customers are from professional occupations, followed by private and government service. Together, these three occupations represent 75% of customers unsatisfied with distance from the residence.

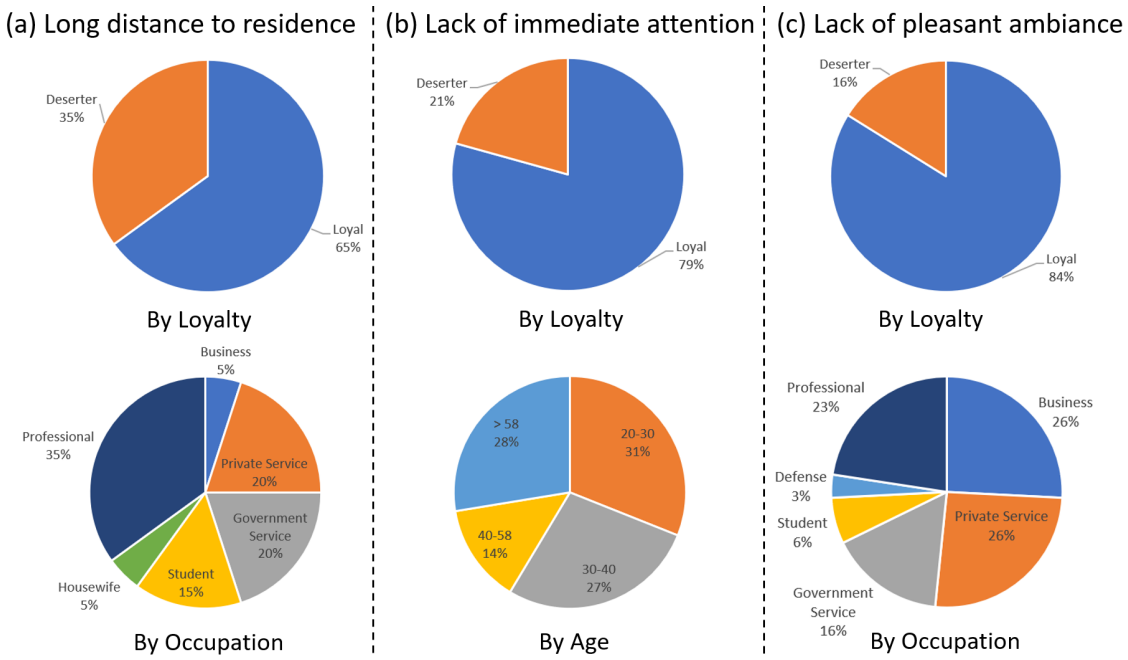


Figure 6.6. Analysis of customer dissatisfaction (score less than 5 in a 0 to 10 scale) for the features distance from residence, immediate attention, and pleasant ambiance

More than half of the customers who identified a lack of immediate attention are young customers (40 years old or younger). Analyzing pleasant ambiance by occupation, we can see that customers from the business occupation and the private service complained the most. In an overall assessment for customer dissatisfaction by loyalty, it is possible to notice that most customers remained loyal regardless of the problem hypotheses under consideration. Even though we are able to extract insights from the dataset, ultimately, there is no evidence

of why customers deserted. For this purpose, Section 6.3.2 demonstrates results of using ML that can potentially provide some evidence.

6.3.2 Prediction Models

To encode and represent knowledge about feature contributions using Supervised ML, we experimented with several ML algorithms, including Linear Regression, Support Vector Machine, Decision Tree, Random Forest, and XGBoost Classifier. XGBoost showed the highest accuracy rate in our experiment. Due to space limitations, only the results from XGBoost are discussed in this section.

The ML segments of the investigation were conducted using Python language and scikit-learn open-source ML libraries [86]. The dataset used for the experiment was a public banking customer churn dataset [82]. After data cleansing, 67% of the data (164 records) were used for model training and 33% (81 records) for testing. Data features used included the customer responses to the survey questions, such as *Pleasant Ambiance*, *Comfortable Seating*, *Immediate Attention*, *Good Response On Phone* and others, on the scale of 0-10. We excluded customer information, such as age and occupation, used separately for data analysis as reported in Section 6.3.1 The resulting prediction model showed an accuracy of 84% (F1 score) on the test dataset, which was better than other ML algorithms in our experiments. The modest accuracy rate was probably due to the small and highly unbalanced dataset that required a data pre-processing step that further reduced the dataset size.

6.3.3 Explainability Model

To extract feature contribution information from the resulting prediction model, we used SHAP (SHapley Additive exPlanations) [58]. This Explainability library uses a game-theoretic approach to explain the output of many ML models. It connects optimal credit

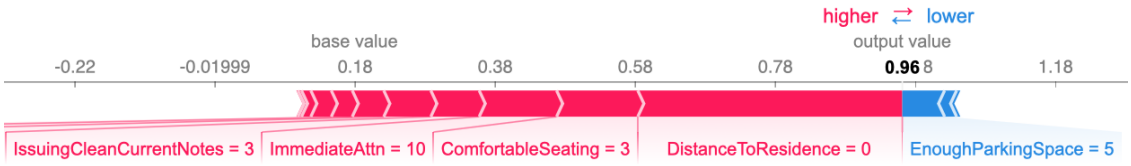


Figure 6.7. Features importance for one churner's responses

allocation with local explanations using the classical Shapley values from game theory and their related extensions.

Fig. 6.7 is a Force Plot produced from a SHAP model (M_p in Fig. 6.5(b)) created from the most accurate XGBoost prediction model (M_e in Fig. 6.5(b)). It gives a visual representation of the influence each feature has on the final output value of 0.96. In this plot, the base value of 0.18 is the average prediction value without any influences from the features, while the output value of 0.96 is the output from the prediction model, where 1 represents a churner customer. The effects of features are represented by the direction towards the output value and width of the corresponding arrow blocks. Here, *DistanceToResidence* feature has the most influence in increasing the output value away from the base value towards the final output value, which is consistent with the score of 0 (least satisfaction) given by the customer. On the other hand, *EnoughParkingSpace* has the most influence in the opposite direction, decreasing the value away from the final output value, which seems consistent with the satisfaction score of 5 (neutral satisfaction) given by the customer. It is interesting to note that *ImmediateAttn* with the value of 10 (most satisfaction) was seen as an influence towards the customer's churner decision. SHAP does explain this counter-intuitive result.

Fig. 6.8(a) plots individual SHAP values for all features and all churner customers. Each dot represents a SHAP value that a feature has in support of increasing the output value towards 1 (Churner label in Fig. 6.5(a)). Visually, it is clear that *DistanceToResidence* has higher positive SHAP values than other features. For this experiment, the more positive SHAP values a feature has, the more influence it has on the prediction outcome. This is supported by Fig. 6.8(b) where *DistanceToResidence* has the highest total(sum) SHAP value.

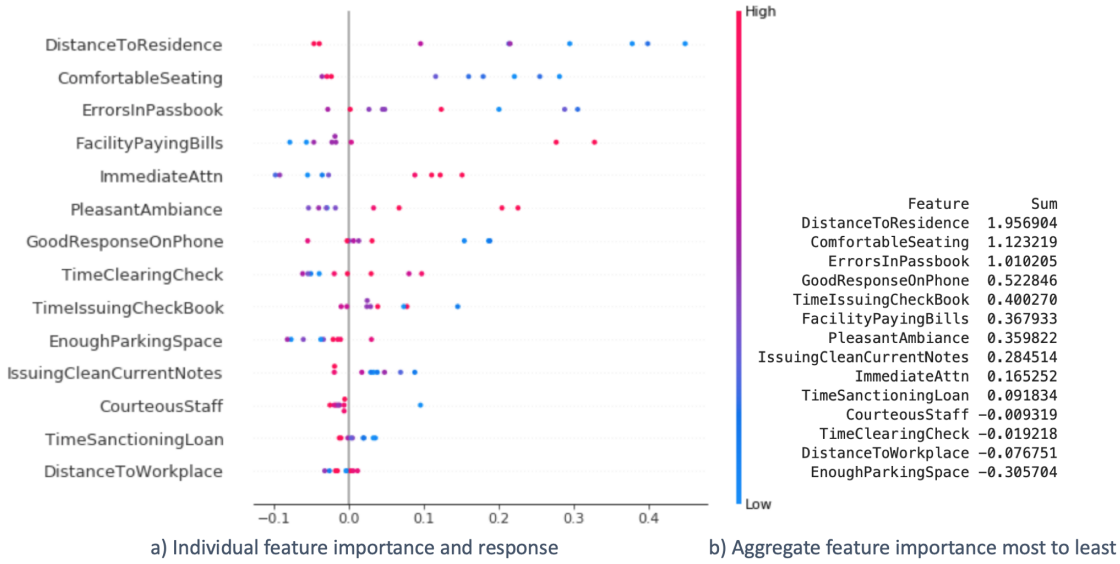


Figure 6.8. Feature importance for all churners' responses

6.3.4 Validating Problem Hypotheses

By following Step 4 of the *Metis* process (Section 6.2.3), we applied Formula 3.4 and 3.5 against the sum SHAP value for the respective feature (see Fig. 6.8(b)), which led to the validation of *Long distance to a residence* problem hypothesis against other features having Poor Facility as the common parent problem hypothesis. Then, Formula 3.8 was applied to validate Poor Service problem hypothesis. Subsequently, Formula 3.2 was applied to validate *Customer Churn* problem hypothesis. The resulting goal-problem model is shown in Fig. 6.9, with check marks to reflect the validation status.

6.4 Related Work and Discussion

We believe this initial work is one of the first to propose an end-to-end, explicit and formal approach that provides traceability between business goals and ML. Most data mining and ML projects in practice are often based on the informal identification of low-level problems [54] that may not have clear relationships with higher-level goals. *Metis* allows ML solutions to be traceable to business at the highest level of business goals and related problems.

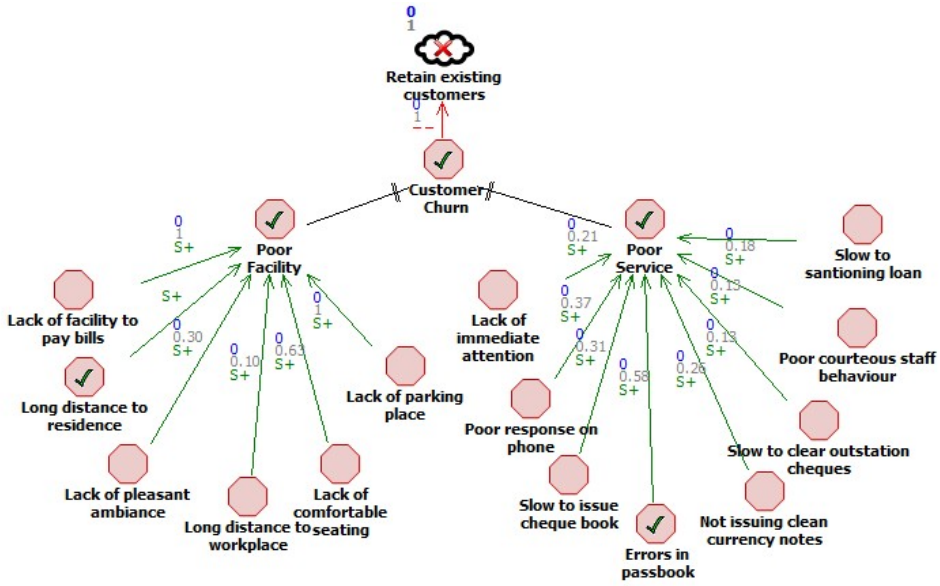


Figure 6.9. Validated customer churn problems

Using data to validate goal-oriented models has been proposed in [40] using questionnaires and statistical hypothesis testing to validate different model elements (e.g., actors, goals, resources) and their relationships (e.g., depends, make, hurt). The statistical method is widely accepted but has been criticized for being difficult to understand [87] and impractical to find evidence in the real world for some hypotheses to test the null hypothesis [88]. This is especially true in the data-rich Big Data environment, where it is difficult to find evidence for both hypothesis and null hypothesis in the available business data. ML allows organizations to utilize the existing data for hypothesis validation that is grounded by the model's accuracy.

Threats to Validity and Limitations.

Regarding threats to internal validity, the dataset used in the experiment was highly relevant to the customer churn problem, but it was a small dataset (i.e., 245 records), leading to biased results. Training and testing data were randomly selected and tested with stratification to reduce this bias. We also ran several ML algorithms but got similar results. For threats to external validity, as we only applied our approach to a customer churn case, the approach

may be too early to be generalized. More experimentation for different domains and datasets is needed.

This chapter has presented a promising initial result with some limitations, including 1) inter-feature AND and OR relationships are not currently supported, 2) it is currently unclear whether the result would be consistent across other ML algorithms and model explainability libraries.

6.5 Conclusions

This chapter has presented *Metis*, a novel approach that uses ML to validate hypotheses of business problems that are captured in the context of business goals. *Metis* uses Supervised ML and Model Explainability algorithms to detect feature importance information from the data. Our initial experiment results showed that *Metis* was able to catch the most influential problem root cause when it was not apparent through data analysis. The most influential root cause was then used to validate higher-level problem hypotheses using the provided formalization.

Future work to address the identified threats to validity and limitations include

1. conducting additional experiments with larger datasets,
2. testing with additional ML algorithms and explainability libraries,
3. investigating solutions for encoding AND/OR relationships in the datasets for model training or exploring ML algorithms internally to extract the relationships if captured by the algorithms.

CHAPTER 7

VALIDATING POTENTIAL PHENOMENA CAUSING THE OCCURRENCE OF WEST NILE VIRUS

7.1 Introduction

Validating the right business problems hindering stakeholders' goals during the requirements engineering process is often more critical than developing solutions. This step helps define system boundaries to develop in the early phase of requirements engineering [2, 89]. If the right problems are identified and solved first, a business can save precious time, cost, and effort to deal with essential problems [3, 20, 12]. Otherwise, problems cannot be solved. At best, harmful or useless solutions may be developed, leading to unintended consequences. For example, the West Nile virus transmitted to humans by infected mosquitoes could cause critical health problems to people in one city. Suppose airborne pesticides to control mosquitoes are sprayed in less critical or wrong locations predicted by an information system. In that case, it could cause significant health problems to some citizens, especially for the elderly.

However, validating elicited business problems hidden in Big data are frequently challenging for business organizations and requirements engineers due to a lack of a systematic methodology [17, 18]. Omitted, overlooked, or unidentified problems due to lack of validation methods frequently lead to a system that is not useful enough to solve important business problems or even is required to redevelop, spending valuable business resources [11, 12, 48].

This chapter presents the improved Metis framework in a goal-oriented and Machine Learning-based approach to help requirements engineers validate elicited business problems and then build the right solutions, such as software architecture and detail design, for the validated problem.

Three technical contributions are made in this paper: 1. The refined Metis ontology, including essential modeling concepts and relationships among those concepts, is presented.

2. A template for a problem hypothesis is presented to help elaborate a problem hypothesis.
3. A problem hypothesis interdependency graph is shown to help visualize and reason about the impacts among problem hypotheses.

We suppose one city offers citizens public services, such as pest controls, parks and recreational services, and water supply. After the first human cases of West Nile virus (WNV) were reported in the city a few years ago, city officials in the health department have monitored occurrences of WNV and tried to control the virus occurrence, such as by spraying pesticides. The city officials want to minimize the spread of the WNV disease by controlling mosquitoes but are unsure precisely what specific phenomena are behind this problem and which ones are valid. This paper applies the proposed method to explore and validate problems of high WNV occurrence. Since this is a hypothetical example, we used the data set of WNV available on the Kaggle [1] to show the applicability of the Metis framework.

The rest of this paper is structured as follows. Section 7.2 presents the Metis framework, and Section 7.3 applies the Metis process to the West Nile virus empirical study. Section 7.4 describes related work, and Section 7.5 discusses observations and limitations. Finally, Section 7.6 summarizes the paper.

7.2 The Metis Framework

The Metis framework adopts a goal-oriented and Machine Learning-based approach to analyze and validate the right business problems. The Metis framework includes a domain-independent ontology, semantic reasoning methods, and a series of processes.

7.2.1 The Metis Ontology

The Metis ontology consists of categories of essential concepts, relationships among concepts, and constraints on the concepts and relationships shown in Fig. 7.1, where boxes and arrows

[1] https://www.kaggle.com/c/predict-west-nile-virus

represent the essential concepts and relationships. The added concepts and relationships are shown in green and red colors. The ontology helps the modeling work of building a problem hypothesis, preparing a data set, using ML models, and using feature importance to validate the problem hypotheses. Also, the ontology helps prevent omission and commission errors in bridging gaps between a goal-oriented approach and an ML-based approach.

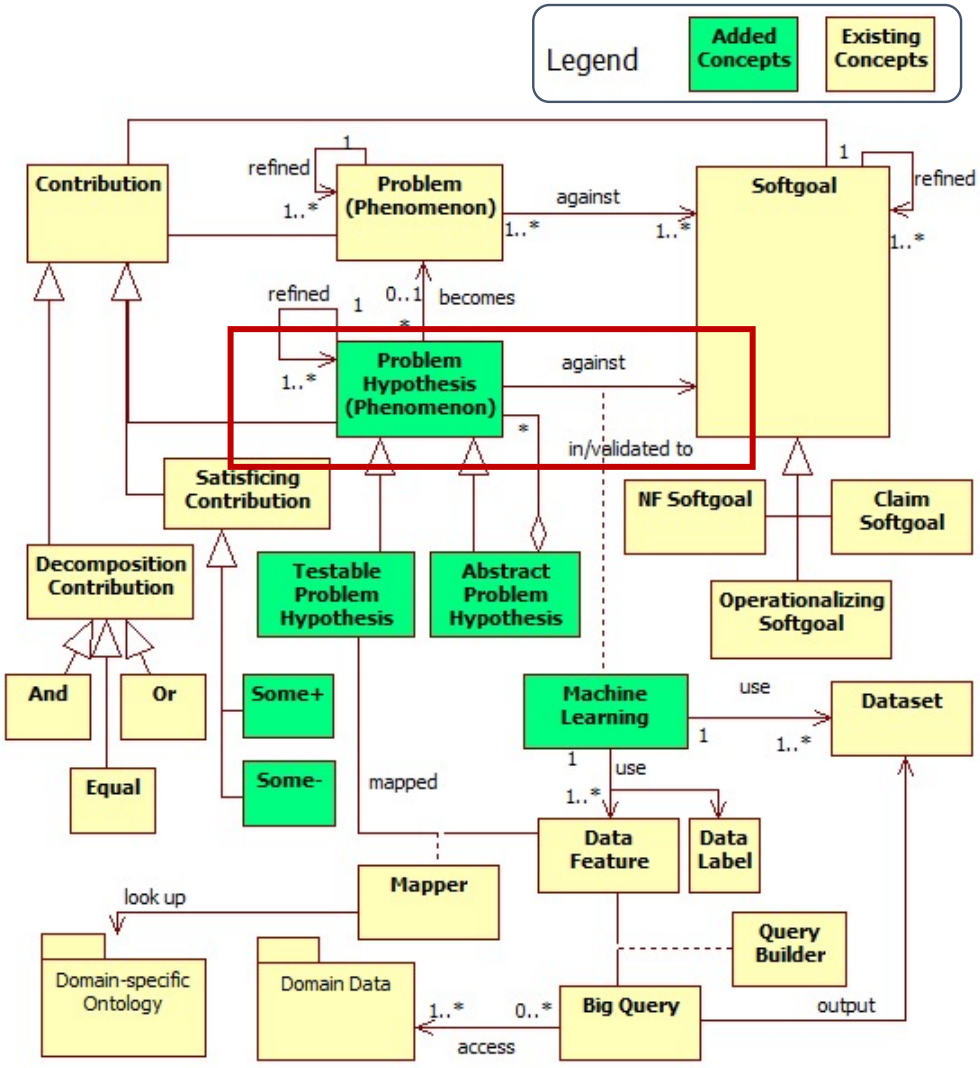


Figure 7.1. The types of essential modeling concepts for validating a problem hypothesis

The categories of essential concepts in the Metis include (Soft-)Goal, (Soft-)Problem, Problem Hypothesis, Data Features, Machine Learning (ML), and others. A (Soft-)Goal

is defined as a goal that may not have a clear-cut criterion. A (Soft-)Problem is defined as a phenomenon against a Goal. A Problem Hypothesis is a hypothesis that we believe a phenomenon is against or some- a Goal. However, we do not know the truth (or label) value of the phenomenon or this proposition. In Metis, the Problem Hypothesis is in/validated using ML. There are two kinds of Problem Hypothesis, an Abstract Problem Hypothesis and a Testable Problem Hypothesis. An Abstract Problem Hypothesis is conceptual, but a Testable Problem Hypothesis is measurable and testable that may be mapped to Data Features in a data source. A more formal definition for a (Soft-)Problem Hypothesis can be defined in a Backus–Naur Form (BNF) in Table 7.1.

Table 7.1. A problem hypothesis in a Baclus-Naur form

```
<Soft-Problem Hypothesis> ::= <Phenomenon>
                    <Hypothesis-Relationships> <Soft-Goal>
<Phenomenon> ::= <Type><Topic>
<Type> ::= Proposition
<Topic> ::= Proposition
<Hypothesis-Relationships> ::= <Likely-Some-Plus>
                    <Likely-Some-Minus> <Unknown>
<Likely-Some-Plus> ::= 'Likely-to-Make' |
                     'Likely-to-Help' | 'Likely-to-Some-Plus'
<Likely-Some-Minus> ::= 'Likely-to-Deny' |
                     'Likely-to-Hurt' | 'Likely-to-Some-Minus'
<Unknown> ::= 'Unknown'
<Soft-Goal> ::= <Phenomenon>
```

The relationships among a Goal, a Problem, and a Problem Hypothesis are modeled with Contribution types. The Contribution types include And, Or, Equal, or Likely to Some+ or Likely to Some-, adopted from the NFR Framework [59, 32]. The relationship between a Problem Hypothesis and a Goal is either validated or invalidated. Once a Problem Hypothesis is validated, it becomes a Problem. Some constraints among Problem Hypotheses

include time-order among source and target Problem Hypotheses, where a source Problem Hypothesis must have occurred before a target Problem Hypothesis [60].

A Problem Hypothesis can be elaborated on with a problem hypothesis template (PHT) in Table 7.2, which helps more formally understand a phenomenon against a goal. We hypothesize that hot weather may cause the increase of mosquitoes infected with the West Nile Virus in Table 7.2. The PHT consists of essential constructs, such as an observation, a general problem hypothesis, a phenomenon, a contribution relationship, a goal, a stakeholder, and a rationale for the problem hypothesis. Problem hypotheses can also be represented in a graph style called a problem hypothesis interdependency graph (PHIG). Please refer to Fig. 7.3 for a WNV example using the PHIG.

Table 7.2. A problem hypothesis template with an infected mosquitoes

Category	Description
Phenomenon Observation	Mosquitoes infected with West Nile Virus are observed frequently in hot and dry days
General Problem Hypothesis	Hot weather may cause the increase of mosquitoes infected with West Nile Virus
A Phenomenon (Problem Hypothesis)	Hot weather
Contributions	Likely to be some minus
Goal	The problem negatively impacts to the goal of good public (citizen) health in Chicago
Stakeholders	Department of Public Health
Rationale for the potential cause	Literature A describes the hot weather as a major factor for the occurrence of mosquitoes

7.2.2 The Formal Semantics for Validating a Problem Hypothesis and Reasoning Methods

A *problem hypothesis* is a hypothesis that a phenomenon is believed to be some- or against a goal. However, we do not know the truth value of the problem hypothesis. We describe where

ML is used, validating a problem hypothesis below. We can formally express a problem and validate a problem hypothesis using ML:

Let e and g be propositions, representing a phenomenon e and a goal g. Then, the problem is expressed as follows according to the NFR framework [59].

$$\begin{aligned} e \xrightarrow{against} g &\equiv against(e, g) \\ &\equiv hurt(e, g) \vee break(e, g) \end{aligned} \tag{7.1}$$

$hurt(e, g)$ is expressed as follows:

$$\begin{aligned} satisficed(e) \wedge satisficed(hurt(e, g)) \\ \rightarrow deniable(g)) \end{aligned} \tag{7.2}$$

$break(e, g)$ is expressed as follows:

$$\begin{aligned} denied(e) \wedge satisficed(break(e, g)) \\ \rightarrow satisficeable(g) \end{aligned} \tag{7.3}$$

Formula 7.2 shows that the predicate $satisficed(e)$ and $satisficed(hurt(e, g))$ should be determined to validate whether the goal g is deniable or not. In other words, to validate the problem hypothesis that is considered to be true, we need to validate both $satisficed(e)$ and $satisficed(break(e, g))$ predicates. If some records represent the occurrence of some phenomenon e in a dataset, we can treat the truth value of e as true (e.g., satisficed or weakly satisficed). However, it is not easy to validate whether the satisficed predicate of the Contribution, $satisficed(hurt(e, g))$ or $satisficed(break(e, g))$ is true or false in the data model. ML is used here to determine the truth value of the satisficed relationships, $hurt(e, g)$ or $break(e, g)$.

Other important formal definitions for the validation of problem hypotheses have been described in our previous work [22]. Some of them, which are used in the Metis process, are described in the following.

Let $validated(P_n)$ be the proposition that the problem hypothesis P_n is validated, for $n \in \mathbb{Z}^+$. For all $i, j \in \mathbb{Z}^+$, let $P_{i+1,j}$ be the jth problem hypothesis directly decomposed from P_i. An offspring hypothesis ($P_{i+1,j}$) can be related to a parent hypothesis (P_i) using a some positive (some-plus) or negative (some-minus) Contribution. If the offspring is validated and the positive Contribution (some-plus) is validated, then the parent hypothesis is validated. If there are many validated offsprings, we use Formula 7.8.

$$validated(P_{i+1,j}) \wedge validated(some - plus(P_{i+1,j}, P_i)) \rightarrow validated(P_i) \tag{7.4}$$

The Metis defines *feature importance value* (I), which is obtained from running ML and ML Explainability model. The feature importance value is associated with a Contribution from a problem hypothesis P_s (source) to P_t (target), $I_{s,t}$, and the following Formula determines Contribution weight and type (e.g., some-plus or S+) .

$$w(P_s, P_t) = I_{s,t} \tag{7.5}$$

$$ctr\_type\big(I_{s,t}\big) = \begin{cases} \text{S+} & \text{if } I_{s,t} \geq 0 \\ \text{S-} & \text{if } I_{s,t} < 0 \end{cases} \tag{7.6}$$

A source hypothesis has a score based on the weight of the targeted hypotheses and their respective contributions. The function $w(P_t)$ describes the importance weight of a target hypothesis. Hence, the overall score for a source hypothesis P_s can be given by the utility function as follows:

$$score(P_s) = \left(\sum_{t=1}^{\#targets} w(P_t) \times w(P_s, P_t) \right) \tag{7.7}$$

After computing the scores for all source hypotheses, the selection process may be carried out in a bottom-up approach [64]. We select the maximum value in the lowest source

hypothesis set to propagate that validation to the target hypothesis set. The source with the highest score is selected towards a parent hypothesis.

$$selection(P_t) = max\Big(score(P_s)\Big)_{s=1}^{\#sources} \tag{7.8}$$

We want to determine which hypothesis in the source set (i.e., hypotheses that originate the contributions) is more relevant to the target set (i.e., hypotheses that receive the contributions) to maximize the validation insights generated by the application of ML models. In this case, validating a hypothesis P_i will now depend on the validation of the selection for P_i. After the lowest source hypothesis set is evaluated, we proceed to the next one until the selection process covers the entire set of hypotheses.

$$validated(selection(P_i)) \rightarrow validated(P_i) \tag{7.9}$$

7.2.3 The Metis Process

The Metis process described in Fig. 7.2 includes four steps. The process helps model business goals and problem hypotheses, prepare a dataset, build ML models, and validate problem hypotheses using the Metis ontology and semantic reasoning methods.

Step 1: Explore Problem Hypotheses. Important business needs of stakeholders are elicited as (Soft-)goals through consultation, interviews, and reviewing key business documents. (Soft-)goals are then used as the context to identify problems.

A phenomenon against the captured goals is analyzed, where a problem hypothesis template may be used to elaborate on the phenomenon and its relationships. The elicited problems are then refined into sub-problems specific enough to test with data. The elicited goals, problems, and sub-problems are modeled in a problem hypothesis interdependency graph (PHIG) in Fig. 7.3.

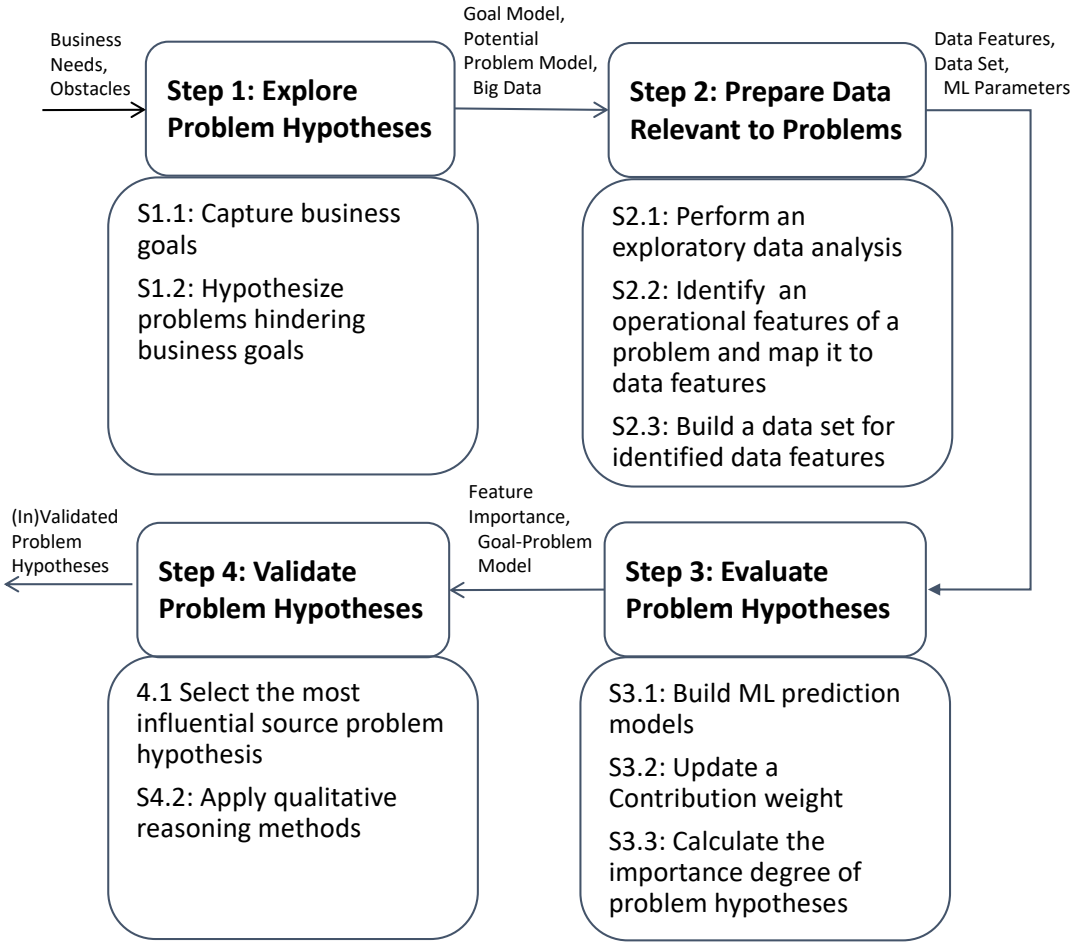


Figure 7.2. The Metis process for validating a problem hypothesis

Step 2: Prepare Data Relevant to Problems. For the analyzed problems, requirements engineers, together with data scientists, perform an exploratory data analysis, identify data features relevant to testable problem hypotheses in a semi-automatic manner, clean or transform data, and build a dataset from a data source through a database/Big query for ML processing.

Step 3: Evaluate Problem Hypotheses. Using the prepared ML dataset, we build Supervised ML models to classify records for a classification label, which could be positively or negatively contribute to a goal [78]. Next, we set up an accuracy criterion of ML models and select the best one to validate problems more accurately. An ML Explainability model

is then utilized to detect the feature importance [85, 58]. The feature importance is used to capture impact relationships among problem hypotheses and goals and identify the most important problem towards a goal.

Step 4: Validate Problem Hypotheses. This step selects the most influential problem hypothesis among offsprings and then uses semantic reasoning methods towards a goal. If the problem hypothesis is validated, then it becomes a problem against a goal.

7.3 The Metis in Action

In this section, the Metis framework is applied to the West Nile virus empirical study. We validate problem hypotheses behind the West Nile virus cases following the Metis process.

7.3.1 Step 1: Explore Problem Hypotheses of Increased Mosquitoes Infected by West Nile Virus

In the WNV example illustrated in Fig. 7.3, we suppose the city has a goal, Minimize Mosquito-Infected Diseases, an NF(Non-Functional) Softgoal, at the top organizational level, which is AND-decomposed to Minimize Occurrence of West Nile Virus and Minimize Occurrence of Zika Virus.

After consulting with city officials, we hypothesize that the Spread of the West Nile Virus is against the Minimize Occurrence of West Nile Virus goal for problem identification. We then hypothesize that Increased Mosquitoes Infected by WNV positively contribute to the Spread of the West Nile Virus problem. The Increased Mosquitoes Infected by WNV is then refined into Optimal Weather, High Occurrence of WNV Species, and High Occurrence Locations sub-problem hypotheses. The High Occurrence Location is further refined into the testable hypothesis, such as unique trap locations from Trap-1 to Trap-136. The other problem hypotheses are further refined into testable hypotheses, as illustrated in Fig. 7.3.

7.3.2 Step 2: Prepare an ML dataset for Hypotheses Validation

We briefly first describe the West Nile virus (WNV) data used in this exploratory study. Three datasets are available: the Main dataset, Weather dataset, and Spray dataset. The Main dataset contains occurrence data of WNV, such as test date, location, trap number, the number of trapped mosquitoes, the species of WNV, presence of WNV species, and others in 2007, 2009, 2011, and 2013. The feature, the presence of WNV species, is the classification label ML models need to predict. The Weather dataset from 2007 and 2014 contains weather conditions of two weather stations in the city. The Spray dataset contains data of spraying work controlling mosquitoes in 2011 and 2013.

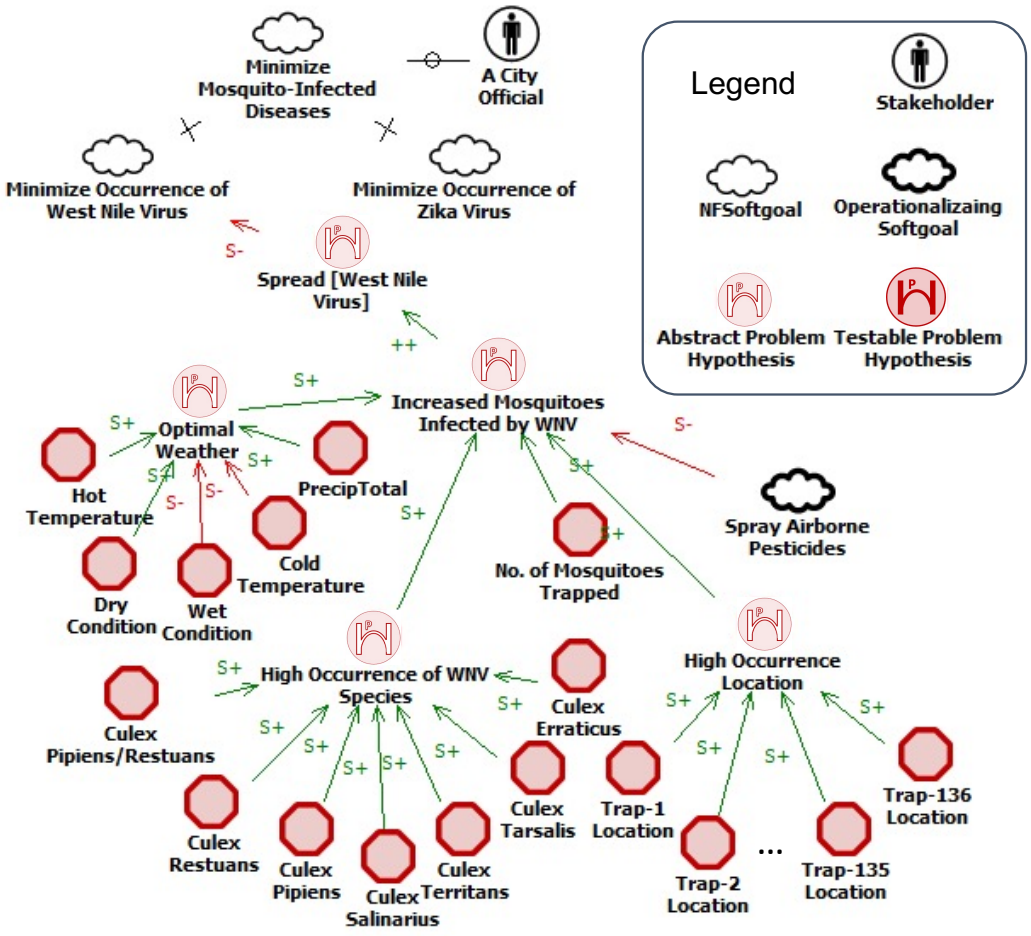


Figure 7.3. A portion of a West Nile Virus problem hypothesis model

After the initial analysis of the WNV dataset, the testable problem hypotheses are mapped to data features in a WNV dataset for ML processing. For example, for the testable problem hypothesis P1, "Dry weather condition is likely to help for the optimal weather to increase WNV infected mosquitoes," we manually map the *dry weather condition* to a *WetBulb* feature in a Weather dataset in Fig. 7.3.

We noticed that the three independent datasets could only be related to *date* and *location* features where the location was identified with a latitude and longitude.

We merged the Main and Spray datasets based on a date and the location, checking whether the trap location in the Main dataset is within the rectangle area. In merging the Main and Weather datasets, we used location data with weather stations' latitude and longitude. We assumed that if the trap location in the Main dataset is near one of two weather stations, the trap location is under the impact of the weather in the station.

If the data type is a category type, such as WNV species and Trap locations, we applied one hot encoding technique to those features and cleaned some data features. We extracted and merged relevant data features from the WNV dataset for the identified data features. The constructed ML dataset contained 10506 records and 160 features.

7.3.3 Step 3: Evaluate Problem Hypotheses about the Spread of West Nile Virus

Three Supervised ML models were built for the WNV prediction. ML models, such as Decision Tree, Random Forest, and eXtreme Gradient Boosting (XGBoost) model, showed an accuracy of 0.93, 0.947, and 0.95. The XGBoost model showed the best accuracy, passing our predefined accuracy criterion (here, accuracy > 0.9). Fig. 7.4. shows the ROC (Receiver Operating Characteristic) curve of the XGBoost model.

Next, we got the feature importance of each feature mapped to the problem hypotheses using a SHAP Explainability model. We analyzed features that positively or negatively impact the presence of mosquitoes infected WNV or the classification label in the Main dataset.

Fig. 7.5(a) shows the average feature impact. Fig. 7.5(b) aggregated feature importance value for all the present WNV cases in the all dataset, where we can observe that station pressure, trap_2, num_mosquitoes features have a higher positive SHAP value than other features. It can be interpreted that these features increase the WNV infected mosquitoes. In contrast, the bottom features, such as average temperature, sea level, are against the cases. We noticed that almost 120 features did not impact the classification output in our experiments.

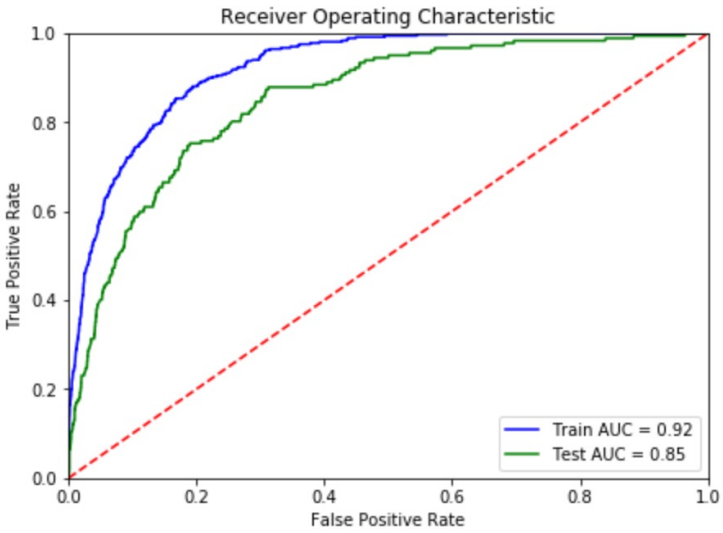
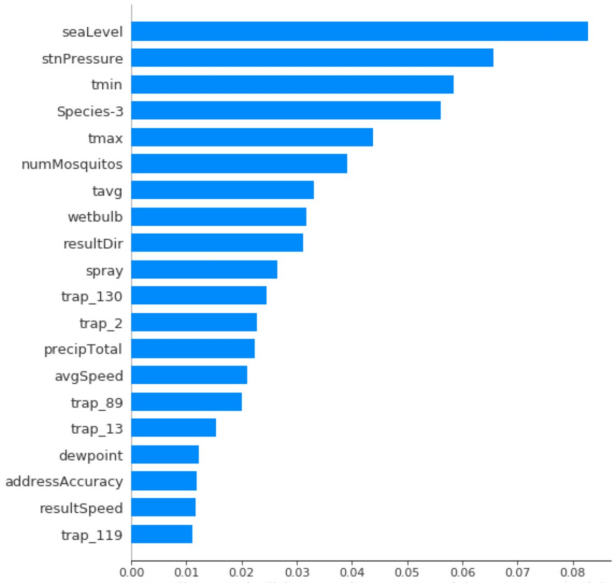


Figure 7.4. The XGBoost ROC Curve for validating a WNV problem hypothesis

We then applied the feature importance (the sum SHAP value) of data features to each leaf-level problem hypotheses' Contribution weight and type using Formula 7.5 and 7.6 in Fig. 7.6. For example, the *Station Pressure* problem hypothesis is updated with the value $(14.226 * 0.2 =)$ 2.845 and $S+$ using Formula 7.7. The maximum value in the source problem hypotheses, here Station Pressure, is selected using Formula 7.8 to propagate that validation to the target hypothesis set. Similarly, the other problem hypotheses are updated, calculated, and evaluated accordingly.



a) Average feature importance to XGBoost

No	Feature Name	Feature Importance
1	stationPressure	14.2260771
2	avgSpeed	1.712537308
3	precipTotal	1.361484005
4	wetbulb	2.33147E-11
5	trap_2	1.39819E-11
6	trap_130	1.27329E-11
7	trap_135	9.38485E-12
8	numMosquitos	8.50015E-12
	...	
153	trap_61	-1.03129E-11
154	trap_119	-1.48059E-11
155	trap_13	-2.27336E-11
156	resultSpeed	-0.447770751
157	avgTemperature	-1.855886578
158	seaLevel	-6.548853986
159	trap_121	-8.44758835

b) Aggregated feature importance

Figure 7.5. The XGBoost ROC Curve for validating a WNV problem hypothesis

7.3.4 Step 4: Validate Problem Hypotheses about the Spread of West Nile Virus

In the WNV example, *Station Pressure*, *Species-1 (Culex Pipiens/Restuans)*, and *Trap-2* were selected by Formula 7.9 among the important hypotheses for each group on the leaf nodes. The selected problem hypothesis is considered a validated problem hypothesis by Formula 7.9, as it is most likely to be the cause for the target problem hypothesis. We applied the qualitative reasoning methods for the other problems against goals, as shown in Fig. 7.6. Over *S+ (Some Plus)* satisficing contribution toward *Optimal Weather*, *High Occurrence of WNV Species*, and *High Occurrence Location*. As the *S+* contributions are satisficed, the parent problem hypothesis' label is *weakly satisficed*. For the other problems, similar methods are applied upward.

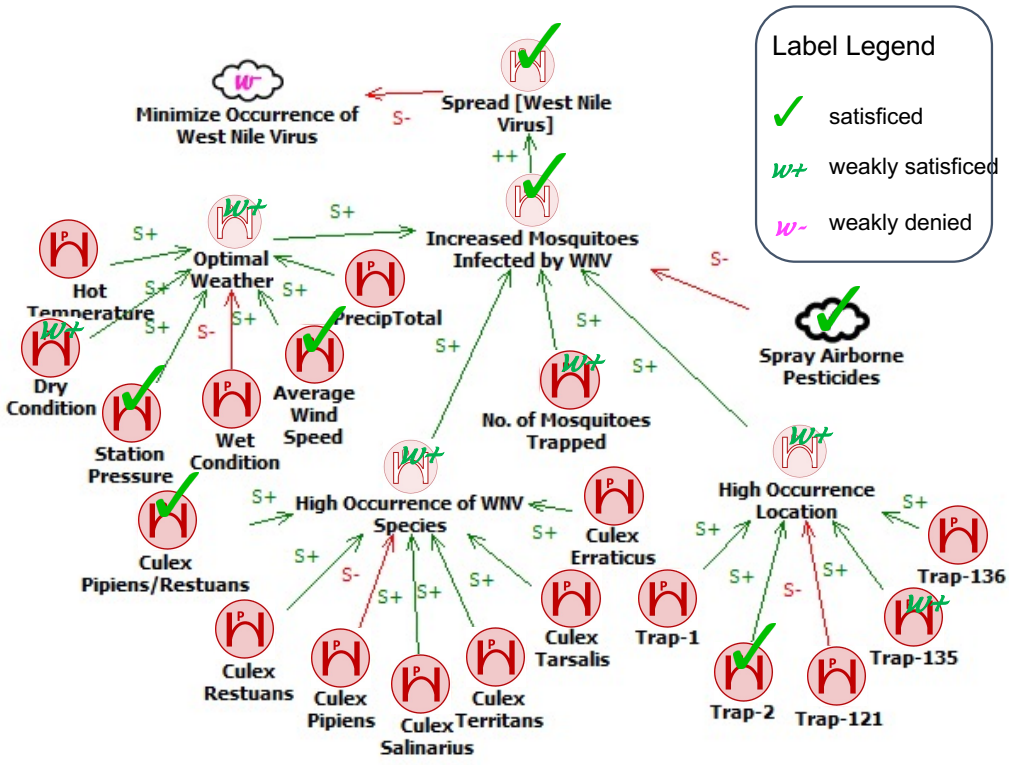


Figure 7.6. Validating the problem hypotheses of West Nile Virus cases

7.4 Related Work

Several problem analysis methods have been used to understand real-world business problems and identify root causes at a conceptual level. A Fishbone diagram helps enumerate possible causes for a problem through an interview or brainstorming with stakeholders [35]. The Fault Tree Analysis (FTA) provides deductive procedures and a logic diagram to help determine failures or errors of software, hardware, and people with a top-down approach [71]. (Soft-)Problem Interdependency Graph (PIG) refines a problem in the context of goals supporting uncertainty relationships [37]. The key difference between our work and the above three methods is their lack of validation methods using ML among a problem and sub-problems encoded in a dataset. The Metis builds a dataset mapped to problem hypotheses and uses ML to get insights into interrelated data features corresponding to problem hypotheses.

Our work also helps elaborate and visualize a problem hypothesis using a problem hypothesis template (PHT) and a Problem Hypothesis Interdependency Graph (PHIG).

7.5 Discussion and Observation

Some key ideas and observations are discussed. Limitations of our work are also described.

7.5.1 Discussion

The improved Metis framework classified a problem hypothesis into an abstract and testable problem hypothesis to better model a problem mapped to data features. The Contribution relationships also include 'likely to some plus' and 'likely to some minus' to reflect some uncertain relationships among problems and goals.

Many different factors in Weather, WNV testing in traps, and spraying pesticide activities may impact mosquitoes infected with WNV. Although the Metis framework selects and validates the most influential hypothesis or feature, different parallel factors may cause a problem. In this case, we may select the top 20% problem hypotheses to understand the situation better using the Pareto principle [90, 91].

As some of the ML data features in the WNV dataset are repeated and unique features are not big, we used the manual approach to map a concept of a problem hypothesis to a data feature for ML. In case a testable problem hypothesis needs to be related to many data features in a complex entity-relationship model, a semi-automatic approach may be utilized [23]. Although the root causes are identified using the Metis methods, caution should be taken as the Metis provides the result based on the problem hypotheses, available dataset, and ML models. Our work may not encompass all the real phenomena and interrelated impacts in the complex environment.

7.5.2 Observation

We believe this paper is one of the first to propose a framework validating a problem hypothesis using ML. The Metis utilize the feature impact of a problem contributing to a goal and connect those insights to problems and goals reasoning. We noticed that some traps (or locations) show negative contributions to observing WNV infected mosquitoes. Requirements engineers may interpret this observation as there have been few WNV cases in those locations. It may mean the city official can utilize this insight to fewer spray pesticides on those locations. Similarly, more than 120 features, such as some trap locations, WNV species, and heat mapped to problem hypotheses, have little impact on the ML model's output. It can be interpreted these features are not important or do not make a difference to the model. The city officials may focus on more important problem hypotheses to mitigate the WNV cases. The validation experiments for the WNV dataset have been supported by Python scikit-learn on Jupyter Lab. The SHAP Explainability Library showed slow performance as the number of data records increased in calculating the SHAP values, which may cause issues in processing and validating Big Data.

7.5.3 Limitations

This paper has some limitations. In case several plausible problem hypotheses against goals are validated and competed, those hypotheses may need to be selected depending on the situation. Multiple selections for the validation of a problem hypothesis have not been studied. Supervised ML models' quality properties, such as recalls, false positives, and false negatives, are not much utilized to support semantic reasoning of problems and goals.

7.6 Conclusion and Future Work

This paper has presented the improved Metis framework that supports the validation of business problems using problem hypotheses. Using problem hypotheses helped explore

and validate phenomena that are likely to cause the West Nile virus's spread. The newly added and refined essential concepts and relationships, a problem hypothesis template, and a problem hypothesis interdependency graph helped capture phenomena against goals and streamlined the Metis process, bridging the gap between goal orientation and ML.

Future work includes in-depth studying about relationships of interrelated features and their analysis, utilizing the quality properties of Supervised ML to identify problems better and explore solutions to mitigate the identified problems.

CHAPTER 8

CONCLUSION

8.1 Summary

This dissertation has presented the Gomphy framework to validate business problem hypotheses with empirical studies validating clients' banking events behind an unpaid loan and customer churn. Business organizations may use Gomphy to confirm whether some potential problems hidden in Big Data are against a business goal or not. Gomphy would help find real business problems and improve business value, especially in Big Data and Machine Learning (ML) projects. Information systems with confirmed problems can explore solutions to take the right actions to mitigate those problems towards achieving business goals. The Gomphy framework supports the validation of a business problem, utilizing a goal-oriented and Machine Learning-based approach. Starting with modeling business goals, we explored potential business events against goals, modeled the events as a testable problem hypothesis entity, identified data attributes along with constraints and relationships, and built an ML dataset from a source database. Next, we discovered critical factors towards a target problem using ML, evaluated the relationships among the critical factors and problems, and selected the most important one as a validated problem hypothesis.

8.2 Contribution

Five technical contributions, as shown in Table 8.1, to help overcome challenges for validating business problems were presented: 1. The domain-independent Gomphy ontology helping prevent omissions and commissions in modeling essential concepts and the Gomphy process were described. Using the Gomphy ontology and process, Gomphy explicitly and formally explores hypothesized problems against goals, prepares a dataset with the identified

testable factors corresponding to problem hypotheses, discovers critical factors towards a target problem using ML, evaluates the relationships among the critical factors and problems, and selects the most important one as a validated problem hypothesis. Gomphy captures categories of essential concepts of business goals, problems, ML, and a dataset. Gomphy also captures categories of essential relationships between the essential concepts, provides traceability from business problems to a data feature and helps validate problem hypotheses. 2. A method of modeling a concept of problem hypothesis as a problem hypothesis entity was presented. The concept in a problem hypothesis is modeled as an entity that consists of an entity name, attributes, constraints, and relationships to identify testable factors. 3. A data preparation method was illustrated, helping identify relevant features to test in a database and build a dataset, mapping a concept of a problem hypothesis to a domain data feature, and extracting a dataset from a source dataset. 4. An evaluation method was presented, detecting the positive and negative relationships among problem hypotheses and validating the problem hypothesis utilizing feature importance and a reasoning scheme. 5. A set of formalized validation rules were described for reasoning about connected problem hypothesis validation in a goal-oriented problem hypothesis model.

8.3 Future Work

Future work includes an in-depth study about the impact of correlated and time-series events and different ML algorithms, such as neural networks on validating problem hypotheses, exploring potential solutions to mitigate the validated problem using ML and a goal hypothesis, and developing a reliable Gomphy assistant tool that would be helpful to automate the mapping between a problem hypothesis entity and a database entity. Tool support, such as a template, helping to manage a problem hypothesis is also needed.

Table 8.1. Challenges for validating business problems and the Gomphy solutions

Challenges	**Solutions**
A lack of understanding about relationships among business problems, goals, data, and ML	The Gomphy ontology helping modeling categories of essential concepts, relationships, and constraints, while preventing omissions and commissions of an application model. The Gomphy process, guiding the validation of a problem hypothesis and providing traceability from goals to problems, ML and data
Determining a testable factor associated with a potential business problem	A method of modeling a concept of a problem hypothesis as a problem hypothesis entity, helping capture business events and identify testable factors
Preparing a relevant dataset corresponding to the business problem	A data preparation method for building an ML dataset by mapping a concept of a problem hypothesis to a data feature and extracting a dataset from a source dataset
Analyzing the impact on the business problem to other problems	An evaluation method of a problem hypothesis, detecting contribution relationships among the business events and a problem, using ML and ML Explainability techniques
Reasoning about inter-connected problems and goals	A set of formalized validation rules for reasoning about connected problem hypothesis validation in a goal-oriented problem hypothesis model

Formalization of the mapping process is planned using first-order logic. We also plan to apply the Gomphy approach to other domains, such as the public health domain, to see the strength and weaknesses of our work.

REFERENCES

[1] D. T. Ross and K. E. Schoman, "Structured analysis for requirements definition," *IEEE Transactions on Software Engineering*, vol. SE-3, no. 1, pp. 6–15, 1977.

[2] B. Nuseibeh and S. Easterbrook, "Requirements engineering: a roadmap," in *Proceedings of the Conference on the Future of Software Engineering*, pp. 35–46, 2000.

[3] D. Pyle, *Data preparation for data mining*. Morgan Kaufmann, 1999.

[4] H. S. Fogler, S. E. LeBlanc, and B. R. Rizzo, *Strategies for creative problem solving*. Prentice Hall, 2008.

[5] J. Brownlee, *Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python*. Machine Learning Mastery, 2020.

[6] P. Drucker, *The practice of management*. Routledge, 2012.

[7] T. H. Davenport, P. Barth, and R. Bean, "How 'big data' is different," *MIT Sloan Management Review*, vol. 54, no. 1, pp. 22–24, 2012.

[8] S. Ransbotham, D. Kiron, and P. K. Prentice, "Beyond the hype: the hard work behind analytics success," *MIT Sloan Management Review*, vol. 57, no. 3, 2016.

[9] S. Ransbotham, P. Gerbert, M. Reeves, D. Kiron, and M. Spira, "Artificial intelligence in business gets real," *MIT sloan management review*, vol. 60280, 2018.

[10] M. H. Jensen, P. A. Nielsen, and J. S. Persson, "Managing big data analytics projects: The challenges of realizing value," in *European Conference on Information Systems*, Association for Informations Systems, AIS, 2019.

[11] R. Bean and T. H. Davenport, "Companies are failing in their efforts to become data-driven," *Harvard Business Review*, pp. 5–8, 2019.

[12] M. Asay, "85% of Big data projects fail, but your developers can help yours succeed. techrepublic," 2017.

[13] S. LaValle, E. Lesser, R. Shockley, M. Hopkins, and N. Kruschwitz, "Big data, analytics and the path from insights to value," *MIT Sloan Management Review*, vol. 52, pp. 21–32, 2011.

[14] V. Grover, R. H. Chiang, T.-P. Liang, and D. Zhang, "Creating strategic business value from big data analytics: A research framework," *Journal of Management Information Systems*, vol. 35, no. 2, pp. 388–423, 2018.

[15] L. NewVantage Partners, "Big data and AI executive survey 2020: Data-driven business transformation connecting data/AI investment to business outcomes (2020)," 2020.

[16] B. Schreck, M. Kanter, K. Veeramachaneni, S. Vohra, and R. Prasad, "Getting value from machine learning isn't about fancier algorithms–it's about making it easier to use," *Harvard Business Review*, 2018.

[17] L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos, "Machine learning on Big data: Opportunities and challenges," *Neurocomputing*, vol. 237, pp. 350–361, 2017.

[18] T. H. Davenport and R. Bean, "Big data and ai executive survey (2020)," *NewVantage Partners (NVP), Tech. Rep*, 2020.

[19] N. P. Repenning, D. Kieffer, and T. Astor, "The most underrated skill in management," *MIT Sloan Management Review*, vol. 58, no. 3, pp. 39–48, 2017.

[20] M. P. Joshi, N. Su, R. D. Austin, and A. K. Sundaram, "Why so many data science projects fail to deliver," *MIT Sloan Management Review*, vol. 62, no. 3, pp. 85–89, 2021.

[21] D. Paradice, "Decision support and problem formulation activity," in *Encyclopedia of Decision Making and Decision Support Technologies*, pp. 192–199, IGI Global, 2008.

[22] S. Supakkul, R. Ahn, R. J. Gonçalves, D. Villarreal, L. Zhao, T. Hill, and L. Chung, "Validating goal-oriented hypotheses of business problems using machine learning," in *Int. Conf. on Big Data*, Springer, 2020.

[23] R. Ahn, R. J. Gonçalves, T. Hill, L. Chung, S. Supakkul, and L. Zhao, "Discovering business problems using problem hypotheses: A goal-oriented and machine learning-based approach," in *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 137–140, IEEE, 2021.

[24] R. Ahn, S. Supakkul, L. Zhao, K. Kolluri, T. Hill, and L. Chung, "Validating business problem hypotheses: A goal-oriented and machine learning-based approach," in *Int. Conf. on Big Data*, Springer, 2021.

[25] R. Ahn, S. Supakkul, L. Zhao, K. Kolluri, T. Hill, and L. Chung, "A goal-oriented approach for preparing a machine-learning dataset to support business problem validation," in *2021 IEEE Intl Conf on Cloud and Big Data Computing*, IEEE, 2021.

[26] P. Berka and M. Sochorova, "Discovery challenge guide to the financial data set, pkdd-99," 1999.

[27] A. Van Lamsweerde, "Goal-oriented requirements enginering: a roundtrip from research to practice [enginering read engineering]," in *Proceedings. 12th IEEE International Requirements Engineering Conference, 2004.*, pp. 4–7, IEEE, 2004.

[28] A. Van Lamsweerde, *Requirements engineering: From system goals to UML models to software*, vol. 10. Chichester, UK: John Wiley & Sons, 2009.

[29] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-functional requirements in software engineering*, vol. 5. Springer Science & Business Media, 2012.

[30] J. Horkoff, "Non-functional requirements for machine learning: Challenges and new directions," in *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pp. 386–391, IEEE, 2019.

[31] J. Mylopoulos, L. Chung, and E. Yu, "From object-oriented to goal-oriented requirements analysis," *Communications of the ACM*, vol. 42, no. 1, pp. 31–37, 1999.

[32] J. Mylopoulos, L. Chung, S. Liao, H. Wang, and E. Yu, "Exploring alternatives during requirements analysis," *IEEE Software*, vol. 18, no. 1, pp. 92–96, 2001.

[33] R. J. Gonçalves, R. Ahn, T. Hill, and L. Chung, "Towards high quality recommendations: A goal-oriented and ontology-based interactive approach," in *The 32nd International Conference on Software Engineering and Knowledge Engineering (SEKE)*, pp. 89–92, KSI Research Inc., 2020.

[34] K. Kolluri, R. Ahn, T. Hill, and L. Chung, "Risk analysis for collaborative systems during requirements engineering," in *The 33rd International Conference on Software Engineering and Knowledge Engineering (SEKE)*, pp. 297–302, KSI Research Inc., 2021.

[35] K. Ishikawa, *Introduction to quality control*. Productivity Press, 1990.

[36] W. Vesely, F. Goldberg, N. Roberts, and D. Haasl, "Fault tree handbook," tech. rep., Nuclear Regulatory Commission Washington DC, 1981.

[37] S. Supakkul and L. Chung, "Extending problem frames to deal with stakeholder problems: An agent-and goal-oriented approach," in *Proceedings of the 2009 ACM symposium on Applied Computing*, pp. 389–394, 2009.

[38] G. Park, L. Chung, L. Khan, and S. Park, "A modeling framework for business process reengineering using big data analytics and a goal-orientation," in *2017 11th International Conference on Research Challenges in Information Science (RCIS)*, pp. 21–32, IEEE, 2017.

[39] G. Park, S. Park, L. Khan, and L. Chung, "Iris: A goal-oriented big data analytics framework on spark for better business decisions," in *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 76–83, IEEE, 2017.

[40] J. Hassine and D. Amyot, "A questionnaire-based survey methodology for systematically validating goal-oriented models," *Requirements Engineering*, vol. 21, no. 2, pp. 285–308, 2016.

[41] D. Dell'Anna, F. Dalpiaz, and M. Dastani, "Validating goal models via bayesian networks," in *2018 5th International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, pp. 39–46, IEEE, 2018.

[42] T. Hill, S. Supakkul, and L. Chung, "Confirming and reconfirming architectural decisions on scalability: a goal-driven simulation approach," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pp. 327–336, Springer, 2009.

[43] L. Chung, T. Hill, O. Legunsen, Z. Sun, A. Dsouza, and S. Supakkul, "A goal-oriented simulation approach for obtaining good private cloud-based system architectures," *Journal of Systems and Software*, vol. 86, no. 9, pp. 2242–2262, 2013.

[44] H. Johng, D. Kim, T. Hill, and L. Chung, "Estimating the performance of cloud-based systems using benchmarking and simulation in a complementary manner," in *International Conference on Service-Oriented Computing*, pp. 576–591, Springer, 2018.

[45] H. Johng, D. Kim, G. Park, J.-E. Hong, T. Hill, and L. Chung, "Enhancing business processes with trustworthiness using blockchain: a goal-oriented approach," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pp. 61–68, 2020.

[46] R. Wirth and J. Hipp, "Crisp-dm: Towards a standard process model for data mining," in *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, vol. 1, Springer-Verlag London, UK, 2000.

[47] B. Rajagopalan and M. W. Isken, "Exploiting data preparation to enhance mining and knowledge discovery," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 31, no. 4, pp. 460–467, 2001.

[48] S. Nalchigar and E. Yu, "Business-driven data analytics: a conceptual modeling framework," *Data & Knowledge Engineering*, vol. 117, pp. 359–372, 2018.

[49] S. García, J. Luengo, and F. Herrera, *Data preprocessing in data mining*, vol. 72. Springer, 2015.

[50] J. Luengo, S. García, and F. Herrera, "On the choice of the best imputation methods for missing values considering three groups of classification methods," *Knowledge and information systems*, vol. 32, no. 1, pp. 77–108, 2012.

[51] J. A. Sáez, J. Luengo, and F. Herrera, "Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification," *Pattern Recognition*, vol. 46, no. 1, pp. 355–364, 2013.

[52] H. Liu and H. Motoda, *Computational methods of feature selection*. CRC Press, 2007.

[53] K. Yu, L. Liu, and J. Li, "A unified view of causal and non-causal feature selection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 4, pp. 1–46, 2021.

[54] M. J. Berry and G. S. Linoff, *Data mining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons, 2004.

[55] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[56] C. Molnar, *Interpretable machine learning*. Lulu. com, 2020.

[57] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should I trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

[58] S. Lundberg, G. Erion, H. Chen, A. DeGrave, J. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable AI for trees," *Nature machine intelligence*, vol. 2, pp. 56–67, 2020.

[59] J. Mylopoulos, L. Chung, and B. Nixon, "Representing and using nonfunctional requirements: A process-oriented approach," *IEEE Transactions on software engineering*, vol. 18, no. 6, pp. 483–497, 1992.

[60] J. Pearl and T. S. Verma, "A theory of inferred causation," in *Studies in Logic and the Foundations of Mathematics*, vol. 134, pp. 789–811, Elsevier, 1995.

[61] P. P.-S. Chen, "The entity-relationship model—toward a unified view of data," *ACM Transactions on Database Systems (TODS)*, vol. 1, no. 1, pp. 9–36, 1976.

[62] B. Carlo, S. Ceri, and N. Sham, *Conceptual Database Design: An Entity-Relationship Approach*. Benjamin/Cummings, 1992.

[63] L. Chung, P. Katalagarianos, M. Marakakis, M. Mertikas, J. Mylopoulos, and Y. Vassiliou, "From information system requirements to designs: a mapping framework," *Information Systems*, vol. 16, no. 4, pp. 429–461, 1991.

[64] S. Supakkul, T. Hill, L. Chung, T. T. Tun, and J. C. S. do Prado Leite, "An NFR pattern approach to dealing with NFRs," in *2010 18th IEEE International Requirements Engineering Conference*, pp. 179–188, IEEE, 2010.

[65] M. Binkhonain and L. Zhao, "A review of machine learning algorithms for identification and classification of non-functional requirements," *Expert Systems with Applications: X*, vol. 1, p. 100001, 2019.

[66] C. Rolland, C. Souveyet, and C. B. Achour, "Guiding goal modeling using scenarios," *IEEE transactions on software engineering*, vol. 24, no. 12, pp. 1055–1071, 1998.

[67] S. Hartmann and S. Link, "English sentence structures and eer modeling," in *APCCM*, vol. 7, p. 2735, 2007.

[68] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: methods and prospects," *Big Data Analytics*, vol. 1, no. 1, pp. 1–22, 2016.

[69] A. Zheng and A. Casari, *Feature engineering for machine learning: principles and techniques for data scientists*. " O'Reilly Media, Inc.", 2018.

[70] J. J. Li and X. Tong, "Statistical hypothesis testing versus machine learning binary classification: Distinctions and guidelines," *Patterns*, vol. 1, no. 7, p. 100115, 2020.

[71] B. Vesely, "Fault tree analysis (fta): Concepts and applications," *NASA HQ*, 2002.

[72] R. Wille, "Restructuring lattice theory: an approach based on hierarchies of concepts," in *Intĺ conference on formal concept analysis*, pp. 314–339, Springer, 2009.

[73] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[74] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, R. Kaluri, D. S. Rajput, G. Srivastava, and T. Baker, "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54776–54788, 2020.

[75] S. Supakkul, L. Zhao, and L. Chung, "GOMA: Supporting Big data analytics with a goal-oriented approach," in *2016 IEEE International Congress on Big Data (BigData Congress)*, pp. 149–156, June 2016.

[76] R. Lukyanenko, A. Castellanos, J. Parsons, M. C. Tremblay, and V. C. Storey, "Using conceptual modeling to support machine learning," in *International Conference on Advanced Information Systems Engineering*, pp. 170–181, Springer, 2019.

[77] M. Jackson, *Problem frames: analysing and structuring software development problems*. Addison-Wesley, 2001.

[78] P. Norving and S. Russell, *Artificial intelligence: a modern approach, Global Edition*. Pearson Education Limited, 2021.

[79] H.-Y. Wu, G.-H. Tzeng, and Y.-H. Chen, "A fuzzy MCDM approach for evaluating banking performance based on balanced scorecard," *Expert systems with applications*, vol. 36, no. 6, pp. 10135–10147, 2009.

[80] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, "Data lifecycle challenges in production machine learning: a survey," *ACM SIGMOD Record*, vol. 47, no. 2, pp. 17–28, 2018.

[81] T. H. Davenport and R. Ronanki, "Artificial intelligence for the real world," *Harvard business review*, vol. 96, no. 1, pp. 108–116, 2018.

[82] P. Tatter, *gpk: 100 Data Sets for Statistics Education*. R package v. 1.0., 2013.

[83] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, pp. 37–37, 1996.

[84] W. Wang, L. Chen, K. Thirunarayan, and A. P. Sheth, "Harnessing Twitter 'Big Data' for automatic emotion identification," in *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pp. 587–592, IEEE, 2012.

[85] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st international conference on neural information processing systems*, pp. 4768–4777, 2017.

[86] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[87] J. Cohen, "The earth is round ($p<.05$)," *American Psychologist*, vol. 49, pp. 997–1003, 1994.

[88] J. Berkson, "Tests of significance considered as evidence," *Journal of the American Statistical Association*, vol. 37, no. 219, pp. 325–335, 1942.

[89] A. Einstein, *The ultimate quotable Einstein*. Princeton University Press, 2011.

[90] R. C. Craft and C. Leake, "The pareto principle in organizational decision making," *Management Decision*, 2002.

[91] R. Sanders, "The pareto principle: its use and abuse," *Journal of Services Marketing*, 1987.

BIOGRAPHICAL SKETCH

Sung Soo (Robert) Ahn was curious why some software systems were built, what components were needed to develop, and how some systems worked well or not when he studied computer science subjects.

He joined a research institute, Korea Institute of Science and Technology Information (KISTI), in Daejon, after he finished his master's study at Hanyang University, Seoul, South Korea. He was responsible for designing and developing a low level storage engine, one essential component of the information retrieval system, KRISTAL 2000 in KISTI, which was widely used as a back end system to provide the search service of R&D papers, reports, and others. He was also a technical staff managing biodiversity data, training an exchange protocol of biodiversity data, and representing a Korean node of the Global Biodiversity of Information Facility (GBIF). He designed and operated a decision support system and helped prepare rules and policies for performing information system projects in KISTI.

When he joined a PhD program in the computer science department at the University of Texas, Dallas, he has worked with Dr. Lawrence Chung. His research areas include validating a problem hypothesis, problem analysis, non-functional requirements in requirements engineering, machine learning, knowledge representation, and model-driven software engineering. During his PhD study, he could answer some questions about a software system he conceived at the beginning of his research.

Mr. Ahn performed and presented research work at NSF Net-centric & Cloud Software & Systems (NCSS) Industry & University Cooperative Research Center (IUCRC) and Software and Security Research Center (S^2ERC).

CURRICULUM VITAE

Sung Soo (Robert) Ahn

October 1, 2021

Contact Information:

Department of Computer Science
The University of Texas at Dallas
800 W. Campbell Rd.
Richardson, TX 75080-3021, U.S.A.

Email: robert.sungsoo.ahn@utdallas.edu

Educational History:

BS, Computer Science and Engineering, Hanyang University, Ansan, South Korea, 1998
MS, Computer Science and Engineering, Seoul, South Korea, 2000
PhD, Computer Science, University of Texas at Dallas, 2021

Validating Business Problem Hypotheses: A Goa-Oriented and Machine Learning-Based Approach
PhD Dissertation
Computer Science Department, University of Texas at Dallas
Advisors: Dr. Lawrence Chung

Employment History:

Senior Research Engineer, Korea Institute of Science and Technology Information, May 2006 – present
Research Engineer, Korea Institute of Science and Technology Information, September 1999 – May 2006

Professional Recognition and Honors:

KISTI - Outstanding Achievement Award, 2004
KISTI - Outstanding Achievement Award, 2008

Professional Memberships:

KIISE - Korea Institute of Information Scientist and Engineers (KIISE), 2000–present
KIPS - Korea Information Processing Society (KIPS), 2000–present