2014

NEW ARTIFACTS FOR THE KNOWLEDGE DISCOVERY VIA DATA ANALYTICS (KDDA) PROCESS

Yan Li
liy26@vcu.edu

NEW ARTIFACTS FOR THE KNOWLEDGE DISCOVERY VIA DATA ANALYTICS (KDDA) PROCESS

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

by

YAN LI

Master of Science in Information Systems
Virginia Commonwealth University, 2009

Bachelor of Science in Chemical Physics,
University of Science and Technology of China, 1999

Director: KWEKU-MUATA OSEI-BRYSON
PROFESSOR, INFORMATION SYSTEMS

Virginia Commonwealth University
Richmond, Virginia
December 2014

Acknowledgement

My journey in the VCU Information Systems PhD program is a long, eventful, and fruitful one. During this journey, I have oriented my career in the direction that integrates research, teaching, and practice in the realm of information science. This dissertation is the result of my intellectual curiosity for data analytics and my passion for designing and building things. It is also the result of many experiences I have encountered at VCU from many extraordinary individuals. I am using this opportunity to express my gratitude to these individuals who supported me throughout the PhD process.

First and foremost I wish to express my deepest gratitude to my dissertation advisor, Dr. Kweku-Muata Osei-Bryson. The inception of this dissertation started six and a half year ago when I took Data Mining, Databases, and Data Warehousing courses from him in the master's program. His has supported me not only by intellectually preparing me with a sound technical background, but also academically and emotionally through the bumpy road to finish this dissertation. He has been a true mentor in shaping who I am today, including emails that he forwarded to me about the academic job openings.

I would like to thank my wonderful dissertation committee: Dr. Jose Dula, Dr. Richard Redmond, Dr. Manoj A. Thomas, and Dr. Heinz Roland Weistroffer. Dr Dula's input helped me restructur my research statement that have been well received by others. Dr. Redmond stopped me the day before Christmas in 2008 to ask about my application to the PhD program, which resulted in my staying at VCU. He has given me tremendous help in making many critical decisions. First time I met Dr. Thomas, I thought he was a random engineering student who was

stealing EMBA food. Later, sharing the same passion for emergent technologies and design science, he becomes my teacher, colleague, and friend. Many of ideas in this dissertation were results of our length discussions and brainstorming. I must also thank Dr Weistroffer for being always supportive as a PhD advisor, and for not letting me quit two years ago.

A very special thanks goes out to Mr. Joe Cipolla, who I truly cherish as a mentor in both academia and industry. His feedback and inputs are always invaluable. I would also like to thank Dr. Wynne, Dr. Kasper, Dr. Andrews, among other VCU school of business faculty who have helped and taught me immensely. I want to thank my peers in the PhD program, especially Yurita Yakimin Abdul Talib who was with me through all ups and downs.

I would also like to thank my family for the support they provided me through the process, and without whose love, encouragement and editing assistance, I would not have finished this dissertation.

Finally, I want to dedicate this dissertation to my daughter, Lianna. She arrived in the middle of my PhD program and has never stopped amazing me and reminding me to appreciate everything in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AR	Association Rules
BI	Business Intelligence
BU	Business Understanding
CBR	Case Based Reasoning
CRISP-DM	Cross Industry Standard Process for Data Mining
DBMS	Database Management System
DM	Decision Maker
DP	Data Preparation
DQ	Data Quality
DSS	Decision Support Systems
DU	Data Understanding
EDW	Enterprise Data Warehouse
GQM	Goal Question Metric
IKDDM	Integrated Knowledge Discovery and Data Mining
IQ	Information Quality
KDDA	Knowledge Discovery via Data Analytics
KDDM	Knowledge Discovery and Data Mining
KM	Knowledge Management
KMS	Knowledge Management Systems
MCDA	Multiple Criteria Decision Analysis
MMS	Model Management System
MOO	Multiobjective Optimization
OLAP	Online Analytical Process
PF	Problem Formulation
PMML	Predictive Modeling Markup Language
RTI	Reproductive Tract Infection
SBO	SmartBoxOne
SDLC	System Development Life Cycle
VFT	Value-focus Thinking

ABSTRACT

NEW ARTIFACTS FOR THE KNOWLEDGE DISCOVERY VIA DATA ANALYTICS (KDDA) PROCESS

By Yan Li,

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 2014

Major Director: Kweku-Muata Osei-Bryson
Professor, Information Systems

Recently, the interest in the business application of analytics and data science has increased significantly. The popularity of data analytics and data science comes from the clear articulation of business problem solving as an end goal. To address limitations in existing literature, this dissertation provides four novel design artifacts for Knowledge Discovery via Data Analytics (KDDA). The first artifact is a Snail Shell KDDA process model that extends existing knowledge discovery process models, but addresses many existing limitations. At the top level, the KDDA Process model highlights the iterative nature of KDDA projects and adds two new phases, namely Problem Formulation and Maintenance. At the second level, generic tasks of the KDDA process model are presented in a comparative manner, highlighting the differences between the new KDDA process model and the traditional knowledge discovery

process models. Two case studies are used to demonstrate how to use KDDA process model to guide real world KDDA projects. The second artifact, a methodology for theory building based on quantitative data is a novel application of KDDA process model. The methodology is evaluated using a theory building case from the public health domain. It is not only an instantiation of the Snail Shell KDDA process model, but also makes theoretical contributions to theory building. It demonstrates how analytical techniques can be used as quantitative gauges to assess important construct relationships during the formative phase of theory building. The third artifact is a data mining ontology, the DM^3 ontology, to bridge the semantic gap between business users and KDDA expert and facilitate analytical model maintenance and reuse. The DM^3 ontology is evaluated using both criteria-based approach and task-based approach. The fourth artifact is a decision support framework for MCDA software selection. The framework enables users choose relevant MCDA software based on a specific decision making situation (DMS). A DMS modeling framework is developed to structure the DMS based on the decision problem and the users' decision preferences and. The framework is implemented into a decision support system and evaluated using application examples from the real-estate domain.

CHAPTER 1 INTRODUCTION

We are drowning in Information and starving for knowledge.

- John Naisbitt

1.1. BACKGROUND

'. . . Knowledge Discovery is the most desirable end-product of computing. Finding new phenomena or enhancing our knowledge about them has a greater long-range value than optimizing production processes or inventories, and is second only to task that preserve our world and our environment. It is not surprising that it is also one of the most difficult computing challenges to do well . . .' Wiederhold (1996)

Knowledge has been widely conceptualized as an important asset for organizational success and competitive advantages. Organizational knowledge management generally consists of four core processes: creation, storage/retrieval, transfer, and application (Alavi et al. 2001). Among various technologies have been developed to support the organizational knowledge management, Knowledge Discovery and Data mining (KDDM) is one of the key enablers for the data-driven organizational knowledge creation. KDDM concerns the entire knowledge creation process through "non trivial process of searching through large amount of computerized data to identifying valid, potentially useful, and ultimately understandable patterns" (Fayyad et al. 1996a).

Recently, the interest in the business application of analytics and data science has increased significantly. This may be attributable to the advancement in information technologies

such as mobile, analytics, big data, social networking, and cloud computing. These techniques serve as both the drivers and enablers for the organizations' adoption and use of analytics. Analytics are defined as "*the analysis of data, using sophisticated quantitative methods, to produce insights that traditional approaches to Business Intelligence are unlikely to discover*" (Sallam et al. 2012). Data science is "*a set of fundamental principles that support and guide extraction of information and knowledge from data*" (Provost et al. 2013).

All three areas (i.e., KDDM, analytics, and data science) are closely related and share some fundamental concepts that deal with extracting/creating knowledge from data to solve business problems. The fundamental concepts of data science are drawn from data analytics and data mining (Provost 2013). In fact, these terms are often used interchangeably, while recent practices give more emphasis on analytics and data science. For example, the SIGKDD (the Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining) has changed its mission statement to "bringing together the data mining, data science, and analytics community". KDnuggets, a widely recognized KDDM community base, has repositioned itself as a "leading site on Business Analytics, Data Mining, and Data Science."

Analytics, however, involves a wider range of quantitative methods than traditional machine learning/data mining methods and algorithms. In addition to the traditional data mining algorithms and techniques such as Tree Induction, Neural Networks, Clustering Analysis, Support Vector Machines, Association Rules, etc., data analytics also include Discrete Event Simulation, Multiple Attribute Decision Analysis (MADA), Mathematical Optimization, and Visualization.

Furthermore, the popularity of data analytics and data science comes from the clear articulation of business problem solving as an end goal. One of the key attributes identified for a

modern data scientist is his or her ability to articulate ill-structured business problem into analytical questions that can be answered by data mining and other analytic techniques. In this dissertation, I adapt the term *Knowledge Discovery via Data Analytics* (KDDA) to describe the knowledge discovery process and practices. The rationale for this adaptation over the traditional KDDM can be summarized as follows:

1. Each different analytical technique has its own unique requirements based on its fit to the business objectives, on its data input and transformation needs, and on its output evaluation and deployment. While the number of analytical techniques continuously grows, a formalized knowledge discovery process shall incorporate all applicable analytical techniques for a specific business problem. KDDA is able to extend the current KDDM practices and capture such a requirement.
2. KDDA provides us with a research lens that focuses on research problems that are relevant to information systems (IS) practitioners and state-of-art of information systems development and implementation technologies. The need for this approach is evident with the increasing demands from companies across industries to hire more people with data analytical skills and the increasing numbers of academic programs to train data scientists. A systematical investigation of the KDDA process shall provide practical inputs for both academia and practitioners.

KDDA solutions are developed and delivered in complex algorithms, metrics, and criteria. In order to use KDDA to solve business problems more efficiently, a process model is desired to translate very technical solutions into business language (Kurgan et al. 2006), which in turn, will be a significant factor in integrating KDDA solutions into organizational business processes. In

the traditional KDDM domain, various process models have been developed in both academia and in industry (Anand et al. 1998; Cabena et al. 1998; Cios et al. 2005; Fayyad et al. 1996a; Rohanizadeh et al. 2009). Among these process models, the CRISP- DM (Cross Industry Process Model for Data Mining) is the most popular one. In the following section, I first present some important definitions for knowledge discovery, data mining, business intelligence, and data analytics. I then discuss the importance of knowledge discovery processes and highlight the need for an updated process model to address the recent changes in the current business and data environment.

1.2. DEFINITIONS

1.2.1. Knowledge Discovery and Data Mining

Knowledge Discovery and Knowledge Discovery in Databases (KDD) is defined as a non-trivial process of identifying potentially useful and ultimately understandable patterns in data (Fayyad et al. 1996a). The discovery process involves different steps and is generally known as data mining. The term data mining has many definitions. According to Merriam Webster dictionary, data mining is "the practice of searching through large amounts of computerized data to find useful patterns or trends." There are two schools of thought on data mining. On one hand, data mining is a set of algorithms, methods, and tools used for analyzing data or extracting patterns. On the other hand, data mining is the process of exploration and analysis of large quantities of data, through computer-based machine learning techniques integrated with statistic algorithms, to discover previous unknown and potentially useful patterns and rules (Linoff et al. 2011). The latter definition refers data mining as the whole KDD process, while the former views data mining as just a step in the KDD process. To incorporate any data source, the term

KDDM has been proposed as the most appropriate term for the overall knowledge discovery process (Cios et al. 2005).

1.2.2. Business Intelligence and Analytics

Recently, the market for Business Intelligence (BI) and Analytics has experienced significant growth and being becoming pervasive in many organizations. BI and Analytics are considered as the top technology priorities for CIOs in the 2013 Gartner Executive Program Survey[1]. However, because of the tendency to interchange the use of two terms, the distinction between BI and Analytics needs to be clarified.

The term BI was coined by Hans Peter Luhn in 1958, as "*the ability to apprehend the interrelationships to presented facts in such a way as to guide action towards a desired goal*." Evolving from Decision Support Systems (DSS), modern BI has been defined by Gartner as "*an umbrella term that spans the people, processes and applications/tools to organize information, enable access to it and analyze it to improve decisions and manage performance*" (Chandler et al. 2011). A typical BI environment includes the following processes:

1. Different types of data from various sources are extracted, transformed, and loaded (ETL) into an Enterprise Data Warehouse (EDW);
2. The EDW is sliced into smaller Data Marts that are oriented towards a specific Line of Business (LOB); and
3. Data Marts provide an information access layer for business users to analyze and report from.

[1] http://www.gartner.com/newsroom/id/2304615

An EDW initiative typically takes six to nine months, and often metrics and BI reporting are not defined properly because end users have not had an opportunity to work with the data. It is also difficult to integrate external data in the existing data mart. Often the BI team faces challenges in answering ad hoc reporting needs and manually looking through data if there are issues related with the reporting result. These manual processes are not scalable when the volumes of data grow exponentially. Hence, many organizations turn into analytics to provide automated knowledge discovery.

Analytics take different forms such as business analytics, data analytics or decision analytics. However, the exact meaning is unclear as it can be used to mean different things in differing contexts. For example, analytics can mean a particular BI technique such as Predictive Analytics; or a business strategy using analytics such as Fraud Analytics; or as an entire analytical domain that includes the hardware, software, personnel, and processes. For this dissertation, I adopt the Gartner's definition of Analytics as "*the analysis of data, using sophisticated quantitative methods, to produce insights that traditional approaches to BI are unlikely to discover*" (Sallam et al. 2012). The data refers to both structured data and unstructured data such as text, image, audio, and video. The traditional approach to BI refers to ad hoc querying, OLAP, and reporting. The examples of sophisticated quantitative methods include advanced statistical analysis, data mining, simulation, optimization, etc.

1.2.3. Different Types of Users

There are different types of users in the KDDA process. It is important to define these users' roles and responsibilities because each type of users has different objectives and different kinds of information needs. First, there are data suppliers, who are responsible for retrieving data

from various resources, preliminary cleaning data, integrating them with certain degree of aggregated form, and transporting them into EDW store. Their role in an organization is usually defined as the ETL developer. The second type of users is BI analysts who interface with the EDW presentation layer to perform BI and data analytics activities. Among these analysts, there are also differences between data analysts and data scientists. Data analysts resume the traditional BI querying and reporting responsibilities. The data scientists or data miners are considered as information power users who actually carry out the KDDA process. The third types of users are end users, or information consumers. They are executives, managers, or Line of Business (LOB) users who have deep business domain knowledge. Traditionally they request and monitor daily reports or dashboards, and occasionally drill down deeper to a specific issue. With vast amounts and variety of data made available at a great velocity, traditional manual BI analysis lacks the capability to fill business needs. The end users, thus, become the main drivers behind KDDA. Figure 1 illustrates how different types of users fit into a modern data-driven decision making environment.

1.3. KNOWLEDGE DISCOVERY PROCESS MODELS

Successful knowledge creation requires an overall approach that describes how to carry out the knowledge discovery process. A process model includes a set of tasks and each task with its inputs and outputs to get the job done (Pressman 2005). A good process model should be effective, maintainable predictable, repeatable, of good quality, improvable, and traceable (Tyrrell 2000). A well-defined and standardized knowledge discovery process model has well recognized advantages. It serves as a blueprint for conducting knowledge discovery projects and

enables the easier deployment (Clifton et al. 2001). It can lead to faster, cheaper, more manageable, and more reliable knowledge discovery realization.

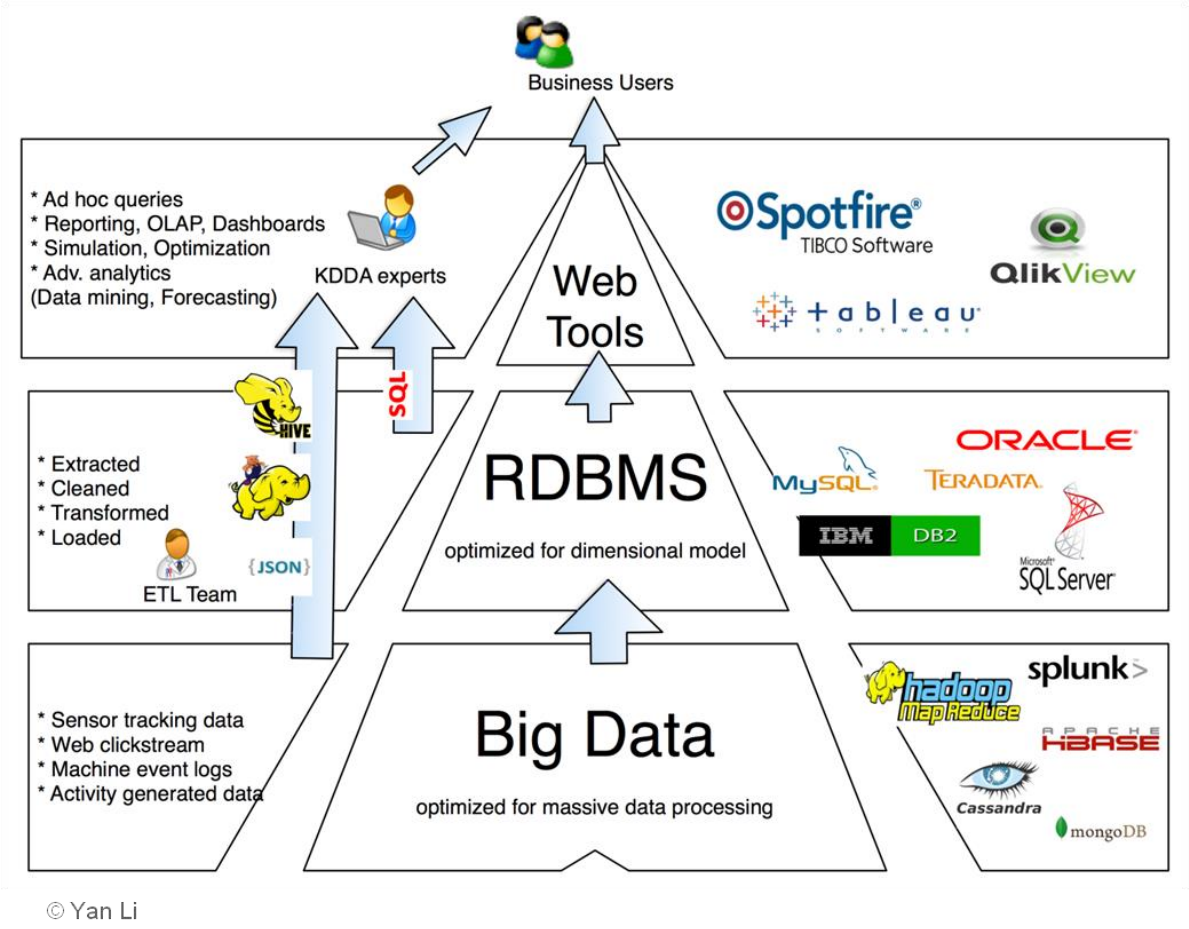

Figure 1: Data Drive Decision Making Environment

Many knowledge discovery process models have been developed in both academia and industry to help organizations understand the knowledge discovery processes and to organize the knowledge discovery projects within a common framework (Marbán et al. 2007). A side-by-side comparison of five major knowledge discovery process models by Kurgan et al. (2006) reveals several common features, although each process model has different number of steps and

different terminologies for each step. For example, the sequence of steps followed in most of the process models is similar, and the processes are iterative in nature.

Among popular knowledge discovery process models, CRISP-DM is the most widely used data mining process model for data mining projects based on the survey conducted by KDnugget.com, a widely recognized Data mining and knowledge discovery community base (kdnuggets.com 2007; Serban et al. 2012). It was first proposed in the year 2000 as an industry-, tool-, and application-neutral standard process model (Chapman et al. 2000)). The CRISP-DM model organizes the DM process into six interdependent phases, namely *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, and *deployment*.

The practical values of CRISP-DM and other comparable knowledge discovery process models are confirmed by its wide acceptance by the industry. However, many changes have occurred in the business applications of KDDM since CRISP-DM was published. These changes include:

1. The new big data environment with large volume, high velocity, and various types of data format demands the scalability of analytical solutions and deployment;
2. The increasing scale of KDDM projects results in the increasing reliance on teams, making it important to educate greater numbers of people on the KDDM processes and best practices associated with data mining and predictive analytics; and
3. The need for real-time analytics posits pressing needs for packaging analytical tasks for non-analytical end users and integrating these tasks in business workflows, including the support for KDDM model creation, maintenance and usage.

Recognizing these changes, in early 2006 the CRISP-DM consortium initiated a call for potential enhancements of CRISP-DM to CRISP-DM 2.0. However, the effort seems to have remained in a frozen stage since no communications were given on CRISP-DM 2.0 since 2007. Nevertheless, the emergent issues mentioned above emphasize the need for an updated process model for KDDA.

1.4. RESEARCH MOTIVATION

KDDA concerns the entire knowledge creation process. In order to perform the successful creation of useful knowledge, an overall model that describes how to carry out the KDDA process needs to be established. Currently, when describing a KDDA project life cycle or a KDDA process, practitioners usually adopt traditional KDDM process models to translate technical solutions (in complex algorithms, matrices, criteria, and so forth) into business language (Kurgan et al. 2006). This is especially true for KDDA initiatives that center on business objectives. An effective KDDA process model can be a significant factor in integrating analytical solutions in organizational business processes.

However, a review of literature on existing KDDM process models reveals certain limitations as well as some missing pieces. Simply adapting a traditional KDDM process model to a KDDA process is not sufficient in addressing the emergent issues identified previously (section 1.3). These limitations and missing pieces call for an update of the existing KDDM process models, which are discussed in the section below.

1.4.1. Lack of Decision Support for Business Understanding Phase

One of the challenging problem in IS research is how to help different types of users avoid many common analytical mistakes by improving the automation of some of the knowledge discovery process (Yang et al. 2006). Serban et al. (2012) surveyed available tools/approaches that provide "intelligent discovery assistant" (IDA) for improving users' experiences in their data analysis tasks. The general idea behind IDA is to build an automated system to advise users in all knowledge discovery stages (Bernstein et al. 2005). However, the IDA approach is limited to knowledge discovery process stages for which there exist automated components and for which their requirements can be well-structured (Bernstein et al. 2005). A more appropriate term, "decision support", is more apt to include semi-automatic support for important but ill-structured business understanding (BU) stages.

BU is considered as the most important phase of any analytical projects and has been highlighted across all existing knowledge discovery process models. BU focuses on understanding business objectives and requirements of a data analytics initiative, and then converting this understanding into a defined analytical problem. Review of existing literature (Bernstein et al. 2005; Charest et al. 2006; Choinski et al. 2009; Engels 1996b; Hilario et al. 2009; Kietz et al. 2010; Lindner et al. 1999; Morik et al. 2004; Panov et al. 2008; Serban 2010) and industry solutions (Weka, KNIME5, Rapid Miner, SAS Enterprise Miner, SPSS Clementine, etc) reveals that current approaches are largely data-centric and modeling technique-centric in providing decision support for the knowledge discovery process. The decision support for the BU is very limited.

CRISP-DM highlights the BU phase with the following key tasks: determining business objectives, assessing the situation, determine data mining goals and produce project plan.

However, the descriptive nature of CRISP-DM only defines what to do (what the outputs of each tasks are), but not how to do (how to facilitate users to obtain these outputs). I first summarize the output areas where decision support functions are needed with very limited research has been done in Table 1, followed by a detailed discussion of why such a decision support is needed and feasible.

Table 1: Limited Decision Supports for Business Understanding Output Areas

Task	Output Area	Decision Support
Determine business objective	Business objectives and business success criteria	Very limited
Determine data mining goals	Data mining goals and data mining success criteria	Very limited
Produce Project Plan	Initial assessment of tools and techniques	Very limited

1.4.1.1. Business Objectives and Business Success Criteria

As mentioned in Section 1.1, KDDA centers on business objectives. Clear articulation of business strategies and objectives are critical to the success of any analytical project (Chandler et al. 2011). The performance of analytical programs is measured by how well they help the business achieve strategic objectives. The need to formally capture business objectives and translate them into business success criteria is not new in IS projects. Most of these needs are carried out by the specific role of business analysts. Business analysis (BA) is defined as a set of tasks and techniques used to elicit requirements from stakeholders to solve a problem or achieve an objectives (Brennan 2009).

There is limited literature on how to provide decision supports to elicit business objectives and define business success criteria based on the business requirements for the KDDA process. In the KDDA process, the organization rarely starts with a clearly defined objective.

The business users are often overwhelmed by the amount of data and hence, require the machine capabilities to discovery problems that human cannot comprehend. However, structured or semi-structured decision support can be provided by utilizing a variety of goal-elicitation techniques, such as influence diagram, Value-Focused Thinking (VFT), and Goal Question Metric (GQM). The ability to capture a KDDA-related business goal in a structured or semi-structured way can facilitate the translation of business objectives to data analytics objectives. If possible, the business objectives can be stored and reused for knowledge management purposes.

1.4.1.2. Data Analytical Objectives and Data Analytical Success Criteria

Business objectives and business success criteria are stated in business terminology, while data analytical objectives and data analytical criteria are stated in technical terms. For example, a business objective might be stated as "to increase the performance or profitability of the customer." The relevant analytical goal might be "to analyze the behaviors of new customers and predict if they are likely to become long-term profitable customers or not", or "to classify the trends of profitable and non profitable customers based on existing purchasing behaviors and their demographic information."

Data mining and other related analytical techniques are knowledge intensive. Current approaches towards providing decision support in the KDDA process are mainly from a knowledge engineer's perspective, which requires a thorough understanding of data analytics algorithms and methodologies. Business analysts, who have skills to elicit business objectives and requirements for analytical projects, are not necessarily experts in the KDDA domain. There is a semantic gap between end users who talk about profitability and churn rate, and knowledge engineers who talk about decision trees and lift values. In addition, there may be subjective

business objectives and business success criteria, which make the translation from business terminology to KDDA terminology more problematic. Such a problem can be accelerated by the impact of real-time analytics when end users are required to make faster and better decisions by applying analytical tasks within their workflows. How effectively turning business requirements into actionable technological solution is a crucial aspect of a successful knowledge discovery (Choinski et al. 2009). Decision support shall be provided in translating business objectives and business success criteria to data analytical objectives and data analytical success criteria, and thus, bridge the semantic gap. If possible, data analytical objectives can be stored with analytical models for retrieval and reuse by end users.

1.4.1.3. Initial Assessment of Tools and Techniques

Initial assessment of tools and techniques is very important in the initial stage of KDDA process. Different tools implement different data mining and analytical algorithms and methods, as well as auxiliary supports for various tasks in the KDDA process such as data understanding (DU), data transformation, data cleaning, sampling, model performance evaluation, and deployment environment. Meanwhile, the selection of tools and techniques is also constrained by outputs from other tasks and stages in the KDDA process. There are hundreds of software for knowledge discovery, data mining, and analytics, from commercial enterprise analytic suites (e.g., SAS Enterprise Miner, IBM SPPS Modeler, SAP KXEN), to free and open source tools (e.g., Weka, R, KNIME). Each tool may provide different analytic techniques. An improper selection of the tools and techniques can be costly and may adversely affect business processes.

The lack of the decision support in selecting and evaluating relevant tools and techniques for various KDDA tasks has been identified in the literature (Charest et al. 2006; Mariscal et al.

2010). Sharma and Osei-Bryson (2009) described how initial assessment of tools and techniques can be performed through interviewing domain experts, reviewing resources in combination with data mining goals, assessing the financial constraints and associated risks, and learning from similar past projects. However, a formalized process to facilitate the tools and techniques selection in KDDA process has not been defined.

1.4.2. Need for an Integrated Knowledge Repository

All types of decision supports require some prior knowledge to be stored in a machine-interoperable format based on a specific decision situation. In the context of KDDA process, the prior knowledge or background knowledge includes but not limited to:

1. Different existing KDDA techniques, or operators, such as operators for data transformation, modeling, evaluation, process control, and so forth;
2. Meta-data on the operators, such as the external properties (e.g., the operators' Inputs, Outputs, Preconditions, and Effects, or IOPE) and the internal properties (e.g., speed, accuracy, complexity) (Mariscal et al. 2010);
3. Meta-data on each KDDA stage (i.e. meta-data about the BU result, input data, data preprocessing, modeling, evaluation);
4. Expert rules about modeling processes (i.e. impute missing values if the data contains missing values); and
5. Previous successful KDDA workflows.

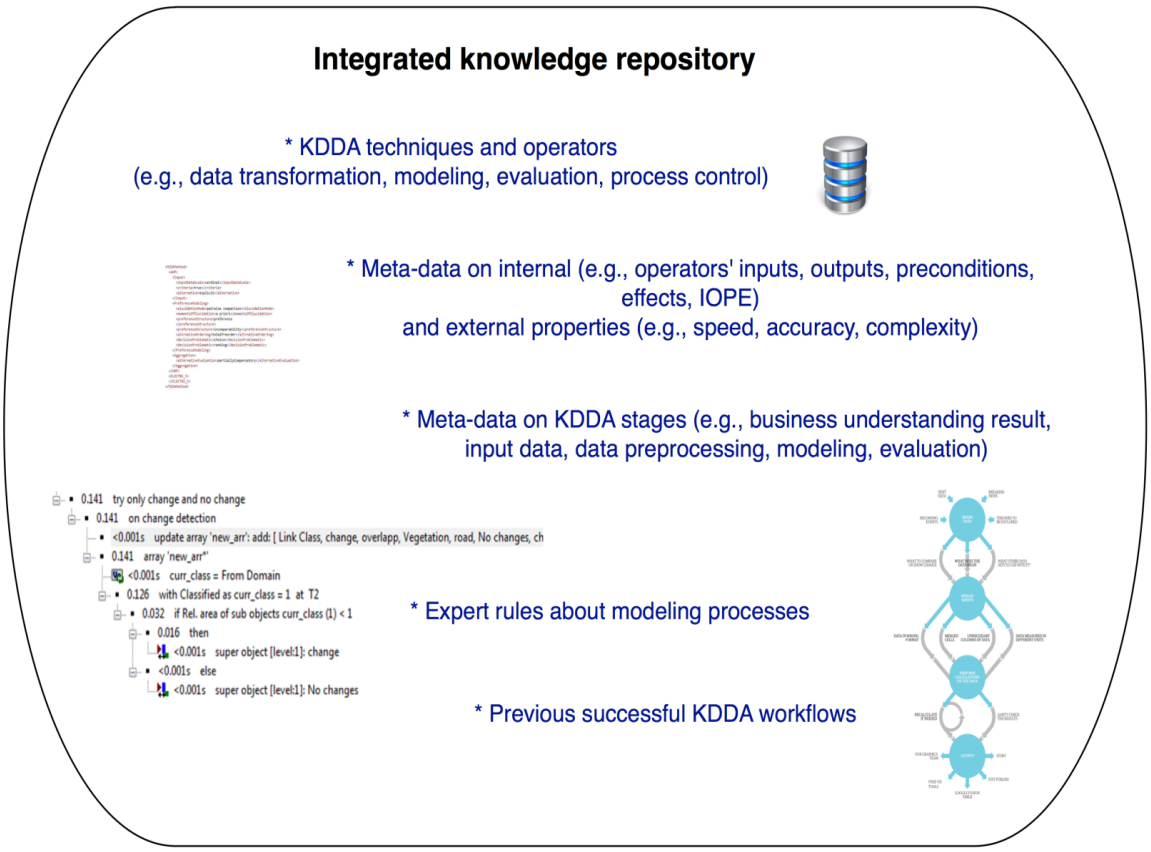

Figure 2: Components of Integrated KDDA Process Knowledge Repository

Figure 2 summarizes the components needed for an integrated knowledge repository for the KDDA process. In traditional KDDM process, all phases and activities are centered on data, which I argue that it should be an integrated knowledge repository. Among these components, existing research has explored the knowledge creation for analytical techniques and operators, the meta-data on internal and external operators, and the storage of previous successful workflows. However, expert rules about modeling processes and meta-data on KDDA stages especially the result from the BU, has attracted only limited attention.

1.4.3. Lack of Decision Support for Data Quality Verification

Once business objectives and KDDA goals have been identified, the next step is data understanding, starting with initial data collection that may be used to fulfill the KDDA project requirements. This requires exploring available data resources (may be both internal and external) that are identified in the BU phase and identifying potential data quality problems. Although the EDW provides a somewhat cleaned and integrated data based on formally defined information requirements, the KDDA process needs to explore data to answer unexpected questions and issues, which are not-predefined in the EDW design. Furthermore, knowledge engineers need to access data both within the EDW and outside of it. There are unavoidable data quality (DQ) issues in the KDDM process.

DQ is best defined as "fitness of use" (Orr 1998; Vassiliadis et al. 2000; Wang et al. 1996). High quality data continuous to be a challenge to ensure that data is fit for use in business processes across the enterprise, ranging from core operations to analytics. Organizations often have DQ assurance programs and DQ steering committees to address DQ issues. DQ is also a well-established research area in academia, with topics ranging from DQ dimensions and assessment (Ballou et al. 1985; Martin 1974; Pipino et al. 2002; Wang et al. 1995), to DQ management methodologies (Jarke et al. 1999; Scannapieco et al. 2004; Wang 1998).

The scale and complexity of DQ issues across organizations require tools to help automate key elements in the DQ management. This has resulted in a substantial market for DQ tools (Friedman 2013) from well established software vendors such as Informatica, IBM, SAP, and SAS. Many ETL tools have designated data quality functions, where data quality rules can be formally captured and integrated within the ETL process. However, the KDDA process often

needs to transport data into an analytical tool (e.g., excel, SAS, R) where the data usually is in a unique format that cannot be integrated with existing DQ management tools.

In addition, different analytical techniques have different requirements in handling data quality issues (e.g., decision trees analysis can incorporate missing values as specific cases while time series analysis does not allow missing values). Traditional DQ evaluation and assessment focus on the physical level of data stores, e.g., edit/imputation methods for maintaining business rules and imputing missing data, and record linkage methods for finding duplicates (Winkler 2004). The analytical aspects of DQ issues are not supported by traditional approaches. The KDDA experts who understand the analytical requirements of DQ are not necessarily data quality experts. The KDDQ process should ideally provide some form of decision support to guide the knowledge engineers in performing DQ analysis tasks.

1.4.4. Missing Model Maintenance and Reuse

Large-scale analytical model development is common in today's business environment. These models are knowledge-intensive information products that are not only very expensive to build, but also very difficult to maintain. As mentioned in Section 1.3, existing KDDM process models follow similar steps, usually starting from BU to DU and preparation, to modeling and deployment. These steps equate to the model conception and creation parts of a model management lifecycle (Krishnan et al. 2000). However, model maintenance and reuse aspects are not captured in the KDDA process. Although there have been attempts to automate the KDDM process to let end users reuse existing models to new data (e.g., SPSS Clementine 5A's process model), they have not proven to be effective and have not been adopted.

While the model maintenance and reuse have not been formally captured in current KDDM process models (such as CRISP-DM), various approaches have been proposed in academia and industry. Liu & Tuzhilin (2008) highlighted three high-level tasks in data mining model management:

1. Model building, which includes semi-automated or automated generation of a large number of models as well as organization and storage of these models in a model repository;
2. Model analyzing, which is to query and analyze models in the model repository; and
3. Model maintenance, which is to keep models in the model repository up to date when the modeling environments or requirements change.

In industry, IBM Intelligent Miner includes model management functionalities in the InfoSphere Warehouse administration console so the users can view, update, export, and delete models in the model repository. SAS Model Manager takes it a step further to register and compare candidate models, as well as model performance monitoring. However, SAS Model Manager is project-based and only one model is registered as a champion per project.

Other challenges in analytical model management include (Ari et al. 2008):

1. Model aging and their predictive performance changes as the business environment changes, and the modeling results need constant updating, performance monitoring and parameter tuning;
2. Hundreds of models are not practical to the manual model management, and hence, certain degrees of automation are desired;
3. The semantic gap between business users and knowledge engineers need to be addressed for better modeling results; and

4. The need for timely communication among business users and knowledge engineers, which would otherwise hinder the value of real-time analytics.

To address these challenges, the current KDDM process models (e.g., CRISP-DM) need to be updated with embedded model management functionalities, especially for model selection, usage, and retire/replacement.

1.5. RESEARCH OBJECTIVE AND SCOPE

In previous sections, I have highlighted the needs for an updated process model for the KDDA to address key deficiencies in existing KDDM models (section 1.4). These key deficiencies are summarized as follows:

1. Lack of decision support for formulating business objectives and business success criteria in KDDM process models;
2. Lack of decision support for formulating data mining goals and data mining success criteria in KDDM process models;
3. Lack of decision support for analytic tools and techniques assessment;
4. Missing an integrated knowledge repository for KDDA background knowledge;
5. Lack of decision support for data quality verifications; and
6. Missing model maintenance and reuse.

The objectives of this dissertation involve two main aspects. First, it is to provide a new KDDA process model based on existing KDDM process models. Second, by breaking down each phase of the KDDA process model, I elaborate on different sets of issues that can be

addressed in each stage and demonstrate how the KDDA process model can be instantiated in certain specific situations. Specifically, I focus on three parts of the KDDA process:

1. How to provide decision support for BU;
2. How to provide model management capabilities in the KDDA process, with a focus on model maintenance and reuse; and
3. How to provide decision support to address the data quality issues during the DU and data preparation process.

Scope: the scope of this dissertation is to address the above three steps. The KDDA phases, modeling, evaluation, and deployment, are excluded from the scope of this dissertation.

1.6. SIGNIFICANCE OF THE RESEARCH

As organizations move towards a data-driven decision making environment, data analytics that includes extensive use of various analytical techniques has become increasingly popular and relevant in IS. Data analytics is a multi-disciplinary approach that is not just concerned with individual analytical techniques or analytical tasks, but a process that involves knowledge discovery. In practice, knowledge engineers and business users desire a formalized process model to guide them in implementing the analytical processes. Without a KDDA process model in place, the traditional KDDM process models will continue to be used even though they have many limitations and weaknesses. The extensive review of literature has also revealed that current decision support for knowledge discovery process mainly covers DU and modeling phases, while the decision support for BU is very limited. The semantic gap between end users

and knowledge engineers also needs to be addressed to accommodate the growing needs for real-time analytics.

This dissertation research will provide a novel KDDA process model that extends the leading KDDM process models. It addresses many limitations within the traditional KDDM models. The significance of this research is two-fold. From an academic perspective, this research will contribute to the IS knowledge base by providing an updated process model for the KDDA process to bridge the current gap between the research and practice. Additional contributions would include as the first attempt to integrate model management capabilities into the knowledge discovery process. From a practical perspective, not only can knowledge engineers take advantage of the KDDA process model and relevant decision support to guide the development and implementation of analytical solutions, but IS management can have a better understanding of KDDA processes, and manage analytical projects within the organization more effectively.

1.7. OUTLINE

The remainder of the dissertation is organized as follows. Chapter 2 provides a literature review of related literature. The literature review includes a review of knowledge discovery process and leading process models, current approaches in decision support for the knowledge discovery process, model management, data quality and data quality management, and two data analytic techniques. In chapter 3, I present the review IS design science methodology and the reason I choose the design science research as research methodology for my dissertation. The guidelines for ensuring the research rigor and relevant are also represented. Chapter 4 to 7 represents the four new artifacts I designed for the KDDA process. Each chapter is written in an essay style. Table 2 summarizes what research objectives that each artifact is designed to address.

Table 2: Summary of New Artifacts

Chapter	Design Artifact	Research Objects
4	KDDA Process Model	To provide a new KDDA process model to address the deficiencies in existing KDDM models and elaborate key tasks in each stage of KDDA process.
5	Methodology for Qualitative Theory Building	To demonstrate how KDDA process model can be instantiated in certain specific situations.
6	DM^3 Ontology	To provide decision support for BU phase, to provide model management capabilities in KDDA process
7	Software and Tool Selection Framework	To provide decision support for BU phase

CHAPTER 2 LITERATURE REVIEW

This chapter describes the related work of this thesis, also providing some related concepts. More specifically, the following topics are discussed:

- Existing process models related to knowledge discovery, described in section 2.1. These existing process models can serve as a baseline design for KDDA.
- Current approaches in providing decision support in the knowledge discovery process, described in section 2.2. This comprehensive review highlights strengths and limitations in existing approaches. The strengths provide the additional insights into various techniques/methods (e.g., ontological approach, case-based reasoning approach, etc.) that can be utilized in the KDDA decision support. The limitations provide motivation for addressing the decision support in the BU phase of the KDDA.
- Section 2.3 provides a review of model management, including its definitions and its modeling life cycle. This section demonstrates that the model management practices can be integrated with the knowledge management process to address the missing model maintenance and reuse in KDDA.
- Data quality, its dimensions and management are reviewed in section 2.4. The rationale to provide a comprehensive review of DQ and related DQ practices is to utilize existing

IS knowledge to provide decision support in data quality verification that is identified in section 1.4.3.

- Section 2.5 and section 2.6 provide a review of two data analytic techniques that do not fall into the traditional KDDM domain. However, they are emergent and important techniques in current KDDA environment. The review demonstrates the unique characteristics of these techniques, as well as their applicability in decision support for KDDA process.

2.1. KNOWLEDGE DISCOVERY PROCESSES AND PROCESS MODELS

Organizational knowledge has been widely conceptualized as an important asset for organizational success and competitive advantages (Nunamaker Jr et al. 2001). Based on the view of organizations as knowledge systems, organizational knowledge management (KM) consists of four socially enacted processes: (1) knowledge creation, (2) knowledge storage and retrieval, (3) knowledge transfer, and (4) knowledge applications (Alavi et al. 2001). The advancement in information technology (IT) provides technical resources for successful (KM) in organizations. Realizing the critical roles of IT in knowledge management (KM) practices, knowledge management systems (KMS) are promoted by IS researchers to support aforementioned organization's knowledge management processes. There is no single information technology can cover support and enhance all aspects of KM.

Rather, a variety of technologies can be used to support different KM processes, including but not limited to decision support systems (DSS) and expert systems (ES), workflow management systems, knowledge repositories and databases, knowledge directories, electronic

bulletin boards and discussion forums, groupware and communication, and data mining (Alavi et al. 2001)

Among those technologies, knowledge discovery and data mining (KDDM) (Reinartz 2002) is a key enabler in knowledge creation. KDDM has well-defined foundations and is a very dynamic and popular research area that is reaching its maturity (Kurgan et al. 2006). In recent years, knowledge discovery and analytics has becomes more relevant for organizations. However, the knowledge discovery involves non-trivial refinement of processes that require not only technical knowledge about methods and algorithms, but also process knowledge about how to correctly and effectively carry out the entire knowledge discovery process. However, there are no existing process models for the KDDA process. Instead, leading KDDM process models are adapted by the practitioners to carry out analytical activities. Despite the limitations, these existing KDDM model have demonstrated their strengths in the field of knowledge discovery. They constitute the knowledge base for the updated KDDA process model. In the following section, I will provide a review of these existing KDDM process models, with an emphasis on CRISP-DM.

2.1.1. CRISP-DM

After an initial proposal of a nine-step development of the standard process model by Fayyad et al (1996a; 1996b), significant progresses in both academics and industry have been made in developing different types knowledge discovery process models. Fayyad's nine-step model (Fayyad et al. 1996a; Fayyad et al. 1996b) is KDD-oriented, with emphasis on data analysis, while several important business issues were omitted. On the other hand, the CRISP-DM is very industry-oriented with a strong focus on the business view. A more recent comprehensive survey of knowledge discovery process models (Mariscal et al. 2010) provided

an evolutionary map of 14 process models, among which the CRISP-DM was a central approach. The CRISP-DM model (Figure 3) organizes the DM process into six interdependent phases. The phases are iterative which means moving back and forth between different phases is always desirable. The inner arrows represent the most important dependencies between phases, while the outer circle illustrates the cyclical nature of data mining. A brief description of each phase is outlined as follows:

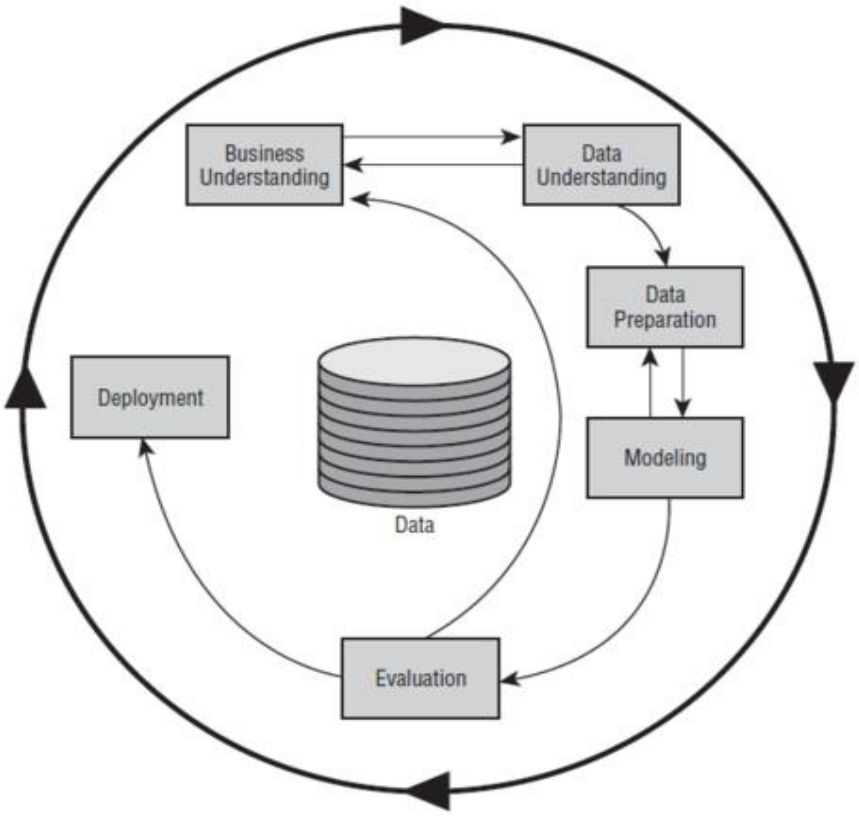

Figure 3: CRISP-DM Process Model

- **Business Understanding**: considering the most important phase of any data mining project, the BU phase focuses on determining business objectives and business success criteria, converting this knowledge into DM objectives and success criteria, and then developing an initial project plan to carry out these objectives. Generally, 50 to 70 percent data mining efforts involves the BU phase (Shearer 2000).

- **Data understanding**: the DU phase starts with initial data collection, and then proceeds with activities related to familiarize the data (e.g., examining data with regard to the relevant requirements, exploring data to gain initial insights, and verifying data quality).
- **Data Preparation**: the data preparation phase evolves all activities in preparing the final data set for modeling, including data selection, data cleaning, data construction, data integration, and data formatting.
- **Modeling**: in the modeling phase, applicable modeling techniques are selected, along with a test design for models' quality and validity, followed by model(s) building and assessment.
- **Evaluation**: before the final model deployment, the data mining result needs to be assessed towards the business success criteria. The modeling process is also reviewed, especially to determine if any important business issues have been overlooked.
- **Deployment:** the last phase in CRISP-DM involves applying learned knowledge in model(s) within organizational decision-making processes. It includes plan deployment, plan monitoring and maintenance, the production of the final report, and the review of the project.

2.1.2. IKDDM

Developed by industrial practitioners, the CRISP-DM model promotes best practices and serves as a blueprint for data mining project. Even though established as the *de facto* standard, the wide acceptance of CRISP-DM in industry confirms its practical values. However, the CRISP-DM model also has its limitations. First of all, because there are numerous dependencies between various phases and tasks in CRISP-DM (Sharma et al. 2009), it is very hard for non-

expert users to create new knowledge from the new data using existing modeling results. In addition, Sharma (2008) identified the following limitations:

- Large number of activities in CRISP-DM is prescribed in a checklist manner that is hard to follow;
- The dependencies between each task are not completely captured, where explication of dependencies can be leveraged as the first step towards semi-automated KDDM process; and
- Lack of decision support towards "how" to implement tasks and activities suggested, especially the lack of decision support in BU phase.

To address aforementioned limitations, Sharma (2008) designed an Integrated Knowledge Discovery and Data Mining model (IKDDM), where all existing task-task dependencies were studied and identified; suggestions were given towards how to execute some of the task dependencies semi-automatically; and a set of clearly defined techniques were proposed for implementing these tasks. The IKDDM process model was presented to guide the business or technical user in end-to-end KDDM process. However, it is impossible to capture a comprehensive list of task-dependencies. In addition, the IKDDM process model did not capture iterations between tasks and phases. Although total sixteen candidate tasks were recommended for automation, the author did not formally implement these automations. Especially, no artifacts have been designed to provide decision support in the KDDM process.

2.1.3. Limitations in KDDM Process Models

One of the challenging problem in KDDM research is how to help users avoid many common data mining mistakes by improving the automation of some of the KDDM process

(Yang et al. 2006). One attempt to address this issue was the five A's (*assess*, *access*, *analyze*, *act*, and *automate*) process model used by SPSS Clementine, but was abandoned by SPSS when joining CRISP-DM consortium. The main contribution of this approach is the introduction of an additional phase in the KDDM process: the Automate phase. The philosophy behind it is to automate KDDM process to let end users use previous obtained models to new data. However, 5 A's model does not establish how to apply the discovered knowledge, and thus, the *automate* phase provides no practical values.

Second, CRISP-DM does not include some of the project management activities, such as quality management or change management (Marbán et al. 2007). The *DMIE* (data mining for industry engineering) process model (Rohanizadeh et al. 2009; Solarte 2002) is the only attempt to add an *on-going support* phase to existing CRISP-DM model, where supports for KDDM model maintenance and updates are available after its deployment. Mariscal et al. (2010) proposed a Refined Data Mining Process that included three high-level processes (analysis, development, and maintenance) and 17 sub-processes based on the synthesis of existing approaches. Nevertheless, the authors focused on the description of sub-processes, without providing a concrete life cycle definition on "how to do" (the order in which tasks to be carried out in each sub-process). All these examples highlight that there is a missing component after deployment phase in CRISP-DM and an update on CRISP-DM is needed.

Furthermore, many changes have occurred to KDDM application in recently years, where there are pressing needs for integrating KDDM tasks for end users in business workflows and for supporting KDDM model creation, maintenance and usage. The use of CRISP-DM has seen a decrease due to the rivalry in-house methodologies developed by KDDM project teams and *SEMMA* (sample, explore, modify, model assess) by SAS institute (Mariscal et al. 2010). This

decrease is largely due to descriptive nature of CRISP-DM, which only defines what to do but not how to do.

2.2. DECISION SUPPORT IN KNOWLEDGE DISCOVERY PROCESS

The KDDA process is not just about data processing and analytical techniques, but also business-centric. The KDDA experts, or knowledge engineers, are desired to have a combination of skill set in analytics, statistics, business, and IT. However, their skills are more skewed towards IT and methodical, process-driven analysis. End users, on the other hand, have extensive business knowledge, but lack of the KDDA domain knowledge. Even knowledge engineers could be overwhelmed by the growing number of methods and techniques available in KDDA. Aggravating the problem is the increasingly large and complex data (recently frequently referred as big data) and the need for real time analytics, where end users sometimes are required to perform analytic tasks on the fly. There is a dire need to support both knowledge engineers and end users in KDDA process.

Currently, various approaches have been proposed in the literature to provide decision support in the knowledge discovery process, such as ontology-based approach, Case-based Reasoning (CBR) approach and workflow management-based approach. However, research in how to provide decision support for the BU phase is still limited. In the following section, I provide a comprehensive overview of research in each of these areas.

2.2.1. Ontology-Based Decision Support for Knowledge Discovery

An ontology is a formal, explicit specification of a shared conceptualization (Gruber 1993). It provides a means to explicitly represent domain-specific knowledge in an interoperable

format that can be understood by humans and machines (Chen 2010). Ontology-based decision support for KDDA has several advantages. First of all, since extensive prior knowledge about KDDA process and techniques needs to be stored and shared, ontologies provide a centralized knowledge presentation and storage (i.e. in a standardized XML/RDF format), and can be extended by others and automatically queried using ontological query language such as SQWRL (Semantic Query-Enhanced Web Rule Language). It can provide a common vocabulary to describe KDDA workflows unambiguously (Mariscal et al. 2010).

Second, as the number of data analytics techniques continuously grow, it is impossible for a human expert to keep up with all the up-to-date knowledge. Rather, a collaborative approach is desired, where individual users can share and upload the background knowledge about KDDA processes. Ontological approach can provide a platform. The DMO (data mining ontology) Foundry[2] can be seen as an initial attempt towards a collaborative KDDA knowledge platform. The goal of the DMO Foundry is to gather different DM ontologies, as well as different DM algorithms and resources that have been developed to support the KDDA process.

Third, KDDA requires cooperation between end users and knowledge engineers, where a semantic gap exists. How to effectively translate business requirements into actionable technological solution is a crucial aspect of a successful knowledge discovery (Choinski et al. 2009). Ontologies can be used as a knowledge representation model to provide a common vocabulary that is shared between different types of users. In addition, they can be integrated with multiple reasoning techniques using declarative logic for effective decision support.

Finally yet importantly, as described in section 1.4.2, an ideal decision support system for KDDA should include an integrated knowledge repository of all required prior knowledge. This repository can be implemented as relational database, or XML databases. XML-based

[2] http://www.dmo-foundry.org/

knowledge storage has its advantages as it can support ontological descriptions of operators, meta-data, and workflows, and allows direct querying with XML queries. Current ontologies are implemented in OWL (Ontology Web Language), which supports XML and RDF schema, but with greater machine interpretability of Web by providing additional vocabulary along with a formal semantics. This interpretability is essential to ensure the extensibility for web-based implementation of KDDA models in a distributed environment (Podpečan et al. 2012).

Currently, there exist several data mining ontologies to support knowledge discovery processes, which are presented below.

2.2.1.1. DMWF3

DMWF3 (Data Mining Ontology for Workflows) is proposed (Kietz et al. 2010) to extract rules from knowledge discovery domain and provide automatic assistant in executing knowledge discovery workflows. It is one of the two data mining Ontologies in eProPlan, an ontology-based planner for planning DM workflows (Kietz et al. 2010). The DMWF stores the IOPE properties of operators in SWRL rules that aim to support the user in checking the correctness of workflows, understanding the goals behind a given workflow, enumeration partial workflows, and retrieval, adaption and repair of previous workflows. The DMWF focuses on the knowledge discovery workflows, which is very data-centric. It concerns "how can data miners navigate the multitude of data mining operators to construct a valid and applicable data mining process (Kietz et al. 2009)?" It does not address the needs from the business users. Also, the automated processes start with DU and ends with modeling, where BU and model management are not included.

[3] http://www.e-lico.eu/dmwf.html

2.2.1.2. DMOP

DMOP (data mining optimization ontology) (Hilario et al. 2009) is the second part of data mining ontology in eProPlan tool. It attempts to present a compendium of knowledge about knowledge discovery tasks, algorithms, data and models, which can be used to support algorithm and model selection. It extends traditional black-boxed meta-learning research (aligning experiments and performance metrics) by adding algorithm features to dataset features as parameters of the algorithm selections. While an ambitious goal is presented to identify the components of inductive bias that characterize each algorithm and algorithm family, such a goal is not quite feasible. Without taking into consideration business objectives that influence the DM objectives, the same set of dataset and algorithm features may result in different algorithm selections if the business objectives are different. Nevertheless, the DMOP and DMWF together provide a controlled vocabulary of semantic annotation of knowledge discovery tools and processes, which can be adapted into ontology design for decision support in KDDA.

2.2.1.3. IDEA

IDEA (the Intelligent Discovery Electronic Assistant) (Bernstein et al. 2005) provides ontology-based decision support in DM process (i.e. phases of preprocessing, modeling and post-processing). The IOPE information of operators is encoded in an ontology. Manually defined heuristics are also encoded, including relative speed of an algorithm, accuracy, and tradeoffs between speed and accuracy. Based on user's objectives, attributes of a set of heuristic functions (such as model accuracy and algorithm speed) are captured and workflows are ranked through a heuristic ranker. The user then can review the result and select a number of workflows to execute. The authors argue that only preprocessing existing variables, induction algorithms, and

post-processing learned models are well-understood domain to be modeled, while other knowledge discovery processes are not well understood and hence, are left out their research scope. In addition, the authors acknowledge that the knowledge discovery process should not be totally automated, as the user-system interactions are critical to successful discovery. I agree with the authors' later assertion that a key in the successful knowledge discovery is to support user-system interaction proactively. However, some of the prior knowledge in the BU phase should and can be modeled.

2.2.1.4. OntoDM

OntoDM (Panov et al. 2008) is a generic ontology designed for describing the DM domain in a set of basic entities:

1. A dataset is presented to have a structure and has data examples; detailed representation is available in figure 1 of the referenced paper;
2. Data mining tasks (i.e. estimation, predictive modeling, pattern discovery and clustering);
3. Generalization which is an output of a data mining task (i.e. predictive model, pattern, a clustering, probability distribution);
4. Data mining algorithms;
5. Components of data mining algorithms (e.g., distance functions, kernel functions, features); and
6. Constraints given which a generalization is to said to be valid.

OntoDM assures the interoperability so it can be easily mapped to other ontologies. In addition, it is possible to use OntoDM to formalize and describe the knowledge discovery workflows (called as KDD scenarios by the authors). However, it is a light-weighted ontology,

which does not include axioms and rules. Hence, it has no-inference abilities. Secondly, collaborative efforts need to be carried out to refine some of the concepts definitions such as generalization and constraints. Last, there are no BU related concepts in the ontology design.

2.2.1.5. Ontological Planning

Ontological Planning (Serban 2010) aims to support automatic knowledge management workflow generation through auto-experimentation of all possible plans to discover heuristics that prune the number of generated workflows. It also aims to allow the user to define some criteria (execution time, accuracy, etc.) to limit the plan search space. It tries to address one of the limitations in existing decision support for the generation of knowledge management workflows, where only limited numbers of operators are supported. Ontology-based planning approach is selected because the ontology offers a hierarchical structure of the DM concepts which is essential in the adopted Hierarchical Task Planning (HTN) (Ghallab et al. 2004). It also enables the user of SWRL rules to define IOPE for knowledge discovery operators. However, how to carry out the auto-experiment remains unanswered. The scalability of the proposed solution is also a concern, even with possible plan-space reduction through user-defined criteria. In addition, the set of workflow selection criteria with related quantitative metrics is not provided. The goal of Ontological Planning is similar to that of DMWF, which is from the knowledge engineer's viewpoints without considerations for business requirements.

2.2.1.6. KDDVM

KDDVM (the Knowledge Discovery in Database Virtual Mart) is a project aiming to provide service-oriented decision support for the design and management of knowledge discovery processes. That is, it should provide an open, collaborative, and distributed support

environment where users can look for implementations, suggestions, evaluations, examples of use of tools as services. In KDDVM, each KDD service is represented in three logic layers (algorithm level, tool level, and instance level). A domain ontology describing algorithms, KDDONTO, is used to provide a broker service for discovery of suitable KDD services.

The key concept in KDDONTO is algorithm, along with other fundamental knowledge discovery concepts as follows: method, phase, task, dataset, parameter, precondition/post-condition, performance (an index and a value about the way an algorithm works), and optimization function. The KDDONTO ontology is implemented in OWL-DL with limited expressiveness. It views the knowledge discovery process as a workflow of algorithms that allows achieving the goal requested by the user. The KDDVM uses a goal-driven procedure to compose the different knowledge discovery processes, namely: (1) dataset and goal definition, (2) process building, and (3) process ranking. Some criteria are defined to rank generated processes: (1) similarity measurement, (2) precondition relaxation, (3) use of link modules, and (4) performance evaluation. This approach is very similar to previous workflow generation approaches with a specific focus on algorithm matching (to find which operator to proceed). Rather than based on heuristic rules, the authors designed an algorithm to iteratively add operators. Although the architecture presented case-based support and versioning support modules, the authors did not give any details of their design. The prototype system seems not be working (http://boole.diiga.univpm.it/KDDDesigner/). The concept of providing support for choosing suitable tools in knowledge discovery is unique, though the authors' definition of "tools" is not clear. It seems that the tools are similar to "operators", rather than DM software tools. Nevertheless, it again did not model any BU activities in its ontology.

2.2.1.7. OLA

OLA (Ontological Learning Assistant) (Choinski et al. 2009) attempts to leverage the business domain and technical knowledge by the uses of ontologies. The authors recognized the need not only to provide technological assistance for business users, but also for knowledge engineers, who is almost impossible to possess all the available knowledge discovery knowledge. An intelligent software platform based on ontologies is proposed to address issues related to communications among business and technology experts and incorporation of structured corporate knowledge and KDDM domain knowledge into the process. This is the only ontology-based approach in this review that attempt to deliver an end-to-end support for the CRISP-DM process from business requirement definition to the model deployment, though this work is rather conceptual (only proposed architecture). Nevertheless, I would like to provide a detail overview of this proposed system architecture, with specific focus on how they address the issue of supporting BU phase in CRISP-DM.

The basic assumption of OLA is the need of close collaborations between domain experts (end users) and technical users (knowledge engineers), where both types of users can take advantage of decision support and real-time advice in knowledge discovery processes. On one hand, business users can focus on their business goals without in-depth technical knowledge of knowledge discovery. On the other hand, knowledge engineers can model the business requirements more comprehensively. The authors give their justification of choosing ontologies as the meta-model for the domain as:

1. Ontologies provide means to describe concepts and their relations;
2. Ontologies are recognized as knowledge representation for IS;

3. Ontologies can be integrated with various reasoning engines for implicit knowledge discovery; and
4. Ontologies can be used to model different user's perspectives, and hence, can be used to present different profiled information to different users based on their specific perceptions.

Hence, even though the authors acknowledge that ontology modeling requires a lot effort, there are no better solutions for their proposed architecture. Three separate ontologies are proposed to model different domain knowledge. First, business knowledge ontology models the enterprise knowledge containing business domains such as finance domain, marketing domain, accounting domain, IT domain, etc. The business knowledge ontology is bounded to the CDM (Corporate Data Model) ontology. The CDM ontology is used to model relevant business rules, business processes, project, strategic initiatives, Key Performance Indicators (KPI), previous analyses and their results, etc. Essentially, all information related to the data selected for analysis should be modeled using ontological semantics. CDM ontology is important because even the most experienced business user may not possess all the relevant enterprise knowledge, or may overlook some important issues in knowledge discovery process. CDM ontology provides an explicit and holistic representation of business domain knowledge, which is essential in knowledge discovery processes.

Second, data mining ontology is used to model taxonomies for DM domain. The authors did not present a new DM ontology design. Rather, they combined the approaches from IDEA (Bernstein et al. 2005) and DAMON (DAta Mining Ontology) (Cannataro et al. 2003). The taxonomies and axioms for DM tasks, methods, algorithms, and software (operators) are

borrowed from DAMON. The sub-concepts of tasks and methods (i.e. preprocessing, induction, and post-processing) are borrowed from IDEA to support CRISP-DM process creation.

Third, CRISP-DM ontology is used to model knowledge discovery processes according to CRISP-DM model. Again, all the ontologies are conceptually proposed. The implementation of those ontological concepts is not an easy task. Without actual implementation, these ontologies are merely conceptual domain models that can be replaced by other representation methods. The real power of ontologies is its inferencing abilities and its interoperability (for both human and machine).

It is to be noted that the authors also incorporate the CBR paradigm (which will be discussed in detail in the next section). Because the successful (or failed) applications of knowledge discovery process are rarely published, it is very hard to capture the semantic knowledge of how to perform a successful knowledge discovery. Rather, the CBR approach can be used to support the process creation, e.g., similar cases from previous projects may be provided to the user in different phases of knowledge discovery.

The authors described how to address the user support in BU phase. First, the OLA will ask the user to document business objectives, data mining goals, project plan and the situation. However, how to guide the user to provide such documentation is not provided (which is rather critical and difficult). While documenting the inventory resources and terminology, the CDM ontology can provide additional insights into the relevant resources. OLA will also check if there is previous project with similar business requirements. The documentation from the BU phase can also feed into DU phase to allow the semi-automatic description of data and verify its quality (based on the input from the users). In a case scenario of churn analysis, the OLA should allow end users to search for all the relevant data through keyword search (such as customer

complains), and browse the result while being presented with relevant business-related characteristics. Approaches in other phases are standard based as mentioned in my previous discussion.

In the propose architecture, business advisor is described as a set of tools to provide intelligent support with relevant business knowledge. The background knowledge of such a support is extracted from the CDA ontology and business ontology. The OLA presents a comprehensive conceptual platform for decision support for the whole knowledge discovery process. However, it is rather ambiguous and the authors did not demonstrate the utility of the proposed architecture. Furthermore, each module in the proposed architecture requires extensive work for implementation.

2.2.2. Case-based Reasoning Decision Support for Knowledge Discovery

CBR is a problem solving approach that is fundamentally different from the traditional rule-based Artificial Intelligent approaches (Aamodt et al. 1994) . A major bottleneck in rule-based approaches is the time-consuming rule assembly process and rule can be in complete. In contrast, CBR approaches assess problem-solving experiences or cases from memory and adapt them to the next new problem-solving situation (Leake 1996). Another important advantage of CBR is that it provides incremental, sustained learning by retaining the new experience each time a new problem has been solved and making it immediately available for future problems (Aamodt et al. 1994). A case-based reasoner has complimentary principles of reasoning by remembering and reasoning is remembered.

The CBR research includes different methods for organizing, retrieving, utilizing and indexing the past cases. Figure 4 shows a general CRB circle where the most similar case or

cases are retrieved and reused, and the solution is then revised based on the reused case(s) where the new experience is retained in the case-base.

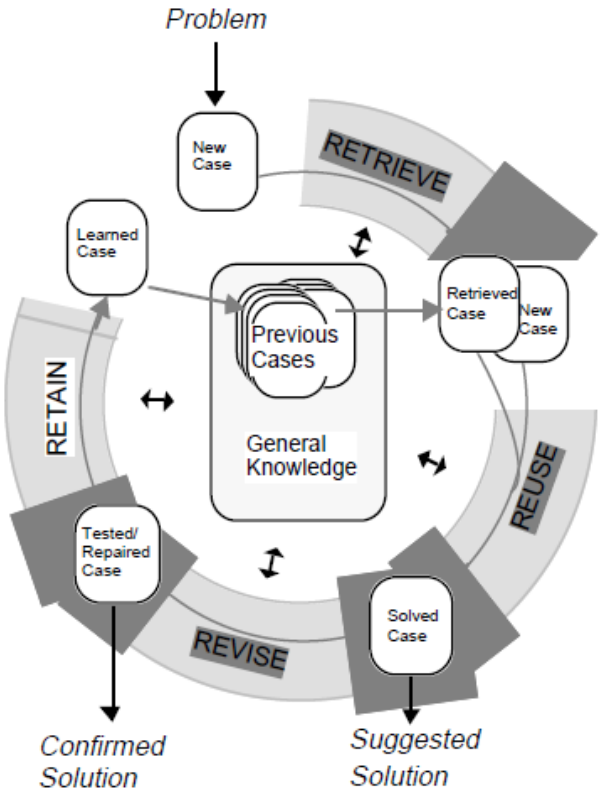

Figure 4: The CBR Circle (Aamodt et al. 1994)

Within the context of decision support for knowledge discovery processes, a general CBR system architecture can be presented as in Figure 5. Human experts are responsible for providing original seeding cases as well as verifying new submitted the cases. When the user's input (including the new problem and user's preferences) is captured through workflow editor, the case-based reasoner retrieves previous workflows from the case base using similar measures between the new problem and prior successful workflows. The user can select one or more of the recommended workflows and make changes if necessary. Upon finishing, the new workflows can then be uploaded into the case base as a new case. This architecture represents a case-based aiding system (Leake 1996) where the case memories provide the experiences that human

experts may lack and suggest successful prior solutions, while human experts maintain the final control. This interactive approach not only avoids the need for automatic case adaption and evaluation, but also increases the user's acceptance of the advices (Leake 1996). In the next section, I will review some CBR-based approaches that provide decision support in KDDM processes.

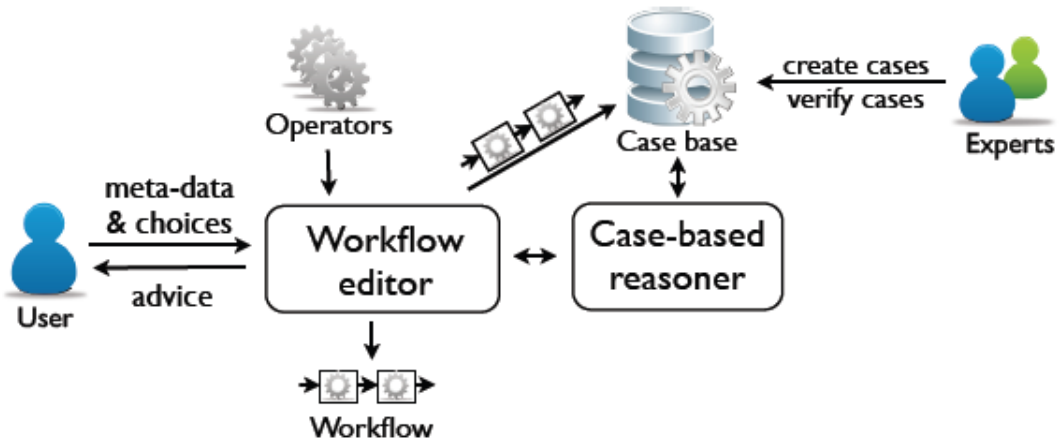

Figure 5: General Architecture for CBR systems (Mariscal et al. 2010)

2.2.2.1. CITRUS

CITRUS (Engels 1996a) is an advisory component for Clementine, now part of IBM SPPS Modeler. The general architecture of CIRRUS extends Clementine by integrating additional algorithms and tools, by providing a user interface for process-oriented support and user guidance, and by providing other interfaces to other software packages.

There are also two important components: information manager and execution server. The information manager supports the DM modeling (especially data selection and preparation tasks) and provides means to retrieve and interpret data and result. Viewing a knowledge discovery workflow (or stream in authors' term) as a special query against the database, the execution server pushes the workflow to the RDBMS server in form of SQL queries and only

result data is sent back the data mining system. A case base of available DM operators and streams (sequence of operators or a workflow) are entered by knowledge engineers and described with pre- and post- conditions. After gathering tasks characteristics from the user, CBR reasoner will load similar cases in the Clementine's workflow editor, where the user can decide to reuse or modify. The CITRUS can also validate the operator selections by removing these who violate any constraints. The CITRUS is the only commercial solution been reviewed in CBR-based approach. However, the details on how cases are stored, retrieved, and reused were not provided. In addition, the focus of CITRUS is limited to data selection and preparation tasks.

2.2.2.2. AST

AST (the Algorithm Selection Tool) is designed (Lindner et al. 1999) to support algorithm selection in knowledge discovery process with a case-based reasoning approach. AST design assumes that the algorithm selection is based on three decision factors: 1) application restriction (user specifies application-specific goals, which in turn, restricts the functionality of a DM application), 2) given data, and 3) existing experience. A case in AST contains the experience on the known applications (e.g., the execution of a special algorithm on a specific dataset), description of the algorithm (i.e. algorithm name, model as interpretable or not, training time from very fast to very slow, testing time from fast to moderate to slow), and data characteristics. The user preference is captured and appended to the data characteristics to form a new case. Three similar cases were then selected using the CBR tool and the user can select to adopt the workflow to the new problem or not. This early approach focuses on algorithm

selection with limited consideration in user preferences and data characteristics. It is a very algorithm-centric approach.

2.2.2.3. MiningMart

The MiningMart[4] (Morik et al. 2004) project is an environment for the knowledge discovery support. The acronym for this approach is sometime presented as KDDSE (OLA uses the similar term to define their architecture). The MiningMart attempts to reuse successful workflows with specific focuses on data selection and data preprocessing in knowledge discovery process. It views the process of knowledge extraction from databases and data warehouses in two themes. The first theme is a meta-model that offers constraints for pre-prosing and pair knowledge discovery tasks with algorithms (or operators) such as feature selection, sampling, transforming, modeling, etc; the second theme is using multi-strategy learning to explore the combination and automatic parameter settings of diverse DM operators for pre-preprocessing. The objectives of MingMart include:

1. Supporting the view of the end-user by the case base, where knowledge discovery tasks are described in business applications as cases and the users can query the case base through MiningMart interface to identify best practice of knowledge discovery from very large and heterogeneous data sets; and
2. Supporting advanced pre-processing by implementing data pre-processing operators (e.g., discretization, attribute aggregation, handling null values), as the improving the quality of data improves the quality of DM results.

The knowledge discovery workflows (for pre-processing) are stored in XML-based language called M^4. It is the meta-model of the meta-data. It is structured in two dimensions:

[4] http://mmart.cs.uni-dortmund.de/caseBase/ChurnPredictionCase/case.html

topic (either data or the case), and abstraction (either conceptual or relational). The case is a sequence preprocessing steps. An ontology is designed to describe all M^4 cases in business terms. The user can search the ontology to select relevant cases.

Three layers of graphical editors are available to map the case to the data stored in a database. The M^4 Relation Editor should be able to map all reasonable conceptual representation of entities to database objects. The M^4 Concept Editor is used to list and edit M^4 conceptual model attributes. The M^4 Case Editor is to support the case designer (a DM expert) to edit workflows by adding or dropping operators. A fixed set of powerful pre-processing operators enables setting up cases, as well as ensuring re-usability of cases. However, MiningMart only focuses on data pre-processing tasks in knowledge discovery process, which highly depend on the outputs of BU phase. One unique idea from this approach is to build a DM case base on the web. In order to collaboratively building knowledge about successful knowledge discovery workflows, a standardized XML-schema needs to be defined to ensure the interoperability across different platform and domains.

2.2.2.4. HDMA

HDMA (the Hybrid Data Mining Assistant) (Charest et al. 2006) takes a hybrid approach to provide decision support in knowledge discovery processes, based on CBR and a formal OWL ontology. This approach has been seen in OLA and MiningMart. It attempts to bright the gap between data mining and decision support in data mining processes. The inherent complexity of data mining posits some challenging questions for novice users (or decision makers in the authors' term), such as which training parameters are most suitable, or which machine learning algorithms should be used. Current software tools provide some wizard-like interfaces (such as

in SAS Enterprise Miner), but the wizard itself requires some background knowledge from the user. Also, the overall data mining process knowledge is tacit and is not directly managed in a form that can be effectively stored, refined and reused. For example, SAS EM simply stores the knowledge discovery process in a flow diagram and stores archived. The knowledge about how the flow diagram is constructed is not stored, and hence, cannot be searched and reused. The authors also make the case for going beyond model selection support (selecting appropriate algorithm for a given tasks) to supporting a user throughout the whole knowledge discovery process. Detailed DM knowledge about "know-how" in knowledge discovery process also needs to be modeled.

2.2.3. Workflow Management Approach

The workflow management approach is the only one that is implemented within the analytics software for decision support. The workflow management here refers to support the user to compose data analytics workflows (and thus, interactively building the analytics process manually) through graphic editing interface, or high-level scripting language. It is different from the workflow management (WFM) defined from the business and information systems perspective, where the workflow management is closely related to reengineering and automating business and information processes in an organization (van der Aalst 1998).

The workflow management in knowledge discovery not only provides a collection algorithm, but also offer some decision supports in workflow constructions (e.g., meta-data propagation, correctness check before execution, operator recommendations, etc.). Two main groups of tools are available to provide workflow management in analytical process. The first group is canvas-based tools, such as IBM SPSS modeler, SAS Enterprise Miner, and the second

group is open source systems (e.g., Weka, Rapid Miner, KNIME). While I am somewhat familiar with SPSS and SAS, the other tools are worth more attention.

2.2.3.1. Weka

Weka (Hall et al. 2009) is an open-source suite of machine learning algorithms (including data processing, classification, association rules, and visualization) for data mining tasks written in Java. The algorithms can be either directly applied to the dataset or called from one's own Java code. It allows the researchers and practitioners to quickly try out and compare different algorithms on the new dataset. Weka has three different types of GUIs. The Explorer interface (Hall et al. 2009) operates in a batch mode, where data can be imported (only flat files are supported), preprocessed, and analyzed using Classify (Supervised Learning), Cluster (Unsupervised Learning), Associate (association rules induction), and Visualize. There is also an additional panel for attribute selection.

Weka also includes a Java-Beans-based knowledge flow GUI for setting up KDDM processes in an incremental manner. This interface is very similar to that of SAS EM and SPPS Modeler. Operators (e.g., data sources, filters, classifiers (or cluster), and evaluators) can be connected graphically and the flow diagram can be saved and reloaded again. The third interface in Weka is called Experimenter, which can be loaded to compare the performance of different learning schemas. Only regression and classification problems are built in the Weka's experiment environment. Evaluation options include cross-validation, learning curve, and holdout. Evaluation result can be written into the file or database. The widespread acceptance of WEKA is evident. For example, Rapid Miner has Weka built-ins and RWeka is built to interface from R to Weka. The open source (therefore free) and light-weighted nature demonstrate the clear

advantages of Weka. However, it has inherently limitations for the number of operators supported, scalability issues (traditional algorithms need to have all data in main member), and as with all open source software packages, reliability issues.

2.2.3.2. Rapid Miner

Rapid Miner is also an open source system, but much more powerful than Weka as standalone software tool. It uses *Processes* to build DM models, where processes are produced from large number of (more than 500) nestable operators (similar to nodes in SAS EM). Processes flow design is described internally by XML. Examples of operators include input, output, data processing (ETL), as well as extensions for text mining, web mining, sentiment analysis (opinion mining), as well as time series analysis. Rapid miner provides a high connectivity to various data sources, such as Oracle, IBM DB2, SQL Server, MySQL, PostgreSQL Excel, Access, and SPSS files, as well as other data formats. Rapid Miner supports windows, Mac, Linux, or UNIX systems though it requires Java Runtime Version 6 or later. An interesting operator in Utility operators group is the sub-process for grouping the sub- processes within processes. Another useful function is the filters for the operator views where one can simply search for the operators. All operators can be drag and dropped into the process view, and can be connected from left to right by click and draw that similar to the SAS EM. It provides the support if IOPE properties are violated, where the operators will not be connected and an error message will show the violation.

2.2.3.3. KNIME

KNIME is anther open source system written in Java and based on Eclipse. It also provides commercial licensing options (KNIME professional) with full technical supports and priorities in enhancement requests. The graphic interface of KNIME is very similar to Rapid Miner and its operators have grown to closer 1000 operators. Examples of KNIME operators can be found on its website[5]. It includes an extension to integrate with Weka. KNIME provides business scenarios with example workflows for download via its public server. Sample business scenarios include Telco churn and retention, credit scoring, social media sentiment analysis, social media text and network analysis, and so forth.

An interesting white paper from KNIME, "Creating Usable Customer Intelligence from Social Media Data: Network Analytics meets Text Mining" presented a case study about how to combine text mining and network mining to provide social media analytics. Text mining has been widely used in social media for sentiment analysis in a given context. However, it does not present the interactions among individuals who express the sentiments. On the other hand, network mining identifies influenced individuals and their followers in the social media through analysis of nodes and connectors. However, it does not provide any contextual information about the people. The two techniques seem to be complementary to each other, but have not been combined until recently. The main reason is that both techniques require unique set of machine learning algorithms, which are often implemented separately. The open source DM platforms provide an opportunity for the user to access both techniques in the same environment.

The new version of KNIME (2.2 or higher) now includes Server Workflow Project view where the users can publish their workflow examples on the public server. The workflows are

[5] http://www.knime.org

organized using different workgroup directories. Each workflow includes meta-information about the background information and required KNIME extensions to run. If annotated properly, these workflows can be integrated with CBR case-bases. Nevertheless, no evidence shows that any metadata about BU phase is captured.

2.3. MODEL MANAGEMENT

A model management system (MMS) is a computerized system that facilitates the creation, storage, retrieval, and utilization of models for the end users' decision making process (Muhanna et al. 1994; Will 1975). The concept of model management systems (MMS) were coined by Decision Support Systems researchers to support decision model creation, execution, storage, retrieval and maintenance (Sprague et al. 1975). The seminal work by Dolk & Konsynski (1984) outlined the purpose of an MMS as to bridge the decision maker's problem domain and a wide variety of models (e.g., linear programming, regression, simulation) without involving the technical and/or procedure aspects of models' implementation. In essence, a MMS is not a user-oriented application, but a reusable component that can embedded into domain-specific applications (Bernstein et al. 2007).

Earlier MMS research mainly focused on model storage and representation. On one hand, recognizing data and models both as organizational resources, where database techniques were adapted store models in a database (Blanning 1982). On the other hand, the knowledge intensive nature of models promote the adoption of knowledge representation techniques to represent models in the model base (Elam et al. 1983). However, those approaches conceptualize MMS as a black box, where models are represented only as inputs (e.g. user's problem statements) and outputs (e.g. appropriate models that matching the problem statement). With its limited

expressive power, the black box approach to MMS did not present unique structures of different models, which leads to problems in model documentation, verification, validation (Fourer 1983), selection (Liang 1988), and integration (Dempster et al. 1991). Aforementioned shortcomings led to the need for providing model management support for the entire modeling life-cycle (Geoffrion 1987).

Table 3: Tasks in the Modeling Life Cycle (Krishnan et al. 2000)

Task	Goal	Mechanism
Problem Identification	Clear, precise problem statement	Argumentation process
Model creation	Statement of the model (s) required to mathematically describe the problem	Formulation ;Integration; Model selection and modification; Composition
Model implementation	Computer executable statement of the model	Ad hoc program development; Use of high-level specialized languages; Use of specialized model generator programs
Model validation	Feedback from validator (*either during model creation or after*)	Symbolic analysis of attributes such as dimensions and units syntax rules
Model solution	Feedback from solver (*it can be part of model creation where a model is either created or failed*)	Solver binding and execution Solver sequencing and control script execution
Model interpretation	Model comprehension Model debugging Model results analysis	Structural analysis Sensitivity analysis
Model maintenance	Revise problem statement and/or model to reflect changes/insight	Symbolic propagation of structural changes
Model versions/security	Maintain correct and consistent versions of models; Ensure authority to access.	Versioning Access control methods

Different fields have their own discipline-specific set of modeling techniques owing to similarities of the technical apparatus they commonly involve (Geoffrion 1989). For example, logic, frames, production rules and semantic network are common modeling techniques in artificial intelligence community (Brachman et al. 1985). Similarly, different data mining modeling techniques (e.g. Decision Tree, Clustering, and Association Rules) are developed in the KDDM community. Nevertheless, each model has a modeling life cycle that spans different stages, from conception, to the creation, use, and eventually retires or replace. Table 3 presents a summary of common tasks in the modeling life cycle. It has some similar tasks as those in CRISP-DM. Viewing DM models as specific type of models, CRISP-DM cover problem identification, part of model creation (model selection, modification, and integration are not covered), validation, and implementation. Model maintenance and security are also not covered in CRISP-DM.

The integration of model management and knowledge discovery process can be viewed from two different perspectives. From the process model perspective, the current knowledge discovery process models (e.g., CRISP-DM) need to be updated with embedded model management functionalities. The notion of PMML (Predictive Modeling Markup Language) provides a mean to model storage, query, and possible selection. However, the meta-data for the updated knowledge discovery process model needs to formally structured and captured to realize the embedded function fully. Thus, PMML schema also needs to be extended. A model management system is not a user-oriented tool (Bernstein et al. 2007). From the tool's perspective, a MMS can be viewed as a reusable component that, if with some customization, can be embedded into a data analytical tool.

2.4. DATA QUALITY

Data Quality (DQ) is a multiple dimensional concept with different proposed definitions. In the early years of information systems research, accuracy was considered to be an important dimension of data quality (Martin 1974). Later researchers further recognized the multi-dimensionality of data quality, and identified other important dimensions such as the data processing quality (Ballou et al. 1985) and systems quality (Wixom et al. 2001).

Instead of the data-centric quality view, a system-design oriented view was applied (Wand et al. 1996) to make a distinction between external and internal DQ views. The external view is concerned with the DQ issues related to the use and effectiveness of the data, such as why the data are needed and how they are used. The internal view of DQ issues is use-independent, and is related to the systems design and operations to achieve the desired functionalities. With the assumption that the external view of DQ is correctly captured by the systems, they focused on identifying a set of intrinsic DQ dimensions that represent internal views of DQ.

Wang (1998) presented an information product view of data quality issues, taking advantage of Total Quality Management (TQM) principles and techniques from the field of product manufacturing. The information is viewed as Information Product (IP) that is the output of the information system using raw data as input, similar to the physical products that are outputs of the assembly lines using raw materials as input.

In this dissertation, the IP view is adopted to present data quality. Within this context, data quality and information quality (IQ) are used inter-changeably. The DQ or IQ can be best defined as "fitness for use" (Orr 1998; Vassiliadis et al. 2000; Wang et al. 1996), which means users are ultimate judgers of the data or information quality provided by the information system.

2.4.1. DQ Dimensions

DQ dimensions are vocabularies to define characteristics of data and information quality. A set of well-defined DQ dimensions is the key in DQ management. Many proposals of quality dimensions have been developed in different context, though there is no general agreement on a fixed set of DQ dimensions.

Based on the assumption that an information system (IS) is a representation of a real-world system as perceived by users, and the distinction of external (use and value) and internal view (design and operation) of IS, DQ dimensions can be related to either internal view or external view. The DQ dimension can also be data-related or system-related. Examples of data-related quality dimensions from an internal view's perspective are accuracy, reliability, timeliness, completeness, etc.

Lee et al. (2002) distinguish four DQ categories: intrinsic (quality in its own right), contextual (quality within the context of the task of hand), representational (quality related to the computer system to present information), and accessibility (quality related to the ability provided by the computer system to access information). An example of intrinsic DQ is Accuracy, contextual DQ is relevance, representational DQ is interpretability, and accessibility DQ is accessibility. Liu and Chi (2002) provided an evolutionary view of DQ through a stage of data collection, organization, presentation, and application. Consequently, the DQ can be classified as collection quality (e.g., from raw data, surveys, observations, recordings), organization quality (e.g., data files, relational databases, data warehouses), representation quality (e.g., web pages, financial reports), and utilization quality (e.g., research data, medical data).

DQ is a subjective phenomenon in that different users might have different quality goals. A goal-oriented approach (Jarke et al. 1999) of DW quality management organizes the quality

dimensions according different types of users in the DW environment. Following the user-centric view of data quality, Bovee et al. (2003) proposed a DQ model to determine the quality of information, the user must: get useful information (accessibility), understand it and find meaning in it (interpretability), applicable to the domain and purpose of the interests in a given context (relevance), and believe it to be free of defects (integrity). In the context of web information integration, Naumann (2002) proposes four sub-categories quality dimensions: content-related (actual data retried); technical (related to the source, network, and user); intellectual (subjective nature of data source); and instantiation–related. Please refer Appendix A for detailed definitions for DQ dimensions from existing literature.

2.4.2. Information Quality and Software Quality

While there seems to be no agreed-upon on data quality dimensions, some software quality requirements are actually standardized. The ISO and ISO/IEC standards related to software quality include the families of 9126 and 14598, within which ISO/IEC 9126-1 specifically defines a quality model for software evaluation. The 9126-1 quality model defines general software characteristics, as well as the different sub-characteristics (Table 4). Each sub-characteristic can be further decomposed into measurable software attributes.

Table 4: ISO/IEC 9126-1 Characteristics and Sub-characteristics

Characteristic	Sub-characteristics
Functionality	Suitability, accuracy, interoperability, security, functionality compliance
Reliability	Maturity, fault tolerance, recoverability, reliability compliance
Usability	Understandability, learnability, operability, attractiveness, usability compliance
Efficiency	Time behavior, resource utilization, efficiency compliance
Maintainability	Analyzability, changeability, stability, testability, maintainability compliance

Portability	Adaptability, installability, replaceability, coexistence, portability compliance

A six-step methodology for building a quality model has been proposed by Franch, et al. (2003) (Figure 6). The six steps are: (1) determining quality sub-characteristics, (2) determining a hierarchy of sub-characteristics, (3) decomposing sub-characteristics into attributes, including some derived attributes that can be further decomposed, (4) decomposing the derived attributes into basic ones, (5) stating the relationships between quality entities (such as dependency, damage, and collaboration), and (6) determining metrics for attributes. From a DQ perspective, the quality characteristics and sub-characteristics can be viewed as DQ dimensions, which can be described using various attributes. The attributes of a quality dimension are defined as quality factors. A quality factor relates a quality value to measurable objects with the quality dimension of the quality factor. Examples of quality factor, quality dimension, and objects are given in the next section.

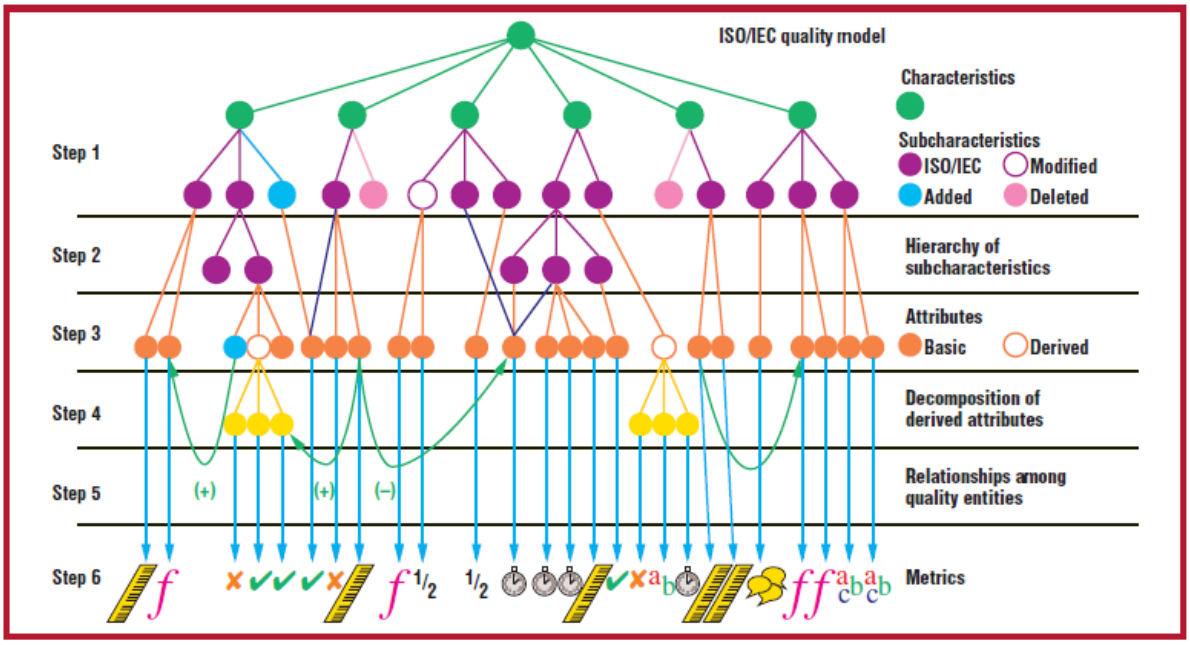

Figure 6: Methodology to build a quality model (Franch et al. 2003)

2.4.3. Data Quality Assessment and Quality Factors

In order to manage DQ, DQ dimension needs to be operationalized so DQ can be evaluated with respect to its dimensions. Traditional DQ evaluation and assessment focus on the physical level of data stores, e.g., edit/imputation methods for maintaining business rules and for imputing missing data, and record linkage methods for finding duplicates (Winkler 2004). The edit/imputation methods verifies if data values satisfy predefined business rules, and the formal mathematical model can be used to ensure all records pass the edit rules without human intervention (Fellegi et al. 1976). Recording linkage, often referred as data cleaning (Elfeky et al. 2002) or object identification (Tejada et al. 2001) uses linkage rules to identify duplicates when unique identifiers are not available. An automatic record linkage generally consists of three different phases: (1) preprocessing to parse data and standardize different spelling or abbreviations in the same format, (2) comparison for matching using string comparators, and (3) using matching algorithms to decide if two compared records are match, not match, or possible match. Both methods are investigated thoroughly in the area of computing statics and operations research. In the traditional DQ architecture, these physical level data is assumed to have been "cleaned" in the ETL process. However, these techniques are still useful when external data sources are incorporated in the DW for decision making.

My research interest in the DQ management is more from an end user's view, with the assumption that all physical level intrinsic DQ issues have been addressed. The concentration is how to capture the subjective DQ requirements or goals from the users, translate these requirements into related DQ dimensions and attributes, and measure the DQ against measurable objects. The user requirement is often ill structured and a structured approach is required to elicit the requirement in a structured format. Furthermore, not all aspects of a DQ goal can be

automatically measured (or computerized). The DQ goal elicitation should be structured in a way to capture the computer-measurable requirements. Goal elicitation is an active research area in requirement engineer, as well as in Multiple Criteria Decision Making (MCDM). Examples of techniques and methods used to help the goal elicitation include influence diagram, value-focused thinking, and Goal Question Metric (GQM). Among these techniques, GQM paradigm has its heart in goal-oriented measurement: "how do you decide what you need to measure in order to achieve your goals?" (Van Solingen et al. 1999) This provides a best fit in the user-focused DQ assessment, which is to bind the measurement problem with the user's DQ goals.

GQM was established in the software engineering to measure software quality. There are two basic assumptions of GQM approach: (1) the measurement program is "goal-based", not "metrics-based", and (2) the definition of goals and measures are individualized (tailored to specific needs). The GQM method contains four phases:

1. The planning phase during which the measurement application is characterized, defined, and planned;
2. The definition phase during which the Goal (conceptual level), Question (operational level), and Metric (quantitative level) are defined and documented;
3. The data collection phase during which the actual data for the measurement are collected; and
4. The interpretation phase during which data is processed with respect to the metric to measurement, the question to answer, and the goal-to-goal attainment.

Briefly, GQM defines a goal, refines the goal into a set of questions, and defines metrics to answer these questions. A template (Basili et al. 1994) is used to define measurement goals in GQM: "Analyze “object of study” in order to "purpose" with respect to “focus” from the point of

view of "point of view" in the context of "environment." The GQM method is incorporated into a quality meta-model (Jeusfeld et al. 1998) for data warehouse quality (DWQ) management , which can be instantiated to assess DQ based on a specific Quality Goal. Appendix B provides a list of commonly used DWQ concepts.

2.4.4. Data Quality Management Methodology

A pragmatic methodology is needed to guide the architecture and implementation of applications for data quality management. In this section, I provide a review of three commonly used DQ management methodologies, which can be incorporated into my design for the decision support for data quality verification.

The framework of Total Data Quality Management, (TDQM) is one of the most commonly adopted DQ methodology (Wang 1998) . TDQM Circle is a pragmatic methodology based on research in production manufacturing quality management and also practical experiences (Wang 1998). In TDQM circle, information product (IP) is the information produced by the system that is the center of quality management. IP has its characteristics and quality dimensions. This is similar to the conception of DW quality management where decision support view (DSV) can be viewed as IPs for decision support purposes. In CIS, the IP can be viewed at owner or local level (each individual organization in CIS network) and enterprise or global level. The global level IP is an integration of local IPs. The TDQM circle includes: Definition (IP and its quality dimensions are clearly articulated), Measurement (information quality metrics are developed based on the quality dimensions), Analysis (measurement result is analyzed to find the root cause for quality issues), and Improvement (key areas for improvement are identified based on the analysis phase result) (Wang 1998).

Extending from the TDQM framework (Wang 1998), TDQM-CIS is a methodology in the context of Cooperative Information Systems (CIS). In a CIS environment, different resources are shared among different organizations and each organization has different levels of quality data and different quality goals. In addition, individual organization in a CIS network usual lacks of control of the others' data sources. In order to achieve a high-level data quality across the CIS network, the quality of exchanged data needs to evaluated and any changes in data quality should be sent to relevant organizations that are interested in those changes. To solve this problem, TDQM-CIS methodology (Bertolazzi et al. 2001) extends TDQM by incorporating an "exchange" phase, where both data and data quality values obtained after "measurement" phase are exchanged among the organizations in the CIS network. Those different organizations in the CIS network can be analogical to different users in the DW, for whom the notifications of quality changes are desirable.

A methodology for data warehouse quality (DWQ) is proposed (Vassiliadis et al. 2000) based on the idea that a goal is operationally defined over a set of questions, which originates from GQM approach (van Solingen et al. 2002a). The DWQ methodology is composed of three phases:

1. A design phase to elicit a quality goal by defining its purpose, the set of questions to solve it, and the set of quality factor (dimensions) to answer these questions;
2. An evaluation phase where values for each quality factor are computed; and
3. An analysis and improvement phase, where the quality goal evaluation result is interpreted and improvement actions are recommended if needed.

2.5. TEXT MINING AS A SPECIAL CASE OF KNOWLEDGE DISCOVERY

Knowledge discovery makes extensive use of data and statistical methods to gain insights and make informed decisions for future business planning. Traditional knowledge discovery focuses on structured data elements, while most estimates show that 80 percent of corporate information originates in unstructured form, primarily in the form of text. The concept of text mining, or text data mining, is a burgeoning new field that attempts to look for patterns in text.

Similar to data mining, text mining is the automatic discovery of new, previous unknown information through unstructured or semi-structured textual data. Text mining is closely related to the research areas of Information Retrieval, Information Extraction and Natural Language Processing (NLP). Text mining applications can have two phases: 1) exploring the textual data for its content and, 2) using discovered information to improve the existing processes. Both are important and can be referred to as descriptive text mining and predictive text mining respectively. Text mining has been applied in search engines, email spam filters, fraud detections, social media analysis, marketing surveys, web content analysis, etc. According to Rexer Analytics 2011 data miner survey summary report, text mining is most often used to analyze customer surveys and blogs/social media, and about a third of data miners incorporate text mining into their analyses, while another third plan to do so.

Text mining can be used in information extraction, such as text summarization and document retrieval. It can also be used in assessing document similarity through text classification and document clustering, etc. Text mining can also take a step further to learn rules from the text. Standard data mining techniques can be employed to generate both prediction rules and association rules. These learned rules could be further applied for automatic textual information extraction. Text mining typically involves the following steps:

(1) Start with a document collection.

(2) Retrieve and pre-process document: document can be retrieved through web-crawler, or some document-bases. The retrieved documents are "parsed" using information extraction and natural language processing (NLP) techniques. Information extraction techniques identify key phrases (terms) and relationships within the text through pattern matching, (e.g., people, places, time, etc). Examples of NLP are parts of speech, synonyms, stemming (treating variation of term as the term itself). The parsed documents then can be transformed into a term-by-frequency (*tbf*) matrix for text analysis. Importance of terms is based on how frequently the terms occur in individual documents and how the terms are distributed in the document collection. Algorithms are designed to determine different types of weighting functions to the term frequency. The basic assumption of term weighting methods is that terms that are useful for categorizing documents are those that occur in only a few documents but many times in those few documents.

(3) Analyze Text: the *tbf* matrix can then be used as input for other data mining techniques. For example, association rules mining can be used to identify commonly associated concepts; concept linkage graph can be used; clustering node can be used to cluster documents into clusters and reports on the descriptive terms for these clusters. Decision Tree and/or other explanatory DM techniques can be used to discover rules within the text. It can also be used (along with some other classification techniques) to classify new documents.

When text mining is applied to the web to crawl text documents on the web, it is sometimes called Web Mining. Text/web mining posits new challenges in KDDM process. In one way, it can be viewed as a special type of DM methods, algorithms, or tools that are used in the KDDM process. The meta-knowledge of this method, algorithm, and tool needs to be modeled and captured. In the other way, text mining can be used as a technique in the BU phase

to capture domain knowledge. For example, text mining can be used to extract terms and concepts from the document corpus on the web and/or corporate document bases, which can be used into domain knowledge generation. OTTO system (Hartmann et al. 2004) is an example of such an application.

2.6. MULTIPLE CRITERIA DECISION ANALYSIS

The field of MCDA can be traced back to Benjamin Franklin in 1700s, who recorded his position on important decisions with the consideration of multiple objectives and trade-offs in making decisions. Earlier history of MCDA also shows influence by researchers from different research areas. For example, mathematicians who provided mathematical foundations of multiobjective optimization (e.g. Georg Gantor), economists (e.g. Francis Edgeworth who developed foundations of utility theory, and Vilfredo Pareto who introduced the concept of Pareto-optimality). Since 1990s, MCDA field has grown significantly, largely due to the increase in computing power that enables application programming for much more sophisticated MCDA methods, which in turn, enable MCDA models to be applied in practice. MCDA research has now penetrated many other disciplines, such as engineering, medicine, etc. MCDA can be viewed as a subfield of Management Science (MS) or Operation Research (OR), or as an important field of its own right (Köksalan et al. 2011).

In the field of IS research, the MCDA research focuses on how to develop and design computerized systems to facilitate the aforementioned MCDA process by implementing the various algorithms/methods proposed in other fields (e.g., operation research). In essence, this objective fits into the decision science (DS) and decision support systems (DSS) research within the IS domain. DSS is a class of information systems and interacts with the other parts of the

overall information system to support the decision making activities of managers and other knowledge workers in the organizations (Sprague 1980). Based on sophisticated MCDA methods and techniques, MCDA software should cover various stages of the decision making process, from problem exploration and structuring to discovering the DM's preferences and the most preferred compromise solution. Research areas that cover both MCDA and IS include application of MCDA methods in different decision situations (such as software evaluation), development of DSS that support MCDM, and more recently, preference modeling in machine learning and knowledge discovery.

2.6.1. MCDA Methods

Classical decision making process model tries to optimize a single objective function over a set of feasible solutions. However, it is well known that naturally the decision is related to a plurality of points of view (or decision criteria), many of which may be conflicting and need to be handled at the same time. As a result, the decision is rather a satisfactory one than an optimal one. In order to understand fundamentals of MCDA process, three basic concepts need to be defined. They are:

(1) *Alternative (or potential actions):* it denotes a set of potential actions that are worth some considerations to a given MCDA process.

(2) *Criteria (or family of criteria):* the first step in MCDM is to build *n* criteria (n>1) so that each potential action according to a point of view can be evaluated and compared against. Different types of scales can be used in evaluating a criterion against a potential action based on views of each decision making situation. The types of scales are characteristics of input

information from the decision makers, which are generally expressed as Ordinal (or qualitative), cardinal (or quantitative), and mixed.

(3) Decision *Problematic*: it refers to the way in which a decision making situation (DMS) is formulated. Roy (1985) suggested four possible decision problematics, which are the description problematic, the choice problematic, the ranking problematic, and the sorting problematic.

The MCDA approach usually consists of four non-linear recursive steps: (1) decision problem structuring, (2) preferences articulation, (3) alternative evaluation aggregation, and (4) solution recommendations (Guitouni et al. 1998). MCDA decision models do not possess a mathematically well-defined optimal solution; therefore, the decision maker (DM) has to find a satisfactory (desirable, acceptable) compromise solution from among many non-dominant (efficient) solutions. Unless the utility function of the DM is known *a priori* and explicitly, interactive solution techniques are imperative to identify the most preferred solution or a manageable set of desirable compromise solutions.

Many MCDA methods are proposed in the literature to solve MCDA problem, most of which require a considerable amount of computation. Increases in computing power have contributed to many of advances in MCDA research. In a comparison of ISI (Institution for Scientific Information) publication in MCDA research (Wallenius et al. 2008) for the years 1970-1990 and 2002-2006 by subtopic areas, the relative share of computer science and information systems research has increased 20%, while relative share of OR/MS decreased by about 40%.

The MCDA methods can be categorized into two groups: multiple criteria design methods and multicriteria evaluation methods (Cho, 2003). The first group of methods is designed to solve multiobjective optimization (MOO) problems, in which a finite number (in

integer problems) or infinite number (in continuous problems) alternatives are i*mplicitly* (assumed to exist but is otherwise unknown) known. The alternatives are defined by a set of finite number of set constraints (criteria) that can be expressed in the form of linear or nonlinear mathematical objective functions. The MOO methods sometimes are referred as continuous MCDA methods. Hundreds of optimization methods are available: each method is intended to solve a specific or more generic MOO problem. Hwang and Masud (1979) provide a systematic classification of MOO methods into four categories based on preference articulation:

1. No preference articulation,
2. A priori preference articulation,
3. A posteriori preference articulation, and
4. Interactive method.

The second group of methods is designed to solve multiple attribute decision analysis (MADA) problems, in which a finite number of explicitly known alternatives are characterized by a set of multiple attributes. The MADA methods sometimes are referred as discrete MCDA methods. Vincke (1989) characterized MADA methods into three categories: (1) the multi attribute utility theory (MAUT) methods, (2) the outranking methods, and (3) the interactive methods. Guitouni and Martel (1998) categorize MADA methods into three similar categories: (1) the single synthesizing criterion approach, (2) the outranking synthesizing approach, and (3) the interactive local judgments with trial-and-error approach. The characterization is based on the multi-criterion aggregation procedures (MCAP), which are considered as the heart of MADA methods (Guitouni, et al., 1998). The single synthesizing criterion approach assumes a value function *U* to model the DM's preferences with the hypothesis that each attribute can be described using a utility function. The aggregation (or calculation) of the function can thus be

obtained in a straightforward manner and the alternatives can be directed ranked. The outranking synthesizing approach has its root in social choice theory (SCT) (Vansnick, 1986), where the alternative is the candidate, the criterion is the voter, the partial preference is the individual preference, and the global reference is the social preference, in SCT, respectively.

2.6.2. MCDA Software

Decision analysis software can assist DMs at various stages of structuring and solving decision problems. These stages can include problem exploration and formulation, decomposition, and preference and tradeoff judgments. Many of the general commercially available decision analysis software have been included in the biennial decision analysis software survey in 'OR/MS Today' since 1993 (Buckshaw 2010), including a range of commercial MCDA software packages. Weistroffer et al. (2005) surveyed both commercial and academic MCDA software solutions based on seven different problem types: problem, structuring, multiple attribute DM, multiple objective DM, sorting problem, portfolio analysis, group decision support, and application specific software. Even though the survey provides a comprehensive set of MCDA software available, it does not provide a general guideline on how to select MCDA software based on a specific DMS.

Many of software packages presented in that survey have been discontinued or not currently supported. Weistroffer and Li (2014) provide an updated overview of the state of MCDA software. This review is structured around several decision considerations when searching for appropriate available software, including decision problem formulation (MADA or MOO) type, MCDA methods implemented, group decision support (GDS) capabilities, and platform supported. It provides an initial pool of candidate MCDA software packages. Pole et al.

(2008) reviewed MOO software available since 1999, focusing on the tools and features that advisable MOO software should contain. The set of features considered in that review include graphical user interface, platform supported, optimization methods, visualizations, capabilities for meta-modeling and validation of models, robust design considerations, parallelization support, and external plug-ins support. Both reviews identified some criteria related to MCDA software selection, though neither provided a formalize approach in MCDA software selection.

From a different angle, an Analytic Hierarchy Process (AHP) based software application (Seixedo et al. 2010) was constructed for selecting MCDA software. In that study, MCDA tools were presented using the similar approach in Weistroffer et al. (2005), and the criteria proposed were similar to those proposed for simulation software selection (Banks 1991). One limitation of that study is that the MCDA-specific selection criteria (e.g., cost, compatibility, user interaction, online help, user manual and tutorials, and free version) are not considered in software selection, e.g., the MCDA methods implemented. Secondly, the selection criteria evaluation is limited to AHP, one of many different types of MCAP. As mentioned in the MCDA method review, AHP is based on single synthesizing criterion approach. If a DM prefers the outranking synthesizing approach or interactive approach to model the MCAP, he or she might not want to use AHP when evaluating the MCDA software. A more general MCDA software selection framework should be able to articulate the decision preferences from the DM and map this preference in the selection result evaluation.

2.6.3. MCDA Software Selection

The growing market of off-the-shelf software packages requires specific considerations in software selection, especially in how to fit the customer's requirements to the software selection

process. An improper selection of software package can be costly and may adversely affect business processes. Literature in software selection mainly focuses on software selection methodology, software selection criteria, and/or software evaluation techniques (Jadhav et al. 2009). Different types of software have been considered in the software selection literature, such as accounting software systems (Adhikari et al. 2004), simulation software (Cochran et al. 2005), ERP systems (Wei et al. 2005), decision support systems (Blanc et al. 1989), COTS (Commercial off-the-shelf) products (Leung et al. 2002), data mining software (Collier et al. 1999), and etc. However, very little research has been done in the area of MCDA software selection.

Methodologies are designed to demonstrate software factors, issues, and processes that need to be taken into consideration during the selection process. A seven-stage generic selection methodology is proposed (Jadhav et al. 2009) based on a comprehensive review of software selection literature. The seven stages are:

1. initial investigation of the availability of packaged software,
2. short listing of candidate packages,
3. eliminating software that do not have required features,
4. using an evaluation technique to evaluate the remaining software,
5. pilot testing the tool in an appropriate environment by obtaining trial copy,
6. negotiate a contract, and
7. purchase and implement most appropriate software packages.

Interested readers can refer that review for a list of software selection literatures. Nevertheless, the methodology should serve as a guideline and not be followed without any

deviation. It should be adapted based on the requirements of the individual decision maker or organization (Patel et al. 2002).

Software evaluation itself falls into the domain of MCDA, where the decision makers make preference decisions over the available alternatives (candidate software packages) based on a set of selection criteria. AHP and weighted sum are two popular MCDA methods applied in the evaluation of software packages (Jadhav et al. 2009). However, as discussed earlier, the software evaluation technique should also map the DM's decision preferences. To best of my knowledge, the preference modeling has not been considered in the literature in software selection and evaluation. A framework for MCDA software selection should not be limited to a specific MCDA technique in the evaluation process. Instead, it should provide recommendations on relevant techniques and let the DM to choose one that fits the situation.

Many literature attempts to provide a hierarchical list of software selection criteria. Selection criteria are compared with different types of requirements from the DMs, such as managerial, political, and software quality requirements (Franch et al. 2003). While managerial and political requirements are often subjective and unique to the individual organization, quality requirements can be standardized. The ISO and ISO/IEC standards related to software quality include the families of 9126 and 14598, within which ISO/IEC 9126-1 specifically defines a quality model for software evaluation. The 9126-1 quality model defines general software characteristics, which in turn include different sub-characteristics. Each sub-characteristic can be further decomposed into measurable software attributes. For a given software domain, a structured quality model can provide a taxonomy of software evaluation criteria and metrics for computing their values (Franch et al. 2003).

CHAPTER 3 RESEARCH METHODOLOGY

Design is essential to the information systems (IS) discipline. Right from the beginning, IT professionals and computer scientists have designed and implemented IT artifacts to manage and support IT or IT-enabled business initiatives. Simply stated, design science is the science of designing artifacts. However, what constitutes the science of design is much more complex. This chapter describes my understanding of design science research in IS and why I adopt design science research methodology to guide my dissertation. More specifically, the following topics are discussed:

- What constitutes the science of design, described in section 3.1
- An overview of existing design science methodological frameworks, described in section 3.2
- The design science approach adopted for this dissertation, described in section 3.3

3.1. THE SCIENCE OF DESIGN

In his seminal book, *The Sciences of the Artificial*, Simon (1969) made a clear distinction between natural sciences and sciences of artificial, or sciences of design, He defined "artificial" as: "produced by art rather than nature; not genuine or natural, affected, not pertaining to the essence of the matter" (1969, p. 4). In his view, the world we live in today is much more an

artificial world than a natural world (p. 4). Artifacts are purposely designed artificial things to satisfy human's goals, and they obey natural laws in their environment. Furthermore, an artifact is an interface between the substance and organization of itself and external environment in which it operates (p. 7). Whenever the environment (either the natural law or the human's purpose) changes, the artifact needs to evolve accordingly.

According to Simon (1969), the ontological assumptions of design science in social science can be viewed as: "'there is an artificial world besides a social world". The artificial world is different not only from the positivistic view of a single and objective reality, but also from the interpretive view of multiple, socially constructed realities. Even though a design science researcher may acknowledge that there are multiple alternatives to a phenomenon, an artifact is grounded to a fix reality for a specific human purpose.

If natural science is knowledge about natural objects and phenomena, Simon (1969) posits "why there cannot also be 'artificial' science addressing the artificial objectives and phenomena?" However, design science problems tend to be intellectually soft, intuitive, and informal, which lead to design being recognized only as a professional activity in the scientific community (Simon 1969, pp. 56-57). Simon argues that the science of design is not only possible but also emergent, which is rooted in two central topics: (1) utility theory and statistical decision theory to define what things "should" be in a set of alternatives; and (2) the body of techniques to search for optimum alternative(s) (Simon 1969, p. 62). Hence, a systematic and formal theory of design science should include both representation of design problems and generation and evaluation of design solutions. In essence, design science believes the knowledge is embedded in both design artifacts and design processes, and the construction of knowledge is through "making" rather than "observing."

3.2. DESIGN SCIENCE IN INFORMATION SYSTEMS RESEARCH

IS discipline is rooted in the creation of information technology (IT) artifacts. While mainstream research in the field of IS still focuses on understanding of phenomena that occur at the intersection of organizations, people, and information technology, the design-based approach to build and evaluate IT facts has gained momentum in the recent years. Simon's theory of design has inspired a paradigmatic school of thoughts to develop a set of methodological framework to guide design science research (Hevner et al. 2004; March et al. 1995; Peffers et al. 2007; Walls et al. 1992). A brief review of each of these design science framework is presented below.

Walls et al. (1992) argue that a prescriptive Information System Design Theory (ISDT) is needed to carry out an effective and feasible design process. They highlighted the differences between natural and social science theories and design theories, and formally defined the ISDT to include two aspects: design as a *Product* and design as a *Process*. When dealing with the design *product*, the design theory should have: (1) a set of meta-requirements to describe the class of goals to which the design applies; (2) a meta-design to describe a class of artifacts hypothesized to meet the meta-requirements; (3) a set of kernel theories from natural or social sciences to govern the design requirements; and (4) a set of testable design product hypotheses that are used to test whether the meta-design satisfies the meta-requirements. When dealing with the design *process*, the design theory should include: (1) a design method that describes the procedure(s) for artifact construction; (2) a set of kernel theories from natural or social sciences to govern the design requirements; and (3) a set of testable design process hypotheses that are used to test whether the design method results in an artifacts that is consistent with the meta-

design. While the components of the design theory were outlined, they did not provide any guidelines on how to carry out design as a scientific research.

March and Smith (1995) proposed four types of designed artifacts as outputs of design science research: constructs, models, methods, and instantiations. Constructs are vocabulary and symbols used to characterize phenomena (the design problems and solutions). Models are orderly constructions of the constructs to describe tasks, situations, or artifacts of the design problem and its solution. Methods are processes to guide the design activities to solve the design problems. Instantiations demonstrate that constructs, models, and methods can be physically instantiated in a working system to perform their intended tasks. They also argue that the IS design research shall intersect both design science and natural science (i.e. the utility of the artifacts as a design science, and theory as a natural science). Thus, they introduced the second dimension of their research framework that is based on the broad types of design and natural sciences research activities. These activities are build, evaluate, theorize, and justify. Build refers to the construction of the artifact; evaluate concerns with the assessment of the utilities of the artifact against its development criteria; theorize refers to constructing theory to explain how and/or why the artifact performs within its environment; and justify concerns with gathering scientific evidence to support or refute the theory. Together, the four types of research outputs and four types of research activities form a four by four framework that describes various design science research efforts. Their framework help to characterize the IS design science research.

Building on March and Smith, Hevner et al (2004) presented seven practical guidelines (Table 5) for design science research in IS and a set of design evaluation methods (Table 6).

Table 5: Design Science Research Guidelines (Hevner et al. 2004)

Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Peffers et al. (2007) proposed a Design Science Research Methodology (DSRM) aiming to provide a commonly accepted framework for Design Science Research and presentation. Building on priori Design Science research, the authors synthesized a DSRM process model that includes six iterative activities: problem identification and motivation, define objectives for a solution, design and development of artifacts, demonstration of the use of artifacts, evaluation of the artifacts' utility, and communication of the design output to the appropriate communities.

Table 6: Design Evaluation Methods (Hevner et al. 2004)

Methods	Examples
Observational	Case Study – Study artifact in depth in business environment
	Field Study – Monitor use of artifact in multiple projects
Analytical	Static Analysis – Examine structure of artifact for static qualities (e.g., complexity)
	Architecture Analysis – Study fit of artifact into technical IS architecture
	Optimization – Demonstrate inherent optimal properties of artifact or provide optimality bounds on artifact behavior
	Dynamic Analysis – Study artifact in use for dynamic qualities (e.g., performance)
Experimental	Controlled Experiment – Study artifact in controlled environment for qualities (e.g., usability)
	Simulation – Execute artifact with artificial data
Testing	Functional (Black Box) Testing – Execute artifact interfaces to discover failures and identify defects
	Structural (White Box) Testing – Perform coverage testing of some metric (e.g., execution paths) in the artifact implementation
Descriptive	Informed Argument – Use information from the knowledge base (e.g., relevant research) to build a convincing argument for the artifact's utility
	Scenarios – Construct detailed scenarios around the artifact to demonstrate its utility

Similarly, Vaishnavi and Kuechle (2007) summarized a general methodology of design science research (Figure 7) that includes five iterative process steps, each step including a research outputs. The five process steps and their outputs are: (1) awareness of problem with a formal or informal research Proposal; (2) suggestion phase where a Tentative Design is envisioned; (3) development phase where the Tentative Design is further developed and implemented into design Artifact; (4) evaluation phase where the artifact is evaluated according

to the criteria implicitly or explicitly presented in the Proposal; and (5) conclusion phase where a "satisficing" Result is consolidated and knowledge gained is summarized.

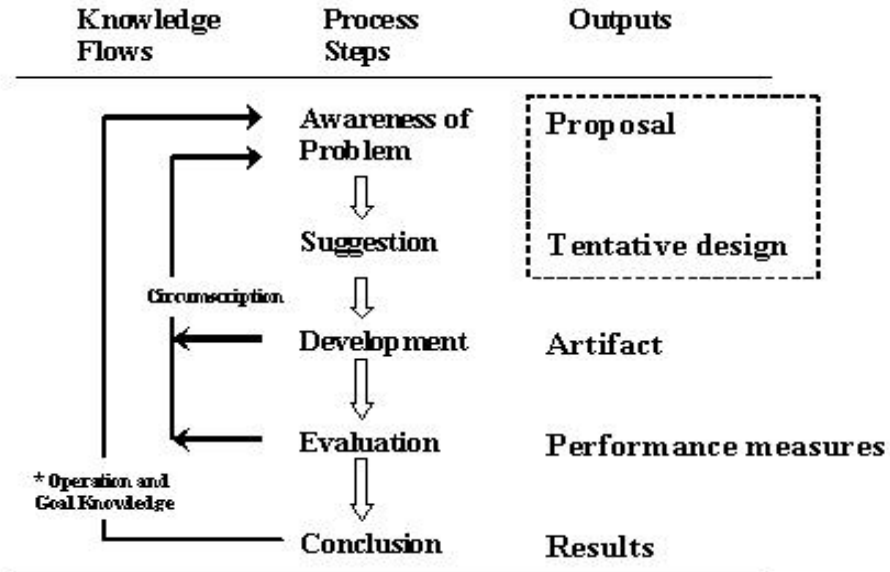

Figure 7: General Methodology of Design Science

3.3. DESIGN SCIENCE RESEARCH FRAMEWORK GUIDELINES

This dissertation undertakes the design science paradigm. In my own word, I summarize the design science research as creating and evaluating IT artifacts to provide a better solution for identified problem to achieve desired human goals. The reason I choose the design science research as my dissertation research methodology is two-folds. The first reason is based on my own philosophical view of the natural of science and the nature of society. I believe that besides the social world, there is also a world of artifacts, or an artificial world. In this artificial world, there are multiple alternatives to solve a specific problem. However, an artifact is grounded to a fixed reality for a specific human goal. I believe that the new knowledge is created through the creation of artifacts, that knowledge creation is through "making" rather than "observing". Second, my own practical view of the field of IS lies in its ability to design and build IT artifacts. While the descriptive IS theories are able to provide a descriptive understanding of the IS

phenomena that intersect organizations, people and information technology, I am more intrigued to expand the boundaries of the known IT solutions and prescribe IT artifacts for a desired future (Boland 2002) .

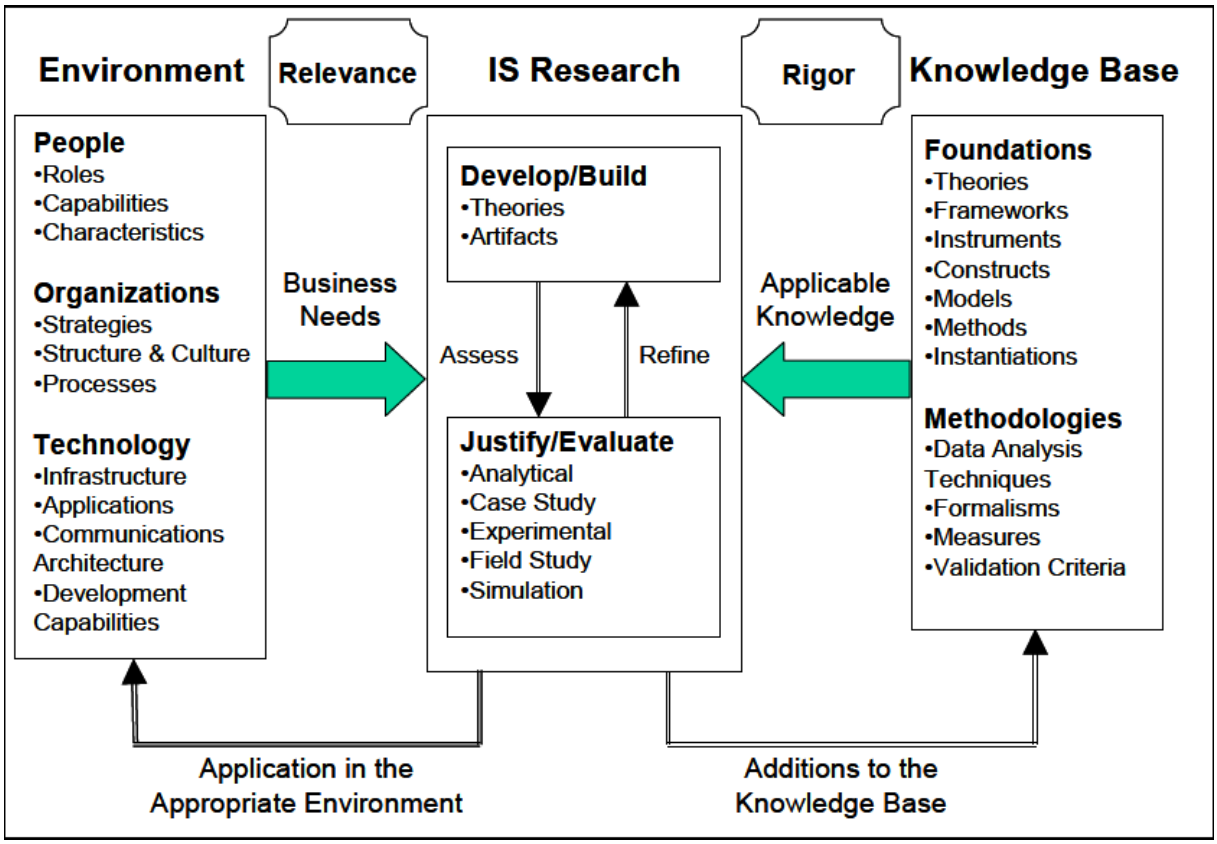

Figure 8: Information Systems Research Framework (Hevner et al. 2004)

This research is guided by the information systems research framework (Figure 8) and seven practical guidelines (Table 5) proposed by Hevner et al. (2004). The research framework illustrates the needs to achieve IS research relevance by framing the research activities to address businesses, as well as the needs to achieve IS research rigor by appropriately applying the existing foundations and methodologies from the knowledge bases. The research activities in design science continuously shift between design as a process (develop/build) and design as an artifact (justify/evaluate). The build-and-evaluate is an iterative and creative process. It is

different from the routine design activities that are normally carried out in organizations, where existing knowledge from the knowledge base are applied to solve organizational problems, such as developing an information system using best practice artifacts. By contrast, a design science research must clearly define its contribution to the archival knowledge base of foundations and methodologies. In the following section, I present a summary on how I will apply the seven research guidelines in this dissertation.

3.3.1. Design as an Artifact

Design-science science research must produce a "*viable*" artifact in the form of a construct, a model, a method, or an instantiation, to address an important organizational problem. The term "*viable*" means that the artifact must demonstrate its effectiveness, by enabling its implementation and application in a specific domain.

Based on the research objective described in chapter 1, an integrated KDDA process model shall be developed from this research. This KDDA process model is an artifact (model) that addresses an important organizational problem, which is the need for a process model to guide the effective, reliable, faster, and cheaper knowledge creation through KDDA. Furthermore, additional artifacts need to be designed to provide decision support in BU and data quality verification, and provide model management capabilities. These artifacts can be constructs (such as designing a KDDA ontology to characterize the KDDA BU and model management domain), methods (such as a detailed process to guide the KDDA activities to solve a KDDA problem), and instantiation (to demonstrate that the proposed models, constructs and methods can be physically instantiated in a working system and to demonstrate their utilities).

3.3.2. Problem Relevance

The objective of design science research is to develop and implement technology-based solutions to address important and relevant business problems. The problem relevance shall be demonstrated through it applicability to a community of practitioners who plans, manage, design, implement, operate and evaluate information systems and technologies that enables the IS implementation.

In the previous chapters, I have demonstrated that the research objectives represented in this dissertation relate to very important and relevant business problems. Its relevancy to practitioners can be summarized as follows:

- Currently, practitioners adapt the existing KDDM process models for data analytics, as there are no existing models. The proposed KDDA process model can be applied by practitioners in real world analytical projects to address many limitations in previous KDDM process models.
- The practitioners can utilize the decision support functionalities for BU and DU to carry out analytical tasks more effectively and efficiently.

3.3.3. Design Evaluation

Design evaluation is a crucial component of the design science research. The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods. As shown in Figure 8, the design activities are iterative and incremental between construction and evaluation, where the evaluation phase provide a better understanding of the problem and valuable feedback to the construction phase so that the quality of the design process and the design artifacts can be refined. The final design artifact must demonstrate its

completeness and effectiveness by evaluating whether it satisfies the requirements and constraints of the problem it intended to solve. The design science artifacts can be evaluated through the evaluation methods presented in (Table 6).

Because the proposed KDDA process model is a new concept, its implementation may require many organizational changes. As a result, the observational evaluation (e.g., case study of the artifact) in a business environment is not feasible within the scope of this dissertation. I will first utilize the relevant evaluation methods to demonstrate the usefulness and viability of the KDDA process model. The relevant evaluation methods that will be used are *descriptive*, *analytical*, and *experimental*. It is reasonable to apply these evaluation methods because the proposed KDDA process model has not yet been adopted by the industry. These methods allow us to test and improve the designed artifacts. Future research can deploy the KDDA process model in real-world setting, which can provide empirical assessment of whether the KDDA process model is useful for practitioners through observational methods.

3.3.3.1. Descriptive Evaluation

To evaluate the KDDA process model, I will use *informed arguments* by utilizing the related research from the knowledge base to build convincing arguments for its utility. *Scenarios* around the KDDA process model will also be constructed to demonstrate its utility. The scenarios will be taken from my real world experience as a data scientist to demonstrate the utility of the KDDA process model in solving real world problems.

3.3.3.2. Analytical Evaluation

The static analysis will be utilized to analyze the artifacts for their statistic qualities. The statistic qualities will be evaluated through two methods: feature comparison and quantitative survey to assess usefulness of the proposed model. Feature comparison is a method of discursive evaluation of the artifact with a checklist of the requirements that it shall meet to be a satisfying solution to a problem (Siau et al. 1998). For example, as identified in section 2.2.1, an ontology-based design can be used to provide the missing decision support in the KDDA process. The designed ontology (a construct) can be evaluated against a set of pre-determined ontology design criteria (design requirements) to demonstrate its utility.

In order to evaluate the KDDA process model, I adopt the evaluation criteria for assessing conceptual model quality proposed by Maes and Poels (2006) to assess its quality. Maes and Poels' (2006) model is based on the Delone and McLean's IS success model (1992) and Seddon's (1997) re-specified model of IS success. The Mae and Poel's (2006) model incorporates the same dimensions as Seddon's (1997), which are *perceived ease of use*, *perceived usefulness*, and *user satisfaction*. However, the information quality dimension of the Seddon's model is replaced with *perceived semantic quality*. The reason of this replacement is that the information quality of a conceptual model will be conceived by the users as the semantic quality, i.e., how valid and complete the semantics of the conceptual model with respect to (the users' perception of) the problem domain. Valid implies that the conceptual model semantics convey correct and accurate information of the problem, whereas completeness means that the conceptual model includes all information about the domain that is considered correct and relevant. Table 7 shows the instruments proposed by Maes and Poels (2006) for Perceived Ease of Use (PEOU), Perceived Usefulness (PU), User Satisfaction (US), and Semantic Quality (PSQ)

construct. The language can be modified to include designed artifacts, such as KDDA process model, instead of the conceptual model in the original instrument.

Table 7: Measurement instruments proposed by Maes and Poels (2006)

PEOU1	It was easy for me to understand what the conceptual model was trying to model.	PU1	Overall, I think the conceptual model would be an improvement to a textual description of the process.
PEOU2	Using the conceptual model was often frustrating.	PU2	Overall, I found the conceptual model useful for understanding the process modeled.
PEOU3	Overall, the conceptual model was easy to use.	PU3	Overall, I think the conceptual model improves my performance when understanding the process modeled.
PEOU4	Learning how to read the conceptual model was easy.	PSQ1	The conceptual model represents the process correctly.
US1	The conceptual model adequately met the information needs that I was asked to support.	PSQ2	The conceptual model is a realistic representation of the process.
US2	The conceptual model was not efficient in providing the information I needed.	PSQ3	The conceptual model contains contradicting elements.
US3	The conceptual model was effective in providing the information I needed.	PSQ4	All the elements in the conceptual model are relevant for the representation of the process
US4	Overall, I am satisfied with the conceptual model for providing the information I needed.	PSQ5	The conceptual model gives a complete representation of the process.

3.3.3.3. Experimental Evaluation

The utility of the artifact can also be evaluated through building a prototype system to test various aspects of the design and illustrate design ideas or features (Sommerville 2007). For example, the decision support functionalities can be instantiated in a prototype, preferably, web-based decision support system. The prototype system constitutes an experimental environment for obtaining early user feedback about the artifact's usability. Building the prototype system can also address various issues being raised in its building process, and may provide valuable feedback to improve the artifact design the prototype intends to present. The usability testing of the prototype system shall be evaluated by KDDA domain experts and users of the artifacts using structured survey instruments (similar to the one listed in Table 7).

The human subjects' input is indispensable for demonstrating the effective decision support capabilities that are realized by means of the information technology. I plan to use a mixed-mode survey using both online and paper questionnaire with multi-item scales. The research design will follow Dillman (2011) principles in ways that reduce errors related to coverage, sampling, measurement, and non-response. The same questionnaire will be presented to two types of subjects: KDDA domain experts (practitioners in KDDA fields) and business domain experts (end users). For the KDDM domain experts, the URL to the web-based prototype systems and the URL to the questionnaire will be sent to the users' emails. Paper questionnaires will be sent out to those who may not respond to the web survey. For business domain experts, a face-to-face tutorial of using the prototype system will first be given. Then paper-based questionnaires will be administrated after the system usage.

For KDDM domain experts, we expect them to be identified from the following sources: (1) VCU MBA students who are working professionals, (2) personal business relationships of

VCU faculty, and (3) my personal business relationships in the analytical space. VCU MBA students will be recruited through email notices sent out with permission from the department of Information Systems (IS), or through a 5 minute briefing session at the beginning or end of a class session (permission from corresponding professor will be sought beforehand). Emails are already available for the subjects that are going be recruited through my personal business relationships. For the subjects that are going to be recruited through the business relationships of the VCU faculty, the faculty member would first get the consent from the potential participants, and then forward their email to me. The prospective subjects will be made aware of the nature of the dissertation project and its specific aims, in addition to providing any supplementary information that may help address any questions they might have. Their consent will be sought and the relevant socio-behavioral consent template will be used to seek and document their consent.

3.3.4. Research Contribution

The effective design science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies. A design science research contribution (March et al. 2008) requires:

1. identifying and clear describing a relevant organizational IS problem,
2. demonstrating that no adequate solutions exist in the IS knowledge base,
3. developing and presenting a novel IS artifact that addresses the identified problem,
4. evaluating the IT artifact rigorously,
5. presenting the contributions to the IS knowledge base and to practice, and
6. explaining the implications for IS management and practice.

In previous chapters, I have identified and clearly described the relevance of research problem, as the current KDDA initiatives lack a formal and effective process model that can serve as a blueprint. The existing KDDM process models from the IS knowledge base are not adequate to address the need of KDDA. I will develop a novel KDDA process model, as well as artifacts that providing decision support in the KDDA process. The designed artifacts will be evaluated rigorously using the evaluation methods identified in section 3.3.3. The model will add value to the IS knowledge base about the knowledge discovery and data analytics process and the decision supports that can be provided in various aspects of this process. The KDDA practice can benefit from a systematic KDDA process model to avoid the *ad hoc* development and implementation of KDDA solutions. The practical implications for the IS management is evident. Not only will KDDA experts, or knowledge engineers, benefit from the design artifacts to perform analytical tasks more completely and rigorous, but the KDDA process model can provide the business users a template to manage an analytical project life cycle more effectively.

3.3.5. Research Rigor

The design science research shall be conducted by applying rigorous methods in both the construction and evaluation of artifacts. The rigor of research will be derived from the use of knowledge base, e.g., applying the proven design science research framework to guide the research process, utilizing proven KDDM process models as a design base for the KDDA process model, and using adequate evaluation techniques as outlined in section 3.3.3.

This design process in this dissertation will use relevant theories, concepts, and best practices in IS knowledge base, including subject areas such as knowledge discovery, data

mining, data analytics, and ontology. The artifacts will be evaluated using appropriate and rigorous evaluation methods to justify as satisfactory solutions toward the research objectives.

3.3.6. Design as a Search Process

Design is essentially an iterative search process to discover an effective solution to a process. The search for an effective artifact requires utilizing available means to reach desired ends while satisfying the laws in the problem domain (Simon 1969). However, it may not be feasible for determining, let alone explicitly describing, all available means, ends, or laws (Vessey et al. 1998). Thus, the design science research often decomposes a problem into simpler sub-problems, and the solutions to the sub-problems can be treated as a starting point for future research to solve large problems by expanding the scope. It is sometimes also impossible to generate and test all possible design alternatives. In these cases, heuristics can be used for constructing artifacts satisfying, i.e. *satisficing* (Simon 1969), the specified class of design problems.

This design process in this research will be developed in a progressive manner. In the beginning, only the BU phase of the knowledge discovery will be investigated thoroughly, where possible solutions will be explored to address the issues identified in the BU phase. There may also some inter-dependencies among tasks in various stages in the initial exploration. These inter-dependencies will be captured and expanded in the next iteration. In each of the iterations, alternative solutions will be tested and evaluated, and feedback will be provided to the next iteration. The search process will be continuous until all identified limitations/definitions have been addressed in the designed artifacts, which represent satisfactory results towards the set research objectives identified in section 1.5.

3.3.7. Communication of Research

The result of design science research must be presented effectively to both technology-oriented and management-oriented audiences. Sufficient details shall be given to the technology-oriented audiences concerning how the artifacts can be constructed and implemented within a specific organizational content, so that they can build upon the design artifacts for future extension and evaluation. In order to present the design research effectively to the management-oriented audience, the importance of the research problem and the novelty and effectiveness of the solution approach need to be emphasized.

The results of my research will be presented to both technology-oriented and management-oriented audiences. The technology-oriented audiences will be presented with detailed information of the KDDA related techniques and concepts, as well as relevant tools, methods, and technologies used in building the artifacts. The management-oriented audiences will be presented with results demonstrating the utilities of the KDDA process model and its effectiveness in solving the set of research objectives.

CHAPTER 4 A SNAIL SHELL KDDA PROCESS MODEL

In this chapter, I propose a KDDA process model based on existing KDDM process models (Sharma et al. 2012; Shearer 2000). The process model is an abstract project life-cycle representation of the KDDA process. This abstract representation is a meta-level model of the KDDA process itself. It provides a collection of key concepts (steps, phases, stages, or guidelines) that describe what happens in the KDDA process. The project life-cycle representation of the KDDA process is important to the business users and management, as it highlights the differences between a KDDA project and traditional software development or data management projects.

Furthermore, a process model (Feiler et al. 1993) shall include a set of process steps (tasks) and process elements (activities). Similar to existing KDDM process models, the KDDA process model consists of two levels of abstractions: phase and generic task. The top level organizes KDDA process into several interconnected phases. The second level describes generic tasks within each phase. The descriptions of generic tasks are presented in a comparative manner, highlighting the differences between the new KDDA process model and previous KDDM process models. The purpose of the KDDA process model is to serve as a blue print for carrying out KDDA projects. Thus, the generic tasks described are intended to cover all possible data

analytic situations. Once the KDDA process model is instantiated as a KDDA process, it can have more specialized tasks and process instances.

The proposed KDDA process framework is evaluated using the evaluation approach identified in section 3.3.3.1, namely *informed arguments* and *scenarios*. During the artifact design process, existing KDDM process models are reviewed and gaps are highlighted. The evaluation approach demonstrates that the proposed KDDA process model addresses the highlighted gaps. It also demonstrates how the KDDA process model differs from existing knowledge management frameworks and how it adds additional contributions to the IS knowledge base. The *informed arguments* are ongoing activities during the design process, providing means of iterative improvements to the artifact design. The KDDA process model is also evaluated using case *scenarios* from real world KDDA projects in a leading media and technology company.

The rest of this chapter is organized as follows. I will first summarize the research motivations for a new KDDA process model, which is mainly to address the change in current business and data environment and limitations in existing KDDM process models. An overview of different types of analytics will be provided to highlight the different analytical problem types and the need for analytical capability assessment. I will then provide an overview of the proposed KDDA snail shell model at the top level. The model is conceptualized as a snail shell to emphasize the highly iterative nature of the KDDA process. Compared to traditional KDDM process models, the KDDA Snail Shell Model includes two additional phases in the life cycle of the KDDA project, namely, problem formulation and maintenance. After describing phases in the KDDA process model, I will provide a description of the generic tasks for each phase and point out how they are different from the traditional KDDM process models.

4.1. NEED FOR NEW KDDA PROCESS MODEL

As discussed in previous chapters, many changes have occurred in business applications since CRISP-DM, IKDDM, and other traditional KDDM process models have been published. Existing KDDM process models may not reflect these changes and suitable for data analytics. I summarize the need for new KDDA process model as follows:

- The popular big data environment imposes a paradigm shift in the traditional data management landscape. The big data environment is characterized by its volume (terabytes or even petabytes of data), variety (structured, semi-structured, and/or unstructured data types), and velocity (ability to collect machine logs, sensors, and web click streams makes frequency of data delivery near real-time). While big data platforms such as Hadoop, Cassandra, MongDb provide the gigantic statistic samples and data with the finest gratuity, there exists many managerial challenges. The big volume challenge can be divided into two categories: SQL analytics directly on databases and advanced analytical techniques. While the formal has been well served in the practitioner world (e.g., Hive, Pig, Jason Scripts, etc.), the later has made little progress. The big variety challenge posits additional data integration requirement and data source governance. The big velocity challenge requires much faster turnaround for analytical projects and much more frequent updates to previous built analytical models.
- There are also increasing need for real time analytics, in areas such as digital marketing, smart homes, online recommendations, real-time intrusion detection, etc. While traditional KDDM process takes considerable amount time to develop data mining models, real-time analytics should enable end users to perform analytic tasks on the fly.

- There are missing model maintenance and reuse components in the existing KDDM process models. Analytic models are knowledge-extensive products that are not only expensive to build, but also expensive to maintain.
- There is the lack of decision supports in the analytic process, especially in problem formulation and BU.
- Existing KDDM process follows traditional SDLC model where majority of the business requirements are gathered in the beginning of the projects. However, the analytic project usually starts with ill-structured business problems. To understand and gather requirements to understand ill-structured business problems often involves multiple iterations. Current KDDM process models are not able to reflect these iterations.
- There are different types of analytics as discussed in the next section. The organization's analytical capability directly influences its ability to answer the kind of analytical questions. For instance, an organization is in the early stage of adopting descriptive analytics (such as dashboard and OLAP report) would not be able to answer the question of "what will happen". Currently, the analytic capability assessment is missing in existing KDDM process models.

4.2. OVERVIEW DIFFERENT TYPES OF ANALYTICS

As defined in chapter 1, analytics uses sophisticated quantitative methods to discover novel insights in data. Generally, there are four types of analytics: descriptive, diagnostic, predictive, and prescriptive. Gartner's analytics capabilities framework (Figure 9) describes the four types of analytics as analytics capabilities. They are different in asking different questions

towards data, using different analytical tools and techniques, and requiring different level of human input to arrive at a decision (Kart et al. 2013).

Descriptive analytics applies mathematics and business logic to data to summarize and reports on what is happening or what has happened. Some example techniques for descriptive analytics include reporting, dashboard, and scorecards. An example of descriptive analytics is using key performance indicators (KPIs) to measure performance. The customer preference survey by Gartner (Hagerty et al. 2012) estimated that 68% of organizations adopted reporting and 43% adopted dashboard to provide analytics. This is what traditional BI platforms have been offering, including big name vendors such as IBM, Oracle, SAP, Microsoft, and MicroStrategy.

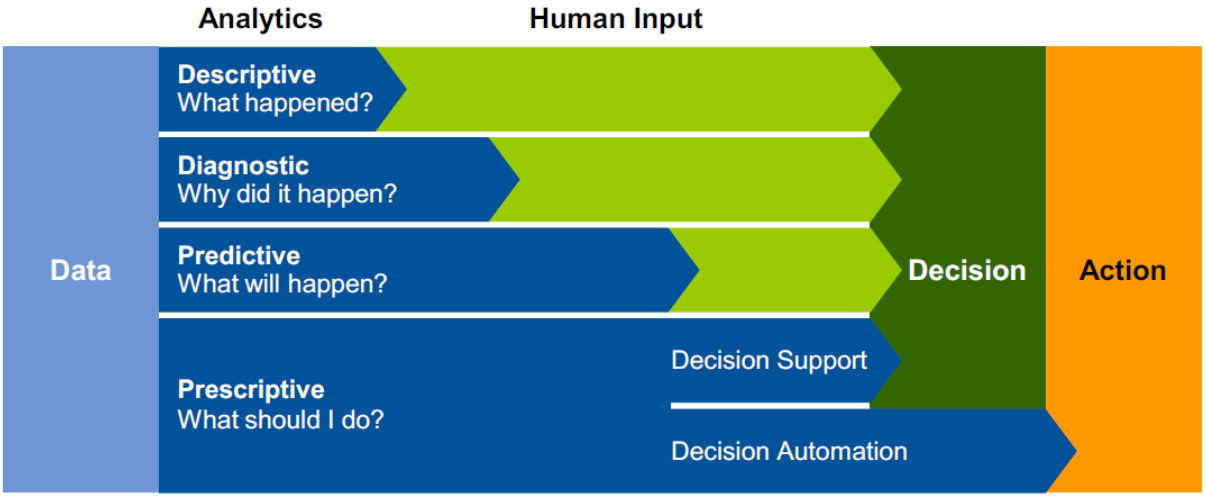

Figure 9: Gartner Analytics Capabilities Framework (Gartner, September 2013)

Diagnostic analytics is a more detailed type of descriptive analytics, requiring drill-down and interacting with data to answer questions about why outcomes, events, or trends occurred and what the key relationships are. Example techniques for diagnostic analytics include OLAP (Online Analytical Processing), interactive visualization, Bayesian networks, correlation and chi-square testing, affinity analysis (or market basket analysis), and other data mining techniques

(e.g., clustering analysis, decision trees, association rules, etc.). Some common application areas for diagnostic analytics are: customer scoring and segmentation, customer profiling, churn analysis, understand leading indicators and drivers for variability, web analytics to understand usage patterns, and sentiment analysis (opinion mining) based on social networks postings. Besides aforementioned traditional BI vendors, advanced visualization for data discovery vendors are gaining popularity recently, such as Tableau Software, Tibco Spotfire, QulikView. Data mining software vendors also fall in this category, such as SAS, IBM SPSS, and R. The same Gartner survey (Hagerty et al. 2012) estimated about 30% of organizations adopted diagnostic analytics.

Driven by the organization's need to understand whether current trends will continue into the future, or to predict a future outcome, predictive analytics are often a natural extension of descriptive and diagnostic analytics. The typical questions that come with predictive analytics are what will happen, what if, and how risky it is. Regression, time series forecasting, neural networks, decision trees (classification and regression trees), support vector machines, genetic algorithms, case-based reasoning, and ensembles, are some examples of predictive analytics techniques. Example use cases for predictive analytics include fraud detection, credit scoring, target marketing/cross-selling, and sales forecasting. Leading vendors in predictive analytics include SAS, IBM SPPS, StatSoft, Actuate, as well as open source software providers such as R, KNIME, and Rapid Miner. While organizations predominantly use analytics to measure the past (reporting and dashboard), or understand the current (interactive visualization and OLAP), only a small percentage of organizations (less than 13%) reported extensive use of predictive analytics (Hagerty et al. 2012).

The descriptive, diagnostics, and predictive analytics presented above require certain degree of human judgment or rules. Instead, predictive analytics outputs a preferred course of action by answering the questions, "what should I do?", "what is the best option?", and "how can I optimize?" Example techniques for prescriptive analytics include optimization (e.g., linear programming, Pareto optimization); multiple criteria decision analysis, Monte Carlo simulation, game theory, and analytical decision management (optimize decisions by combining predictive analytics, business rules, scoring, and optimization). Use cases for prescriptive analytics include price optimization, financial portfolio optimization, and risk management. Vendor groups that provide prescriptive analytics are significant smaller than the previous three types of analytics vendor groups, including many niche players such as General Algebraic Modeling System (GAMS), Palisade, and Wolfram Mathematica. Gartner estimated that less than 3% of the organizations adopt prescriptive analytics (Hagerty et al. 2012).

4.3. THE SNAIL SHELL KDDA PROCESS MODEL

The Snail Shell KDDA Process Model consists of eight phases as shown in Figure 10. The process model is highly iterative that there are no defined sequences between phases, though most KDDA project starts with the problem formulation phase. Each phase includes different tasks, and the outcome of each task determines which phase or particular tasks of a phase to be performed. In the following sections, I briefly outline each phase, and provide a comparison of differences between the KDDA phases and the traditional KDDM phases.

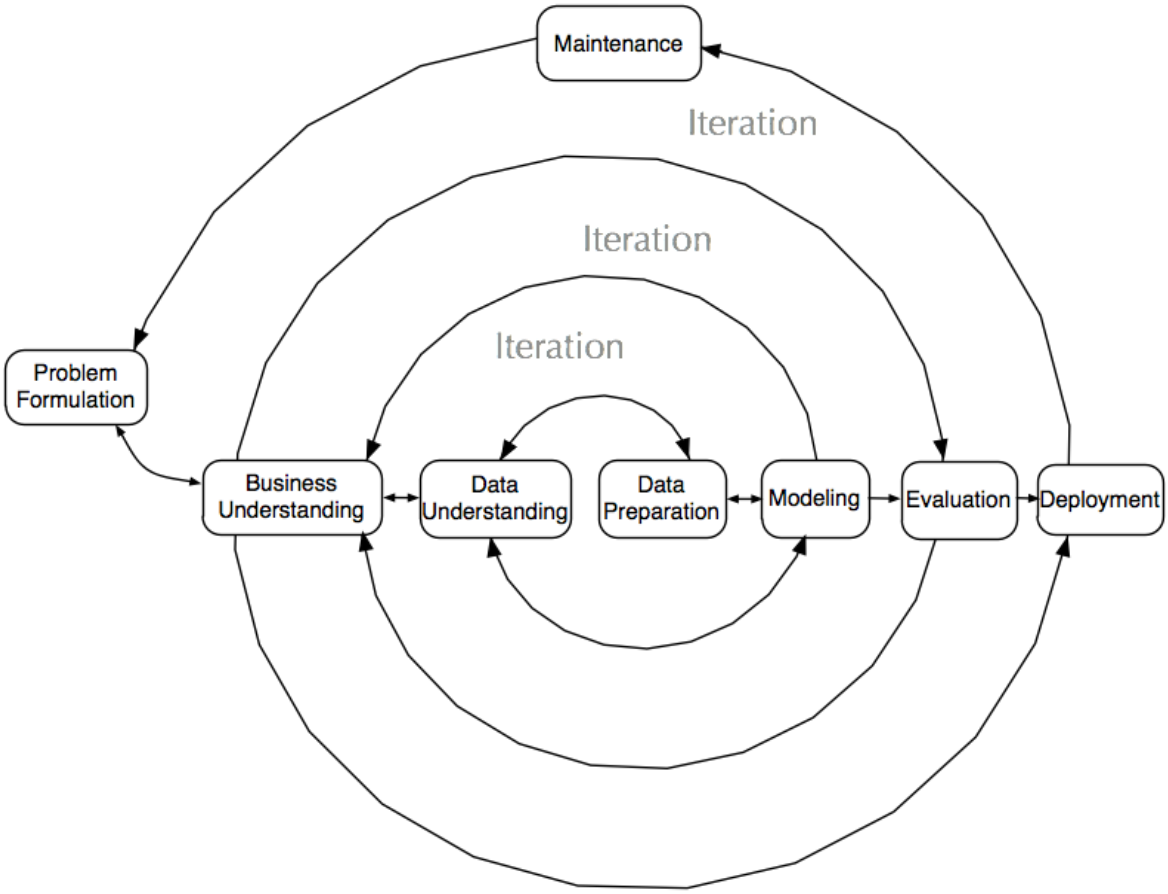

Figure 10: The Snail Shell KDDA Process Model

4.3.1. Problem Formulation

This phase focuses on formulating what business problem(s) the KDDA projects should address, and then transform the business problem statements into actionable analytical problem statement. A problem can be best defined as an undesirable situation that is expected to be altered or completed in a desired manner, while it is believed to be solvable with some difficulty (Agre 1982). "The formulation of a problem is often more essential than its solution..." (Einstein et al. 1938 :92). Problem formulation has been well recognized as the most important aspect of decision process (Mintzberg et al. 1976; Newell et al. 1972). However, at a conceptual level, it is different from the traditional concept of "decision making" that involves making a choice of

identified alternatives. Problem solving focuses on resolving "the difference between some existing situation and some desired situation" (Pounds 1965). Thus, the two concepts, "problem solving" and "decision making", are similar at a cognitive process level, but denote different bodies of research into human thought (Smith 1988).

The quality of a well-formulated business problem can potentially affect the results of succeeding phases in the KDDA process. Problem formulation is different from the formulation of business objectives and data mining goals in the KDDM BU phase, though they share some similar characteristics. While previous research mainly focuses on describing and solving well-defined analytical problems, the business problems in the area of analytics are often ill structured and complex. Literature discerns four types of problem formulation processes: as it relates to the clarity of the goal state, based on characteristics of its problem space, based on the set of problem-relevant knowledge, and reference to the problem solving process. Inadequately defined goal can cause problems in validating if a proposed solution is acceptable. The problem space is a formal, explicit representation of the problem. A well structured problem (Simon 1977) shall include – a problem space that includes initial state, goal state, and all possible intermediate state; the problem space shall represent all attainable state change or transformations; the problem space shall represent all relevant knowledge; and the problem space shall be isomorphic to the problem involving real world actions. The assumption of a knowledge-based problem formulation is that the problem solver lacks knowledge in determining the problem structure, relevant states and transformation. The process-based problem formulation approach assumes the problem solver lacks an effective solution procedure, as expressed by Newell's (1969): "a problem solver finds a problem ill structured if the power of his methods that are applicable lie below a certain thresholds."

Various problem formulation strategies can be adopted, such as formal representation of the problem in models (i.e. specification of elements and relationships to be included in a problem) (Morris 1967), reformulation (e.g. opening or closing constrains till adequate presentation of problem) (Duncker et al. 1945), decomposition (i.e. factoring complex problems to manageable small ones) (Simon 1977), and heuristics (Smith 1988). The heuristic process of problem formulation depends on the current extend of knowledge relevant to the problem domain, the existing repository of problem solving methods, and the solver's cognitive strategy (Smith 1988).

Smith (1998) provides a problem taxonomy of problem categories and problem types so that it can provide means of decomposing the complex problems into sub-problems that match up with specific problem solving solution techniques. Smith (1998) proposed four general problem categories: state change (the need to change some unsatisfactory state or to achieve some goal), performance (the need to improve performance of some function or system), knowledge (the need to acquire certain knowledge), and implementation (the need to put some action into effect). Within these categories, the problem type for KDDA process is related to knowledge. The relevant problem types relate to KDDA process are: description (determining what happens to be the case), evaluation (assessing the worth of entity against one's preferences or external standards), diagnosis (providing explanations of why things are what they are), prediction (predicting future or unknown current states of affairs) and design (demining what one should do to achieve a desired state).

IKDDM (Sharma 2008) suggested a four-step guideline towards formulate business objectives:

1. Apply Value Focused Thinking (VFT) (Keeney 2009) to simulate discussion about business objectives,
2. Apply Goal Question Metrics (GQM) approach (Van Solingen et al. 2002b) to generate preliminary statement of business objectives,
3. Assess preliminary statement of objectives against SMART (Doran 1981) criteria, and
4. Refine step 2 statement based on output from step 3.

The proposed steps provide a structured approach towards formulating business objectives. However, it does not well fit into the structured ill-structured decision context. Nevertheless, GQM approach can be adopted to establish measurable goals. The SMART criteria can be used to assess the organizational objectives: **S**pecific, **M**easurable, **A**chievable, **R**elevant, and **T**ime-bounded. In the context of assessing business objectives, IKDDM (Sharma 2008) outlined the SMART business objective for KDDM as follows:

- The business objective shall be **s**pecific to result in an observable action, behavior, or outcome that is measurable in quantitative terms;
- It shall be **m**easurable in either quantitative or quantitative terms;
- It shall be **a**chievable within the constraints of available resources, knowledge, and time;
- It shall be **r**elevant to a higher order of organizational objective.
- It shall have a clear **t**imeline for achieving the objective through the project.

Table 8 provides a summary of tasks for the problem formulation phase in the KDDA process.

Table 8: Problem Formulation Tasks Summary

KDDA Tasks	Description	KDDM Comparison
Determine business objectives and success measures	Various techniques can be used to facilitate goals and objects determination, including, Value-focus	Similar to CRISP-DM; IKDDM provides a list of applicable

	thinking, GQM, SMART, Mean-end analysis, etc. The objectives shall be measurable	techniques.
Deploy problem formulation strategies (Volkema 1983)	Determine boundaries, factor complex problems into sub-problems; focus on controllable components of a decision situation, reformulation, heuristics	Not available
Define business problem	The business problem shall has a problem type that are related to what, why, and how questions	Not available
Determine KDDA problem, goals, and success measures	The KDDA problem type and goal is determined based on the business problem and objectives. The analytic goal needs to be measurable, which results formally defined analytical success measures. The analytical success measure is the input for evaluating final analytical models.	Similar in KDDM BU phase

4.3.2. Business Understanding

The BU phase focuses on business requirement elicitation that ultimately helps to translate the high-level executive requirements into very technical analytic solutions. BU is one of the most important phases in KDDA process, where its output feeds into all other phases. One of the key tasks in BU phase is enterprise knowledge acquisition. There are different types of enterprise knowledge, which resides in multiple sources. Multiple perspectives need to be considered in the knowledge acquisition process to avoid potential oversight. Example of perspectives include knowledge vis-à-vis data and information, objects that are stored and

manipulated, state of knowing and understanding, process of applying expertise, condition of access information, and capability to influence action (Alavi et al. 2001). Distinction between tacit-explicit and individual-collective knowledge also need to be considered (Nonaka 1994; Spender 1996). Explicit knowledge is knowledge that being articulated, codified, and communicated (Alavi et al. 2001). Examples of enterprise explicit knowledge can be found in Enterprise Content Management (ECM), ETL processes, existing BI reports (usually with embedded key metrics and business logic, and other documentations. Tacit knowledge is rooted in individual's action, experience and involvement in a specific context that refers to individual's cognitive and technical knowledge (Nonaka 1994). In order to acquire tacit knowledge from individuals, the researcher has to have some shared knowledge base with these individuals (Alavi et al. 2001). Similarly, background tacit knowledge is also required for the researcher to acquire and interpret explicit knowledge.

Another key task in BU phase is the analytical capability maturity assessment. The analytical capability includes three different types: organization analytic maturity describes the analytical environment of the organization, data maturity describes if data is suitable for analytics, and decision style maturity describes if the business users' decision styles are mature enough to use the analytical result. The organizational analytical maturity can be best assessed by asking the question based on the Gartner analytics capabilities framework as shown in Figure 9. A more comprehensive analytical capability maturity model shall be developed to guide the analytical process improvement.

The data maturity assessment focuses on the available, stability, and quality of the data for analytics. Data stored in the big data platform and/or EDW does not mean they are ready for analytics. One of the biggest pitfalls of implementing big data solution is it becomes a data vault.

For example, Cassandra databases can be adopted to capture transactional big data (web click streams, machine event logs, sensor tracking data, etc.) However, without pre-implementation planning, these data are not suitable for analytics as they are not queryable. NoSQL databases are designed to store schema-less data, but it does not mean data modeling shall be ignored. On the contrary, data modeling is more challenging in the big data environment, where the design of analytical data access path requires special consideration. More importantly, NoSQL databases are designed to complement other technologies. They are not to replace existing RDBMS, especially in the analytical environment where big data requires additional dimensions of the traditional small data (mostly are stored in a relational or dimensional format for querying and analysis).

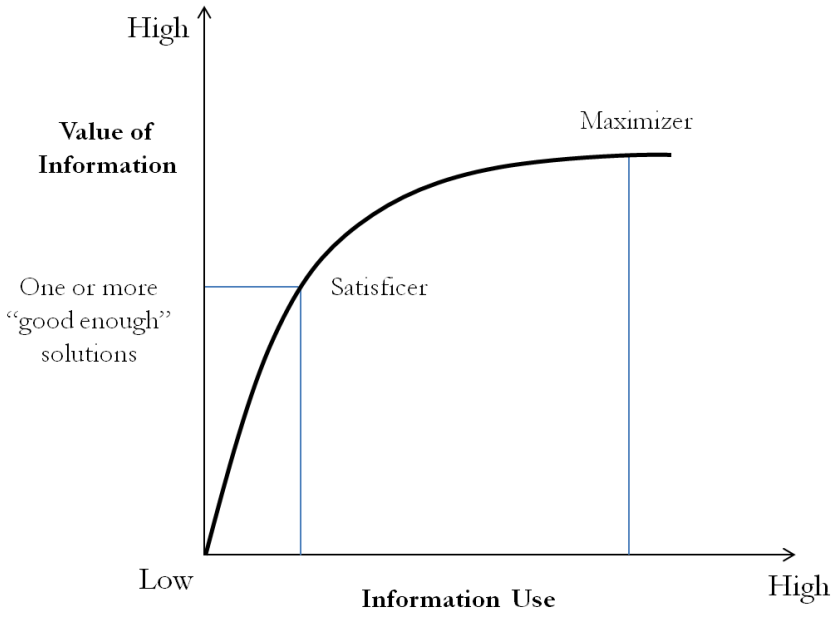

Figure 11: Information Usage Styles (Driver et al. 1998)

Driver et al. (1998) proposed a dynamic decision style model based on two independent factors: information use (i.e. the amount of information actually considered in making a decision) and focus (the number of alternatives identified when reaching decisions). Two decision patterns (Figure 11) are related to the information usage: the satisfier who tries to get "good enough"

situation based on just enough information, while the maximizer who wants to get all relevant information before making a decision.

Differences in focus result in two difference patterns: the person with unifocus lens uses information to produce one solution, and the person with multifocus lens uses several options. Putting the four dimensions together, the decision style model (Driver et al. 1998) defines five decision styles, as shown in Figure 12. The decisive style is a satisfier with unifocus who derives a clear solution based on minimum amount information. The flexible style is a satisfier with multifocus who also moves fast to make decision but focusing on adaptability and keeping options open. The hierarchic style is a maximizer with unifocus who carefully designs a very detailed solution for the problem based on a lot of information. The integrative style is a maximizer with multifocus who although uses a lot of information, explores the problem from multiple perspectives. The systemic style is someone who combines both hierarchic and integrative styles. The person with a systemic style usually approaches the problem through a two-stage process: first an integrative view of the problem and then hierarchic. As noted by the authors (Driver et al. 1998), each decision style has its strengths and weaknesses that fit or do not fit a decision situation. However, in an analytical project, it is easier to present the business cases to the maximizer than the satisfier. The satisfier usually trusts his or her instincts for good enough solutions. As a result, the KDDA expert has to build a strong business case that the analytical solution would provide much higher value of information than the satisfier's initial solution(s).

	Satisfier	**Maximizer**	
Unifocus	Decisive	Hierarchic	Systemic
Multifocus	Flexible	Integrative	

Figure 12: Five Decision Styles (Driver et al. 1998)

A thorough understanding of the organization's processes, including business process, existing analytical process, and ETL processes are also critical in this phase. Existing KDDM process models assume the traditional SDLC methodologies for KDDM projects, which requires producing a project plan at the end of business understand phase. The iterative nature of the KDDA process, however, requires an analysis of appropriate project management methodologies. Based on the nature of project, the KDDA project team may choose waterfall, agile, or mix methodologies. Relevant tasks for BU phase are summarized in Table 9.

Table 9: Business Understanding Summary Tasks

KDDA Tasks	Description	KDDM Comparison
Establish business case (BABOK 2.0)	Identify and quantify costs and benefits; identify requirement, assumptions and constraints, risks and contingencies; identify inventory of resources; present business case to executive sponsor	Under assess situations, but do not include requirement and a separate business case.
Analytical Capability Maturity Assessment	It includes three different types of analysis, namely, data maturity analysis, organization maturity analysis, decision style maturity analysis	Not available
Enterprise Knowledge Acquisition	It includes explicit knowledge acquisition in existing document, business process, ETL process, queries, BI reports, relevant matrices, data quality requirement and matrices, etc. It also includes tacit knowledge acquisition from individuals.	Not explicitly stated
Determine project management methodology	One needs to understand nature of the projects as well as organizational culture.	Implicitly defined as iterative SDLC

Initial tools and techniques selection	Software selection framework can provide some decision support. It is constrained by the business case output.	No software selection framework provided

4.3.3. Data Understanding

DU phase involves familiarizing with data from various data sources. Different from traditional KDDM process where data analyst collects initial data before describing data, the KDDA process calls for exploring data in its original data sources first. With the increasing complexity and variety of data, it is almost impossible to collect initial data without a thorough understanding of the data structure, size, and format of these data. There may exist formal ETL processes within organization to integrate data from various resources into a centralized repository. Advanced analytical databases are now available to provide optimized native support for easier exportation, integration, and visualization of data from multiple sources. However, the KDDA expert still needs to identify the possible data sources and data elements that are needed to solve the analytical problem. The pre-designed data integration strategy often falls short in answering these questions. The KDDA expert will most likely explore multiple sources first to understand the existing data. The findings of the data exploration will help the KDDA expert to formulate the data requirement for analytics.

In addition to data exploration in database, visualization tools are recommended for out-database exploration. Human has known cognitive limitations with too much information presented in an inappropriate way. Visualization analytics is "the science of analytical reasoning facilitated by interactive visual interfaces" (Thomas et al. 2006). Integration of visualization and modeling provides interactive decision support in the analytical process as shown in Figure 13 (Keim et al. 2008).

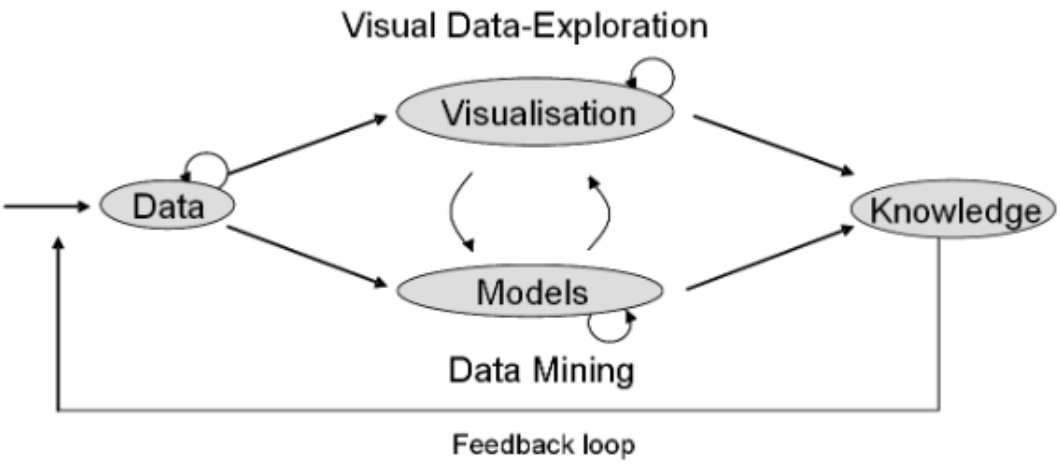

Figure 13: Integration of Visual Data Exploration and Modeling (Keim et al. 2008)

The DU has two different purposes in this phase: one as part of business requirement elicitation, and the other as part of modeling requirement. Detail tasks that are needed in the DU phase vary based on the analytical tools and techniques chosen for the KDDA project. Data quality is always a concern in analytical project. Depending on the business requirements, data quality measures may be different. Data quality measures also rely on the analytical requirements. For example, time series analysis does not allow missing data point for the time interval, which will not be an issue in regression analysis. Data description is very important in understanding initial characteristics of the data. The description should not only include data, but also its metadata, such as source systems, update frequencies, etc. Table 10 lists the set of tasks that are related to the DU phase of the KDDA process.

Table 10: Data Understanding Summary Tasks

KDDA Tasks	Description	KDDM Comparison
Within-DBMS data exploration	SQL can be used in relational databases, and Pig, Hive, or other NoSQL query languages can be used in the big data platforms.	Only available after initial data collection

Out-DBMS exploration	Advanced visualization tools are recommended.	Similar
DU for business logic/requirements	Many of the business requirements and or business logics reside in data.	Not available
DU for modeling requirement	Depending on the modeling techniques selected, different types of DU tasks need to be performed.	Not available
Verify data quality	Data quality depends on the business requirements, as well as the analytical techniques selected.	Similar
Describe Data	Data description should include its source, its owner, and its update frequency, as well as other attributes.	Similar

4.3.4. Data Preparation

The data preparation covers all activities in preparing data for the modeling phase. The analyst needs to extract, transform, and integrate various formats of data from various sources. Based on the data mining problem statement and output from problem formulation phase and BU phase, the analyst first creates an initial dataset integration requirements, including identifying how each data element in the dataset shall be sourced and transformed. Initial data integration effort shall be performed in a testing environment, where iterations of DU – DP – Modeling – BU – DU – DP – Modeling may be performed until an acceptable dataset for modeling purpose.

Data quality is always a concern in the data preparation process. All quality-related problems identified in DU phases should be address by data transformation tasks. Data source related quality issues could be done out of the analytical tool or within the analytical tool. The actual strategy is contingent upon each unique DMS. The same data quality problem may be addressed differently based on the DMS and the quality requirements of the selected analytical

techniques. For example, if time series technique is selected, missing values have to be replaced, as Time Series analysis does not accept NULL. On the contrary, regression analysis and decision tree analysis are both robust towards missing values. As a result, it is recommended to record the data quality issues in DU phase first, and only approach it after a modeling technique has been selected.

Data preparation activities shall be formally documented and often need to feed into organizational ETL processes for deployment and maintenance purpose. The tasks for the DU phase are summarized in Table 11.

Table 11: Data Preparation Summary Tasks

KDDA Tasks	Description	KDDM Comparison
Data integration requirement	Based on BU and DU outputs, the requirement of the dataset output shall be created and communicated. How each data element shall be sourced and in what format should be formally captured. A data integration strategy should also be defined (e.g., whether integrate data on the fly or create a new ETL process).	Similar, however, it separates into dataset and select data.
Data transformation based on quality requirement	Data cleaning and transformation are closely related, where data transformation is more suitable name for this task. The same data might need different types of transformation based on the analytical techniques selected.	Similar to clean data, though some differences
Data transformation based on business requirement	Transformation may be needed based on the business requirements, such as normalization, and aggregation.	Not available
Data transformation based on modeling	Depending on the selected modeling techniques, different types of data	Similar to construct data

requirement	transformation may be needed (e.g. log transformation to remove skewness in the data). Integrated knowledge repository with expert rules can provide decision support in this task.	
Data integration	Integrate data based on formally defined and approved data integration.	Similar

4.3.5. Modeling

Based on the analytical problem statement, various modeling techniques are applicable. The selected modeling techniques are constrained by the data mining tool(s) selected in the BU phase. For example, one class Supported Vector Machine (SVM) is a popular abnormality detection technique, however, it is not available in many analytical packages, including SAS Enterprise Miner and SPPS modeler. Each modeling technique has its own process requirement. Expert rules about the modeling process shall be followed. After final tuning parameters of each model, an initial assessment of modeling results generates a set of candidate models for the evaluation phase. Tasks related to the modeling phase are summarized in Table 12.

Table 12: Modeling Summary Tasks

KDDA Tasks	Description	KDDM Comparison
Select modeling technique	Based on the analytical problem, as well as requirements in BU phase, suitable modeling techniques are selected.	Similar, but KDDA covers a wider range of techniques.
Describe expert rules for the modeling technique	Each selected technique includes set of expert rules. Integrated knowledge repository can provide decision support in this task.	Not available
Define training and testing strategy	Define how analytical models will be trained and how the result can be tested.	Similar
Build model	Expert rules shall guide the process of	Similar

	building model. Any additional insights shall be documented and may be used to update expert rules in the future. Expert rules are stored in the centralized knowledge repository.	
Assess model	The models are assessed using previously defined criteria, and candidate models are chosen for further evaluation.	Similar

4.3.6. Evaluation

In this phase, candidate models are evaluated against the business objectives and the formulated business problems. While analytical models can be evaluated within the tool using objective measures such as accuracy, evaluation against business objectives is usually less clear. In addition to objective measures, testing scenarios can be constructed and evaluation criteria can be defined. The initial model can be tested in the real-world application, but in a much small scale. The analytical process is also to be reviewed to determine whether any factors or tasks are overlooked factors or tasks, and to understand how the modeling requirements will influence existing business, data, and analytical processes. For example, what kind of ETL changes is needed in order to deploy the model in production? Communication of the evaluation results with the executive sponsors and stakeholders are critical. Once the model(s) and modeling process are reviewed and signed off, the next step is to determine whether to move the modeling result to deployment, or need additional iterations. The evaluation result may reveal that the formulated business problem is not adequate, and hence, a new problem formulation is needed. Table 13 summarizes the relevant tasks in the evaluation phase.

Table 13: Evaluation Summary Tasks

KDDA Tasks	Description	KDDM Comparison
Evaluate Result	Evaluate result based on business requirement. If direct evaluation of results is not feasible, a field test may be needed.	Similar
Conduct field test	Create test cases and test the model in a testing environment.	Did not consider when results cannot be directly evaluated.
Review analytical process	Are there additional insights that are beneficial to the organization? Are there changes needed in the business process, data process, or analytical process?	Similar
Communicate results	The result shall be communicated effectively with executive sponsors, and it shall refer back to the business case presented in the beginning of the BU phase.	Not available

4.3.7. Deployment

The deployment strategy shall be considered in the beginning of KDDA project as part of the BU phase. It is important to make sure that stakeholders are aware of the deployment plan and resources are available for the deployment. Similar to the traditional KDDM process, a deployment plan is an output of this phase to summarize deployment strategy, and steps to perform them. The deployment plan shall also document all non-functional requirements (i.e., security requirements, performance requirements) and functional requirements (e.g., systems, database, network needs). Previous KDDM models embed model monitoring and maintenance within the deployment phase. However, it only focuses on planning-related activities, such as identifying possible changes, describing how model performance will be monitored, determining

when to update or retired the modeling results, and documenting the business problems. Model maintenance covers much more than just planning for the corresponding changes in the environment or data. It also includes how to track the analytical model life cycles, including its construction, evaluation, certification, deployment, usage, and retirement. In addition, collective performance of groups of models also need to be monitored (Liu et al. 2008). Thus, a maintenance phase is desired in the KDDA process. The deployment phase tasks are summarized in Table 14 below.

Table 14: Deployment Summary Tasks

KDDA Tasks	Description	KDDM Comparison
Deployment plan	The deployment plan shall explicit document functional and non-functional requirements.	Similar, but does not highlight requirement documentation.
Produce final project report and final presentation	Each project shall be well documented, and presentation to the management is critical.	Similar
Review Project	Any expert rules about model process shall be documented and stored in the centralized KDDA knowledge repository.	Similar, but does not call for the expert rules about modeling process.

4.3.8. Maintenance

The maintenance phase includes all model management activities, including model selection, usage, and retire/replacement. The analytical model is described in rich PMML language and store in the organization's knowledge repository. A formal model maintenance process needs to be established with assigned accountabilities of roles in the maintenance process. Clear defined process is needed to capture when and how to capture changes in the usage of analytical model(s). The possible change can include the model performance

deterioration, data environment change, or business environment change. Based on the functional and non-functional requirements documented in the deployment phase, model usage shall be monitored and feedback from the users shall be collected. Table 15 below summarizes the tasks in the maintenance phase of the KDDA process model.

Table 15: Maintenance Summary Tasks

KDDA Tasks	Description	KDDM Comparison
Describe and store analytical results	Analytical models are semantically described, including its data input and transformation, modeling technique and parameters, its model performance measures and business performance measures.	Not available
Create a model maintenance process	A process towards model maintenance shall be defined, which includes initiate maintenance cycle, delivery model result, check modeling performance, prepare change report, authorize model update, perform model update, and communicate update with the business owners. Each activity in the process execution shall include roles with accountabilities.	Not available
Define change initiation	Explicitly describe how a change shall be captured, including model performance changes, business environment changes, or data environment changes.	Not available
Monitor model usage	The model usage shall fit its security requirement. End users' feedback on the model usage shall be included, which may initiate additional changes that are not formally defined in the change initiation.	Not available

4.4. CASE STUDY 1: DEVICE ABNORMALITY BEHAVIOR DETECTION

In this section, I present the application of KDDA process model for a real world problem. The KDDA process model is used to guide an analytical project in a data-driven decision making environment as shown in Figure 1. The analytical project is highly iterative with constant move between phases and tasks. Only major iterations are highlighted in this case study. Constraint by the organization's policy, all identifiable information in this case is either removed or anonymized.

The study site is a new product division of a multi-billion dollar telecommunications company. The new product, SmartBoxOne (SBO), is viewed as the company's strategic move towards its high-tech service offering. The division is one of the early adapters of big data platforms to store and process data. Cassandra databases are used to store SBO log files and Splunk Enterprise is used to index them. The SBO product division has a BI team that is responsible for providing reporting on the SBO for multiple business units within the same division, as well as across various organizational units. The BI team includes a team of ETL developers who focus on data extraction, transformation, and loading, and a team of analysts who focus on reporting. The executive director of the BI team, who has realized that the team will soon exceed its capability in supporting the exponential growth of SBO, championed the analytical project.

At the time the analytical project was initiated, Splunk was indexing 7-8 TB log data per day. The ETL team had just implemented a unique Change Data Capture (CDC) process to retrieve SBO backend machine data from the enterprise Cassandra databases. One limitation of Cassandra data store is that it is not easily queryable at the aggregate level for analytics. The CDC allows ingest data from Cassandra to Hadoop HDFS (Hadoop Distributed File System)

through Apache Flume, a highly scalable log collection framework. Splunk and Hadoop connect is used to schedule and manage the Splunk Indexing output to Hadoop. Pig scripts are written to coalesce data into a conformed structure. Thus, the log files stored in HDFS are enriched, cleaned, and downsized into a suitable size and queryable format for a traditional RDBMS Data Warehouse.

4.4.1. Problem Formulation

At the time I joined the team, the director has not yet had a clearly articulated business problem at hand. The main business objective was identified as to "find out what went wrong in the SBO environment from data". The executive director acknowledged the complexity of the decision environment: "SBO environment has many variables that could impact its performance, from hardware, software services, to customer's home characteristics, and even weather. Sometime SBO performance deteriorated and I know that one environment variable could have caused that change. However, the problem is that we collect so much data on so many different dimensions that we are not able to determine which one changed and hence we cannot take any actions. We only react to an event (e.g., many customers called after seeing errors on the screen) by looking through all the available data and try to identify causes. It could be caused by a bad software release, but we do not know. Using the *needle in a haystack* metaphor, the SBO is the haystack and we would like to find a way to find these needles (something went wrong) in the haystack." The business problem was communicated, in a more explanatory form, as "What have changed in the SBO environment that cause unwanted events?"

The first step in problem formulation is to deploy problem formulation strategies. As noted by the director, the complexity of the SBO environment prevented him to provide a precise problem statement for the project. Thus, the boundaries of the problem need to be clearly defined,

which can be done by asking the key questions of what, when, where, how, and who? Answers to these questions are related to enterprise knowledge acquisition, which leads to the BU Phase enterprise knowledge acquisition. After extensive knowledge acquisition, the boundary of the problem was first limited to two direct, obtainable measures of SBO platform: Error Logs and Reconnect Logs. So the problem was structured in a more declarative form: "how Error Logs and Reconnect Logs can be used to identify these needles (SBO environmental changes)?" Further BU of Error Logs and Reconnect Logs were needed as described in section 4.4.2. This second round BU was also paired with the task of DU for business requirements (see section 4.4.2 and 4.4.3). After the BU and DU, it was apparent that the problem could be factored into two sub-problems: (1) "how Error Logs can be used to detect the SBO environmental changes?" and (2) "how Reconnect Logs can be used to detect SBO environmental changes?" The business user further rated the Error Logs as the first priority, followed by Reconnect Logs, as Error Logs seem to contain more information related to the state of a device.

These two questions still lack clarity and actionability for carrying out KDDA project, as there were no clearly defined business goals and objectives. Multiple conversations were carried out with the director and BI teams regarding the objective of this project. Currently the team acts in a reactive mode, where after a major event analysts would provide intensive analysis over the past data to try understand "what has happened?" They considered themselves "pretty good at it right now", though sometimes the investigation would be long and take up many resources. The exponential growth of the data would soon outpace the limited human resources. Thus, the first objective at the high level is to be able to reduce the resource times spending in diagnosis past events. However, this objective is neither **s**pecific nor **a**chievable. To help formulate the business objective more structurally, a template based on GQM technique was used, as shown in Table 16.

Table 16: Goal Formulation Template

Objective/Goal	Description
Object under analysis:	SBO devices
For the purpose of:	Identify the change in the environment so pro-active actions can be taken before it becomes too big
With the focus on:	Behaviors that are abnormal from the usual state
From the viewpoint:	BI analysts
Within the context:	The SBO environment where the analysis shall be based on the data currently available

Further analysis of the business objective arrived at an objective to "identify SBO devices that are abnormal from their usual state near real time so that the analysts can focus on investigating these devices (the *needles*) rather than querying the whole device pool (the *haystack*)". This objective was future time-bounded at 2-month for an initial result. The objective evaluation was evaluated using SMART criteria as shown in Table 17.

Table 17: Evaluation of Objective using SMART Criteria

Criterion	Description
Specific	The abnormal devices can be observed as quantitative numbers.
Measurable	The measurement of achieving the business objective must be concrete.
Achievable	It is achievable through machine logs to describe normal state and then identify the abnormal state.
Relevant	It is related to the high-level organizational objective as to reduce resource times spent on investigating pass event.
Time-Bounded	It shall have an initial result within two months of time.

The self-assessment of the team's current analytical capability is, as remarked by the director, "we're stuck in ***descriptive*** and edging into ***diagnostic.***" A formal analytical capability assessment was later carried out in BU phase. Based on the business objective as described

above, the analytical problem is to estimate a device's current state as being Normal or Abnormal. The business users clearly stated the requirement for an analytical model to be:

- The model is desired to be easy to build and deploy.
- The model is desired to be stable with very little human intervention needed.
- The modeling process needs to be flexible so it can be expended to other dimensions if needed.

The above formal requirement was communicated with business users and formally documented as the analytical project success criteria.

4.4.2. Business Understanding

4.4.2.1. BU Iteration 1

Immediately after the project was initiated and the need to determine problem boundaries in the problem formulation phase was identified, the task of enterprise knowledge acquisition was carried out. The company has an enterprise ECM for knowledge management, where all SBO related environment variables and their definitions were identified first. There are *m* different manufacturers and *n* different types of devices. Each device and manufacturer combination provides a different hardware environment, as well as a different software version. In addition, the SBO environment has front-end services where customers interact with the SBO device and back-end services where different types of software services (more than dozen) were delivered. There are extensive documentations about software services types and releases, the combination of which results in a different software environment. After examination of the ECM, conversations with business domain experts were carried out to understand which SBO environment that they are most interested. Two direct measures were emergent in the

conversation: Error Logs and Reconnect Logs. An Error Log is a log file written back by the SBO backend application in an event of failure. A Reconnect Log is a log file written back by the SBO backend application in an event of SBO device attempts to reconnect the server after an offline event. A log file is written in a standard format that includes application name, minor version, major version, timestamps of the event, event duration, physical and logic addresses of the device (e.g. MAC address, IP address), hardware type, hardware model, preceding event, response code, device physical locations, etc.

Based on the analytical problem that is defined in the problem formulation phase, an analytical capability assessment was performed to assess if the organization is mature enough to carry out the project. From the organizational perspective, the organization positions itself as a leading technological company and has a deep data-driven decision-making culture that ranges from top executives to different business units directors. However, the SBO BI team is still in an initial stage of descriptive analytics, where data are primarily used to explain what has happened. The BI team analysts are well prepared with writing ad-hoc queries, building interactive Tableau reports and dashboards. However, they lack of understanding of advance analytics tools and techniques, as well analytical processes. From the organization maturity perspective, the analytical capability of the SBO business unit is considered as low to medium.

From the data maturity view, significant progress has been made in capturing and integrating machine data for analytic purposes. As mentioned earlier, the log files are loaded from Cassandra to HDFS through flume and Splunk has provided some indexing over the log files. Pig scripts were used to understand the data structure and guided the design of data schema that Hive can partition the files based on the defined schema. While date and time are obvious choice for the partitioning, the more useful schema shall be defined by the analytical needs.

Currently, the Error Logs and Reconnect Logs have defined schema and corresponding Hive tables. Further, HDFS are loaded into and integrated with the EDW data through an automatic ETL process. Various business logics are applied in the integration process, which is acquired by reviewing the ETL process packages. For example, the EDW uses a different type of physical IDs to identify unique device, while HDFS captures another. In addition, log files are device-specific, while analytics often require account-specific information.

In order to understand if Error Logs and/or Reconnect Logs can actually be used to answer the question being raised in 4.4.1, the DU is needed. The second round of BU started with conversations with senior BI analysts, who understand the available data sources and structures. The analysts provided three different data sources for Error Logs and Reconnect Log data. First, Cassandra stores the most recent one month of log files in the most comprehensive way (that is, including all payloads of the log file). Log files can be explored and indexed using Splunk Enterprise. Second, HDFS stores the past one and half years (since HDFS was in production) of Error Logs and Reconnect Logs in a structured format. Some information is removed, such as different versions of services. The rationale behind this is simple – even with HDFS, it is still too much information to store. Third, EDW stores one month Error Logs and Reconnect Logs in an integrated format that include account information, device type, and device manufacturers. The one-month truncate period was chosen as most of the business reporting only concerns most recent history.

After identifying the three data sources, within-DBMS data exploration in DU phase was performed to assess the data maturity level for analytics. The DU process is closely integrated with BU phase, where any intermediate results are communicated with the business users to understand its meanings and to decide the next step. As mentioned in 4.4.1, the first step is to

answer the question "how can we use Error Logs to detect the environmental changes?" The ultimate goal is to identify devices with abnormality so that the resources can focus on investigating these devices with dimensions that have issues, rather than looking at everything. Based on feedback from the business users and results from the DU phase, the KDDA expert hypothesized that integrating Error Logs and/or Reconnect Logs with the software versions would provide a means to identify changes in the environment (e.g. a software service update results in a higher number of failures related to one type of error codes). The device location and hardware may provide additional useful information in answering the question.

Within-DBMS exploration (from DU) revealed that Splunk has the required data elements but only one-month history. Such data elements were missing in HDFS and EDW. However, operational change management documents were available that recorded the historical software service releases. The software release was pushed in a rolling base of 20%, 50%, 70% and 100%. Each phase of release usually pushes the software service to one server stack to mitigate the risk. An EDW table used to capture which device connects to which server stack. However, a quick DU of that table revealed that the table has not been updated for a few months. The DU result was communicated with multiple business teams and identified an organizational process change that the application feeds to populate this EDW table was rewritten. A process change was then initiated to start capturing the stack information again. However, the history was again lost. The DU phase also brought some frequent data structure changes, which were considered as expected by the business user and BI team. The organization's big data platform does not have structured planning and data governance. The data modeling was at the hand of application developers, who may rewrite one line code or one application process that would change the current structure of data completely. There were lack of communications between

application developers and the business users who try to use the data to facilitate decision making, which have been a constant struggle. The current data have been good enough for reporting purposes when compared to a few year back. However, the data maturity level for analytics is still very low.

The decision making style of the director is a mix of decisive and systemic. In a situation where there is a time pressure to make quick decision on less complex problems, he is able to make speedy, efficient, and quick decisions by settling on good enough information. In a more complex situation, he would solve the problem in a systemic style. He has a deep understanding of the business, and is technically savvy in using data to find answers. His decision style represents a medium to high maturity level for analytics. However, the business environment poses time pressures more often than not. Thus, he frequently makes decisive decisions. The other leaders' decision styles are also very similar. Table summarizes the initial analytical capability assessment result. The organization should have the ability to answer the question of "what has happened", as posited in the problem formulation phase.

Table 18: Analytical Capability Maturity Assessment Result

Analytical Capability Maturity	Assessment	Description
Organization maturity	Low to medium	In descriptive stage and edging into diagnostic.
Data maturity	Low	Good amount of data, but lack of integrated view and data governance.
Decision style maturity	Medium to high	Data driven decision style that is favorable towards analytics, though it is constrained by decision making situations.

Because the director is the executive sponsor of the project, he has already made the business case for the need of analytics. However, this project is considered as a proof of concept (POC) project, where limited resources can be allocated. Only one resource (the KDDA expert)

understands the KDDA process and advanced analytical techniques, while other resources from the BI team and operations team can provide support. Excel spreadsheet and Tableau visualization tool are two main tools for presenting BI results to end users. There are no analytical tools licensing currently and the purchase of an analytical tool (such as SAS) was not in the near future. The KDDA expert was constrained to select from open source software packages. Three open source solutions were investigated: KNIME, R, and Rapid Miner. The KDDA expert has priori experience with R and Rapid Miner, but Rapid Miner only offers limited features in its free edition. R is also favorable by the director and BI team because some of them have had certain exposure to R. Thus, R is selected as the main analytical Tool. Techniques relevant in solving the posited analytical question include Decision Trees, Regression, Association Rules, and Support Vector Machines (SVM). Basic mean-standard deviation approach is also applicable if data is normally distributed. R provides implementation of all these techniques, however, R runs in memory and its scalability is a concern. The risks and contingencies of using R were documented. The selection of the actual technique(s) shall be based on the BU output as well as outputs from the DU phase.

The next step is to determine the project management methodology. The BI team is not a process-oriented team. The team has been operating in a fast-paced environment, where 70% of the resources were utilized to answer ad-hoc requests. There is a daily standup meeting where the team meets to provide status update to team members and a backlog of items for requests that were submitted and assigned to different team members. However, the activities were task-based and there was no formal methodology for the everyday operation. The business problem to be solved was ill structured and a formalized schedule was not practical. Based on the culture of the team and the nature of the business problem, a semi-agile methodology was adopted as a mutual

agreement between the KDDA expert and the BI team. Time-box iteration concept from Agile Methodology (Beck et al. 2001) was adopted which involves four iterations of task planning, development, demo and retrospective. The time-box is kept as one week and tasks are adjusted accordingly.

4.4.2.2. BU Iteration 2

The ECM has a detailed description of different type of Reconnect logs. Each reconnect code has a reconnect reason and its description. The actual meaning of each reconnect reason was discussed with the application development team as well as business users. The existing organizational dashboard and reports that are related with Reconnect Logs were also reviewed. There is a scheduled reconnect event between local time 2:00 am and 6:00 am to push software updates. The scheduled reconnect has an assigned reconnect reason code 1. Similar to Error Logs, Reconnect Logs reside in three different places. The ETL process that moves Reconnect Log from HDFS to EDW was reviewed and its business logic was recorded. DU task was carried out to understand the basic distribution of the reconnect logs. Six reconnect reason codes were selected as the focus of the study. The analytical problem was refined again to "how to indentify SBO environment change using the six types of reconnects?"

Previous BU-DU-DP-BU iteration has indicated that the integration of software services with devices was not feasible currently. The scheduled reconnects (reconnect reason code 1), between 06 and 11 UTC should not be included in the final dataset. The business user expressed interested in looking at reconnect reasons one at a time. Thus, a separate model for each reconnect reason is desired. Based on both data and business knowledge, reconnect data was determined to be time-variant. Hourly reconnects were key metrics on the organizational

dashboard. Device location is hierarchical as gateway -> branch -> sub-region -> region -> division. In order to achieve the high-level business objective of reducing resource time, analysis on gateway and branch levels would produce too many results. At region level, it would be coarse that meaningful changes might be overlooked. Thus, it was recommended to aggregate the reconnects at the sub-region level. These set of considerations provided the input for the data integration plan in DP iteration 2.

4.4.3. Data Understanding

4.4.3.1. DU Iteration 1

The DU task started right after initial business problem was formulated. Within-DBMS data exploration was the first task performed in this project. As described in Section 4.4.2, three data sources were identified for Error Logs and Reconnect Logs that are different in their gratuity level and historical load level. First task focuses on understanding Error Logs. There were more than one hundred different error codes. Quick Splunk query (Figure 14) was written to review statistics and indexes related to error logs. One Splunk macro was written to ingest an operational dashboard that includes top device errors with related device software versions and other software services. However, the query was summarized by services and software versions, and not in the device level gratuity needed.

```
sourcetype="sbpGuide" "Guide startup handleRetry ERROR_CODE"|rex "Guide startup handleRetry
ERROR_CODE#(?<error_code>\d+)"|rex "ERROR_STRING:(?<error_string>([a-zA-Z0-9_])+)" |dedup
error_code error_string
```

Figure 14: Splunk Query to Retrieve Error Logs

Data exploration in HDFS revealed that the Error Logs were stored with much less information. Only device ID, time, hour, error code and error string (i.e. error code description)

were stored and partitioned. It also revealed that although there were more than one hundred error codes, 12 error codes constitute 99.99% of all errors. Query performance is a concern in querying HDFS, where a closed date range was used to limit the number of MapReduce jobs at a time and thus reduce the query running time. To gain a comprehensive view of data, eight months of total Error Logs were examined in summarization, one independent query for each month. The distribution of error codes identified two data abnormalities:

- One error code (Application failure) occurred in October2014 that accounted for 70% of total errors in that month versus normal 10% -15% of total errors.
- Another error code (Home backend network failure) occurred in September 2014 that accounted for 99% of errors in that month, after which this error code was almost negligible.

In addition, one error string does not have an error code associated with it. The KDDA expert went back to BU phase to understand these error codes and which are more meaningful to business users. The issues with data abnormality were also raised. The error string with NULL error code was not considered as an error, and subsequently removed from the data integration requirement.

The Error Logs in EDW have some useful dimensions, such as the device hardware versions, location and account information. However, the software-related information was missing, similar in HDFS. Splunk was the only place that this information was formally captured. However, only one-month history was available in Splunk, not enough for current analytical purpose. The KDDA expert went back to BU phase and created a data integration plan based on the hypothesis. How to correlate the software services with Error Logs is documented.

4.4.3.2. DU Iteration 2

The DU iteration 2 focuses on understanding the Reconnect Logs. Direct Hive queries were written to understand the Reconnect type distribution. The query was guided by the ETL process business logic to ensure consistency. For example, multiple applications send Reconnect Logs. Only Reconnects related to "core" application were retrieved. The device identifier has a certain naming strategy to define physical SBO devices. Other devices such as mobile devices may also send Reconnect application through applications, which were not included in the final analysis. Six reconnect reason codes consist of 99% of unscheduled Reconnect Logs. The DP - Modeling plan is to prepare one dataset for each reconnect reason code at a time.

Once the integrated dataset was prepared in EDW, within-DBMS queries were run to understand the distributions of each variable. There were four device manufacturers and three different types of devices. However, two device manufacturers only produce one type of device, one device manufacturer produces two types of devices and one device manufacturer produces three types of devices. It makes seven types of device hardware combinations. One device manufacturer has only started to produce the device with less than 60 days date available. One device type is also in its testing stage with limited number of devices.

Visualization is an important analytical technique that facilitates visual data exploration before modeling. Excel and Tableau are both available for visualization tasks, though tableau handles large dataset faster, and is more interactive. After a preliminary analysis of the initial dataset, the data was imported to Tableau for visual inspection. Interactive visual exploration of data reveals that one manufacturer-device combination has a clear up trend recently. In addition, the sub-regions with very small number of device types showed abnormal reconnect rates. The explanation was that for the random error(s) were magnified by dividing total number of devices.

A cut-off minimum number of devices were chosen based on visualization result. In addition, different sub-region has different performances – there were clearly sub-regions with low reconnect rates, while some were among highest. This confirmed that the reconnect rate is correlated with the location. The reconnect rates are also quite different across different types of devices. Figure 15 presents an example of Tableau visualization result.

Visual inspection revealed two different periods, where the reconnect connects (only one reconnect reason at one time) were much higher in the first period then the second period. Consulting this result with business users indicated that significant application process improvements were implemented to reduce the reconnect rates. As a result, the data was truncated to include the stable period (most recent 9 weeks).

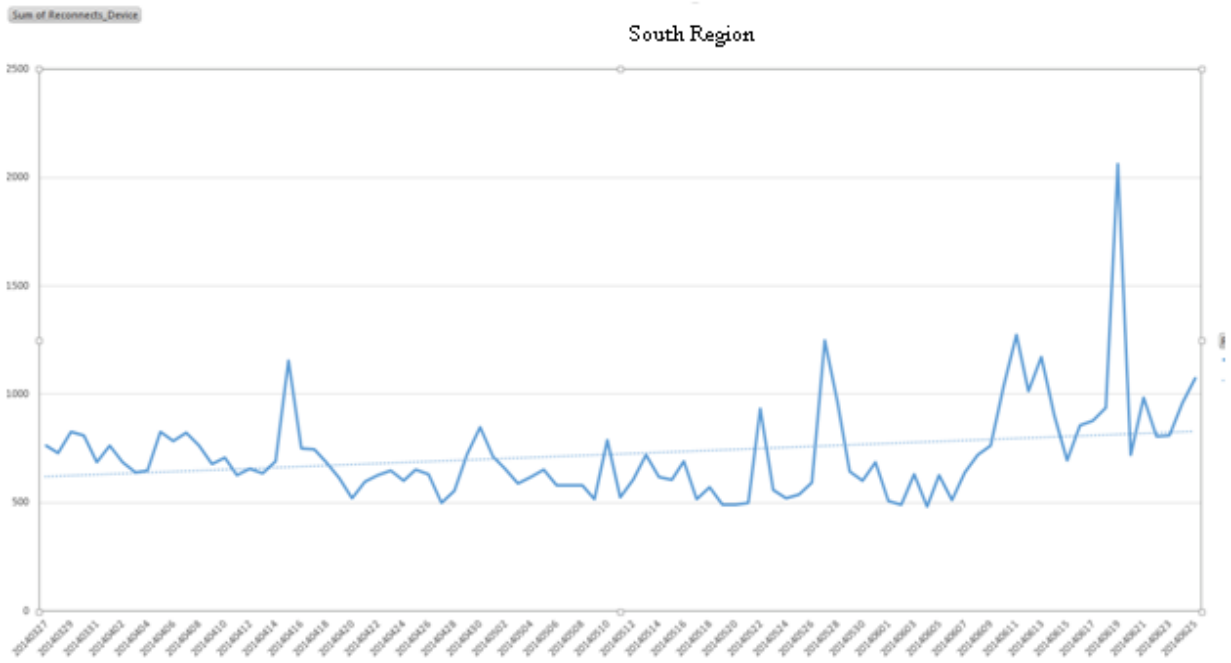

Figure 15: Tableau Visual Inspection Example

Visualization also revealed some potential data quality issues. Two sub-regions "South" and "North" were sourced in system in two different types: "South" versus "South Region", and "North" versus "North region". The "South" and "North" were data source errors that resulted in

very small number of different types of devices. The data quality issue was reported to the ETL team. Before ETL team can implement the process to correct this error, the KDDA expert implemented a process in R to correct the data issue. Other data quality related issues, such as missing values and extreme values, were inspected in R. How to handle these data quality issues are specific to the technique selected.

After visual inspection, data was loaded into R through RJDBC connection for DU based on modeling requirement. Data transformation tasks were first performed based on the modeling requirement and business requirements, which are discussed in DP iteration 2. Regression analysis assumption of normality was checked. Figure 16 shows the histogram of the reconnect rate for reconnect reason code 1. It shows a skew normal distribution. The skewness was mediated by log transformation of the reconnect rate, as shown in Figure 17.

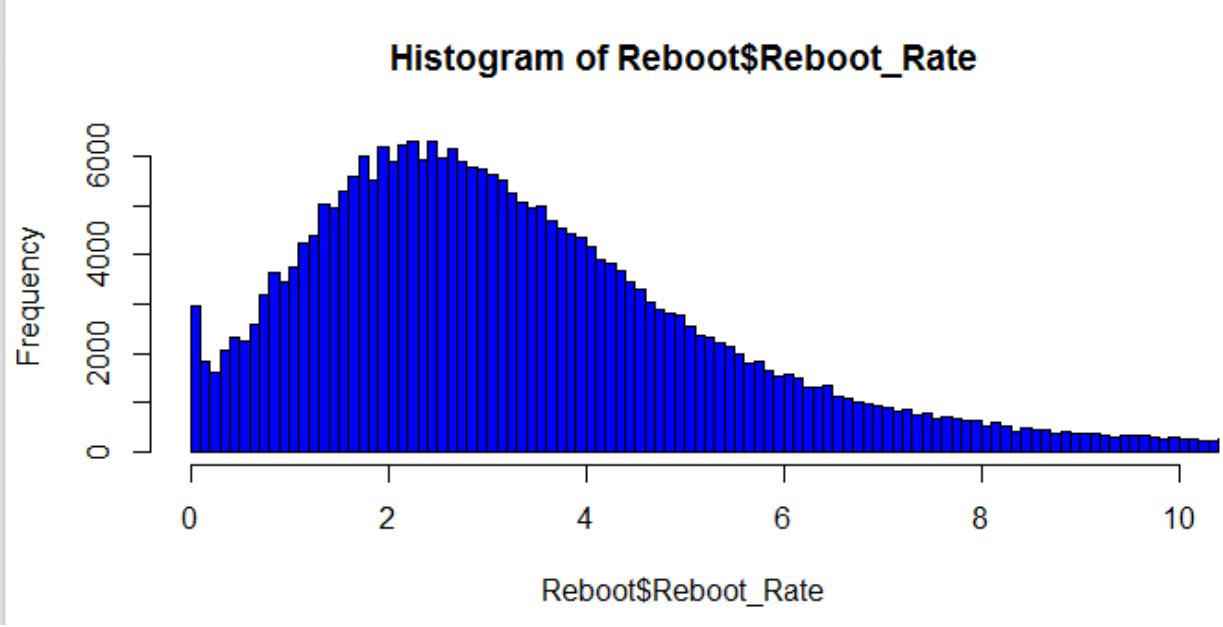

Figure 16: Before Reconnect Rate Transformation

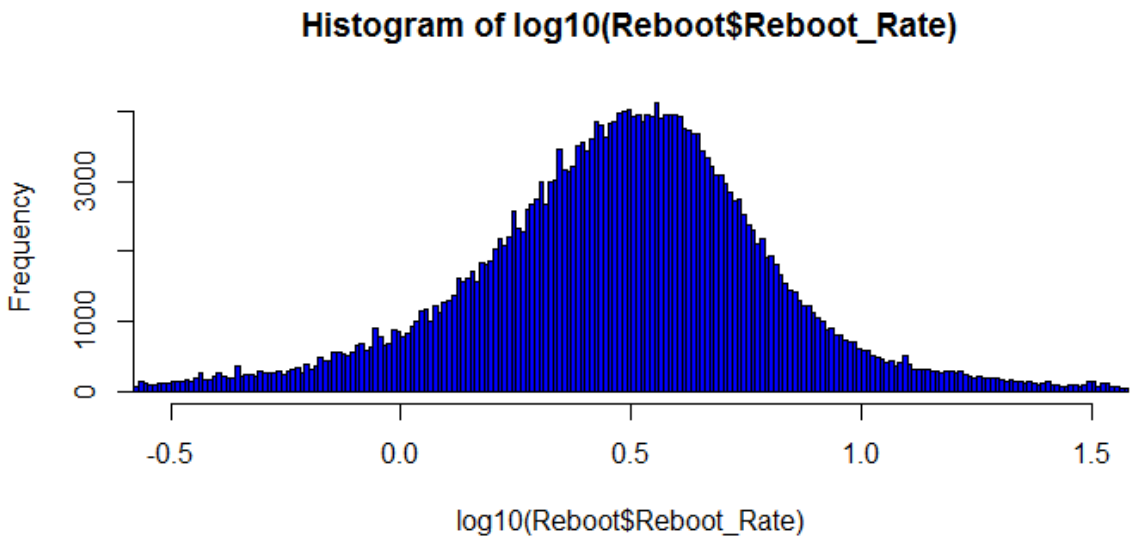

Figure 17: Log Transformation of Reconnect Rate

Boxplot (Figure 18) was used to inspect if there were differences between group means for Sub-Region, Hours, as well as Device Types. Density plot (Figure 19) was used to test density by groups as well. Multiple ways of DU indicated that the reconnect data was close to normal distribution.

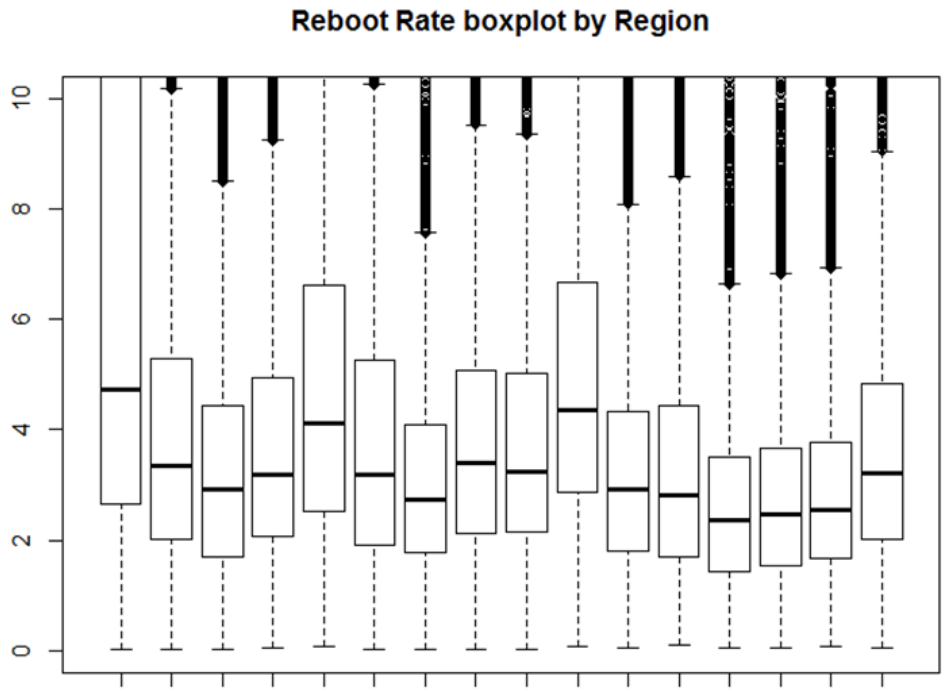

Figure 18: Example of Boxplot to Compare Group Mean

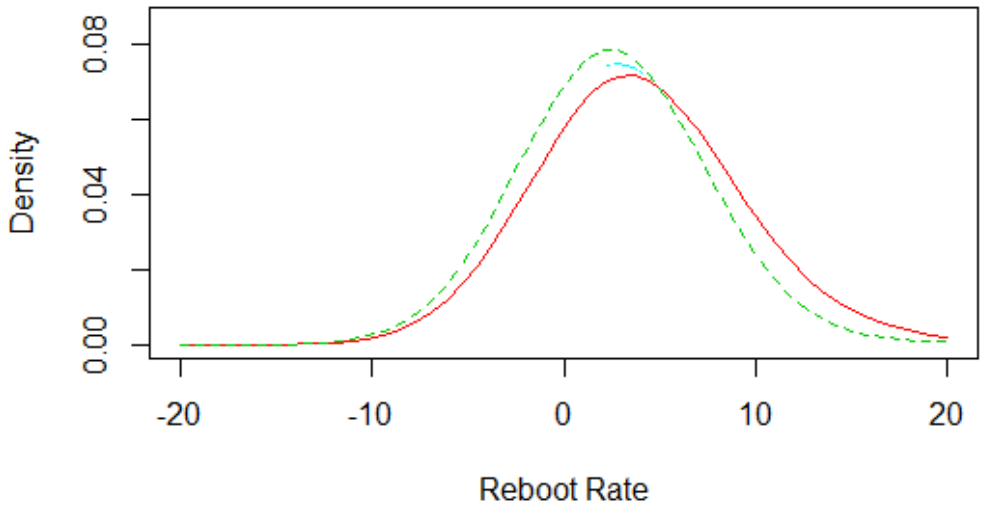

Figure 19: Example of Density Plot by Device Type

Above DU tasks confirms that grouping reconnects by hour, location, and device type was a valid approach. It also provided validation of modeling approach, control chart approach, selected in modeling iteration 2. Central limit theorem states that the sampling distribution of any statistic will be normal or nearly normal, if the sample size is large enough. The rule of sum for sample size is greater than 40 without outliers. Currently sample size is 63 (days) and thus fit to the central limit theorem. However, outliers (extreme values) need to be replaced. Missing values were also inspected. Missing values are observed when there was no reconnect for a specific (hour + device manufacturer type + sub-region) combination. R `sqldf` function was used to find distinct class variable instances and then calculate expected observations. For example, if there was 41 sub-region IDs, 140 date keys, and 24 hours, total number of observations should be 137760. The actual data had 134485, equivalent of 2.4% percent of missing data. It is considered acceptable, especially it was expected that the higher of "missing values" the better since the ultimate goal is to reduce the reconnect rate. DP steps specific to the control chart approach was performed subsequently.

4.4.4. Data Preparation

4.4.4.1. DP Iteration 1

Based on the hypothesis for Error Logs in Section 4.1.1, the KDDA expert proposed a data integration plan that should collect device-level Error Log data with software, hardware, and locations related to the device. The DU output was also presented to the business user, where the top 10 error codes were identified as the point of interests. An initial data integration requirement was created, including the data sources and data elements needed. There were limited resources on ETL team to support the data integration task. Only when the modeling results provided some added business value that the data integration process would be formalized in the ETL process. Based on the analysis, the KDDA expert created a data integration plan. The first step was to create software versions (e.g., platform version, software services versions, application versions, and so forth) based on the change management file. Heuristics were chosen to assign the software versions to devices. Most recent 16 weeks of Error Logs in HDFS were retrieved and ingested in EDW through SQL Loader. Multiple queries were written to join Error Logs with other dimensions. The final data was joined with a random sampled device set of 10% with roughly 300K devices. The integrated dataset has total 33 variables, one of which is the ID field, ten of which are error logs as target variables, and the rest are input variables. Before moving to the modeling phase, the data integration result was presented to the director and BI team. The director raised the issue that the data integration effort was not sustainable. In addition, the error codes were collected in different granularity level. One type of errors includes more detailed error descriptions, while two other types of errors do not. Even if interesting result can be found to detect an error increase due to input variables changes, no applications can be taken to peruse

the result. Thus, the first question was put on hold. The KDDA expert went back to BU phase to understand Reconnect logs.

4.4.4.2. DP Iteration 2

The output of BU iteration 2 and DU iteration 2 guided the KDDA expert to produce a data integration plan that looks at one reconnect reason a time. A data retrieval plan was designed to retrieve data from HDFS, load in Oracle to add device related dimensions, namely device type, device manufacturer, hour key, date key, sub-region and total devices. Same sample device table was used to create sample reconnect data. Reconnect reason code 1 has the highest reconnects. Its integrated dataset was slightly more 200K rows, which was considered manageable for R processing. The total devices are important as the total reconnects need to be normalized for the associated region. The reconnect rate was a calculated continuous variable. The KDDA experts then moved back to DU phase to explore the data before modeling.

As discussed in DU iteration 2, a quality issue was identified related to the sub-region names. A process in R is designed as follows to replace the wrong name with the correct names. Figure 20 provides an example of the R code to implement the process.

1. Subset the dataset to exclude data with the four sub-regions: South, South Region, North, and North Region;
2. Create a dataset for South only and a dataset for South Region only.
3. Create a union of South and South Region using `Rbind` function.
4. Replace the sub-region name South with South Region.
5. Create final South Sub-Region Date Set using sum reconnects group by sub-region names.
6. Repeat step 2-5 for North Sub-Region.

7. Combine dataset in step 1, 5 & 6 to the new dataset.

```
1. Reconnect1<-subset(Reconnect_1, Subregion_ID!=2 & Subregion_ID!=3 & Subregio
n_ID!=11 & Subregion_ID!=12)

2. SouthRegion<-subset(NENew, Subregion_ID==3)
   South<-subset(NENew, Subregion_ID==12)
3. SountRegion<-rbind(SountRegion, South)

4. SouthRegion$SUBREGION_NAME[SouthRegion$SUBREGION_NAME=="SOUTH"]<-"South Regi
on"
SouthRegion$Subregion_ID[SouthRegion$Subregion_ID==12]<-3

5. SouthRegion<-sqldf("select sum(Reconnects) as RECONNECTS, sum(TOTAL_DEVICES)
 as TOTAL_DEVICES, DATE_KEY, HOUR, DEVICE_TYPE, SUBREGION_NAME, REGION_NAME, DE
VICE_MANUFACTURER, Subregion_ID, DIVISION_NAME from SouthRegion group by  DATE_
KEY, HOUR, DEVICE_TYPE, SUBREGION_NAME, REGION_NAME, DEVICE_MANUFACTURER, Subre
gion_ID, DIVISION_NAME")

7. Reconnect_Clean <-rbind(SouthRegion, Reconnect1)
```

Figure 20: Example R Code for Data Cleaning

Some DP tasks were performed based on business requirement. For example, the device manufacturer and the device type are considered as a unique combination of the device hardware. A derived variable Manufacturer Type was created to reflect seven unique combinations. If without this transformation, the unique combinations would be 12, 5 of which would result in NULL values. Hour was loaded in R as interval variables, which was transformed to categorical variable.

4.4.4.3. DP Iteration 3

In this iteration, data was prepared to conform to the modeling requirement. The key DP steps were documented: removed sub-regions with very small number of devices (per device manufacturer type), transform Hour to categorical using R `as.factor` function (Figure 21), log transformation reconnect rate, and remove/replace outliers.

```
'data.frame': 236921 obs. of  11 variables:
 $ RECONNECTS          : num  1 1 1 1 1 1 1 1 1 1 ...
 $ DATE_KEY            : num  20140221 20140221 20140221 20140221 20140221 ...
 $ HOUR_KEY            : Factor w/ 24 levels "0","1","2","3",..: 13 16 16 17 18 22 23 24 1 3
```

Figure 21: Hour Key as Categorical Variable

Two strategies were considered for outliers: replacement and removal. The replace strategy was to replace the outliers with mean, which was not well suited in this case. Therefore,

the removal strategy was considered. Although some R packages for finding and replacing extreme-value were available, none of them performed well on the dataset. R codes (Figure 22) were then written to remove outliers.

```
x0<-Reboot [0,]
for (i in 1:3936) {
x1<-subset(x[[i]], x[[i]]$Log_Reboot>=(mean(x[[i]]$Log_Reboot)+3*sd(x[[i]]$Lo
g_Reboot)))
x0<-rbind(x0, x1)
}
Reboot.Trim<-sqldf ("select * from Reboot except select * from x0")
```

Figure 22: R Codes for Outlier Removal

4.4.5. Modeling

4.4.5.1. Modeling Iteration 1

The DP iteration 2 outputs a dataset with a continuous target variable. The goal is to estimate the normal number of reconnects for a given day, hour, device type, in a given sub-region. Regression, Regression Tree, Regression Splines, Artificial Neural Network and K-Nearest Neighbor are all relevant advanced analytical techniques. However, since the goal is to give the analysts the ability to track down the abnormality, the model shall be explanatory. Regression and regression tree techniques were selected initially.

After DP iteration 2, regression analysis was the first modeling technique used. Confidence interval at 95% was chosen to select the range of normal values. Any reconnect rates out of the confidence interval were considered abnormal. The first run on regression (reconnect_rate ~ hour +sub-region+ DeviceManufacturerType) did not produce significant result. Notably hour, sub-region, and DeviceManufacturerType were all categorical variable. Not all hours or sub-regions had β values that were significant at the alpha level of 0.05. The 95% confidence intervals of the estimation was too narrow and too many false negatives. The KDDA expert then went back to the BU phase to discuss initial findings to the business users. A

traditional control chart approach was suggested, which was to find group mean and standard deviation. The UC (Up Control) limit was set as two standard deviations away from the mean, and LC (Lower Control) limit was set to zero.

4.4.5.2. Modeling Iteration 2

The second modeling approach was to find the aggregate group mean. By the time of the modeling iteration 2 started, two additional weeks of data were already available for testing purposes. All 63 days of date were used in finding group mean and standard deviations. R aggregated function was used. The final modeling process was only a few simple lines of R codes, as shown in Figure 23. The modeling result was loaded in a CSV file for future development.

```
GroupMean<-aggregate(Log_Reboot~MANUFACTURER_TYPE+REGION_NAME+HOUR_KEY, RECON
NECT, mean)
colnames(x)[4]<-"mean"

GroupSd<-aggregate(Log_Reboot~MANUFACTURER_TYPE+REGION_NAME+HOUR_KEY, RECONNE
CT, Sd)
colnames(y)[4]<-"Sd"

GroupResult<-merge(GroupMean,GroupSd)
GroupResult<-merge(GroupResult, Market, all.x=TRUE)
```

Figure 23: Control Chart Modeling

As described in section 4.4.1, the requirements for the analytical model include - it should be easy to build and deploy, it should be stable with very little human intervention needed, and the process needs to be flexible so it can be expended to other dimensions if needed. The first requirement was satisfied, as the model was very easy to build. It can also be deployed by a simple query. Requirement 3 was also satisfied as it presented a repeatable simple process. As long as the data normality is satisfied, additional dimensions (such as other reconnect reasons, error dialogs, among others) can be added as needed. Requirement 2 was contingently satisfied.

The control chart approach was quite robust if the data environment did not change much. However, the existing approach does not capture the trend in the data. Hence, the modeling performance needs to be monitored closely. An automatic process for refreshing group means and standard deviations were recommended. Changes in the modeling output, such as very high abnormal devices or very low abnormal devices, could indicate changes in the data environment and update to the model would be needed.

4.4.6. Evaluation

The first modeling result (reconnect reason code 1) was assessed using the recent two weeks of data. Some sub-regions were identified as abnormal. A BI team member designated to contact the sub-regions that have high abnormality. Special considerations were given to sub-regions that had continuously high reconnect rates for more than three consecutive hours. This specific approach was selected because a direct test of the modeling result was not feasible. There was no single indication of the cause of the abnormal behavior. This evaluation approach is ongoing until some correlations between the modeling result and SBO environment changes can be identified. Nevertheless, the model did correlate a major storm in one of the sub-regions, where reconnect reason code 2 became abnormally higher for consecutive hours.

The project was completed in 6 weeks, ahead of scheduled delivery time. Documentation was written and a presentation was given. The modeling requirement indicates that one-month Reconnect Logs are not large enough statistically. ETL process was changed to include 90 days of Reconnect Logs in EDW. It makes the model maintenance easier. In addition, the model sparked an interest in looking into reconnect reasons and a new question, "can we use reconnects

data, or some other machine data, to identify a device in an unhealthy state?" This question became a second analytical project, highlighting the iterative nature of the KDDA process.

4.4.7. Deployment

The plan for deployment was simple because the chosen modeling technique was aimed for easy deployment. A modeling table was created to store all means and standard deviations for all types of machine code. The table has the following columns: ID, MachineDataType, Code, Code Description, Hour, Device Type, Device Manufacturer, Sub-Region Name, Region Name, and Division Name. A deployment SQL statement was written to run against EDW and results were written to a table in EDW to track history. Interesting findings related to the KDDA process were well documented and shared in the organization's ECM. Examples of expert rules include the investigation towards sub-regions with very small number of devices and data types that are read in R shall be checked against the data types defined by the business rules.

4.4.8. Maintenance

Currently, the organization just started its analytical initiatives with no centralized model repository available. Once the organization increases its analytical maturity level, the model shall be semantically described and stored for future reuse. A model maintenance process was designed specific to the control-chart modeling approach. The model maintenance starts with the initial deployment. An hourly reconnect control chart was created. If more than 10% of sub-regions were identified with abnormal device behaviors, the model assumptions need to be checked. In addition, it was recommended to refresh the mean and standard deviation rates once

a week to capture the trend. Current the model is only used by internal BI team. It was implemented in the EDW, which inherits the security measures of the EDW.

Soon after the model was deployed (less than 3 weeks), one of the reconnects (reconnect reason 2) was disappeared from the control chart, while another (reconnect reason 1) alerted more than 50% sub-regions with abnormities. The BI team quickly looked into the reconnect data and traced down the issues. It turned out that application developers rewrote one application process to redirect reconnect reason calls. As a result, reconnect reason 2 was captured as reconnect reason 1. From a business point of view, reconnect reason 1 and 2 represented two different types of important user experiences. They are considered two important leading metrics for measuring SBO device health. However, it will still be a continuous battle with the application developers without a data governance plan in place. The modeling result was suspended until the correct reconnect reasons can be captured.

CHAPTER 5 APPLICATION OF KDDA PROCESS MODEL – A METHODOLOGY FOR THEORY BUILDING BASED QUALITATIVE DATA

In this chapter, I present a novel application of KDDA process model towards a methodological approach for theory building based qualitative data. A preliminary version of this work is published as a book chapter (Li et al. 2014). Theories can be broadly defined as coherent descriptions or explanations of the reality (Gioia et al. 1990). According to Dubin (1978, p.26), "*A theory tries to make sense out of the observable world by ordering the relationships among elements that constitute the theorist's focus of attention.*"

Theory building is a process by which the theoretical presentation is generated, tested, and/or refined. Gioia and Pitre (1990) identified four steps towards theory building: *opening work, data collection, analysis, and theory building*. *Opening work* involves research topic identification and research design. Based on the research design, *data collection* involves gathering data that are relevant to the research using techniques such as surveys, archival data, interviews, observations, etc. *Analysis* can take different forms based on the nature of research design and data collected. Traditional positivist approaches uses deductive reasoning and causal analysis to evaluate the significance of the data, while interpretive approaches advocate inductive

reasoning to identify emergent concepts and relationships. The last step is *theory building* where the findings related to the phenomenon of interest are summarized as a theoretic representation.

The four steps presented by Gioia and Pitre (1990) are applicable across different research paradigm, namely interpretivist, radical humanist, radical structuralist, and functionalist (Burrell et al. 1979). Eisenhardt (1989) describes a process of building theory from case study research. The proposed theory building process includes 8 steps: *getting started, selecting cases, crafting instruments and protocols*, *entering the field*, *analyzing data*, *shaping hypotheses*, *enfolding literature,* and *reaching closure*. *Getting started* is to provide an initial definition of the research question and possibly a priori constructs. Similar to the hypothesis-testing research, selecting cases involves the selection of an appropriate population from which the research sample is to be drawn. However, different from the hypothesis-testing research, if sampling of cases is needed, it relies on theoretical sampling (Glaser et al. 1967b) instead of statistical sampling. *Crafting instruments and protocols* step combines multiple data collection methods (e.g., interviews, observations, archival sources, quantitative laboratory data), and different types of data (qualitative, quantitative, or both). *Entering field* is the process of data collection that frequently overlaps with data collection. *Analyzing data* is the key of building theory from case studies, and the most difficult one. Eisenhardt (1989) noticed the limited attention to qualitative data analysis and identified several key features of analysis (i.e. within-case analysis, search for cross-case patterns, and overall impressions). *Shaping hypotheses* is a highly iterative process to compare theory and data towards a theory, which closely fits the data. *Shaping hypotheses* includes both shaping constructs and verifying the emergent relationships between constructs. *Enfolding literature* is the comparison of the emergent concepts, theory, or hypotheses with

similar or conflicting literature. *Reaching closure* is to decide when to stop adding cases and when to stop iterating between theory and data.

Limitations of existing theory are exposed when a particular phenomenon arises that is not explainable by the theory. Under such circumstance, positivist approaches often fall short as they rely on verification or falsification of the hypothesis and consequently limited to incremental revision or extension of the original theory. On the other hands, theory building based on qualitative methods center directly on the juxtaposition of contradictory evidence to provide novel insights (Eisenhardt 1989), thereby addressing the above shortcomings of quantitative approaches. The strength of qualitative methods in theory development and refinements relies in the rich knowledge captured in the qualitative data. It helps to develop theories that are relevant, rich, and dynamic in their explanations of social processes (Fine et al. 2000).

Although the richness of qualitative data is invaluable, it can also pose a challenge to the theory development process. In the quest to explain the data, a qualitative researcher can easily go in the direction of making theoretical propositions that are rich in detail, yet lacking in simplicity, a key dimension for good theory (Weick 1979). The nature of the qualitative data precludes the researcher from using quantitative techniques such as statistical testing to identify strong relationships between concepts that are identified through coding process. In addition, the researcher faces the daunting task of developing persuasive arguments to justify the findings. A systematic approach towards qualitative data analysis is therefore compelled to facilitate developing propositions, and establishing the strength and consistency of the findings. To best of our knowledge, this aspect of analysis has not been addressed. For example, grounded theory (Glaser et al. 1967a), a widely used theory building methodology in social science, introduces

constant comparisons as its data analysis strategy to explain patterns in qualitative data. However, the grounded theory building is a descriptive process rather than prescribing how. It is left to the researcher to justify when theoretical saturation is achieved (i.e., further refinement of the concepts and their relationships add little new to the conceptualization). Yin (2003) has recommended several qualitative data analysis tactics (i.e. pattern matching, explanation building, addressing rival explanations, and logic models) to test validity and reliability of the qualitative research design. However, these tactics only test internal validity for explanatory or causal studies, whereas studies in theory building are exploratory in nature. Miles and Huberman (1994) have presented three types of qualitative data analysis activities (i.e., data reduction or coding, data display in matrix or graphs, and conclusions drawing and verification). Their focus is on managing and representing qualitative data without losing their meanings through intensive coding. Eisenhardt (1989) has recommended common qualitative data analysis techniques such as within-case analysis and cross-case pattern search when building theory from case study research. None of these aforementioned data analysis techniques assists the systematic identification and justification of concepts relationships.

The objective of this chapter is to demonstrate how researchers can take advantage of KDDA process model and quantitative data analysis techniques such as association rules (AR) mining to identify strong concept relationships using qualitative data. The underlying philosophical differences between qualitative and quantitative research do not prevent the combination of the two. There are several contexts (Denzin et al. 1994; Gioia et al. 1990; Jick 1979) where both have been used in conjunction during theory building. However, these contexts focus on either research design or data collection, where the data analysis part receives little attention. The rest of this chapter is organized as follows. I first provide an overview of

association rules induction, followed by a description of the proposed methodological framework for qualitative theory building using both quantitative and qualitative techniques. The KDDA process model guides the design of the proposed theory building methodology, which is compared to the qualitative theory building process and steps identified by Gioia and Pitre (1990). The proposed methodology is illustrated using a case study in the public health domain. Finally, I conclude by stating the contributions of this approach.

5.1. ASSOCIATION RULE INDUCTION

Association rules (AR) mining is a popular pattern discovery method in knowledge discovery and data mining (KDDM). It was first introduced by Agrawal et al. (1993) to mine large transactional databases. A transactional dataset for AR analysis can be defined in the following general terms. Let I = $\{i_1, i_2, \ldots i_m\}$ be a set of distinct items, and T = $\{t_1, t_2, \ldots t_k\}$ be a set of k subsets of I. Each t_i is a transaction such that $t_i \subseteq I$. For example, in market basket analysis, each basket is a transaction that contains the set of items purchased from one register transaction, and the set I consists of the items stocked by a retail outlet.

The objective of AR mining is to find items that imply the presence of other items in the same transaction. It can be expressed as A=>B (for e.g., bread => Peanut Butter & Jelly), where A and B are sets of items in a given transaction t_i, and A=>B meets both the minimal support and minimal confidence constraints. Support specifies the probability that a transaction t_i contains both item A and B. Confidence specifies the conditional support, given that the transaction already contains A. It should be noted that an AR does not always imply causation. Both support and confidence constraints are probability-based measures.

Table 19: Some Interestingness Measures for AR

Measure	Formula
Support	$P(AB)=\frac{n(AB)}{N}$
Confidence	$P(B\|A)=\frac{P(AB)}{P(A)}$
Coverage	$P(A)=\frac{n(A)}{N}$
Lift	$\frac{P(B/A)}{P(B)}$
Collective Strength	$\frac{P(AB)+P(\neg B\|\neg A)}{P(A)P(B)+P(\neg A)\times P(\neg B)} \times \frac{1-P(A)P(B)-P(\neg A)\times P(\neg B)}{1-P(AB)-P(\neg B\|\neg A)}$
Expected Confidence	$P(B)$
Reliability	$P(B/A)$-$P(B)$

The advantage of AR mining lies in finding all possible associations between relevant factors and presenting results in a simple and understandable manner. It has been applied to uncover interesting patterns in different application areas such as market basket analysis (Agrawal et al. 1994), web mining (Liu et al. 2004), safety science (Montella 2011), medical records analysis (Chang 2007), and questionnaire analysis (Chen et al. 2009), etc. While AR mining has its key strength in its understandability and completeness (Liu et al. 1999), not all ARs are interesting. The candidate AR set often contains a large number of associations, making it difficult, and sometimes impossible to comprehend. Additional forms of rule interestingness measures have been developed to evaluate and select ARs based on their potential interestingness to the user (Geng et al. 2006). Examples of these interestingness measures include: coverage (Piatetsky-Shapiro et al. 1991), lift (Brin et al. 1997), collective strength (Aggarwal et al. 2001), and reliability (Ahmed et al. 2000). The reliability measure proposed by Ahmed et al. (2000) measures the difference between confidence and expected confidence of an AR, which is the effect of A on the probability of B. Because the reliability measure is a probability, it can be used in classical hypothesis testing. Table 19 shows the mathematical representation of objective measures, where n (A) denotes the number of transactions that

contains A, n (AB) denotes the number of transactions contains both A and B, N denotes the total number of transactions, P (A) denotes the probability of A, P (¬A) denotes probability of not A, and P (B/A) denotes the conditional probability of B.

5.2. THEORY BUILDING BASED ON QUANLITATIVE DATA METHODOLOGY

As mentioned earlier, the objective of this chapter is to demonstrate how KDDA process model can be applied to facilitate qualitative theory building. This can be viewed as a special instantiation of KDDA process model in a specific context. More specifically, analytical techniques such as AR can be applied to discover strong associations among concepts identified from qualitative data. In chapter 4, I provide the description of phases and tasks in KDDA process model, which appears as discrete steps in a specific order. In practice, however, the instantiation of the KDDA process model may have tasks in different orders and may not include all the tasks being identified. Table 20 summarizes the process of theory building based on qualitative data by its phases and tasks. Each theory building phase is mapped to the proposed KDDA process model phase. Notably, the deployment of proposed theory is the theory testing, which is not part of theory building. The change initiation in the maintenance of deployed theory should be captured when the theory testing has conflicting results. Thus, the deployment and maintenance phases of KDDA process are not mapped.

Table 20: A Methodology for Theory Building Based on Qualitative Data

Theory Building Phases	KDDA Phases	Theory Building Tasks	Description
Research Question Formulation	Problem Formulation	Define research goals and objectives	Qualitative theory building falls in interpretive paradigm with its goal to describe and explain the phenomena in order to diagnose and understand
		Deploy problem formulation strategy	Determine the contextual boundary of the research, and factor complex problems into sub-problems.
		Define research question	A research question is a statement that makes explicit the specific area of interest within the area of general concern (Lewis et al. 1987).
		Determine data analysis goal	The goal of data analysis determines applicable analytical techniques. The relevant evaluation criteria for data analysis shall also be identified.
Research Background Understanding	Business Understanding	Understand study background	The research background should include the understanding the theoretical background and contextual background.
		Define research case	The research case should include all assumptions, limitations, constraints, as well as resources needed. Initial study site shall be identified.

		Knowledge acquisition (literature review)	Literature review allows the researcher to have a firmer empirical grounding for the emergent theory.
		Initial tools and technologies selection	This includes both data collection, and data analysis tools and techniques.
		Create a research plan	The research plan shall include stages, duration, resources required, etc.
Data Collection	DU	Select cases and collect data	Select case study site and collect data.
		DU for contextual case requirement	Understand data based on the contextual requirements of the case studies.
		DU for analytical requirement	Detailed tasks vary based on the analysis techniques selected.
Data Preparation	DU	Verify data quality	Verify the data completeness, correctness, accuracy, consistency, etc.
	Data Transformation	Data Transformation (Qualitative to Quantitative)	The transformation is based on the analytical technique selected, as well as nature of the research questions.
	Data Integration	Prepare dataset for analysis.	The dataset requirement is based on analytical technique(s) and tool(s) selected.
Data Analysis	Modeling	Build model	Use quantitative analytical techniques to analyze qualitative data.
		Assess Model	Assess models using previous

			defined
Theory Building	Evaluation	Evaluate results	The result is evaluated using existing literature.
		Propose Theory	Communicate the results to the research community.

5.3. ILLUSTRATION OF PROPOSED METHODOLOGY

In this section, a cross-disciplinary research (Thomas et al. 2014) in the public health domain is used to illustrate the proposed methodology. The research aims to reduce the burden of ill health in low-income communities by developing Health Management Information Systems (HMIS) as an intervention.

5.3.1. Research Question Formulation

The first step to develop health intervention programs in the public health domain is to trace the health behavior (Huda 2006). Extensive literature review in related theories to explain individual health behavior is conducted (see next section). All behavior predicting models in the public health domain are conclusively drawn from empirical studies of population in developed nations. The models were developed based on data collected from the strata of society that have easy access to preventive and diagnostic services. Even the popular Health Behavioral Model (HBM) (Figure 24) finds greater applicability to middle class groups than lower status groups (Rosenstock 2005). The existing models thus fail to predict the health behavior among the under-served communities. It highlights the need for refining the theoretical conceptualizations to account for those living in the very low-income and low-resource societies. The research goal is

to describe and explain health behavior in these marginalized communities in order to understand underlying theoretical concepts and their relationships.

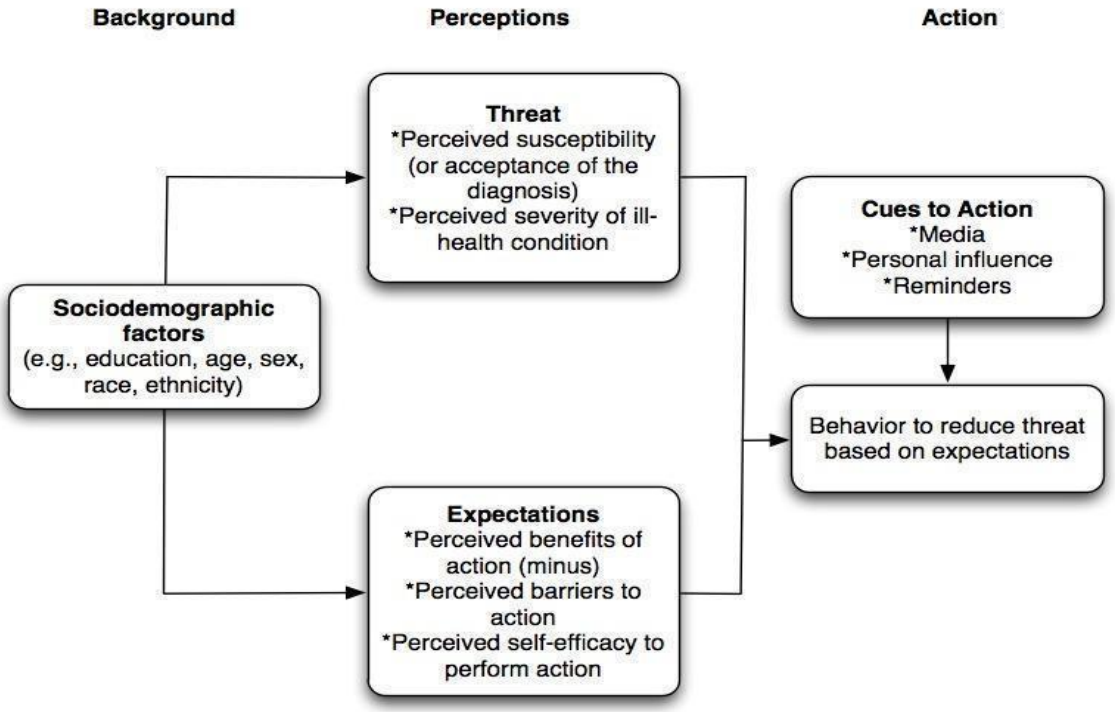

Figure 24: The Health Belief Model (adapted from Rosenstock et al., 1994)

Based on the research goal, the authors defined the contextual boundary of the research as health seeking behavior of women in low-income low-resource communities of developing countries, focusing specifically on Reproductive Tract Infections (RTI) prevention. The reason for focusing on RTI-related health in women is also the result of research understanding phase. First of all, literature observes that women in general report significantly lower levels of illness compared to men (Robeyns 2003; Sen et al. 2000). Secondly, the risks of women exposed to RTI in developing countries are high, and the mortality and morbidity of RTIs are serious. With a clear defined boundary, the research question is to develop a theory to explain the women's health-seeking behavior in marginalized communities of developing countries.

To answer this research question, a qualitative research method – ethnographic methodology was adopted. The ethnographic methodology is the trademark of cultural anthropology (Schwartzman 1993) , which tries to understand human action in natural attitude of everyday life, or within cultural circumstances. The authors then carried out more research understanding tasks, including understanding research background, defining research case, and additional knowledge elicitations (see details in the next section). Based on the output of these tasks, an attempt to explain the high incidence of specific illness such as reproductive tract infections (RTI) requires identifying key associations among the relevant contextual factors. Traditional qualitative data analysis techniques fall short in assisting the systematic identification and justification of concepts relationships. Instead, AR mining technique was selected as it fits the explorative nature of the study by providing a means to analyze the qualitative data using predetermined objective measures.

5.3.2. Research Background Understanding

The first step in this phase is to provide a theoretical background on the research problem. Established theories from the public health domain served to set the focus for the problem analysis. Many models have been proposed to explain the notion of health behavior in individuals. Munro et al. (2007) conducted a detailed literature review covering scholarly databases, electronic libraries and citations to identify theories and models developed for the domain of 'health and behavior'. The study identifies nine prominent theories - *Behavioral Learning Theory* (BLT), *Health Belief Model (*HBM*), Social Cognitive Theory (SCT), Theory of Reasoned Action (*TRA*), Protection Motivation Theory (*PMT*), Theory of Planned Behavior (TPB), Information Motivation Behavioral (*IMB*) skills model, Self Regulatory Theory (*SRT*),*

and *the Transtheoretical Model (*TTM*)*. The models have been applied in varying settings ranging from developing interventions for cervical cancer prevention among Latina immigrants (Scarinci et al. 2011) to identifying the variables that can influence a smoker's motivation to quit (Norman et al. 1999).

Among them, HBM (Rosenstock et al. 1994) is perhaps the most widely used theory to explain health related behavior (Urrutia 2009). The main components of the HBM and the key variables under each category are shown in Figure 24. It takes a cognitive perspective to explain and predict preventive health behavior (Hayden 2008). The six main constructs in HBM are: *Perception of Susceptibility to A Disease Or Condition, Perceived Severity, Perceived Benefits of Care, Cues to Action, Self-Efficacy,* and *Barriers to Preventive Behavior* (Strecher et al. 1997). The model suggests that the *Perceived Seriousness* and *Susceptibility to a Disease* influence an individual's perception of the threat of disease. It posits that the likelihood of engaging in a recommended health behavior is based on an assessment of the benefits of a health modifying action to the barriers in place (Munro et al. 2007; Urrutia 2009). *Self-Efficacy* is the person's conviction to produce behavioral outcome (Hayden 2008; Scarinci et al. 2011). *Cues* are the bodily (e.g., identified symptoms) or environmental events (e.g., information sourced by media, reminders, incentives, or information imparted by peers and family members) that prompt an individual to adopt health modifying actions (Hayden 2008; Strecher et al. 1997).

To understand the contextual background of the research problem, the authors identified few attempts that are made to study the health-seeking behavior of women living in marginalized communities in India (Patel et al. 2003; Prusty et al. 2012). Initial study site identified by the authors is a marginalized community comprising of 510 families, of which 79% of the men are employed in the fisheries sector. Due to the seasonal nature of employment and small scale of

operations, men work a maximum of five months in a year. The women (approximately 200) mostly work as contract laborers in fish processing jobs. The socioeconomic welfare of the respondents in the region is low (the average income ranges from $200- $450 per-annum). Although the Panchayat (village) census lists all the houses in the ward with male household heads, primary data reveals that women are the virtual heads of the family. Thus, the burden of running the family rests on the shoulders of the women. To a large extent, unequal opportunities to act account for the prevalence of communicable and non-communicable diseases that go unchecked. The community is an exemplar of a subsistence economy. Open access mode of resource use, technological dualism, lack of educational attainment and class identity, and the absence of strong socio political movements from within the community are push factors that retain them as a marginalized group (KDR 2008).

Ethnography requires the researchers to study a native on its natural attitude of everyday life by participating in native's daily life, watching what happens, listening what they say and asking questions relates to the objectives of the study. Thus, face-to-face interviews, focus group discussions, iterative follow-up meetings were identified as initial data collection techniques. The analysis tools are constrained by their availability. Because I have access to both SAS Enterprise Miner (academic research licensing) and Rapid Miner (open source) and both tools implement AR mining technique, they are both chosen for analysis. A research plan was created, which included the need for ethical approval, estimated duration of the project, as well as preliminary data analysis plans.

5.3.3. Data Collection

After obtaining the ethical approval from a multi-disciplinary, multi-sectored Ethics Committee, the ethnographic study was conducted over a period of eight months starting from mid-2011. The project commenced with a medical camp organized in the community. To encourage women to attend the medical camp and to assure participation, the camp was promoted by volunteers by means of a door-to-door campaign. The camp had a pediatrician, general physician, and gynecologist. Sixty-eight women attended the camp. Forty-five women (66%) were identified with a confirmed diagnosis of RTI using the syndromic approach. The medical camp helped develop a rapport with the women and gain acceptance in the community, following which the researchers made numerous visits to the community to gain access to the social network and initiate discussions regarding RTI. After the post-camp interactions, a series of neighborhood visits were made to meet with the women individually, as well as in-group settings.

After gaining the confidence of the participation, eight personal interviews were conducted during this time. The interviews were with the women who had participated in the medical camp and had confirmed symptoms of RTI. In addition, seven focus group discussions (FGD) were conducted with those who were identified with symptoms of RTI during the medical camp and the post-camp interactions. The FGDs reinforced the notion of the 'culture of silence' regarding RTIs that is prevalent among the women. The questions in the interview evolved around attitudes towards RTIs and realization of symptoms, perceived threat of RTIs, stigma and related disclosure issues, impact on social and marital relations, and barriers to pursuing treatment. All interviews were anonymized and transcribed before analysis.

After collecting data through face-to-face interviews, focus group discussions and iterative follow-up meetings, four subject experts systematically analyzed statements from 8 subjects and 7 focus groups. The analysis followed the coding technique suggested by Hammersley & Atkinson (2007) and Denzin (1997). Total 158 statements were evaluated in this manner.

5.3.4. Data Preparation

The first step in data preparation is to apply content analysis to identify the key concepts from the narrative data. Based on the HBM, the interview statements were coded across two dimensions consisting of 10 concepts - four subjective constructs and six behavioral factors. The subjective constructs are the belief, desire, intention, and likelihood of action. The factors influencing behavioral outcomes are *Susceptibility*, *Severity*, *Benefits*, *Barriers*, *Cues to Action*, and *Self-Efficacy*. In addition, the descriptive properties of the environment (socio-economic and cultural structure) that shape the likelihood of adopting a health behavior change were also coded by the researchers.

The second step is to transform data into an integrated format for AR mining. AR mining requires preprocessing data into a transactional dataset that includes multiple transactional items in the same transaction. In our case, the transaction is the equivalent of one coded statement by one researcher, and the transactional item is the equivalent of the individual concept in each statement. To uniquely represent each coded statement, an ID is assigned to the coding result in the format of S_x-R_y-Z, where S_x is the subject ID, R_y is the research ID, and Z is the statement ID. The next step is to transform the coded items by assigning a '1' for the concept identified in S_x-R_y-Z, and '0' otherwise.

The data quality is verified by creating a probability measure, which is defined as the proportion of concept-researcher combination in which a given link appeared. If a link appeared multiple times in a given concept-researcher combination, it is counted as one. The overall probability of two item relationships is summarized as in Table 21. Inter-rater reliability is calculated using Cosine similarity between researchers as shown in Table 22. It can be seen from the table that R2 could be a possible outlier. Additional review of the coded data and researcher background reveals that the R2 is strongly biased and the coding result from R2 is subsequently removed.

Table 21: Two-item Relationship Probability

Relationships	Probability
Belief-Severity	0.52
Belief-Susceptibility	0.57
Belief-Barrier	0.73
Desire-Cues	0.15
Desire-Benefit	0.33
Desire-Barrier	0.55
Intention-Cues	0.17
Intention-Benefit	0.37
Intention-Barrier	0.37
Intention-Efficacy	0.33

Table 22: Inter-rater Reliability Measure

	Similarity: Cosine			
	R1	R2	R3	R4
R1	1.00	0.53	0.64	0.75
R2	0.53	1.00	0.45	0.49
R3	0.64	0.45	1.00	0.61
R4	0.75	0.49	0.61	1.00

The final dataset includes a total of 474 transactions and 1331 transactional items. Table 23 shows the summary of transaction counts. The input data requirement for Rapid Miner is different from SAS Enterprise Miner. Enterprise miner requires the dataset to be constructed in two-column format – one ID field and one transaction field for each item coded as 1. The final dataset has total 1331 rows. Rapid Miner accepts the dataset in matrices, where each transaction has an ID and all 10 items with each item is marked as one or zero. The final dataset for Rapid Miner has total 474 rows.

Table 23: Transaction Count Summary

Concept	Count		Concept	Count
Belief	226		Susceptibility	89
Desire	136		Severity	133
Intention	123		Benefits	100
Likelihood	64		Barriers	258
			Cues	87
			Efficacy	116

5.3.5. Data Analysis

The data analysis includes two steps: rule discovery and evaluation. Each step is elaborated as follows.

Step 1: Association Rules Discovery

I first use SAS Enterprise Mining 9.3 as the AR mining tool. The *Association* node is used to extract candidate ARs with parameter settings of 10% minimal confidence, 5% minimal support, and leaving all other settings as default. Figure 25 displays the rule matrix for all candidate ARs before pruning. The candidate AR set includes 91 2-item or 3-item ARs, with the *Confidence* between [15.63, 69.32] and *Lift* between [0.25, 4.32].

I then use Rapid Miner 5 as the AR mining tool with the same parameter settings. Retrieve operator is used to read the dataset into Rapid Miner. FP-Growth operator is used to efficiently calculate all frequent items sets from a data set by building an FPTree. Comparing to the Apriori approach in SAS, Rapid Miner should provide a performance advantage if the dataset is large. In my analysis, performance issue is not a concern because the dataset is rather small. Create Association Rules operator is then used to generate a set of AR for a given measure (only one measure is supported at a time).

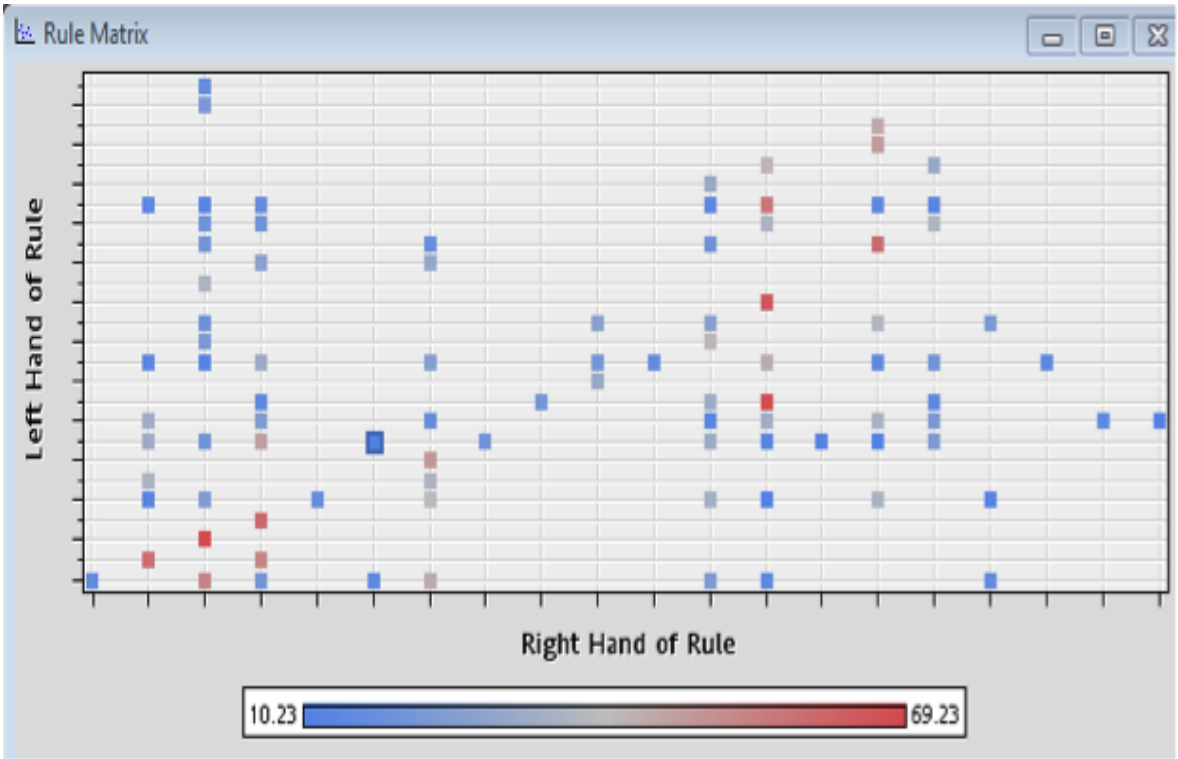

Figure 25: Candidate Rule Matrix

However, the Rapid Miner provides more measures than SAS Enterprise Miner. In addition to Support, Confidence, and Lift that are supported in SAS Enterprise Miner, Rapid Miner also provides these measures: LaPlace, Gain, p-s, and Conviction. If a researcher wants to use these additional measures to assess the AR result, Rapid Miner should be used. In addition, Rapid Miner provides better linkage graph then SAS Enterprise Miner, as shown in Figure 26.

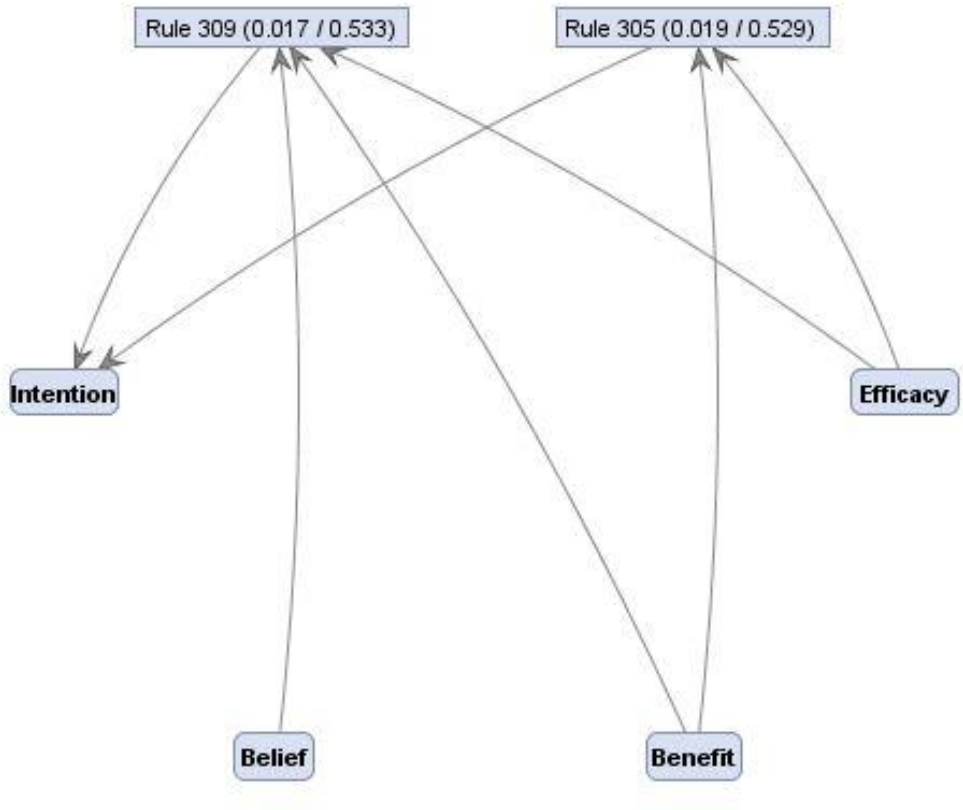

Figure 26: Graphic View in Rapid Miner AR Rule Analysis Result

Step 2: AR Pruning

Sub-Step 2a: Pruning using Reliability

To determine the ARs that are significant, the Reliability score and t statistic for all candidate ARs are first calculated. The reliability is calculated using the *Confidence* (*C*) and *Expected Confidence* (*EC*) from the output table, where *Reliability* = (*C* – *EC*). Based on this, the following population proportion hypothesis testing is performed:

H_0: The difference between the confidence of the AR and the *Expected Confidence* of the AR is not statistically significant.

H_1: The difference between the *Confidence* of the AR and the *Expected Confidence* of the AR is statistically significant.

Table 24: Final Result ARs (*p<0.01)

AR ID	Confidence (%)	Support (%)	Lift	Transaction Count	AR	Reliability (%)	T
1	68.06	10.43	1.72	49	Susceptibility ==> Belief	28.48	246.35*
2	26.34	10.43	1.72	49	Belief ==> Susceptibility	11.02	181.45*
3	60	14.68	1.48	69	Desire ==> Barrier	19.36	194.38*
4	36.13	14.68	1.48	69	Barrier ==> Desire	11.66	170.14*
5	56.76	8.94	1.43	42	Cues ==> Belief	17.18	137.60*
6	22.58	8.94	1.43	42	Belief ==> Cues	6.84	102.48*
7	53.13	7.23	2.4	34	Likelihood ==> Efficacy	31	339.07*
8	32.69	7.23	2.4	34	Efficacy ==> Likelihood	19.08	280.15*
9	46.59	8.72	2.09	41	Benefit ==> Intention	24.25	289.51*
10	39.05	8.72	2.09	41	Intention ==> Benefit	20.32	271.14*
11	43.75	5.96	2.34	28	Likelihood ==> Benefit	25.03	275.91*
12	31.82	5.96	2.34	28	Benefit ==> Likelihood	18.2	242.58*
13	43.75	10.43	1.11	49	Severity ==> Belief	4.18	36.12*
14	26.34	10.43	1.11	49	Belief ==> Severity	2.51	31.47*
15	33.33	7.45	1.4	35	Intention ==> Severity	9.5	100.52*
16	31.25	7.45	1.4	35	Severity ==> Intention	8.91	98.27*
17	31.82	5.96	1.34	28	Benefit ==> Severity	7.99	75.57*
18	25	5.96	1.34	28	Severity ==> Benefit	6.28	69.20*
19	24.04	5.32	1.08	25	Efficacy ==> Intention	1.7	15.83*
20	23.81	5.32	1.08	25	Intention ==> Efficacy	1.68	15.78*

A one-tail t-test is then performed and forty-nine (49) ARs were found to be statistically significant at significance level α=0.01 (supporting H_1), which means that those ARs have a greater confidence level than expected. The remaining 42 ARs were pruned.

Sub-Step 2b: Pruning using Understandability

The example presented is an exploratory case study of a problem domain that is inherently complex. To comprehend the interactions among the concepts, the *understandability* of the AR is a fitting interestingness measure. The number of items in a given AR has been shown to be such a measure (Freitas 1999), where ARs with fewer antecedents (fewer items in A) are considered to be easier to understand. Similarly, Gen and Hamilton (Geng et al. 2006) consider the conciseness of an AR as an important perspective in rule interestingness measures. A concise AR contains relatively fewer items and is thus easier to assimilate. Hence, all ARs that have more than two items were pruned. This leaves total of 22 ARs.

Sub-Step 3c: Pruning using Lift

The premise of this study is to find associated concepts that are departing from independence and positively correlated. *Lift* is a measure of departure from independence (Brin et al. 1997). A lift value greater than 1 means that A and B appear more frequently together than expected under independence, and vice versa. Thus, all ARs that have lift value less than or equal to 1 are pruned. This results in 20 ARs, which is shown in table 3.

Research experts familiar with the research setting will be particularly interested in concepts with strong relationships. The ARs presented in table 3 are organized in pairs, where each pair contains the same rule items, but the antecedent and the consequent are switched. Thus, each pair can be expressed as (A=>B and B=>A). As mentioned previously, an AR does not imply causality. This indicates ten pairs of strong ARs that will require careful interpretation.

For better understandability of the results, we only consider two-item ARs in the illustrative example presented above. AR mining may not have identified certain expected associations. Furthermore, the illustrative example presented here only uses objective interestingness measures (confidence, lift, simplicity and reliability) in the AR pruning step. This may not identify the most important association patterns to the researcher (McGarry 2005). Including subjective interestingness measures (such as actionable, unexpected and novelty) in the evaluation can provide a deeper understanding of the problem domain.

5.3.6. Theory Building

The output of AR pruning (Step 2) results in a set of strong ARs, each of which can be a candidate proposition. The candidate proposition may substantiate into a theoretical proposition only if a persuasive explanation can be provided by the researcher. This assessment is achieved by the researcher's constant reflection on the phenomenon under the study, the evidence gathered from the case, and the existing literature and theory. The candidate propositions may quite likely be expected relationships or unexpected relationships. Hence, the researcher needs to first revisit the case evidence to determine why expected candidate propositions hold. For example, the AR pair Benefit & Intention form two candidates propositions (AR9 and AR10 from Table 24). By reflecting on data acquired from the fieldwork, it is observed that the women in the marginalized communities tend to pursue a health improving behavior when the benefits of such action (e.g., benefits for the immediate family members assured by one's health as a motivating factor to seek medical care) are seen to reduce the disease threat. Hence, AR9 is suitable for further proposition development and AR10 is eliminated. The higher reliability of AR9 (reliability = 24.25) provides

additional support for this selection. In case of unexpected relationships, the case evidence can provide the researcher with many opportunities and flexibility to gain new insights.

Another essential step is to compare the candidate propositions against existing literature to determine whether they are similar to, or contradict previous studies. Understanding the theoretical reasons as to why specific relationships exist demonstrate the internal validity of the findings. Where conflicting relationships are found, the researcher has to seek additional evidence to discern the plausible reasons, and reconcile the findings. For example, folk psychology (Malle et al. 1997) suggests a significant association between intention and likelihood of action, though this association did not emerge as a strong AR from the analysis. This motivates the research to go back to the data and perform additional AR mining to include three or more items in ARs. For instance, the SAS linkage graph shows the possibility of association between intention and likelihood with efficacy as an influencing factor (Figure 3).

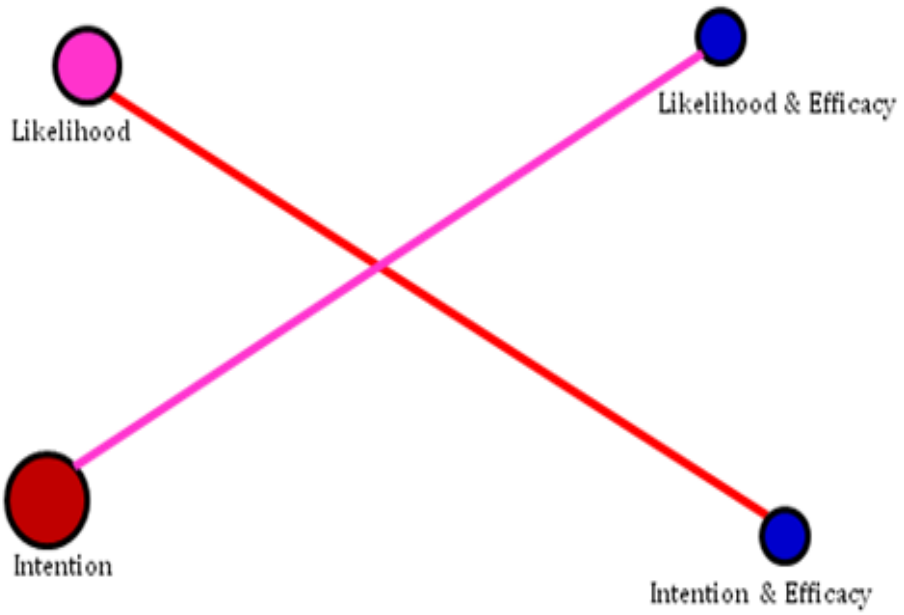

Figure 27: SAS Linkage graph Result for Intention & Likelihood of Action

Only after these iterative steps are complete, a set of theoretical propositions can be expounded. The set of theoretical propositions can then be summarized into a logical, systematic explanatory schema for future theory testing (Glaser et al. 1967b).

5.4. CONCLUSTION

This chapter presents a novel application of KDDA process model in theory building based on qualitative data. The methodology presented makes theoretical contributions towards theory building. AR is used to demonstrate how analytical method(s) can be used to determine the associations among constructs identified via content analysis of qualitative data. Researchers working with voluminous qualitative data often struggle to find an even ground between the richness of the observations and the simplification of the findings. AR provides a quantitative gauge to assess the important construct relationships during the formative phase of theory building. Future research shall explore three-item rules and propose a testable theory.

CHAPTER 6 DATA MINING MODEL MANAGEMENT ONTOLOGY

Organizational knowledge and its management are widely recognized as critical factors for organizational success and competitive advantage (Zack et al. 2009). With the advancement of information technologies (IT), knowledge management systems (KMS) are prescribed by information systems (IS) community to support and enhance the organizational processes of knowledge creation, storage, retrieval, transfer and application (Alavi et al. 2001). Among these technologies, knowledge discovery and data mining (KDDM) (Fayyad et al. 1996a) is a key enabler in knowledge creation. However, current KDDM research mainly concerns the knowledge creation process, where data mining models are built, evaluated, and applied in a specific domain. What goes beyond the deployment of data mining models has received little attention.

Data mining models are knowledge intensive information products that are not only expensive (Leavitt 2002) to build but also hard to maintain. A model management environment is thus desired to support model building and reuse (Liu et al. 2008). Approaches for data mining model management (DMMM) have been proposed in academics and in industry (e.g., IBM intelligent miner model management console, SAS model manager, etc). However, the current model management practice follows the traditional closed world approach, where data mining models are built and stored in a centralized repository. This limits the ability to share and use

models outside of the application. To address this limitation, the data mining group (DMG) developed the Predictive Model Markup Language (PMML), an XML-based open standard language for representing data mining models. The PMML enables model sharing between PMML-compliant applications. However, current PMML schema only describes characteristics of the data mining models. Knowledge from the BU phase in the KDDM process is not captured in the PMML. The issue is escalated in today's big data environment that calls for placing the power of data and knowledge discovery in the hands of business users and providing ad-hoc business queries on the fly. This need is evident in the Business Intelligence (BI) landscape, where self-service BI is gaining significant momentum.

According to Gartner's IT glossary (Gartner 2013), self-service BI is defined as "end users designing and deploying their own reports and analyses within an approved and supported architecture and tools portfolio". Similarly, self-service knowledge discovery is desirable to enable the business users to query and deploy their own analytics from an approved and supported repository of data mining models. Various research efforts for model selection include intelligent model selection support through meta-learning (Matijaš et al. 2013; Vilalta et al. 2002), and mechanisms for selecting appropriate models in the model evaluation process (Osei-Bryson 2012). These approaches are mainly from a knowledge engineer's perspective, which requires a thorough understanding of data mining processes and algorithms (Zorrilla et al. 2013). There exists a semantic gap between knowledge engineers who discuss decision trees and misclassification rates, and business users who converse customer profitability prediction, shun rate, etc. The knowledge engineer-focused approach thus posits two major limitations. First, the business requirements from business users are often lost in the modeling process. Second, business users may not have the statistical or technical knowledge to perform knowledge

discovery tasks themselves. The problem is accelerated by the need for real-time analytics when they are required to make better and faster decisions. What is missing is a DMMM capability that enables business users to query available data mining models based on specific decision criteria and discover appropriate data mining model(s) that produce desired decision outcomes.

The interests of using the web (both intranet and internet) for knowledge management and knowledge sharing have been stressed by several research communities, such as database, intelligent systems, knowledge engineering, and machine learning (Schwartz 2003). While intranets are widely used by organizations as a means for enterprise knowledge sharing, the internet has been viewed as an open knowledge base. The semantic web technology provides a common framework that allows distributed knowledge sharing and querying. The semantic web itself represents a dynamically growing knowledge based system (Kaufmann et al. 2010). Collaboration of researchers from the Semantic Web, social network analysis and machine learning communities has contributed many data mining ontologies for knowledge sharing and knowledge reuse. Current ontology-based research in KDDM mainly focuses on data mining modeling and evaluation, while DM model selection and reuse have received little attention. In this chapter, I address these issues by developing a DM^3 ontology to enable self-service knowledge discovery. I illustrate the use of the DM^3 ontology to translate the business user's requirements into model selection criteria and measurements, and automatically retrieve relevant models from the model repository.

This chapter makes two contributions. First, I present a novel design of DM^3 ontology. To best of our knowledge, this is the first attempt to develop an ontology that serves as a user-centric semantic model for data mining model selection and reuse. The DM^3 ontology includes additional annotations to describe data mining models across PMML-compliant applications.

Second, I provide a knowledge-sharing architecture that utilizes the DM^3 ontology to support self-service knowledge discovery.

The rest of this chapter is organized as follows. I first provide an overview of ontology. I then outline the ontology design methodology adapted in this research, followed by the description of DM^3 ontology design. The deployment of the ontology within an organization's intranet for self-service and discovery of data mining models is presented. The DM^3 ontology design is then evaluated using pre-defined ontology design criteria. The usability and utility of DM3 ontology are illustrated using a private student loan case. This is followed by concluding remarks and directions for future research.

6.1. ONTOLOGY BACKGROUND

An ontology is a formal, explicit specification of a shared conceptualization (Gruber 1993). It provides a means of explicitly representing domain-specific knowledge in an interoperable format that can be understood by both humans and machines (Chen 2010). An ontology-based approach can provide a formal representation of DMMM concepts, their attributes and relationships for model selection and reuse. Using ontology as the knowledge model allows different types of users to share their common understanding of DMMM, and thus bridge the semantic gap.

A popular standard for semantic knowledge representation is the Ontology Web Language (OWL). OWL uses the normative RDF/XML format for ontological context representation and meta-language definition that can be used for reasoning. In OWL, the ontology is defined as a set of classes, individuals, and properties:

- `Classes`: a class is a set of naturally occurring things in a domain of discourse. Classes are defined following simply hierarchy, class: subClassOf. Each user-defined class is implicitly a subclass of owl: Things
- `Individuals`: individuals are actual entities that can be grouped into classes.
- `Properties`: properties are binary relations that assert general facts about classes and specific facts about individuals. Two types of properties are distinguished:
- `Object properties` assert relations between individuals of two classes, i.e. properties link two individuals together.
- `Datatype properties` assert relations between individuals of classes and XML data types.

In the rest of the chapter, ontology specific terms are shown in `courier new` (for e.g., `distinct class`, `inverse object relationship`, etc.) For more information on ontology modeling, the reader may please refer to Smith, et al. (2002).

6.2. DM^3 ONTOLOGY DESIGN CONSIDERATIONS

My first design consideration was to decide the starting point of my ontology design. Although there exists no 'best' design principles for ontology design, the ontology generation can be started from scratch, from existing ontologies, from a corpus of information sources, or a combination of the latter two approaches (Ding et al. 2002). A comprehensive of literature review reveals that current data mining ontologies are mainly from the knowledge engineer's perspective, where KDDM-domain specific knowledge such as DU, DM model building and deployment are captured. My search did not reveal any previously published ontologies that accurately describe the complexity of the data mining model selection and querying. I therefore

choose to build my DM3 ontology from scratch, while re-using some concepts from the well-developed data mining ontologies.

Our second design consideration is to determine a formalized methodology for ontology development. Several ontology development methodologies have been reported in the literature for ontology building (Fernández López et al. 1999; Noy et al. 2001), and ontology merging, reusing and learning (Maedche et al. 2001). However, the proposed methodology by Uschold & Gruniger (1996) does not cover certain requirements that are relevant to our design. Notably, ontology deployment was not included in the original proposed methodology. After the domain-specific knowledge is captured and explicitly presented in the ontology artifacts, and the ontologies are evaluated, the knowledge gained must be applied within an organization's semantic applications. Depending on the purpose and scope for which the ontology is designed, the deployment phase can span a wide range of functions such as generating namespaces for shared vocabulary (Peroni et al. 2012), semantic linking and annotation of metadata (Baker et al. 2013; Compton et al. 2012), and ontology-based querying and inferencing (Bouamrane et al. 2009). In addition, ontology integration is not relevant to the design since I am building the DM^3 ontology from scratch. Thus, in the light of maturity of ontology design and use, I incorporate additional design considerations to address the missing components in Uschold & Gruniger's (1996) methodology.

Figure 28 presents our adaptation of the ontology design methodology suggested by Uschold & Gruniger (1996). It consists of four iterative phases, namely purpose and scope identification, ontology building, ontology evaluation, and ontology deployment. The ontology building phase consists of two steps: ontology capturing and ontology coding. Newer ontology editors now enable the ontology developer to simultaneously capture and code the domain

conceptualizations. However, it is beneficial to separate the two conceptually. Throughout the four phases, the ontology design is guided by a set of design criteria and documentation. The inner arrows represent important dependencies between phases.

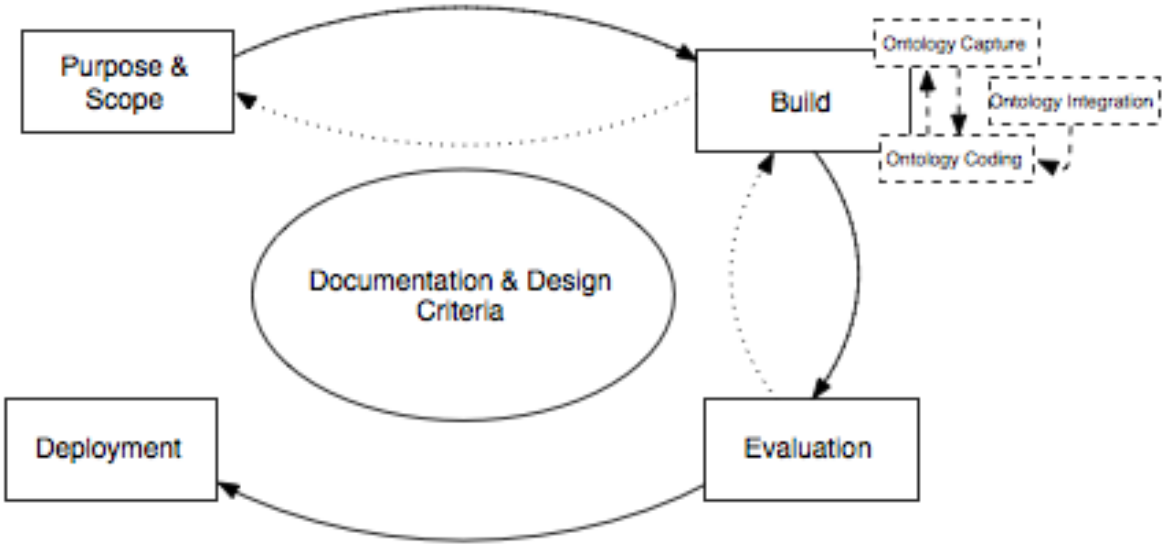

Figure 28: Ontology Design Methodology (Adapted from Uschold et a. 1996)

The skeletal methodology proposed by Uschold & Gruniger (1996) also does not cover all the tasks within each phase that are relevant to our process-based design needs. In my development framework, I incorporate these additional tasks based on the literature. Table 25 outlines tasks within each phase of my ontology design process.

Table 25: DM^3 Ontology Development Framework

Phase		Tasks
Purpose and scope		• Why the ontology is being built? • Who will use and maintain the ontology? • What are the characteristics of the users? • What is the domain that the ontology will cover? • What types of questions should the ontology provide answers?
Ontology Building	Capture	• Identify key concepts and relationships. • Produce precise definitions for concepts and relationships.

		• Define classes, properties, and relationships. • Create individuals.
	Coding	• Commit to the classes, properties, and relationships. • Choose a representation language. • Writing the code.
Ontology Evaluation		• Evaluation by the ontology developer(s) using ontology design criteria. • Evaluation of ontology usability and utility by end users.
Ontology Deployment		• Integration of ontology within the semantic applications.
Design Criteria		• Clarity • Coherence • Extensibility • Minimal ontological commitment • Minimal encoding bias

6.3. DM^3 ONTOLOGY

The DM^3 ontology (available at http://webprotege.vcu.edu:8080/webprotege) is organized conceptually in the following manner. The core concepts and relations are developed based on the popular CRISP-DM model, with an emphasis on model management capabilities. Objective properties and data properties are defined to provide reasoning capabilities to support self-service knowledge discovery. Logical constraints, such as domain and range on the object properties are added only when strictly required. Finally, the DL query rules are developed and tested against use cases. The DM^3 ontology was developed over a period of one year. To inform the work, I had formal and frequent discussions with data mining experts and members of academic community to ensure that the ontology is consistent with the KDDM domain. The ontology is OWL 2 DL compliant, allowing decidability and computational inference by

reasoner engines such as Pellet and RacerPro. In the next section, I describe each phase of the design process.

6.3.1. Purpose and Scope

Based on the tasks as shown in Table 25, I identified the intended users are business users within the organization, who have deep business knowledge and are able to provide business-driven analytical queries. Even though they are the drivers for analytics, they may lack sufficient technical knowledge and skills regarding the KDDM processes, techniques, and tools. I further identified two purposes that the DM^3 ontology should serve. First, it should provide an ontological representation of data mining goals based on the business user's descriptive statements. Second, it should serve to help the business user determine the suitability of the data mining models stored in the model repository based on a desired data mining goal. Queries can then be written to inference data mining models that best suite the analytical needs of the business user. Thus, to formally define the ontology modeling requirements, the DM^3 ontology should answer the following questions:

- What knowledge is required to describe a data mining goal?
- What are needed to support DM model selection based on the data mining goal?

To answer the first question, the specific goals of the business user need to be captured. Goal Question Metrics (GQM) approach (Van Solingen et al. 2002b) is a popular goal elicitation technique to characterize, categorize, decompose and structure goals and related measures. In GQM, the goal formulation requires information about five different components: (1) *purpose* (motivation behind the goal); (2) *focus* (quality attribute under study); (3) *object* **(**entity under study); (4) *viewpoint* (entity from whose perspective the goal is designed); and (5) *context* (scope

or environment) (Basili et al. 1994). Within the context of the DM^3 ontology, the *viewpoint* is the business users who use the DM model selection tool and the *context* is the DM model repository. As such, these two components remain the same for my DM goal conceptualization and hence, do not need to be formally represented. Based on the GQM approach, additional set of questions are identified as:

- What are required to describe a *purpose*?
- What are required to describe an *object*?
- What are required to describe a *focus*?

To answer the second question, DM models need to be described in the ontology. Inference capabilities also need to be built into the ontology so that, once the business user's data mining goal is captured, the relevant data mining models can be queried from the model repository.

6.3.2. Ontology Building

The ontology is developed using Protégé Knowledge Acquisition System (Protégé 2007), a free open source ontology editor and knowledge-base framework developed by the Stanford University School of Medicine. RacePro reasoner plug-in (RacerPro 2012) is used for inferencing. Because I am using an ontology editor, the ontology capture and coding are integrated, though they are conceptually separate. In this section, I provide the rationale behind the development of the main concepts and relationships in the DM^3 ontology, guided by the purpose and scope defined in section 6.3.1.

To capture the data mining goals in ontology, I first use ontological concepts to characterize and categorize the first three components. Purpose is represented by `DMPurpose`

class, focus is represented by `ModelSelectionCriteria` class and object is represented by `DMObject` class. The semantic meaning of the descriptive problem statement as inferred from `DMPurpose`, `DMObject` and `ModelSelectionCriteria` is characterized by the `DMGoal` class.

6.3.2.1. Data Mining Purpose (Class: DMPurpose)

DM purpose is related to DM problem types. The widely accepted classification of DM problem types falls in six categories - classification, estimation, prediction, association rules, clustering, and visualization (Berry et al. 2004). In the DM[3] design, `DMPurpose` is a class that consists of six distinct individuals (Figure 29).

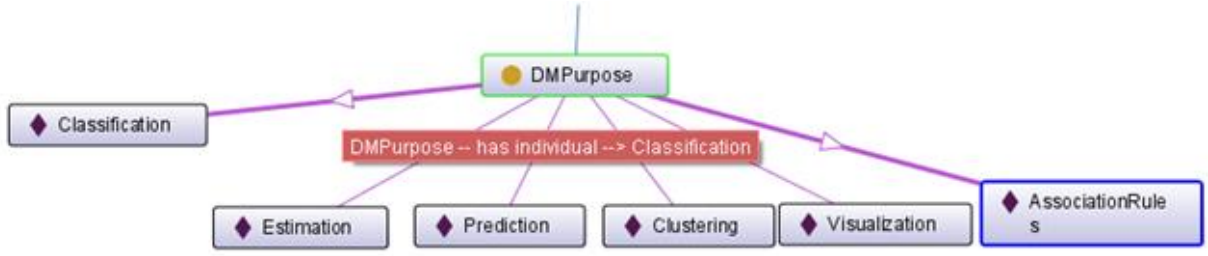

Figure 29: DMPurpose class and its individuals

Each individual in `DMPurpose` class corresponds to one of the six specific DM problem types, and is described using the data property:`hasDMPurposeOf` (Figure 30). For example, an individual AssociationRules is expressed in Protege-OWL as - AssociationRules `hasDMPurposeOf` "AssociationRules".

```
<!-- http://www.semanticweb.org/ontologies/2012/1/17/Ontology1329510839343.owl#AssociationRules -->

<NamedIndividual rdf:about="&Ontology1329510839343;AssociationRules">
    <rdf:type rdf:resource="&Ontology1329510839343;DMPurpose"/>
    <Ontology1329510839343:hasDMPurposeAs rdf:datatype="&rdfs;Literal">AR</Ontology1329510839343:hasDMPurposeAs>
    <rdfs:comment>to determine which things go together.</rdfs:comment>
</NamedIndividual>
```

Figure 30: OWL snippet of individual and data property in DMPurpose class

6.3.2.2. Data Mining Object (Class: DMObject)

DM object is the business process under investigation. It is similar to fact tables in a data warehousing environment which naturally correspond to business process measurement events (Kimball et al. 2011). Examples of data mining objects include customers, products, transactions, etc. One aspect of the BU phase in KDDM is to choose a specific data mining object for which the sample data are extracted. For example, when building predictive models for call centers, the measurement event is customer calls that each call center captures.

In the DM[3] ontology, the DM object concept is represented by `DMObject`, an upper level class with no subclasses. When the data mining models are stored in the repository, they are annotated with the corresponding data mining objects. Each instance of the data mining object is represented as an individual in the `DMObject` class using the data property `hasMiningObjectAs`. For example, if a set of DM models is built using student loan data, the model object is annotated as: `Student hasMiningObjectAs` "Student". If models are built for call center data, and calls may come from different regions, the model object can be annotated as `Call hasMiningObjectAs` "WestCalls", where WestCalls is the object dataset extracted from the calls from the West Region.

6.3.2.3. Data Mining Model Selection Criteria (Class: ModelSelectionCriteria)

Before model deployment, it is important to evaluate the model result to assess if the model meets the business objectives. Translated from data mining goals, data mining success criteria (DMSC) are quantified measures used to evaluate the model results. When the DM models are stored in the repository, the DMSC are annotated with the model. The DM^3 ontology incorporates a shortlist of DMSC (Sharma 2008, p.193) for different problem types, such as accuracy, simplicity, lift, sensitivity, specificity, etc. Each data mining technique has a set of relevant measures. However, not all software packages provide all relevant measures for a given data mining technique. For example, SAS decision tree does not provide ROC (Rate of Change) curve, but it is available in Rapid Miner. Table 26 shows some examples of DMSC and

Table **27** shows some of DMSC definitions for classification trees. The list is not exhaustive. Additional DMSC may be added when new models are created and annotated.

Table 26: Model Selection Criteria for Different Modeling Techniques

Modeling Techniques	Selection Criteria
Classification Tree	Accuracy, Profit and Loss, Lift, Simplicity, Stability
Regression Tree	Accuracy, Simplicity, Stability, Sensibility, Specificity
NN	Accuracy, Stability
Clustering	Variable Importance Vector, Number of Clusters
Association Rules	Confidence, Support

Table 27: Model Selection Criteria Definition for Classification Tree

Selection Criteria	Definition
Accuracy	Proportion correctly classified
Simplicity*	The length of the rule (* it is not relevant for non-exploratory model)
Lift	The change in concentration of a class when the model is used to select a group from the general population
Stability	Generalization of the modeling results over different population.

In the DM[3] ontology, ModelSelectionCriteria is an upper level class with no subclasses. Each selection criterion is modeled as an individual in the ModelSelectionCriteria class and described using data property hasMeasureAs. For example, an accuracy measure for classification is defined as - Accuracy hasMeasureAs "Accuracy".

6.3.2.4. Data Mining Goals (Class:DMGoal)

DMGoals translates the business user's data mining goals. It captures the semantic meaning of the descriptive problem statement as inferred from DMPurpose, DMObject and ModelSelectionCriteria. In the DM[3], DMGoal is an upper level concept that is inferred based on the formalism:

> DMGoal ≡ (hasDMPurpose some DMPurpose) and (hasMiningObject some DMObject) and (hasSelectionCriteria some ModelSelectionCriteria)
>
> The above assertion is equivalent to the following SWRL Rule:

```
DMObject(?o), DMPurpose(?s), ModelSelectionCriteria(?m),
Thing(?p), hasMiningObject(?p, ?0), hasDMPurpose(?p, ?s),
hasSelectionCriteria(?p, ?m) -> DMGoal(?g)
```

For example, Figure 31 shows that when an individual DMGoal1 is defined as (hasMingObject Student) and (hasSelectionCriteria Accuracy) and (hasDMPurpose Prediction), it is inferred as an individual in DMGoal class. In Figure 31, DMGoal is shown with an equivalence symbol in the icon, indicating that it is a defined class.

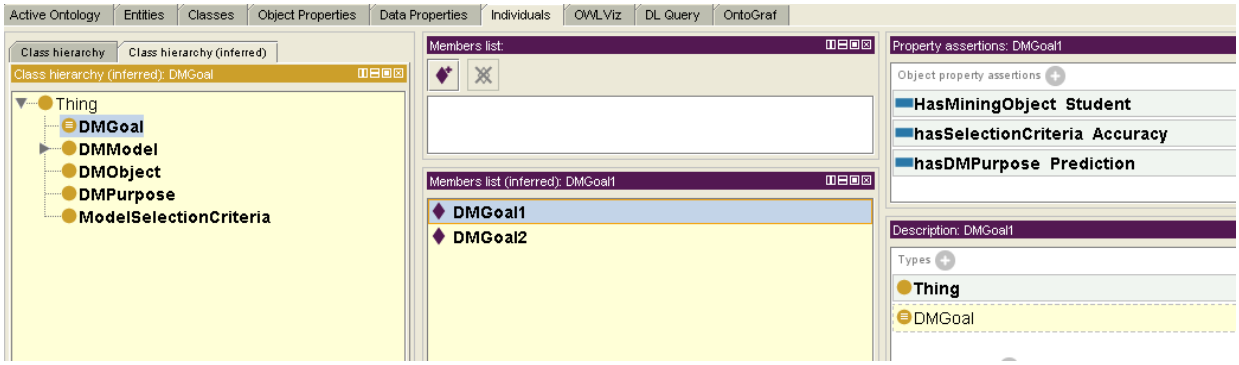

Figure 31: Inferred Individuals in DMGoal Class

6.3.2.5. Data Mining Models (Class: DMModel)

Each DM model is built using one or more data mining technique. In this study, I exclude models with more than one data mining techniques, such as ensemble models. Different DM techniques can be applied to different types of data mining problems. Table 28 summarized some common DM techniques that can be used for the various data mining problem types.

Table 28: DM Problem Type with Relevant DM Techniques

DM Problem Type	Relevant DM Technique(s)
Classification, Prediction with Discrete Target Variable	Logistic Regression, ClassificationTree, Neural Network (Smith, et al.), Association Rule induction (AR), Discriminant Analysis (DA), Case-Based Reasoning (CBR), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Naives Bayes
Estimation	General Regression, RegressionTree, NN, KNN
Prediction with Interval Target Variable	General Regression, RegressionTree, NN, KNN, Time Series (TS)
Clustering	Hierarchical Clustering, K-means Clustering, NN
Association Rules	AR

End users are rarely experts in DM techniques. They may not understand the DM technique that is suitable for the problem-at-hand. The DM^3 ontology can be used to bridge this semantic gap. In the DM^3 ontology, `DMModel` is modeled as an upper level class with different

data mining techniques as subclasses. Individuals in `DMModel` subclasses are the data mining models. Figure 32 shows the `DMModel` class, its subclasses and individuals. Individuals (and their properties) in the `DMModel` subclass are populated from the model repository.

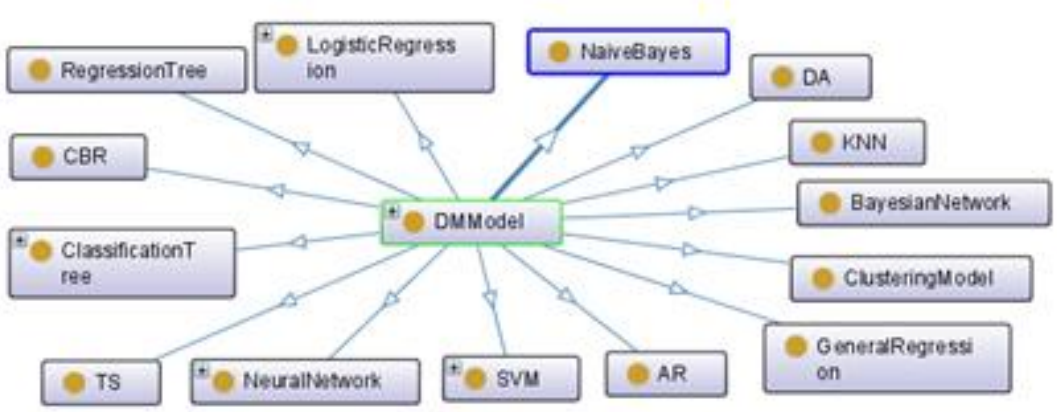

Figure 32: OntoGraph representation of DMModel class and its subclasses

Characteristics of an individual in the `DMModel` subclass are described using object properties and data properties. There are three types of object properties to describe relations of individuals in DM Model subclass to `DMPurpose`, `DMObject`, and `ModelSelectionCriteria` (Figure 33).

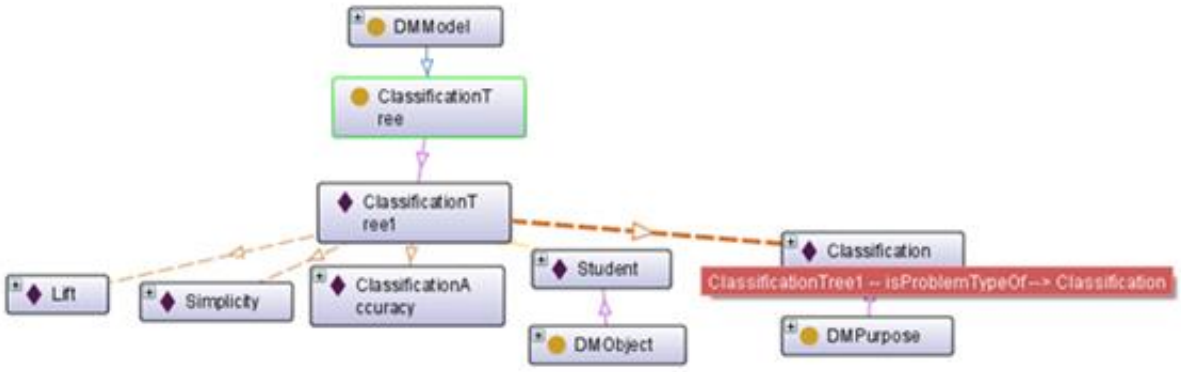

Figure 33: OntoGraph representation of individual object property

The object properties are as follows:

- The object property `hasProblemTypeAs` describes the relation between individuals in `DMModel` and `DMPurpose`. For example, a classification tree model is defined as: ClassificationTree1 `hasProblemTypeAs` Classification, where the ClassficationTree1 is an individual in `DMModel ClassificationTree` subclass and Classification is an individual in `DMPurpose` class. The open world assumption of ontology requires defining an inverse object property: `isModelTypeOf` ≡ `hasProblemTypeAs_`. This definition allows to automatically infer that an individual in `DMPurpose` Class `isModelTypeOf` is an individual in the `DMModel` subclass. Using aforementioned example, the following assertion is inferred: Classification `isProblemTypeOf` "ClassificationTree1".
- The object property `hasMiningObject` describes the relation between individuals in the `DMModel` subclass and `DMObject`. For example, ClassificationTree1 `hasMiningObject` Student, where the ClassficationTree1 is an individual in `DMModel ClassificationTree` subclass and Student is an individual in `DMObject`. Similarly, an inverse object property is defined as `isMiningObjectOf` ≡ `hasMiningObject-`.
- The object property `hasSelectionCriteria` describes the relation between individuals in the DMModel subclass and `ModelSelectionCriteria`. For example, ClassificationTree1 `hasSelectionCriteira` Lift, where the ClassficationTree1 is an individual in `DMModel ClassificationTree` subclass and Lift is an individual in `ModelSelectionCriteria`. As discussed above, each model may have a set of relevant DMSC that needs to be asserted as an object property for the model. For example, figure 5 shows the model ClassificationTree1 has three `hasSelectionCriteria` that are Lift, Simplicity,

and `ClassificationAccuracy`. In addition, an inverse object property is defined as `isSelectionCriteriaOf` ≡ `hasSelectionCriteira`$^{-}$.

Besides object properties, data properties are used to capture the values of the model's performance measures. For example, if the model ClassificationTree1 has an accuracy measure of 90%, it can be described as: ClassificationTree1 `hasAccuracyValue` "0.9"^^decimal, where `hasAccuracyValue` and decimal are the data properties. For each object property `hasSelectionCrteira` asserted in a model, a corresponding data property value is also asserted.

6.3.3. Ontology Evaluation

The third phase in the ontology design is evaluation. The effectiveness of the ontology relies on its quality, which in turn requires its formal evaluation. Three popular approaches identified in literature for evaluating ontologies are gold standard evaluation, criteria-based evaluation, and task-based evaluation (Gangemi et al. 2006; Yu et al. 2009). The gold standard evaluation compares an ontology to a benchmark ontology. This approach is mainly used for assessing the accuracy of automatically or semi-automatically generated ontologies (Lozano-Tello et al. 2004; Yu et al. 2009). Criteria-based evaluation focuses on the characteristics of the ontology in isolation from its application. Various evaluation criteria have been proposed in the literature (Gómez-Pérez 1996; Gruber 1995; Grüninger et al. 1995; Guarino et al. 2002). Depending on the criterion, the evaluation may be either automatically checked by utilizing an ontology tool, by applying quantified ontology measures, or via a manual inspection process. Task-based evaluation is used to judge whether the competency of the ontology satisfies the

needs of the application or task (Yu et al. 2009). It is the assessment of the utility of the ontology within the context of the application.

As illustrated in Table 25, I use two approaches to evaluate the DM^3 ontology: evaluation against a set of pre-determined ontology design criteria (criteria-based evaluation) and evaluation of the ontology utility (task-based evaluation). Gold standard evaluation is not applicable in my context, as there exists no other benchmark ontology that is lexically or conceptually similar to the DM^3 ontology. The criteria-based evaluation is demonstrated in this section, and the task-based evaluation is demonstrated in section 0 using an illustrative example. To be consistent with my choice of the ontology design methodology, I choose the five design criteria based on Ushold et al. (1996). These five design criteria are the same as those proposed by Gruber et al. (1995) and widely used in ontology evaluation.

The first design criterion, clarity, requires that all ontology conceptualizations should be defined objectively and unambiguously (Gruber 1995). There are no suggested quantifiable measures for clarity. It is left to the ontology engineer to determine if the concepts, individuals, and relationships have been objectively defined in the ontology (Yu et al. 2009). Each aspect of the DM^3 ontology was closely considered to ensure clarity of concepts definitions, instance properties, axioms, and relationships. For example, the WordNet search of Model returns 16 different definitions. It could mean different things such as a type of product, an exemplar, or someone that wears clothes to display fashion. To avoid ambiguity, a prefix "DM" is added to my concept of model, which is used to capture data mining model instances. In addition, a clear definition of data mining model is documented under the `DMModel` class annotation: a DM model is a set of rules or formulas extracted from the source data using data mining techniques to represent valid, non-trivial, previously unknown, interesting patterns, and to enable analytical

tasks against new data. Another example is the `ModelSelectionCriteria` class, where Accuracy, Lift, Loss, Profit, RSquare, etc., are well-accepted measures. In addition, detail annotations are given for each measure for clarity.

The second design criterion evaluated is coherence. Both formally defined axioms and informally defined concepts in an ontology should be logically coherent in natural language and documentation (Gruber 1995). Coherence requires that all inferences are consistent with the definitions and axioms. In the process of building DM^3 ontology, Racer Pro inferencing engine (reasoner) is used to check for inconsistencies between the asserted and inferred definitions. This ensures that the DM^3 ontology is logically coherent. Furthermore, all the concepts and properties are represented consistently, including the documentations. For example, each object property requires a corresponding inverse object property to "close" the open world assumption of the ontology design. All the properties and inverse properties are defined in a coherent style, using `hasPropertyAs` and `isPropertyOf`.

The third design criterion evaluated is extensibility. My design of the DM^3 ontology considers future extensions, where new subclasses can be defined using the existing vocabularies without the need for revising definitions. For example, an organization that adopts the DM^3 ontology might employ a data mining technique that has not been captured, such as regression spine. The organization can add a new subclass in the `DMModel` class called RegressionSpine and populate individuals as needed. Similarly, the organization can also add a specific DMSC by creating an individual in the `ModelSelectionCriteria` class using the data property `HasMeasureAs`.

Minimal encoding bias is the fourth design criterion evaluated. It requires the conceptualizations to be coded at the knowledge representation level and be independent from

the symbolic level implementation (Gruber 1995). All classes, individuals, relationships, and axioms in DM^3 ontology are encoded using OWL 2 DL, and can be shared across different representation systems. An organization can implement the DM3 ontology within their intranet applications (as described in the next section).

Minimal ontological commitment is the final design criterion evaluated. While ontological commitment allows agreed upon vocabularies, an ontology should "make as few claims as possible" (Gruber 1995) to support knowledge sharing. In keeping with the minimal ontological commitment requirement, my ontology design is guided by many discussions about possible extensions of DM, similar to the approach used in the SKOS ontology design by Baker et al. (2013). In cases where the inclusion of any conceptualization was questionable, I generally choose to exclude it. Logical constraints, such as domain and range on object properties are added only when strictly required. For example, the DM^3 ontology data properties have no explicit domain constraints. In addition, as already demonstrated in the evaluation of the extensibility, the DM^3 ontology allows the addition of sub-classes and individuals without changing the core ontology design. This further demonstrates that DM^3 addresses the minimal ontological commitment requirement.

6.3.4. Ontology Deployment

In this section, I demonstrate a scenario for the deployment and use of the DM^3 ontology. Figure 34 shows how DM^3 ontology can be integrated within an organization's intranet for self-service knowledge discovery. Security considerations related to the DM model repository and DM^3 ontology are not included in the scope of this deployment scenario.

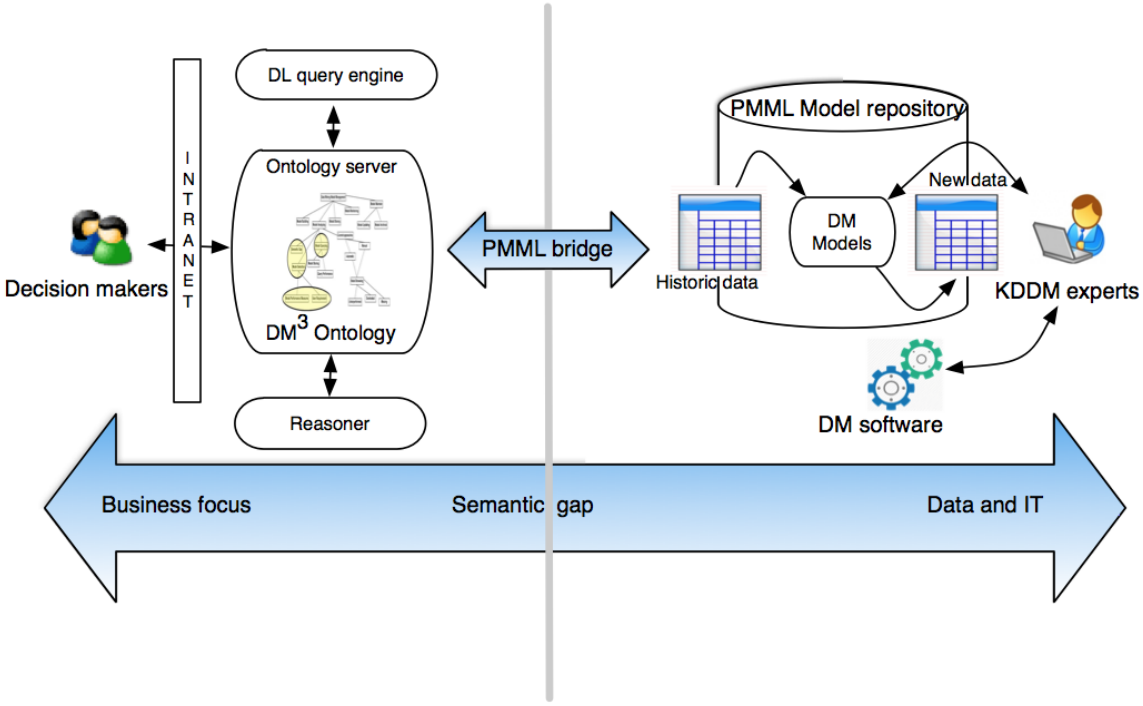

Figure 34: Ontology Deployment Architecture

Following the methodological approach such as CRISP-DM, the KDDM experts use DM software and techniques to mine historical data. They build DM models that represent interesting, actionable, and unexpected patterns in the existing data. The DM models are evaluated against the DM goals established in the BU phase of the KDDM process. Once the DM models satisfy the business needs, they are used to score the new data and stored in the organization's PMML compliant DM model repository.

The data and IT focus in the scenario I described above is the current KDDM approach. The KDDM experts have a thorough understanding of DM processes and algorithms, but lack the understanding of the decision context and the business objectives for which the DM models are applied. The business users who rely on the KDDM experts for the DM results are usually unable to verify their strengths and weaknesses within the business context. Deploying an ontology-based DMMM to support self-service knowledge discovery can bridge the semantic

gap between the business users and the KDDM experts.

In the ontology-based approach, the `DMModel` instances are populated from the DM models stored in the model repository. The repository uses an enhanced PMML template to include the mining purpose (functionName), DM Techniques (modelElement), mining object (miningObject), performance criteria (e.g., accuracy, lift) and performance measures. Figure 35 shows the PMML representation for a decision tree classification model with mining object name as Student. This model has three performance criteria: accuracy, lift (measured at 10 percentile, 20 percentile, and 30 percentile), and simplicity (measured by number of rules). The PMML bridges loads the DM models from the model repository to populate the `DMModel` instances in the DM^3 ontology.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<PMML xmlns="http://www.dmg.org/PMML-4_2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="4.2">
  <Header copyright="SAS Enterprise " description="generated by SA">...</Header>
  <DataDictionary numberOfFields="22">...</DataDictionary>
  <TreeModel modelElement="DecisionTree" functionName="classification" missingValueStrategy="lastPrediction" modelName="Tree" noTrueChildStrategy="returnLastPrediction"
    <MiningSchema>...</MiningSchema>
    <mingObject>
      <Dataset datasetName="StudentLoan"> </Dataset>
      <Object objectName="Student"></Object>
    </mingObject>
    <PerformanceMeasures>
      <Performance accuracy="0.941" lift_10="1.550" lift_20="1.550" lift_30="1.550" stable_till="55" ruleNumber="7">
      </Performance>
    </PerformanceMeasures>
    <Output>...</Output>
    <Targets>...</Targets>
    <Node recordCount="0" score="good    ">...</Node>
  </TreeModel>
</PMML>
```

Figure 35: PMML representation of a DM model

Based on emerging business needs, the decision maker uses the web interface to input the business query using a structured question-answer wizard. The inputs are translated into asserted facts (instances of `DMObject`, `DMPurpose`, and `ModelSelectionCriteria`) in the DM^3 ontology, which in turn, triggers the reasoner to infer a `DMGoal` individual. The DL query engine is then triggered to return all applicable DM models in the model repository that fit this specific DM goal. The query result is presented to the decision maker on the web interface. If there are no existing models that fulfill the DM goal, a new DM goal is created and sent to the

knowledge engineers as a new modeling requirement. The ontology server is periodically updated as new data mining models are built or outdated data mining models are retired.

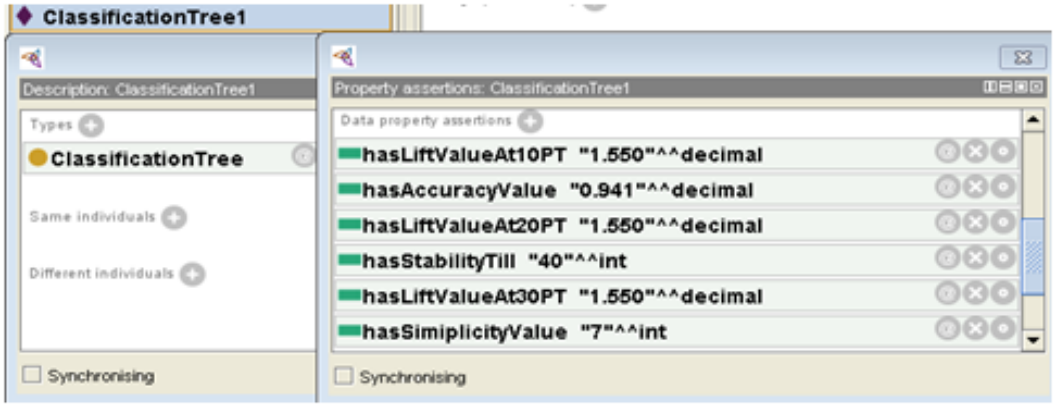

Figure 36: DMModel individual for a ClassificationTree

6.4. EXAMPLES AND USE OF DM^3 ONTOLOGY

In this section, I present an application of the DM^3 ontology to demonstrate its task-based utility. The functionality and applicability of the DM^3 ontology are illustrated in a use case for a private student loan company. The company has acquired data mining to help better understand student loans, improve lending program management, and reduce the incidence of loan defaults. A DM model repository stores candidate models, each registered in the standardized PMML industry format. The application is coded to load the candidate models in the model repository to the ontology. Each candidate model becomes an individual of the appropriate subclass in the DMModel class. For example, the description and property assertion for the decision tree model described in Figure 35 is asserted in the DM^3 ontology as shown in Figure 36 .

To demonstrate the utility of the DM^3 ontology, consider a business user from the loan company who is interested in selecting a model to predict students that may default. The business user should be able to discover and reuse the DM models that are already stored in the repository.

Data source mapping is performed between the new score dataset and the DM training dataset. This identifies a set of candidate models that are tagged with mining object as 'Student' and target event as 'Payment Due' = 'positive'. If the user is interested in selecting a model with accuracy greater than 90%, then the application triggers the reasoner engine to infer a DM goal (`DMGoal1`) based on the following ontological axiom:

`DMGoal1` ≡ (`hasDMPurpose` Prediction) and (`hasMiningObject` Student) and (`hasSelectionCriteria` Accuracy)

The application executes the following Descriptive Logic (DL) query:

`DMModel` and `hasMiningObject` value Student and `hasProblemTypeAs` value Prediction and `hasAccuracyValue` some decimal [>0.9]

The equivalent SWRL rule is:

`DMModel` (?m)^`hasMiningObject` (?m, Student)^`hasProblemType` (?m, Prediction) ^ `hasAccuracyValue` (?m, ?Accuracy) ^swrlb: greaterThan (?Accuracy, 0.9)-> sqwrl: select (?m, ?Accuracy)

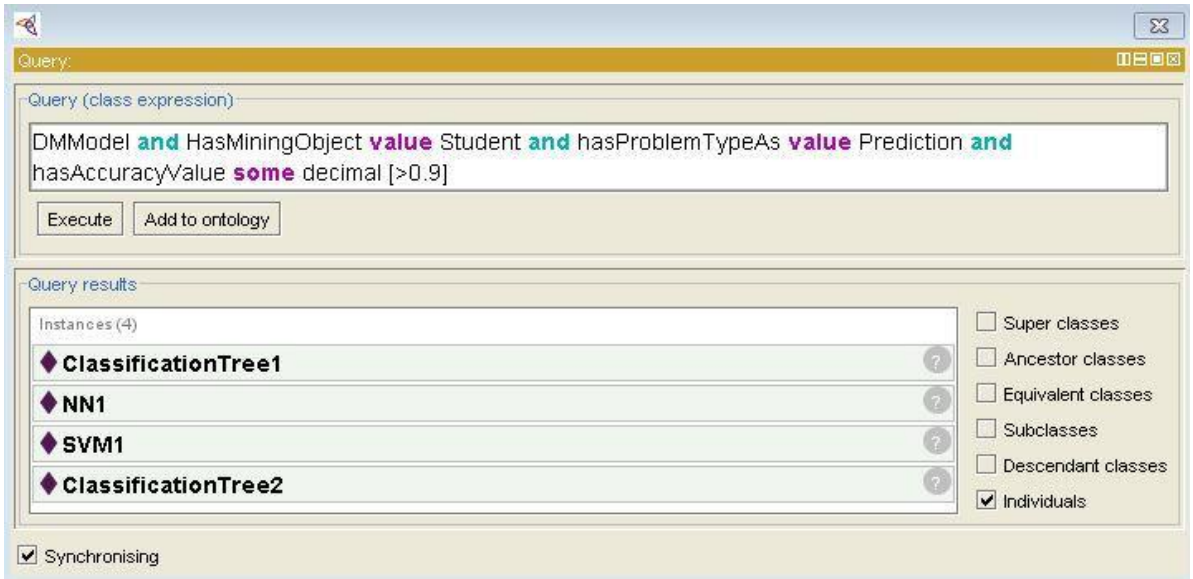

Figure 37: Inferred query result for DMGoal1

The DL query returns four models (Figure 37) that matches the stated DM goal. Furthermore, let's assume that the user wants to identify models with explanatory power, and would like to have a simple rule with rule sizes between 4 and 8. The reasoner would trigger a DM goal (DMGoal2) based on the following ontological axiom:

`DMGoal2` ≡ (`hasDMPurpose`) and (`hasMiningObject` Student) and

(`hasSelectionCriteria` Accuracy) and (`hasSelectionCriteria` Simplicity)

The application executes the following DL query:

`DMModel` and `HasMiningObject` value Student and `hasProblemTypeAs` value

Prediction and `hasSimiplicityValue` some int [>= 4, <= 8]

Of the four models retrieved from the previous DL query, only two are explanatory models: ClassificationTree1 and ClassificationTree2. Between these two models, ClassificationTree1 has the simplicity measure of 7, while the ClassificationTree2 has the simplicity measure of 9. Therefore, the above query returns only ClassificationTree1. Figure 38 shows the Ontograph for the ontology inference that facilitates the model selection results based on the analytical needs of the business user. Appendix A summarizes concepts and relationships to capture the goals of the business user and the description of DM models in the context of the loan company example.

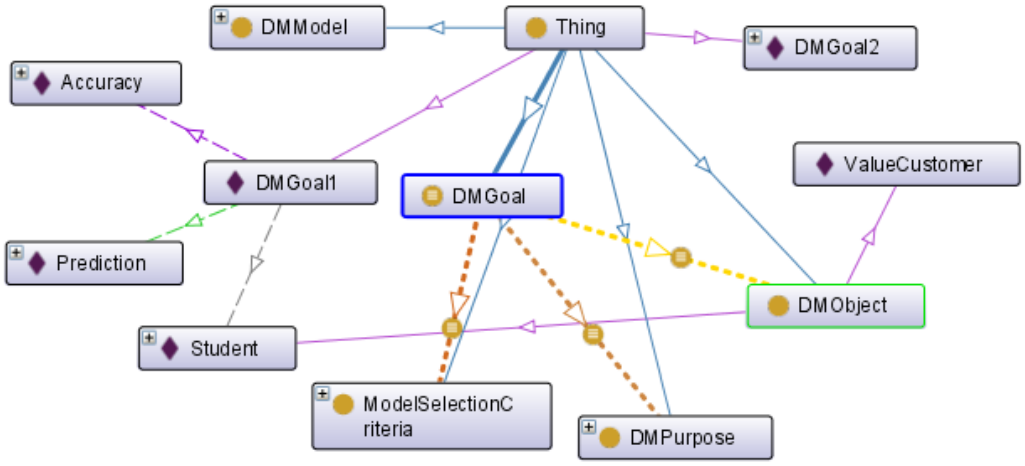

Figure 38: Ontograph Representation of DM^3 Ontology Inference

6.5. FUTURE WORK AND CONCLUSION

This chapter makes practical contributions in areas of KDDM and ontology design. The integration of DM model management in the KDDM process adds practical value for the sharing and reuse of knowledge products. Across many industries (e.g., financial, retail, healthcare, manufacturing and banking) (Davenport 2006), there is an increasing need for self-service knowledge discovery that enable business users to query and deploy analytics from an approved repository of data mining models. The study provides a baseline for developing an ontology-based self-service knowledge discovery model. It reduces the semantic gap between the business users who perform knowledge discovery tasks and the knowledge engineers who build models. In this chapter, I demonstrate that the use of DM^3 ontology enables business users to discover and select appropriate DM model(s) for desired decision outcomes. In the area of semantic web, the DM^3 ontology provides an extensible ontology that serves as a semantic model for data mining model selection and reuse. Both researchers and practitioners can take advantage of the Web Protégé implementation of the DM^3 ontology to build cumulative knowledge towards self-

service knowledge discovery.

This chapter also makes valuable theoretical contributions. In the KDDM domain, both academia and industry have identified a critical missing component after model deployment phase. Recognizing this limitation, few data mining software vendors now include DMMM capabilities such as model maintenance and sharing in their software packages. However, the vendor-based approaches only cover the modeling phase of the KDDM, while model selection based on the business user's decision criteria has not yet been addressed. To the best of my knowledge, this research is the first attempt to add model selection and reuse to the KDDM process. In the area of ontology, this research extends existing ontology design methodology (Uschold et al. 1996) to include deployment phase after ontology evaluation, an important step for the evaluation of ontology within the organization's semantic applications. In addition, a detail list of tasks within each phase is identified to guide the ontology design process. This research thus offers additional prescriptive knowledge towards ontology design that can benefit other ontology researchers.

From the process model perspective, the current knowledge discovery process models (e.g., CRISP-DM) need to be updated with embedded model management functionalities. The notion of PMML provides a means to formalize model storage, query, and possible selection. Another contribution of my research is the proposed extension to the current PMML schema. The extension enables a formal representation of the knowledge embedded in the BU phase of the KDDM process. It thus enables data mining model sharing not just across applications, but also across different organizational units.

Several improvements to the DM^3 ontology can be pursued. First, different data mining models manifest tradeoffs in DMSC measures. The DM^3 ontology does not consider these

tradeoffs in the model selection process. Multiple criteria decision analysis techniques such as weighted sum, analytic hierarchy process, or multi-attribute value theory can be leveraged to evaluate the criteria trade-offs and find a satisfactory solution. Second, the inferred data mining goals are currently not included in the proposed PMML schema extensions. The business queries are also stored in the ontology. They are both externalized organizational knowledge from the business users. Future research can extend the current design to integrate the business queries and inferred data mining Goals for reuse. Although model management systems (MMS) were not traditionally user-oriented tools (Bernstein et al. 2007), organizations may soon consider them integral components to their knowledge discovery initiatives. Future research can address these important decision support functions.

CHAPTER 7 FRAMEWORK FOR SOFTWARE SELECTION

This chapter presents a MCDA software selection framework. This framework can be extended to addressing the issues raised in section 1.4.1.3, where there is lack of support for KDDA tools and techniques selection. A preliminary idea related to this research was presented at the 47th Hawaii International Conference on System Sciences (2014). In this chapter, I build upon the previous research to include a Software Quality Evaluation Model and a DMS database in the framework. I further instantiate the framework into a web-based DSS for MCDA Software Selection that is driven by a backend database. Additional case scenarios from the real estate demonstrate how differences in DMS would result in different MCDA software recommendations.

An abundance of multiple criteria decision analysis (MCDA) methods have been proposed in the literature, most of which require substantial amounts of computation. The methods differ in the way the decision criteria are assessed and operationalized (De Montis et al. 2000). Many software packages have been developed to implement all or parts of these sophisticated methods and techniques. They cover various stages of the decision making process, from problem exploration and structuring to ascertaining the decision maker's preferences and identifying the most preferred compromise solution. The wide variety of methods and software posits a challenge for business users in choosing the MCDA solution that best suits their needs.

Selecting an inappropriate MCDA tool for a specific decision problem not only leads to wasted time and resources, but also the opportunity cost of responding to spurious results. Thus, it is highly desirable to provide decision support to select the appropriate MCDA methods and software for the decision problem at hand. Currently, there exists no systematic decision support to assist business users in this regard.

The issue is further complicated when the unique requirements of specific decision making situation (DMS) have to be taken into consideration. A DMS is a decision context that affects or is affected by the decision maker's (DM) decision making process. Factors such as preference articulation mode (Arbel 1989; Lee et al. 2011), alternatives assessment (Larichev et al. 2002), and conceptualization and modeling process (Corrente et al. 2012; Kiss et al. 1994; Tsoukiàs 1991) influence the DMS. Choosing the appropriate MCDA method(s) for a given DMS has remained a perpetual concern (Marler et al. 2004; Turskis et al. 2011). Although tentative guidelines (Guitouni et al. 1998; Ozernoy 1992) have been proposed for modeling the DMS, methodological approaches have not yet been developed or implemented to gather the DM's inputs and decision preferences.

In many cases, the difficulty to classify, evaluate and compare MCDA methods results in the decision maker choosing MCDA software based on the familiarity and affinity with the tool and the MCDA method(s) implemented within (Ozernoy 1992). The lack of decision support often leads to the DMS conditioned by the MCDA methodology employed in the tool. For example, for a specific DMS, the DM may prefer direct rating for preference elucidation. However, the tool selected by the DM may only provide pairwise analysis. The mismatch may compel the DM to adapt the DMS to the undesirable MCDA method employed in the tool. Ideally, the DM should first structure the DMS, and then based on the nature of DMS, select the

appropriate software that implements the correct MCDA technique. Currently, there is limited literature on addressing this issue.

This chapter seeks to address the paucity of research by developing a methodological approach to provide decision support for the selection of the MCDA methods and software. The study makes three important contributions. First, I develop a DMS modeling framework to structure the decision problem and the DM's decision preferences in the MCDA methods selection process. This framework extends the general guidelines for MCDA methods selection proposed by Guitouni and Martel (1998). Second, I identify a comprehensive set of MCDA software meta-data to enable the selection of the appropriate software. To the best of my knowledge, this is the first attempt to develop a MCDA software knowledge base that includes both MCDA methods and software evaluation quality. Third, utilizing the DMS modeling framework and the MCDA software knowledge base, I propose a decision support framework for MCDA software selection for a specific DMS, which can be implemented in a Decision Support System (DSS). I demonstrate the utility of the framework by implementing into a DSS. I then evaluate the framework using real world application examples from the real-estate domain.

The rest of this chapter is organized as follows. In the next section, I provide background on MCDA methods, process, and software, and decision support for software selection. I then describe the decision support framework for MCDA software selection. The framework is implemented into a decision support system (DSS) as described in Section 4, followed by the real work application examples from the real-estate domain. I conclude the paper by summarizing directions for future research.

7.1. SOFTWARE SELECTION REVISIT

In 2.6.3, a seven-stage generic software selection methodology is reviewed, which is summarized in Table 29. In section 2.4.2, a quality model for software evaluation is presented, which include a list of characteristics and sub-characteristics as listed in Table 4. A MCDA software quality model can be structured to include a taxonomy of MCDA software evaluation criteria and metrics for computing their values (Franch et al. 2003). The MCDA software quality model can be integrated into a decision support system to evaluate candidate software as described in the generic software evaluation methodology (step 4 in Table 29).

Table 29: Generic Software Selection Methodology

Step	Description
1	Initial investigation of the available software
2	Short listing of candidate packages
3	Eliminating software that do not have required features
4	Use an evaluation technique to evaluate the remaining software
5	Pilot testing the tool in an appropriate environment by obtaining trial copy
6	Negotiating a contract
7	Purchasing and implementing

7.2. MCDA SOFTWARE SELECTION FRAMEWORK

In this section, I propose a general framework (Figure 39) for MCDA software selection based on the decision maker's specific decision context. The framework only includes steps that can be automated or semi-automated in the aforementioned general software selection methodology (step 1 through 4 in Table 29). As mentioned in the section 2.2, the MCDA process starts with business understanding phase, where an initial assessment of the DMS is carried out. The framework assumes that the decision problem and relevant stakeholders have been

identified, and a preliminary set of alternatives and criteria has been created. They are utilized to formally structure the DMS, and select appropriate MCDA software packages that fit the DMS.

The framework consists of three stages: (1) develop a MCDA software knowledge base to provide an initial pool of MCDA software packages and their meta-data; (2) use a DMS modeling framework to structure a specific DMS and shortlist candidate software packages that implement the appropriate MCDA methods; (3) provide software recommendations by using the DM's preferred MCAP evaluation techniques. In the following section, I provide a detailed discussion of each stage.

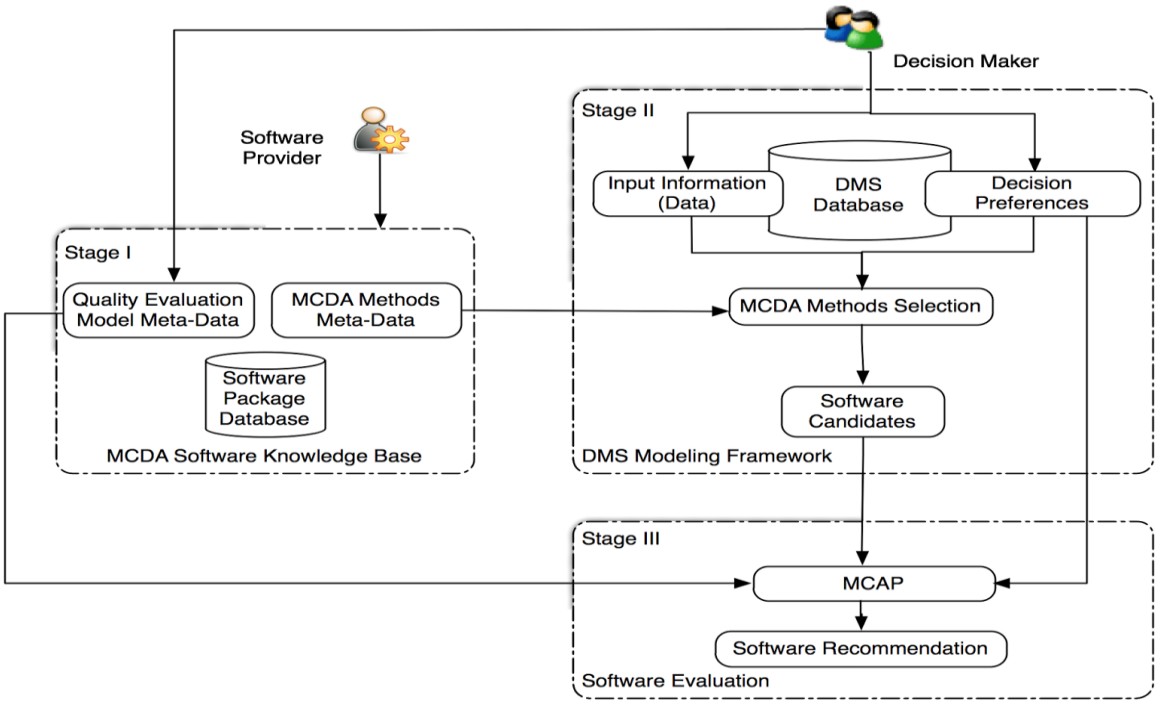

Figure 39: MCDA Software Selection Framework

7.2.1. Stage I: Building MCDA Software Knowledge Base

The first stage of software selection requires the creation of an initial pool of software packages (step 1 in Table 29). As mentioned in section 2.3, previous research has generated

cumulative knowledge on MCDA software. However, a comprehensive knowledge base of available MCDA software packages and their meta-data has yet to be constructed. The MCDA software survey by Weistroffer & Li (2014) provides an initial pool of 69 candidate MCDA software packages, which serves as a baseline for developing such a knowledge base. The MCDA software knowledge base constitutes of two parts: a software quality evaluation model and a meta-data model for the MCDA software. The MCDA software knowledge base constitutes of two parts: a MCDA software meta-data model and a software evaluation quality model.

A software quality evaluation model is required to compare the DMs' software quality requirement to the software quality attributes. There are different types of requirements in the software selection process, such as managerial, political, and quality requirements (Franch et al. 2003). While the managerial and political requirements are often subjective and unique to the individual organization, the quality requirements can be standardized. The standards related to software quality include the ISO and ISO/IEC families of 9126 and 14598, within which ISO/IEC 9126-1 specifically defines a quality model for software evaluation. I adopt the ISO/IEC 9261-1 standard as the software quality evaluation model. The ISO/IEC 9126-1 quality model defines six general software characteristics and 27 sub-characteristics. Each sub-characteristic can be further decomposed into measurable software attributes. The ISO/IEC 9261-1 standard is a generic model with high-level concepts that can be tailored to a specific software domain (e.g., MCDA software). The hierarchies of quality attributes are suitable for comparing user's software evaluation requirements with the software capabilities. The software evaluation quality model can be integrated into a DSS to evaluate the candidate software as described in the software evaluation methodology (step 4 in Table 29).

While the ISO/IEC 9216-1 quality model provides a taxonomy of software evaluation criteria, it does not describe how these criteria can be measured. The ISO/IEC 9216-2 defines external metrics to be used with the ISO/IEC 9126-1 quality model. The external metrics measure the external quality of the software, which suites the software evaluation objective. I therefore adopt relevant ISO/IEC 9126-2 for software product quality external metrics as the quantitative measures for the selection criteria. The software selection criteria in the quality evaluation model can be either vendor provided that are objectively assessed (e.g., platforms support, license cost, etc.) or DM selected that need to be subjectively scored (e.g., ease of use, adaptability, completeness, etc.). Both model selection criteria and their metrics are captured and stored in the knowledge base. Figure 40 shows a snippet of the xml schema for the quality evaluation model.

```
<?xml version="1.0"?>

<SelectionCriteria>
  <functional>
    <includedFunctions> risk and uncertainty </includedFunctions>
    <adatability> high level of customization </adatability>
    <openness> open to additional internal development </openness>
    <interoperability> capability to integrate with other tools and applications</interoperability>
    <security> user authentication  and auditing </security>
  </functional>
  <quality>
    <personalizability>
        <verticalSolution>  customerized version     </verticalSolution>
        <customizableField>  number of customizable field     </customizableField>
        <customizableReport>   no customerizable report   </customizableReport>
        <programmingLanguage>   personalized modules supported by programming language   </programmingLanguage>
    </personalizability>
    <portability>...</portability>
    <maintanability>...</maintanability>
    <usability>...</usability>
    <reliability>...</reliability>
  </quality>
</SelectionCriteria>
```

Figure 40 Snippet of Quality Evaluation Model Meta-data

The metadata schema for the MCDA software is stored in XML format. The XML schema aggregates the software selection criteria characteristics and sub-characteristics defined by the ISO/IEC 9126-1, and the meta-data for MCDA methods (e.g., information input,

elucidation mode, decision problematic, alternative aggregation evaluation, etc.) Figure 41 shows a snippet of MCDA methods meta-data stored in XML format. Currently, all required meta-data for the software identified by Weistroffer and Li (2014) are populated and stored in the knowledge base. The software providers may update knowledge base as needed. Since new MCDA methods and tools are frequently introduced and others may be outdated, it is impractical for a single entity to maintain and update the MCDA software knowledge base. In future, the knowledge base can be maintained as a collaborative effort from the MCDA community. A web-based environment is suitable for such a collaborative effort. The XML-based schema representation makes it easy to build a web-application for collecting new software meta-data or querying the knowledge base, as well as implementing the MCDA software selection framework on the web. To assist DMs with the quality evaluation, a wiki for MCDA software has also been created.

```
<MCDAMethod>
  <AHP>
    <Input>
      <inputDataScale>cardinal</inputDataScale>
      <criteria>true</criteria>
      <alternative>explicit</alternative>
    </Input>
    <PreferenceModeling>
      <elucidationMode>pairwise comparison</elucidationMode>
      <momentsOfElucidation>a priori</momentsOfElucidation>
      <preferenceStructure>preference
      </preferenceStructure>
      <preferenceStructure>incomparability</preferenceStructure>
      <altrnativeOrdering>totalPreorder</altrnativeOrdering>
      <decisionProblematic>choice</decisionProblematic>
      <decisionProblematic>ranking</decisionProblematic>
    </PreferenceModeling>
    <Aggregation>
      <alternativeEvaluation>partiallyCompensatory</alternativeEvaluation>
    </Aggregation>
  </AHP>
  <ELECTRE_I>
  </ELECTRE_I>
</MCDAMethod>
```

Figure 41: MCDA Method (AHP) Meta-data

7.2.2. Stage II: DMS Modeling

In order to shortlist a set of candidate software packages from the software pool, the first step is to explicitly model the DMS based on the DM's preferences and determine the appropriate MCDA methods based on these preferences. Hence, a systematic approach to elicit the preferences from the DM to determine appropriate MCDA methods is needed. Guitouni et al. (1998) presented a set of guidelines to help choose appropriate MCDA methods. Drawing upon these guidelines, stage II comprises of three steps (described below) to elicit the preferences from the DM, which can be mapped to the appropriate MCDA methods.

***Step II.a*: Input Data:** The input capability of a MCDA method are the information accepted (i.e., cardinal or nominal, ambiguous or unambiguous, uncertain or certain), the criteria (i.e., true criteria, quasi-criteria, pre-criteria, pseudo-criteria), and the alternatives (i.e. implicit or explicit). Different MCDA methods may handle different types of information. For example, if the DM defines the input data as ordinal, then the utility-based methods such as MAVT, SMART and AHP are not considered as optimal options. These methods are conceived to handle cardinal information, which require the conversion of original ordinal scales into abstract ones with an arbitrarily imposed range. When there is uncertainty associated with the input information, one should only choose a subset of MCDA methods that can handle uncertain information.

The decision preferences and the characteristics of the input information prescribe the criteria to be built, which can be classified as true criteria, pre-criteria, pseudo-criteria or quasi-criteria. For true criteria, the binary relation between two alternatives is either indifference (denoted as I) or strict preference (denoted as P). The pre-criteria convey the binary relation being either a strict preference or a weak preference (denoted as Q) with a preference threshold

for each criterion. The pseudo-criteria introduce a graduation of preferences that include both indifference threshold and preference threshold. The quasi-criteria are a particular case of pseudo-criteria without weak preferences. If the alternatives are expressed as implicit known, the DMS is a multi-criteria design problem and hence only MOO software packages can be included.

***Step II.b*: Decision Preferences:** This step elicits the DM's decision preferences towards the MCAP, which include preference modeling and aggregation evaluation. The elicitation process is driven by the guidelines and choices summarized in Appendix C. Regardless the output of step II.a, there are four different preference elucidation modes: (1) tradeoffs, (2) direct rating, (3) lotteries, and (4) pairwise comparison. The DM can select one or more of the preferred elucidation modes, which also feeds into Stage III. Furthermore, there are three types of the moments of elucidation: a priori, progressive, or a posterior. All existing MADA methods have a priori moments of elucidation. If the output of Step II.a indicates a multi-criteria design problem, four preference articulation categories will be presented (see previous discussion on MOO methods classification): (1) no preference, (2) a priori preference, (3) a posteriori preference, and (4) interactive method. If the output of Step *II.a* indicates a MADA problem, five preference structure will be presented: (1) indifference, (2) preference, (3) weak preference, (4) incomparability, and (5) outranking. The DM can select one or more of the preferred structure. In addition, there are four different decision problematics: (1) description, (2) choice, (3) sorting, and (4) ranking, where the DM can select one or more based on the DMS. Finally, the alternative aggregation evaluation includes three different types of compensation logic: compensatory, non-compensatory, and partially compensatory.

***Step II.c*: MCDA methods selection**: The output of Step II.a and Step II.b are mapped to the MCDA methods metadata in Stage I. The MOO software packages are characterized by the preference articulation modes. If the output of Step II.a refers to a multi-criteria design DMS, then the next step will be to map the preference Elucidation mode from Step II.b with the preference articulation mode in MOO software metadata, and retrieve the matching software. If the output of Step II.a refers to a MADA DMS, the captured DM's decision preference (e.g., the case example described in the next section) will be mapped to the MCDA-method metadata model in the knowledge base, and a shortlist of MCDA software that implement the applicable method(s) will be provided.

7.2.3. Stage III: Software Evaluation

Once the candidate software packages are generated from the knowledge base, the DM can review the vendor supplied MCDA software selection criteria and their measures. Software that do not have required objective features can be eliminated (step 3 in Table 29). The MCDA software knowledge base integrated in the DSS can then prompt the DM to determine the subset of criteria that require further evaluations. For example, some academic researchers may consider the software cost as a more influential factor, and less concerned with the efficiency. Once the set of software evaluation criteria is selected, the DM can then proceed to further investigate the software.

7.3. FRAMEWORK IMPLEMENTATION

In this section, I demonstrate how the proposed decision support framework for MCDA software selection can be implemented into a DSS, the MCDA Software Selection Center. A DSS typically comprises of three main components - (i) simple, yet powerful user interface, (ii)

modeling functions, and (iii) data management capabilities of internal databases and external data sources (Shim et al. 2002). The user interface of the MCDA Software Selection Center implementation involves two web fronts: one for the software provider and the other for the DM. Figure 42 shows the use case diagram of how the software provider and the DM interact with the Selection Center.

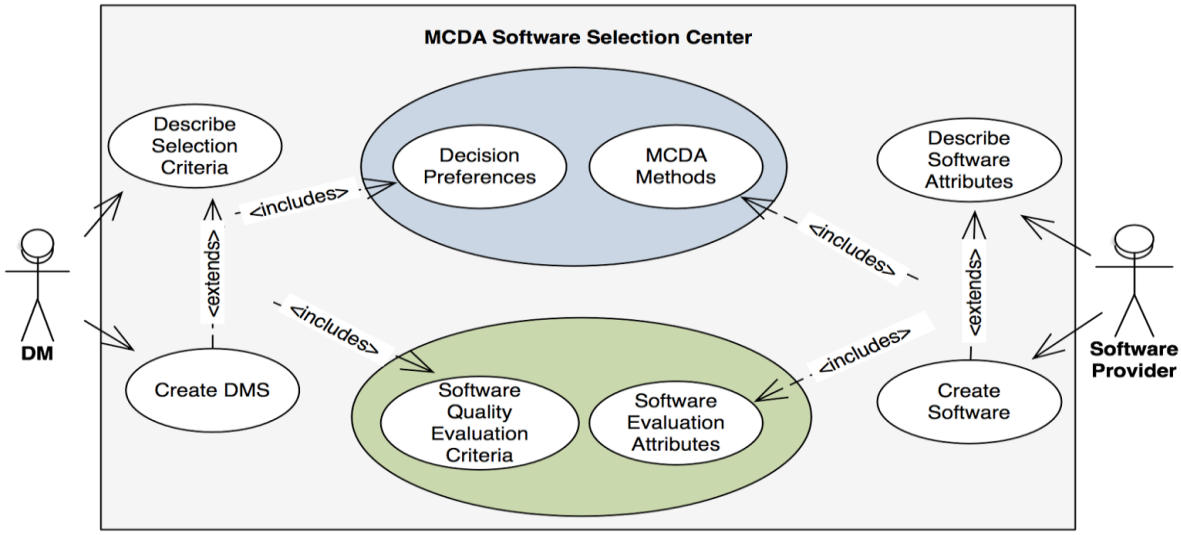

Figure 42: MCDA Software Selection Center Use Case Diagram

Using the first web interface, the software provider creates an instance of the MCDA software in the software package database (Figure 39). The software provider also supplies the software evaluation attributes and MCDA method meta-data as described in section 7.2.1 (Stage I: Building MCDA Software Knowledge Base). For example, the software provider for MakeItRational (www.makeitrational.com) may provide software evaluation attributes such as user authentication and auditing for security sub-characteristics of the functional characteristics, and indicate no customizable report for customizable report sub-characteristics of the personalizability characteristics (see Figure 40). The MakeItRational may also input AHP as its deployed MCDA method.

The second web interface enables the DM to create the DMS, which will be stored in the DMS database for future retrieval and reuse. The DM describes the MCDA software selection criteria that is specific to the DMS. The software selection criteria include two components: (1) software quality evaluation criteria as defined by the software quality evaluation model described in section 3.1, and (2) the DM's decision preferences as described in section 3.2. For example, a DM may desire *a high level of customization* for *adaptability* sub-characteristics of the *functional* characteristics, and specify p*ersonalized modules supported by programming language* for *programming language* sub-characteristics of the *personalizability* characteristics.

The modeling functions of the MCDA Software Selection Center are embedded in the web application, and are driven by the DMS decision modeling framework (Stage II) and the MCDA software knowledge base. Figure 43 shows the MCDA Software Selection Center logical data model, which is implemented in a relational database in the third normal form. In the interest of brevity, only tables relevant to the modeling functions are shown in Figure 43. Tables highlighted in blue enable the mapping between the DM's decision preferences and the MCDA method metadata (shown in the blue circle in Figure 42). Tables highlighted in green enable the mapping between the DM's software quality evaluation requirements and the MCDA software capabilities (shown in the green circle in Figure 42).

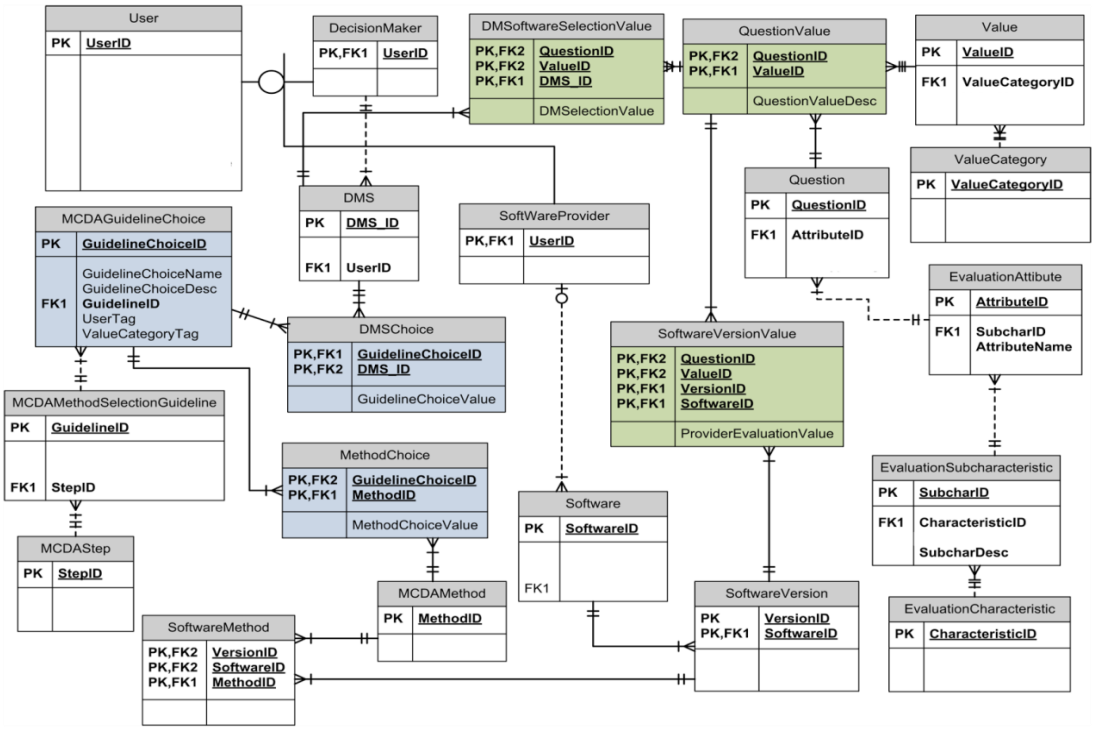

Figure 43: MCDA Software Selection Center Logic Data Model

Design evaluation is a crucial component of the design science research. The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods, such as analytics, case studies, experiments, or simulations (Hevner et al. 2004). Because the decision support framework for MCDA software selection is a novel artifact, its real world implementation may require many organizational changes. As a result, the observational evaluation (e.g., case study of the artifact) in a business environment is not feasible within the scope of this paper. Meanwhile, the proof-of-concept evaluation for novel artifacts is recognized as sufficient by design science researchers (Gregor et al. 2013). The implementation of a DSS provides a proof-of-concept to demonstrate that the utility of the proposed artifact can be realized by the means of information technology. Future research may provide in-depth evaluation in behavioral research projects after the utilities of the decision support framework have been validated.

7.4. APPLICATION EXAMPLES

Numerous websites such as zillow.com, realtor.com, etc. provide buyers and sellers with relevant information pertaining to houses such as listing price, description, payment estimates, pricing history, tax history, and neighborhoods. Finding houses or rentals are a classic case of MCDA problem, where the comprehensive assessment of houses is conducted against a set of selection criteria. Currently, no websites provide any form of MCDA support for buyers. A website that provides MCDA support would effectively attract more potential customers.

In this section, we present a case study of how two real-estate companies (Company One and Company Two) can utilize the framework to investigate MCDA solutions that can be embedded in their website. Using two scenarios, I demonstrate that the framework can help to formally assess and structure the DMS based on a preliminary set of criteria and alternatives. I further demonstrate that the framework captures the differences in the DMS in the BU phase, and consider these differences for the recommendation of the appropriate MCDA software.

7.4.1. Case Scenario 1

For Company One, the DM provides the following initial assessment of the DMS. The input data scales are cardinal, indicating that the input data (such as list price, number of bedrooms) need to either have a meaning between two degrees, or be converted into abstract values with an arbitrary range. The initial set of criteria has absolute discriminating power, and it is a MADA problem given that there are a finite number of candidate houses to compare. Furthermore, the DM selects pairwise comparison (i.e. to compare houses in pairs) as the preferred alternative comparison mode (preference elucidation mode). The preference is elucidated a priori. The decision preference between alternatives is modeled to be either strictly

preferred or indifference. That means for a given criterion (such as the house price), where a and b are respective price values for two alternatives (houses A and B): if a > b, then house A is strictly preferred; if a <b, then house B is strictly preferred; and if a=b, the house A and B are considered indifferent.

The decision problematic is modeled as either a ranking or choice problem, and a total preorder of the alternatives is preferred (i.e., there is no pair of items that is incomparable). The preferred alternative aggregation evaluation mode is partially compensatory, which implies that there is some compensation accepted between the different criteria. For example, if a house has a good open floor plan, it may compensate for the smaller square footage. However, the number of bedrooms is a criterion that may be not compensated at all. The out of the initial assessment of DMS is shown in Figure 44.

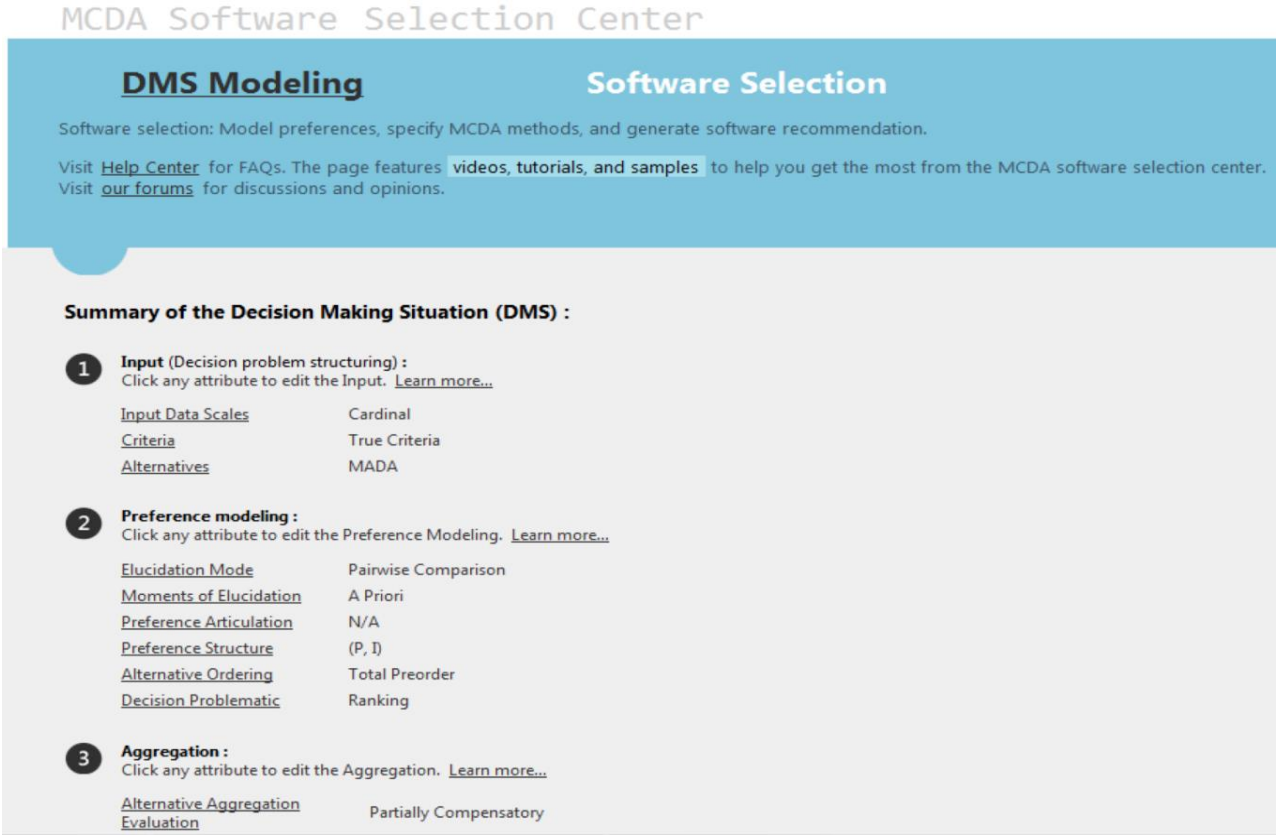

Figure 44: DMS Modeling Output of Company One

The mapping of the decision preferences with the MCDA software meta-data model identifies a relevant MCDA method, *AHP*, which is a systematic procedure to model MCDA problems in multilevel hierarchical structures and derive ratio scales from pairwise comparison of the hierarchical elements. The mapping also provides a list of candidate software packages that implements the *AHP* method, as shown in Figure 45.

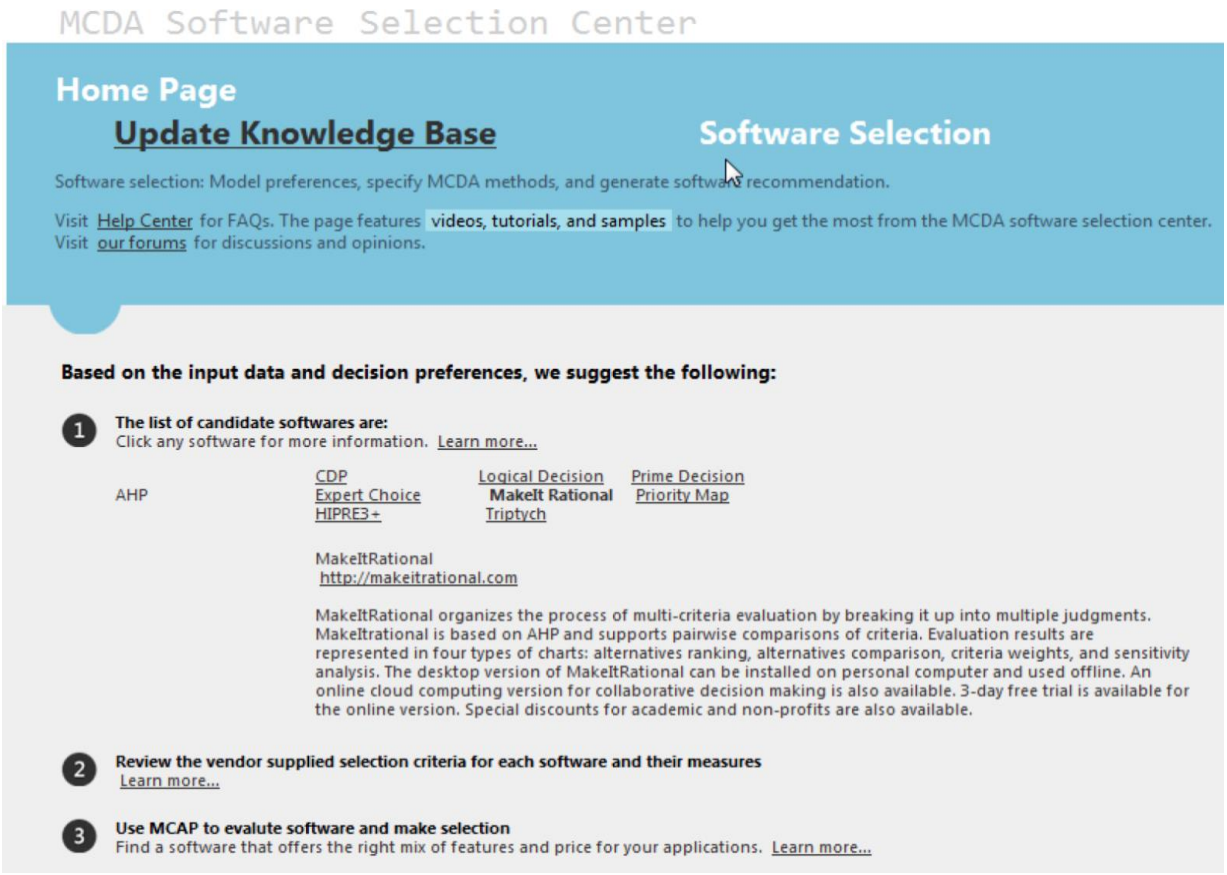

Figure 45: Candidate Software Packages for Company One

A short description of each software package is available and the DM can drill down to review vendor provided evaluation criteria. The DM can review the software packages and exclude those that do not have the required features. For example, the DM can browse through the candidate software description and recognize that *Priority Map* is a GIS-integrated application and *Triptych* is an excel-based application, both of which are not desired in the web

application for Company One. Thus, the two software packages can be excluded. For the software that requires further investigation, the DM can obtain evaluation copies and use the set of software-specific evaluation criteria to evaluate the software. It is out of the scope of this research to provide MCAP support in the MCDA software evaluation process. However, the evaluation technique used in the evaluation (step 4 in Table 29) should match the result of decision preference modeling in *step II*.

7.4.2. Case Scenario 2

Similar to Company One, the DM from Company Two indicates that the alternatives are finite, the moments of elucidation is *a priori*, the preferred alternative comparison mode is pairwise, and the alternative aggregation evaluation is partially compensatory. The DM from Company Two, however, provides the following different inputs after their initial assessment of the DMS. The Company Two wants to select a MCDA tool that can accept purely ordinal scales. For example, the difference between two neighborhood schools may not have an arbitrary range. The Company Two also wants to give their website users the choice of indifference, strict preference, and weak preference. In this case, when comparing the price values (a & b) of two alternatives (house A & house B, respectively), the user can set an indifference threshold q (i.e., q=\$5000) and a preference threshold p (i.e., p=\$7500), as illustrated in Figure 46.

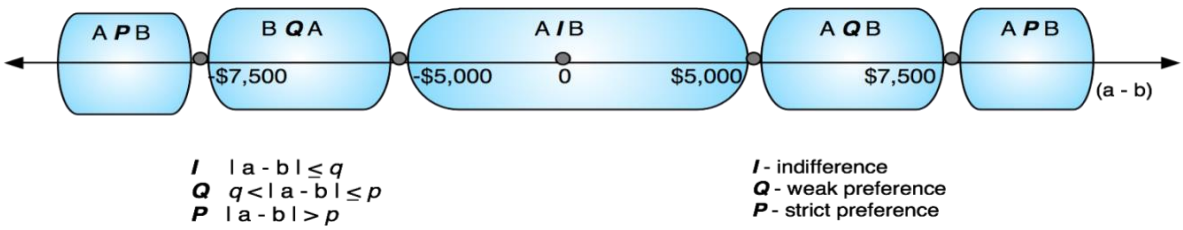

Figure 46: Example of Outranking Preference Structure

If the price difference between house A and B is within $5000 (i.e., |(a-b)|≤ $5000), the two houses are considered as indifferent. If the price difference is higher than $5000 but below $7500 (i.e., $5000<|(a-b)|≤$7500), there is a hesitation between house A and B (i.e., house A is either preferred or indifferent to house B). If the price difference is higher than $7500, (i.e., |(a-b)|>$7500), one house is strictly preferred over the other.

The Company Two also wants to select a MCDA method without the assumption that the decision preference is transitive (i.e., if A>B and B>C, then also A>C). In this case, the alternative ordering choice is set to be partial semi-order. In addition, the DM from Company Two chooses a ranking problematic, which means all the comparable houses are to be ranked from the best to the worst. The mapping of the decision preferences with the MCDA software meta-data model identifies four relevant MCDA methods – ELECTRE II, ELECTRE III, ELECTRE IV, and PROMETHEE. All four methods belong to a family of MCDA methods that is based on the principle of outranking (Roy 1973). While all outranking methods are motivated by decision efficiency for pairwise comparison of all options, the selected methods are applied to ranking problems. The MCDA Software Selection Center then provides a list of candidate software packages that implements the aforementioned outranking methods, which the DM can then precede to investigation further.

7.5. CONCLUSION

Increases in computing power have been at the heart of substantial growth in applications of MCDA. A variety of sophisticated MCDM methods proposed in the literature have been implemented on an ad hoc basis to solve a specific problem situation (Weistroffer et al. 2005). Though previous research have surveyed the state of art of the MCDA software (Poles et al.

2008; Seixedo et al. 2010; Weistroffer et al. 2005), guidelines for MCDA software selection have not yet been provided. In this chapter, I identify the challenges facing DMs in MCDA software selection, and demonstrate the need for decision support in the selection process that takes the specific nature of the DMS into consideration. I propose a decision support framework for MCDA software selection that is implemented into a DSS. To best of my knowledge, this research is the first methodological approach to provide decision support for selecting MCDA software based on a specific DMS.

This chapter also provides two additional design artifacts. First, I develop a comprehensive set of MCDA software meta-data for software quality evaluation and MCDA methods that can be integrated into a MCDA software knowledge base. The software quality evaluation criteria are based on the established ISO/IEC 9126 standards. This enables the standardization of the process of maintaining the MCDA software knowledge base. As new MCDA tools and software are introduced, a collaborative effort from the academic and commercial software providers in the MCDA community is needed to keep the MCDA software knowledge based up to date. The second artifact is a DMS modeling framework that can be used to structure the DMS based on the decision problem and the DM's decision preferences. The DMS modeling framework extends the general guidelines for MCDA methods selection proposed by Guitouni and Martel (Guitouni et al. 1998). In this chapter, I utilize the two artifacts to generate a candidate set of software recommendations for the DM. The research thus breaks new ground by addressing the challenges of modeling the decision maker's decision preferences and the preference input prior to choosing the appropriate MCDA method and software for a given DMS, a concern that been recognized by researchers in the past (Marler et al. 2004; Turskis et al. 2011).

This research provides the basis for several future research directions. First, the framework assumes a single DM for simplicity, while the DMS may involve multiple DMs with different decision preference structures and conflict objectives. Thus, additional considerations are required in the group decision making (GDM) to aggregate different individual preferences into group judgment (Limayem et al. 2000). To provide group decision support capabilities in the framework, a process of creating a group satisfactory DMS based on multiple DMs' input and decision preferences is needed. There are three commonly used preference relations to model multiple DM preferences in GDM: multiplicative preference relations (Ma et al. 2011), fuzzy preference relations (Hatami-Marbini et al. 2011; Orlovsky 1978), and linguistic preference relations (Herrera et al. 1995; Pang et al. 2012). There also exist MCDA software packages that provide group decision support capabilities (Weistroffer et al. 2005). Future research can utilize these approaches to include group decision support functions in the framework.

Second, the current DMS modeling framework primarily provides guidelines and choices (Appendix A) that are related to the characteristics of MADA methods, while MOO methods are characterized by the preference articulation categories (Hwang et al. 1979). Hundreds of multi-objective optimization (MOO) methods are proposed in the literature to solve specific or more generic MOO problems, and there is an array of software that implements these methods. Different MOO approaches are designed to address different types of MOO problems such as linear, non-linear, continuous, discrete, mixed, fuzzy, etc. Each MOO technique design different strategies to search for the Pareto Optimal solution that satisfies the DM's subjective decision preferences. The search strategy is mostly mathematical programming based, and the DMs may not necessarily be familiar with the mathematical formulation of the DMS. How to map the

DM's decision preferences and DMS with the search strategies of the MOO methods is worth consideration.

The decision support framework presented in this research may be extended to other software domains. For instance, KDDA process involves hundreds of analytics tools, each implementing different set of modeling techniques. Business understanding is considered as the most important phase of any KDDA project (Shearer 2000). It focuses on determining business objectives and business success criteria, and converting them into KDDA objectives and success criteria as part of the initial project plan. Similar to the issues in the MCDA software selection, the initial project plan includes the selection of analytical tools and techniques. The modeling techniques need to suite the DMS, which may be further constrained by the business requirements (e.g., politics, money, time, knowledge inventory, etc.). Future research will address extending the current decision support framework to support the tools and techniques selection task in the BU phase of the KDDA process.

CHAPTER 8 CONCLUSION

Business application of KDDA has undergone profound changes in the recent years. Driven by global IT connectivity, advancements in business database solutions, and ubiquity of the World Wide Web, the speed of data generation and creation have increased exponentially (Chen et al. 2012). This results in vast amount of data with high velocity from a variety of sources. This dissertation aims to address key deficiencies in existing KDDM process models by designing several new artifacts for the KDDA process.

The first artifact designed in this dissertation is the snail shell KDDA process model (Chapter 4) that addresses many limitations in existing KDDM process models. In chapter 4, I also highlight the iterative nature of the proposed KDDA process model, and summarize the differences between the snail shell KDDA process model and existing KDDM process models. The KDDA process model is evaluated using informed argument and case scenarios. Two cases illustrate how KDDA process model guides the real world KDDA projects. Future research will explore the evaluation of the effectiveness and efficiencies of the snail shell KDDA process model through controlled experiment. Group of analytical master students will be instructed to use the CRISP-DM or the snail shell KDDA process model to guide their projects. Static quality of the snail shell KDDA process model will be surveyed using a set of survey questions.

Statistical test can be used to compare the performance of the CRISP-DM and the snail shell KDDA process model.

The second artifact designed in this dissertation is a theory building methodology based on qualitative data (chapter 5). The richness captured in qualitative data is a key strength of the qualitative approach to theory building. However, given the nature of qualitative data, it is typically not apparent to qualitative researchers as to how quantitative techniques could be used to facilitate the identification of strong relationships between concepts that are embedded in the data. This often leads to the formulation of theoretical propositions that are rich in detail, yet lacking in simplicity. In addition, the researcher faces the daunting task of developing persuasive arguments to justify findings. The proposed methodology provides a systematic procedure towards qualitative theory building using quantitative data analysis techniques (e.g., AR mining) to facilitate developing propositions. Specifically, I demonstrate how researchers can take advantage of quantitative data analysis techniques such as association rules (AR) mining to identify strong concept relationships from qualitative data. The proposed methodology is illustrated using a case study in the public health domain. Future research will explore three-item rules in the data analysis and propose a competing theoretical model to for deployment.

The third artifact designed in this dissertation is the DM^3 ontology to provide a user-centric semantic approach for DM model selection and reuse (chapter 6). Despite the fact that analytical models are valuable organizational assets with high development costs, improving model sharing and reuse remains a pressing issue. The semantic gap between knowledge engineers who develop analytical models and business users who lack the technical knowledge to perform knowledge discovery tasks further accelerates the problem in today's real-time analytic environment. The DM^3 ontology for DMMM helps translate the business requirements

into model selection criteria and measurements (the ontology is available at http://128.172.188.35:8080/webprotege). Ontology-based deployment architecture is presented to provide a baseline approach for self-service knowledge discovery. The DM^3 ontology is evaluated using criteria-based and task-based approaches. I also propose extensions to current PMML schema to formally represent the BU phase of the KDDA. The DM^3 ontology makes valuable contribution to both practitioners and researchers in the areas of KDDM and ontology development. Currently DM^3 ontology is evaluated using illustration examples. The next step is to evaluate DM^3 ontology using analytical evaluation approach as outlined in section 0. A structured survey instrument (Appendix D) is designed and approved by VCU Internal Review Board (IRB). Data will be collected based on procedures described in section 0.

The fourth artifact designed in this dissertation is a MCDA software selection framework (chapter 7). With the gaining popularity of MCDA among researchers and practitioners, a variety of software packages that implement sophisticated MCDA methods and techniques is now available. However, there exists no systematic approach to assist the DM in MCDA software selection. Furthermore, the decision problem structuring and the DM's preference modeling are not currently considered in this selection process. I propose a decision support framework to enable DMs choose relevant MCDA software based on a specific decision making situation (DMS). A DMS modeling framework is developed to structure the DMS based on the DM's decision preferences and the decision problem. A comprehensive set of MCDA software meta-data for software quality evaluation and MCDA methods is developed and integrated into a MCDA software knowledge base. The framework is evaluated by implementing into a decision support system and using application examples from the real-estate domain. Currently the proposed framework is limited to MCDA software selection. Future research will extend the

framework to provide decision support for analytical tools and techniques selection. The evaluation of the framework will also be enhanced using analytical evaluation approach as outlined in section 0. A structured survey instrument (Appendix E) is designed and approved by VCU Internal Review Board (IRB). Data will be collected based on procedures described in section 0.

This dissertation also provides many potential future research opportunities, described as follows:

- Propose a formalized analytical problem formulation approach by utilizing the business analytics body of work (BABOK) and strategic problem formulation literature.
- Develop an analytical capability maturity model that can provide an objective means of evaluating organizational analytical capabilities and guide the organizations to improve their analytical capabilities in the KDDA process.
- Propose an agile framework for KDDA projects and compare with the traditional SDLC analytical framework.
- Design an artifact to facilitate data quality management in the KDDA projects by incorporating the concept of quality factory service to measure data qualities.
- Design an integrated KDDA knowledge repository that provides background knowledge for carrying out the KDDA process.

REFERENCES

Aamodt, A., and Plaza, E. 1994. "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI Communications* (7:1), pp 39-59.

Adhikari, A., Lebow, M. I., and Zhang, H. 2004. "Firm characteristics and selection of international accounting software," *Journal of International Accounting, Auditing and Taxation* (13:1), pp 53-69.

Aggarwal, C. C., and Yu, P. S. 2001. "Mining associations with the collective strength approach," *Knowledge and Data Engineering, IEEE Transactions on* (13:6), pp 863-873.

Agrawal, R., Imieli, T., and Swami, A. 1993. "Mining association rules between sets of items in large databases," *SIGMOD Rec.* (22:2), pp 207-216.

Agrawal, R., and Srikant, R. Year. "Fast algorithms for mining association rules," 1994 International Conference on Very Large Databases, Institute of Electrical & Electronics Engineers (IEEE), USA, 1994, pp. 487-499.

Agre, G. P. 1982. "The concept of problem," *Educational Studies* (13:2), pp 121-142.

Ahmed, K. M., El-Makky, N. M., and Taha, Y. 2000. "A note on "beyond market baskets: generalizing association rules to correlations"," *SIGKDD Explor. Newsl.* (1:2), pp 46-48.

Alavi, M., and Leidner, D. E. 2001. "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues," *MIS Quarterly* (25:1), pp 107-136.

Anand, S. S., and Büchner, A. G. 1998. *Decision support using data mining*, (Financial Times Management.

Arbel, A. 1989. "Approximate articulation of preference and priority derivation," *European Journal of Operational Research* (43:3), pp 317-326.

Ari, I., Li, J., Kozlov, A., and Dekhil, M. 2008. "Data mining model management to support real-time business intelligence in service-oriented architectures."

Baker, T., Bechhofer, S., Isaac, A., Miles, A., Schreiber, G., and Summers, E. 2013. "Key choices in the design of Simple Knowledge Organization System (SKOS)," *Web Semantics: Science, Services and Agents on the World Wide Web* (20), pp 35-49.

Ballou, D. P., and Pazer, H. L. 1985. "Modeling Data and Process Quality in Multi-Input, Multi-Output Information Systems," *Management Science* (31:2) Feb., pp 150-162.

Banks, J. 1991. "Selecting simulation software," in *Proceedings of the 23rd conference on Winter simulation*, IEEE Computer Society: Phoenix, Arizona, pp. 15-20.

Basili, V. R., Caldiera, G., and Rombach, H. D. 1994. "Goal question metrics paradigm," in *Encyclopedia of software engineering*, pp. 528-532.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., and Jeffries, R. 2001. "Manifesto for agile software development,").

Bernstein, A., Provost, F., and Hill, S. 2005. "Toward intelligent assistance for a data mining process: An ontology-based approach for cost-sensitive classification," *Knowledge and Data Engineering, IEEE Transactions on* (17:4), pp 503-518.

Bernstein, P. A., and Melnik, S. Year. "Model management 2.0: manipulating richer mappings," Proceedings of the 2007 ACM SIGMOD international conference on Management of data, ACM2007, pp. 1-12.

Berry, M. J., and Linoff, G. S. 2004. *Data mining techniques: for marketing, sales, and customer relationship management*, (Wiley Computer Publishing.

Bertolazzi, P., and Scannapieco, M. Year. "Introducing data quality in a cooperative context," 6th International Conference on Information Quality (IQ'01), Boston, MA, 2001.

Blanc, L., and Jelassi, M. 1989. "DSS software selection: a multiple criteria decision methodology," *Information & Management* (17:1), pp 49-65.

Blanning, R. W. Year. "Data management and model management: a relational synthesis," the 20th annual Southeast regional conference, ACM1982, pp. 139-147.

Boland, R. 2002. "Design in the punctuation of management action," in *Managing as designing: Creating a vocabulary for management education and research*, Frontiers of Management Workshop, Weatherhead School of Management, pp. 106-112.

Bouamrane, M.-M., Rector, A., and Hurrell, M. 2009. "A hybrid architecture for a preoperative decision support system using a rule engine and a reasoner on a clinical ontology," in *Web Reasoning and Rule Systems*, Springer, pp. 242-253.

Bovee, M., Srivastava, R. P., and Mak, B. 2003. "A conceptual framework and belief‐function approach to assessing overall information quality," *International journal of intelligent systems* (18:1), pp 51-74.

Brachman, R. J., and Levesque, H. J. 1985. *Readings in knowledge representation*, (Morgan Kaufmann Publishers: Burlington, MA.

Brennan, K. 2009. *A Guide to the Business Analysis Body of Knowledger*, (Iiba.

Brin, S., Motwani, R., and Silverstein, C. 1997. "Beyond market baskets: generalizing association rules to correlations," in *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, ACM: Tucson, Arizona, USA, pp. 265-276.

Buckshaw, D. 2010. "Decision Analysis Software: 10th biennial survey," *OS/MS Today* (37:5).

Burrell, G., and Morgan, G. 1979. *Sociological paradigms and organisational analysis*, (London: Heinemann.

Cabena, P., Hadjinian, P., Stadler, R., Verhees, J., and Zanasi, A. 1998. *Discovering data mining: from concept to implementation*, (Prentice-Hall, Inc.

Cannataro, M., and Comito, C. 2003. "A data mining ontology for grid programming," *Proceedings of (SemPGrid2003)*), pp 113-134.

Chandler, N., Hostmann, B., Rayner, N., and Herschel, G. 2011. "Gartner's Business Analytics Framework," Gartner Group, Stamford, CT

Chang, C.-L. 2007. "A study of applying data mining to early intervention for developmentally-delayed children," *Expert Systems with Applications* (33:2), pp 407-412.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., and Wirth, R. 2000. "CRISP-DM 1.0," in *CRISP-DM Consortium*.

Charest, M., Delisle, S., Cervantes, O., and Shen, Y. Year. "Invited Paper: Intelligent Data Mining Assistance via CBR and Ontologies," 17th International Workshop on Database and Expert Systems Applications, IEEE, krakow, Poland, 2006, pp. 593-597.

Chen, H., Chiang, R. H. L., and Storey, V. C. 2012. "Business intelligence and analytics: From big data to big impact," *MIS Quarterly* (36:4), pp 1165-1188.

Chen, Y.-L., and Weng, C.-H. 2009. "Mining fuzzy association rules from questionnaire data," *Knowledge-Based Systems* (22:1), pp 46-56.

Chen, Y. J. 2010. "Development of a method for ontology-based empirical knowledge representation and reasoning," *Decision Support Systems* (50:1), pp 1-20.

Choinski, M., and Chudziak, J. A. Year. "Ontological learning assistant for knowledge discovery and data mining," International Multiconference on Computer Science and Information Technology (IMCSIT'09), IEEE, Mrągowo, Poland, 2009, pp. 147-155.

Cios, K. J., and Kurgan, L. A. 2005. "Trends in data mining and knowledge discovery," in *Advanced Techniques in Knowledge Discovery and Data Mining*, L. C. Pal and N. Jain (eds.), Springer-Verlag London, pp. 1-26.

Clifton, C., and Thuraisingham, B. 2001. "Emerging standards for data mining," *Computer Standards & Interfaces* (23:3), pp 187-193.

Cochran, J. K., and Chen, H.-N. 2005. "Fuzzy multi-criteria selection of object-oriented simulation software for production system analysis," *Computers & Operations Research* (32:1), pp 153-168.

Collier, K., Carey, B., Sautter, D., and Marjaniemi, C. Year. "A methodology for evaluating and selecting data mining software," System Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on, IEEE1999, p. 11 pp.

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

Corrente, S., Greco, S., and Słowiński, R. 2012. "Multiple criteria hierarchy process in robust ordinal regression," *Decision Support Systems* (53:3), pp 660-674.

Davenport, T. H. 2006. "Competing on analytics," *Harvard Business Review* (84:1), p 98.

De Montis, A., De Toro, P., Droste-Franke, B., Omann, I., and Stagl, S. 2000. "Criteria for quality assessment of MCDA methods," *Transitions Towards a Sustainable Europe*), pp 1-30.

Delone, W. R., and McLean, E. R. 1992. "Information systems success: the quest for the dependent variable," *Information Systems Research* (3:1), pp 60-95.

Dempster, M., and Ireland, A. 1991. "Object-oriented model integration in a financial decision support system," *Decision Support Systems* (7:4), pp 329-340.

Denzin, N., and Lincoln, Y. 1994. *Handbook of Qualitative Research*, (Sage: Thousand Oaks, CA.

Denzin, N. K. 1997. *Interpretive Ethnography - Ethnographic Practices for the 21st Century*, (Sage Publications: Thousand Oaks, California.

Dillman, D. A. 2011. *Mail and Internet surveys: The tailored design method--2007 Update with new Internet, visual, and mixed-mode guide*, (John Wiley & Sons.

Ding, Y., and Foo, S. 2002. "Ontology research and development. Part 1-a review of ontology generation," *Journal of Information Science* (28:2), pp 123-136.

Dolk, D. R., and Konsynski, B. R. 1984. "Knowledge representation for model management systems," *Software Engineering, IEEE Transactions on*:6), pp 619-628.

Doran, G. T. 1981. "There'sa SMART way to write management's goals and objectives," *Management review* (70:11), pp 35-36.

Driver, M. J., Brousseau, K. R., and Hunsaker, P. L. 1998. *The dynamic decision maker: Five decision styles for executive and business success*, (iUniverse.

Dubin, R. 1978. *Theory Building*, (Free Press: New York.

Duncker, K., and Lees, L. S. 1945. "On problem-solving," *Psychological monographs* (58:5), p i.

Einstein, A., and Infeld, L. 1938. "The Evolution of Physics," New York, Simon and Schuster.

Eisenhardt, K. M. 1989. "Building theories from case study research," *Academy of Management Review* (14:4), pp 532-550.

Elam, J. J., and Henderson, J. C. 1983. "Knowledge engineering concepts for decision support system design and implementation," *Information & Management* (6:2), pp 109-114.

Elfeky, M. G., Verykios, V. S., and Elmagarmid, A. K. Year. "TAILOR: A record linkage toolbox," 18th International Conference on Data Engineering IEEE2002, pp. 17-28.

Engels, R. Year. "Planning tasks for knowledge discovery in databases: performing task-oriented user-guidance," International Conference on Knowledge Discovery & Data Mining AAAI-Press, Portland, OR, 1996a, pp. 170-175.

Engels, R. Year. "Planning tasks for Knowledge Discovery in Databases; Performing Task-Oriented User-Guidance," KDD1996b, pp. 170-175.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. 1996a. "The KDD process for extracting useful knowledge from volumes of data," *Communications of ACM* (39:11), pp 27-34.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. Year. "Knowledge discovery and data mining: Towards a unifying framework," 2nd International Conference on KnowledgeKnowledge Discovery and Data Mining, Portland, OR, 1996b, pp. 82-88.

Feiler, P. H., and Humphrey, W. S. Year. "Software process development and enactment: Concepts and definitions," Software Process, 1993. Continuous Software Process Improvement, Second International Conference on the, IEEE1993, pp. 28-40.

Fellegi, I. P., and Holt, D. 1976. "A systematic approach to automatic edit and imputation," *Journal of the American Statistical Association* (71:353), pp 17-35.

Fernández López, M., Gómez-Pérez, A., Pazos Sierra, A., and Pazos Sierra, J. 1999. "Building a chemical ontology using methontology and the ontology design environment,").

Fine, G. A., and Elsbach, K. D. 2000. "Ethnography and experiment in social psychological theory building: Tactics for integrating qualitative field data with quantitative lab data," *Journal of Experimental Social Psychology* (36:1), pp 51-76.

Fourer, R. 1983. "Modeling languages versus matrix generators for linear programming," *ACM Transactions on Mathematical Software (TOMS)* (9:2), pp 143-183.

Franch, X., and Carvallo, J. P. 2003. "Using quality models in software package selection," *IEEE Software* (20:1), pp 34-41.

Freitas, A. A. 1999. "On rule interestingness measures," *Knowledge-Based Systems* (12:5–6), pp 309-315.

Friedman, T. 2013. "Magic Quadrant for Data Quality Tools," Gartner Group, Stamford, CT.

Gangemi, A., Catenacci, C., Ciaramita, M., and Lehmann, J. 2006. "Modelling ontology evaluation and validation," in *The Semantic Web: Research and Applications*, Springer, pp. 140-154.

Gartner, I. 2013. "Gartner IT Glossary," *Technology Research*).

Geng, L., and Hamilton, H. J. 2006. "Interestingness measures for data mining: A survey," *ACM Comput. Surv.* (38:3), p 9.

Geoffrion, A. M. 1987. "An introduction to structured modeling," *Management Science* (33:5), pp 547-588.

Geoffrion, A. M. Year. "Reusing structured models via model integration," the 22nd Annual Hawaii International Conference on System Sciences, IEEE, Hawaii, 1989, pp. 601-611.

Ghallab, M., Nau, D., and Traverso, P. 2004. *Automated Planning: Theory & Practice*, (Morgan Kaufmann: San Francisco.

Gioia, D. A., and Pitre, E. 1990. "Multiparadigm perspectives on theory building," *Academy of Management Review* (15:4), pp 584-602.

Glaser, B. G., and Strauss, A. 1967a. *The Discovery of Grounded Theory : Strategies for Qualitative Research*, (Aldine Publishing: Chicago, IL.

Glaser, B. G., and Strauss, A. 1967b. *The Discovery of Grounded Theory: Strategies for Qualitative Research.*, (Aldine, Chicago.

Gómez-Pérez, A. 1996. "Towards a framework to verify knowledge sharing technology," *Expert Systems with Applications* (11:4), pp 519-529.

Gregor, S., and Hevner, A. R. 2013. "Positioning and presenting design science research for maximum impact," *MIS Quarterly* (37:2), pp 337-356.

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

Gruber, T. R. 1995. "Toward principles for the design of ontologies used for knowledge sharing?," *International Journal of Human-Computer Studies* (43:5), pp 907-928.

Grüninger, M., and Fox, M. S. 1995. "Methodology for the Design and Evaluation of Ontologies,").

Guarino, N., and Welty, C. 2002. "Evaluating ontological decisions with OntoClean," *Communications of the ACM* (45:2), pp 61-65.

Guitouni, A., and Martel, J.-M. 1998. "Tentative guidelines to help choosing an appropriate MCDA method," *European Journal of Operational Research* (109:2), pp 501-521.

Hagerty, J., Sallam, R. L., and Richardson, J. 2012. "Magic quadrant for business intelligence platforms," Gartner Group, Stamford, CT.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. 2009. "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter* (11:1), pp 10-18.

Hammersley, M., and Atkinson, P. 2007. *Ethnography - Principles in Practice*, (3 ed.) Routledge, Taylor and Francis Group.

Hartmann, J., and Sure, Y. Year. "A knowledge discovery workbench for the Semantic Web," International Workshop on Mining for and from the Semantic Web2004, p. 56.

Hatami-Marbini, A., and Tavana, M. 2011. "An extension of the Electre I method for group decision-making under a fuzzy environment," *Omega* (39:4), pp 373-386.

Hayden, J. A. 2008. "Health Belief Model," in *Introduction to Health Behavior Theory*, Jones and Bartlett Publishers.

Herrera, F., Herrera-Viedma, E., and Verdegay, J. L. 1995. "A sequential selection process in group decision making with a linguistic assessment approach," *Information Sciences* (85:4), pp 223-239.

Hevner, A. R., March, S. T., and Park, J. 2004. "Design science in information systems research," *MIS Quarterly* (28:1), pp 75-105.

Hilario, M., Kalousis, A., Nguyen, P., and Woznica, A. Year. "A data mining ontology for algorithm selection and meta-mining," ECML/PKDD09 Workshop on 3rd generation Data Mining (SoKD-09) 2009, pp. 76-87.

Huda, Z. 2006. "Study design for the measurement of gynaecological morbidities and health seeking behaviour," in *Investigating reproductive tract infections and other gynaecological disorders- A multi-disciplinary research approach*, S. Jeejebhoy, M. Koenig and C. Elias (eds.), Cambridge University Press.

Hwang, C. L., and Masud, A. S. 1979. *Multiple objective decision making, methods and applications*, (Springer-Verlag: Berlin/New York.

Jadhav, A. S., and Sonar, R. M. 2009. "Evaluating and selecting software packages: A review," *Information and Software Technology* (51:3), pp 555-563.

Jarke, M., Jeusfeld, M. A., Quix, C., and Vassiliadis, P. 1999. "Architecture and quality in data warehouses: An extended repository approach," *Information Systems* (24:3), pp 229-253.

Jeusfeld, M. A., Quix, C., and Jarke, M. 1998. "Design and Analysis of Quality Information for Data Warehouses," in *Conceptual Modeling – ER '98,* T. W. Ling, S. Ram and M. L. Lee (eds.), Springer-Verlag: Berlin Heidelberg, pp. 349-362.

Jick, T. D. 1979. "Mixing qualitative and quantitative methods: Triangulation in action," *Administrative Science Quarterly* (24:4), pp 602-611.

Kart, L., Linden, A., and Schulte, W. R. 2013. "Extend Your Portfolio of Analytics Capabilities," Gartner Group, Stamford, CT.

Kaufmann, E., and Bernstein, A. 2010. "Evaluating the usability of natural language query languages and interfaces to Semantic Web knowledge bases," *Web Semantics: Science, Services and Agents on the World Wide Web* (8:4), pp 377-393.

kdnuggets.com 2007. "KDnuggets Polls : Data Mining Methodology ".

KDR 2008. "Kerala Development Report," Academic Foundation, New Delhi.

Keeney, R. L. 2009. *Value-focused thinking: A path to creative decisionmaking*, (Harvard University Press.

Keim, D., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., and Melançon, G. 2008. *Visual analytics: Definition, process, and challenges*, (Springer.

Kietz, J.-U., Serban, F., and Bernstein, A. Year. "eProPlan : A Tool to Model Automatic Generation of Data Mining Workflows," ECML Workshop on third generation data mining: Towards service-oriented knowledge discovery (SoKD-2010), Barcelona, Spain, 2010.

Kietz, J., Serban, F., Bernstein, A., and Fischer, S. Year. "Towards cooperative planning of data mining workflows," the Third Generation Data Mining Workshop at the 2009 European Conference on Machine Learning (ECML 2009)2009, pp. 1-12.

Kimball, R., and Ross, M. 2011. *The data warehouse toolkit: the complete guide to dimensional modeling*, (John Wiley & Sons.

Kiss, L. N., Martel, J.-M., and Nadeau, R. 1994. "ELECCALC — an interactive software for modelling the decision maker's preferences," *Decision Support Systems* (12:4–5), pp 311-326.

Köksalan, M., Wallenius, J., and Zionts, S. 2011. *Multiple Criteria Decision Making: From Early History to the 21st Century*, (World Scientific Publishing Company Incorporated.

Krishnan, R., and Chari, K. 2000. "Model management: survey, future research directions and a bibliography," *Interactive Transactions of OR/MS* (3:1).

Kurgan, L. A., and Musilek, P. 2006. "A survey of knowledge discovery and data mining process models," *Knowledge Engineering Review* (21:1), pp 1-24.

Larichev, O. I., Kortnev, A. V., and Kochin, D. Y. 2002. "Decision support system for classification of a finite set of multicriteria alternatives," *Decision Support Systems* (33:1), pp 13-21.

Leake, D. 1996. "CBR in Context: The Present and Future," in *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, D. Leake (ed.), AAAI Press/MIT Press: Menlo Park.

Leavitt, N. 2002. "Data mining for the corporate masses?," *Computer* (35:5), pp 22-24.

Lee, D.-H., Kim, K.-J., and Köksalan, M. 2011. "A posterior preference articulation approach to multiresponse surface optimization," *European Journal of Operational Research* (210:2), pp 301-309.

Lee, Y. W., Strong, D. M., Kahn, B. K., and Wang, R. Y. 2002. "AIMQ: a methodology for information quality assessment," *Information & Management* (40:2), pp 133-146.

Leung, K. R., and Leung, H. K. 2002. "On the efficiency of domain-based COTS product selection method," *Information and Software Technology* (44:12), pp 703-715.

Lewis, I., and Munn, P. 1987. *So You Want To Do Research! A Guide for Teachers on How To Formulate Research Questions*, (ERIC.

Li, Y., Thomas, M., and Osei-Bryson, K.-M. 2014. "Using Association Rules Mining to Facilitate Qualitative Data Analysis in Theory Building," in *Advances in Research Methods for Information Systems Research*, Springer, pp. 79-91.

Liang, T.-P. 1988. "Development of a Knowledge-Based Model Management System Special Focus Article," *Operations Research* (36:6), pp 849-863.

Limayem, M., and DeSanctis, G. 2000. "Providing decisional guidance for multicriteria decision making in groups," *Information Systems Research* (11:4), pp 386-401.

Lindner, G., and Studer, R. 1999. "AST: Support for algorithm selection with a CBR approach," *Principles of Data Mining and Knowledge Discovery*), pp 418-423.

Linoff, G. S., and Berry, M. 2011. *Data Mining Techniques:For Marketing, Sales, and Customer Relationship Management* (2nd ed.) Wiley Publishing, Inc: indianapolis IN.

Liu, B., Grossman, R., and Zhai, Y. 2004. "Mining web pages for data records," *Intelligent Systems, IEEE* (19:6), pp 49-55.

Liu, B., Hsu, W., and Ma, Y. 1999. "Pruning and summarizing the discovered associations," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM: San Diego, California, United States, pp. 125-134.

Liu, B., and Tuzhilin, A. 2008. "Managing large collections of data mining models," *Communications of the ACM* (51:2), pp 85-89.

Liu, L., and Chi, L. Year. "Evolutionary data quality," Proc. 7th International Conference on Information Quality (IQ 2002)2002.

Lozano-Tello, A., and Gomez-Perez, A. 2004. "ONTOMETRIC: A Method to Choose the Appropriate Ontology," *Journal of Database Management (JDM)* (15:2), pp 1-18.

Ma, L.-C., and Li, H.-L. 2011. "Using Gower Plots and Decision Balls to rank alternatives involving inconsistent preferences," *Decision Support Systems* (51:3), pp 712-719.

Maedche, A., and Staab, S. 2001. "Ontology learning for the semantic web," *IEEE Intelligent systems* (16:2), pp 72-79.

Maes, A., and Poels, G. 2006. "Evaluating quality of conceptual models based on user perceptions," in *Conceptual Modeling-ER 2006*, Springer, pp. 54-67.

Malle, B. F., and Knobe, J. 1997. "The Folk Concept of Intentionality," *Journal of Experimental Social Psychology* (33), pp 101-121.

Marbán, Ó., Mariscal, G., Menasalvas, E., and Segovia, J. 2007. "An Engineering Approach to Data Mining Projects," in *Intelligent Data Engineering and Automated Learning - IDEAL 2007*, H. Yin, P. Tino, E. Corchado, W. Byrne and X. Yao (eds.), Springer Berlin Heidelberg, pp. 578-588.

March, S. T., and Smith, G. F. 1995. "Design and natural science research on information technology," *Decision Support Systems* (15:4), pp 251-266.

March, S. T., and Storey, V. C. 2008. "Design science in the information systems discipline: an introduction to the special issue on design science research," *MIS Quarterly* (32:4), pp 725-730.

Mariscal, G., Marbán, Ó., and Fernández, C. 2010. "A survey of data mining and knowledge discovery process models and methodologies," *Knowledge Engineering Review* (25:2), p 137.

Marler, R. T., and Arora, J. S. 2004. "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization* (26:6), pp 369-395.

Martin, J. 1974. *Privacy, Accuracy and Security in Computer Systems*, (Prentice-Hall, Inc: Englewood Cliffs, NJ.

Matijaš, M., Suykens, J. A., and Krajcar, S. 2013. "Load forecasting using a multivariate meta-learning system," *Expert Systems with Applications* (40:11), pp 4427-4437.

McGarry, K. 2005. "A survey of interestingness measures for knowledge discovery," *The Knowledge Engineering Review* (20:01), pp 39-61.

Miles, M. B., and Huberman, A. M. 1994. *Qualitative Data Analysis*, (Sage: Thousand Oaks, CA.

Mintzberg, H., Raisinghani, D., and Theoret, A. 1976. "The structure of" unstructured" decision processes," *Administrative Science Quarterly*), pp 246-275.

Montella, A. 2011. "Identifying crash contributory factors at urban roundabouts and using association rules to explore their relationships to different crash types," *Accident Analysis & Prevention* (43:4), pp 1451-1463.

Morik, K., and Scholz, M. 2004. "The miningmart approach to knowledge discovery in databases," *Intelligent Technologies for Information Analysis*), pp 47-65.

Morris, W. T. 1967. "On the art of modeling," *Management Science* (13:12), pp B-707-B-717.

Muhanna, W. A., and Pick, R. A. 1994. "Meta-modeling concepts and tools for model management: a systems approach," *Management Science* (40:9), pp 1093-1123.

Munro, S., Lewin, S., Swart, T., and Volmink, J. 2007. "A Review Of Health Behaviour Theories: How Useful Are These For Developing Interventions To Promote Long-Term Medication Adherence For TB and HIV/AIDS? ," *BMC Public Health* (7:104).

Naumann, F. 2002. *Quality-driven query answering for integrated information systems*, (Springer-Verlag.

Newell, A., and Simon, H. A. 1972. *Human problem solving*, (Prentice-Hall Englewood Cliffs, NJ.

Nonaka, I. 1994. "A Dynamic Theory of Organizational Knowledge Creation," *Organization Science* (5:1), pp 4-37.

Norman, P., Conner, M., and Bell, R. 1999. "The theory of planned behavior and smoking cessation.," *Health Psychology* (18:1) Jan 1999, pp 89-94.

Noy, N. F., and McGuinness, D. L. 2001. "Ontology development 101: A guide to creating your first ontology," Stanford knowledge systems laboratory technical report KSL-01-05 and Stanford medical informatics technical report SMI-2001-0880.

Nunamaker Jr, J., Romano Jr, N., and Briggs, R. Year. "A Framework for Collaboration and Knowledge Managemen," Proceedings of the 34th Annuual Hawaii International Conference on System Sciences (HICSS-34), Hawaii, 2001, pp. 1060-1071.

Orlovsky, S. 1978. "Decision-making with a fuzzy preference relation," *Fuzzy Sets and Systems* (1:3), pp 155-167.

Orr, K. 1998. "Data quality and systems theory," *Communications of ACM* (41:2) Feb., pp 66-71.

Osei-Bryson, K. M. 2012. "A context-aware data mining process model based framework for supporting evaluation of data mining results," *Expert Systems with Applications* (39:1), pp 1156-1164.

Ozernoy, V. M. 1992. "Choosing the best multiple criteria decision-making method," *Infor* (30:2), pp 159-171.

Pang, J., and Liang, J. 2012. "Evaluation of the results of multi-attribute group decision-making with linguistic information," *Omega* (40:3), pp 294-301.

Panov, P., Dzeroski, S., and Soldatova, L. Year. "OntoDM: An ontology of data mining," IEEE International Conference on Data Mining Workshops, 2008 (ICDMW'08) IEEE, Pisa, Italy, 2008, pp. 752-760.

Patel, D. A., Burnett, N. M., Curtis, K. M., Hillis, S. D., and Marchbanks, P. A. 2003. *Reproductive tract infections*, (US Department of Health and Human Services, Centers for

Disease Control and Prevention, National Center for Chronic Disease Prevention and Health Promotion, Division of Reproductive Health.

Patel, N., and Hlupic, V. Year. "A methodology for the selection of knowledge management (KM) tools," 24th International Conference on Information Technology Interfaces (ITI), IEEE, Cavtat, Croatia, 2002, pp. 369-374.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. 2007. "A design science research methodology for information systems research," *Journal of Management Information Systems* (24:3), pp 45-77.

Peroni, S., and Shotton, D. 2012. "FaBiO and CiTO: ontologies for describing bibliographic resources and citations," *Web Semantics: Science, Services and Agents on the World Wide Web* (17), pp 33-43.

Piatetsky-Shapiro, G., and Fayyad, U. (eds.) *Knowledge Discovery in Databases* AAAI Press, MIT 1991.

Pipino, L. L., Lee, Y. W., and Wang, R. Y. 2002. "Data Quality Assessment," *Communications of the ACM* (45:4), pp 211-218.

Podpečan, V., Zemenova, M., and Lavrač, N. 2012. "Orange4WS Environment for Service-Oriented Data Mining," *The Computer Journal* (55:1) January 1, 2012, pp 82-98.

Poles, S., Vassileva, M., and Sasaki, D. 2008. "Multiobjective Optimization Software," in *Multiobjective Optimization*, J. Branke, K. Deb, K. Miettinen and R. Slowinski (eds.), Springer: Berlin / Heidelberg, pp. 329-348.

Pounds, W. F. 1965. "The process of problem finding,").

Pressman, R. 2005. *Software Engineering: A Practitioner's Approach* (McGraw-Hill Science: New York.

Protégé 2007.

Provost, F., and Fawcett, T. 2013. "Data Science and its Relationship to Big Data and Data-Driven Decision Making," *Big Data* (1:1), pp 51-59.

Prusty, R. K., and Unisa, S. Year. "Reproductive tract infections and treatment seeking behavior among married adolescent women in India," Population Association of America Annual Meeting Program, San Francisco, CA2012.

RacerPro 2012. "Protégé 4.x Reasoner Plugin for RacerPro."

Redman, T. C. 1997. *Data Quality for the Information Age*, (Artech House, Inc.: Boston.

Reinartz, T. 2002. "Stages of the discovery process," in *Handbook of Data Mining and Knowledge Discovery*, W. Klosgen and J. Zytkow (eds.), Oxford University Press, Inc.: New York, pp. 185-192.

Robeyns, I. 2003. "Sen's Capability Approach and Gender Inequality: Selecting Relevant Capabilities," *Feminist Economics* (9:2-3), pp 61-92.

Rohanizadeh, S. S., and Moghadam, M. B. 2009. "A Proposed Data Mining Methodology and its Application to Industrial Procedures," *Journal of Industrial Engineering*).

Rosenstock, I. M. 2005. "Why People Use Health Services," *The Milbank Fund Quarterly* (83:4).

Rosenstock, I. M., Strecher, V. J., and Becker, M. H. 1994. "The Health Belief Model and HIV Risk Behavior Change," in *Preventing AIDS: Theories and Methods of Behavioral Interventions*, R. J. DiClemente, and Peterson, J. L. (ed.), Plenum Press: New York.

Roy, B. 1973. "How outranking relation helps multiple criteria decision making," in *Topics in Multiple Criteria Decision Making*, J. Cochrane and M. Zeleny (eds.), University of South Carolina Press, pp. 179-201.

Roy, B. 1985. *Méthodologie multicritere d'aide ala décision* (Économica: Paris.

Sallam, R. L., and Cearley, D. W. 2012. "Advanced Analytics: Predictive, Collaborative and Pervasive," Gartner Group, Stamford, CT.

Scannapieco, M., Virgillito, A., Marchetti, C., Mecella, M., and Baldoni, R. 2004. "The DaQuinCIS Architecture: a Platform for Exchanging and Improving Data Quality in Cooperative Information Systems," *Information Systems* (29:7), pp 551-582.

Scarinci, I. C., Bandura, L., Hidalgo, B., and Cherrington, A. 2011. "Development of a Theory-Based (PEN-3 and Health Belief Model), Culturally Relevant Intervention on Cervical Cancer Prevention Among Latina Immigrants Using Intervention Mapping.," *Health Promotion Practice* (12:3).

Schwartz, D. G. 2003. "From open IS semantics to the Semantic Web: The road ahead," *IEEE Intelligent systems* (18:3), pp 52-58.

Schwartzman, H. B. 1993. *Ethnography in organizations*, (Sage.

Seddon, P. B. 1997. "A respecification and extension of the DeLone and McLean model of IS success," *Information Systems Research* (8:3), pp 240-253.

Seixedo, C., and Tereso, A. P. 2010. "A multicriteria decision aid software application for selecting MCDA software using AHP," in *2nd International Conference on Engineering Optimization*: Lisbon, Portugal.

Sen, G., and Iyer, A. 2000. "Health Sector Changes and Health Equity in the 1990s," Humanist Institute for Cooperation with Developing Countries, Netherlands.

Serban, F. 2010. "Auto-experimentation of KDD workflows based on ontological planning," in *Proceedings of the 9th international semantic web conference on The semantic web - Volume Part II*, Springer-Verlag: Shanghai, China, pp. 313-320.

Serban, F., Vanschoren, J., Kietz, J.-U., and Bernstein, A. 2012. "A survey of intelligent assistants for data analysis," *ACM Computing Surveys*).

Sharma, S. 2008. *An integrated knowledge discovery and data mining model*, Virginial Commonwealth University, Richmond.

Sharma, S., Osei-Bryson, K.-M., and Kasper, G. M. 2012. "Evaluation of an integrated Knowledge Discovery and Data Mining process model," *Expert Systems with Applications* (39:13), pp 11335-11348.

Sharma, S., and Osei-Bryson, K. M. 2009. "Framework for formal implementation of the business understanding phase of data mining projects," *Expert Systems with Applications* (36:2), pp 4114-4124.

Shearer, C. 2000. "The CRISP-DM model: the new blueprint for data mining," *Journal of Data Warehousing* (5:4), pp 13-22.

Shim, J. P., Warkentin, M., Courtney, J. F., Power, D. J., Sharda, R., and Carlsson, C. 2002. "Past, present, and future of decision support technology," *Decision Support Systems* (33:2), pp 111-126.

Siau, K., and Rossi, M. Year. "Evaluation of information modeling methods-a review," System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on, IEEE1998, pp. 314-322.

Simon, H. A. 1969. *The Sciences of the Artificial*, (The MIT Press: Cambridge, MA.

Simon, H. A. 1977. "The structure of ill-structured problems," in *Models of discovery*, Springer, pp. 304-325.

Smith, G. F. 1988. "Towards a heuristic theory of problem structuring," *Management Science* (34:12), pp 1489-1506.

Smith, M. K., McGuinness, D., Volz, R., and Welty, C. 2002. "Web Ontology Language (OWL) Guide Version 1.0 ".

Solarte, J. 2002. *A Proposed Data Mining Methodoloy and Its Aplication to Industrial Engineering*, University of Tennessee, Knoxville.

Sommerville, I. 2007. *Software engineering*, (8th ed.) Addison Wesley: New York.

Spender, J.-C. 1996. "Organizational knowledge, learning and memory: three concepts in search of a theory," *Journal of Organizational Change Management* (9:1), pp 63-78.

Sprague, R. H. 1980. "A framework for the development of decisoin support systems," *MIS Quarterly* (4:4), pp 1-26.

Sprague, R. H., and Watson, H. J. Year. "Model management in MIS," the 7th National AIDS1975, pp. 213-215.

Strecher, V., and Rosenstock, I. M. 1997. "The Health Belief Model," in *Health Behavior and Health Education: Theory, Research and Practice*, K. Glanz, Lewis, F. M., Rimer B.K., and Viswanath, K. (ed.), Jossey-Bass: San Francisco.

Tejada, S., Knoblock, C. A., and Minton, S. 2001. "Learning object identification rules for information integration," *Information Systems* (26:8), pp 607-633.

Thomas, J. J., and Cook, K. A. 2006. "A visual analytics agenda," *Computer Graphics and Applications, IEEE* (26:1), pp 10-13.

Thomas, M., and Narayan, P. 2014. "The Role of Participatory Communication in Tracking Unreported Reproductive Tract Issues in Marginalized Communities," *Information Technology for Development*:ahead-of-print), pp 1-17.

Tsoukiàs, A. 1991. "Preference modeling as a reasoning process: A new way to face uncertainty in multiple criteria decision support systems," *European Journal of Operational Research* (55:3), pp 309-318.

Turskis, Z., and Zavadskas, E. K. 2011. "Multiple criteria decision making (MCDM) methods in economics: an overview," *Technological and economic development of economy*:2), pp 397-427.

Tyrrell, S. 2000. "The many dimensions of the software process," *Crossroads* (6:4), pp 22-26.

Urrutia, M. T. 2009. *Development and Testing of a Questionnaire: Beliefs about Cervical Cancer and Pap Test in Chilean Women.*, University of Miami, Coral Gables, Miami.

Uschold, M., and Gruninger, M. 1996. "Ontologies: Principles, methods and applications," *The knowledge engineering review* (11:02), pp 93-136.

Vaishnavi, V. K., and Kuechler Jr, W. 2007. "Introduction to Design Science research in Information and Communication Technology," in *Design science research methods and patterns: innovating information and communication technology*, CRC Press: Boca Raton, FL, p. 20.

van der Aalst, W. M. 1998. "The application of Petri nets to workflow management," *Journal of circuits, systems, and computers* (8:01), pp 21-66.

van Solingen, R., Basili, V., Caldiera, G., and Rombach, H. D. 2002a. "Goal Question Metric (GQM) Approach," in *Encyclopedia of software engineering*, John Wiley & Sons, Inc., pp. 528-532.

Van Solingen, R., Basili, V., Caldiera, G., and Rombach, H. D. 2002b. "Goal question metric (gqm) approach," *Encyclopedia of software engineering*).

Van Solingen, R., and Berghout, E. 1999. *The Goal/Question/Metric Method: a practical guide for quality improvement of software development*, (McGraw-Hill: London.

Vassiliadis, P., Bouzeghoub, M., and Quix, C. 2000. "Towards Quality-oriented Data Warehouse Usage and Evolution," *Information Systems* (25:2), pp 89-115.

Vessey, I., and Glass, R. 1998. "Strong vs. weak approaches to systems development," *Communications of the ACM* (41:4), pp 99-102.

Vilalta, R., and Drissi, Y. 2002. "A perspective view and survey of meta-learning," *Artificial Intelligence Review* (18:2), pp 77-95.

Volkema, R. J. 1983. "Problem formulation in planning and design," *Management Science* (29:6), pp 639-652.

Wallenius, J., Dyer, J. S., Fishburn, P. C., Steuer, R. E., Zionts, S., and Deb, K. 2008. "Multiple criteria decision making, multiattribute utility theory: Recent accomplishments and what lies ahead," *Management Science* (54:7), pp 1336-1349.

Walls, J. G., Widmeyer, G. R., and Sawy, O. A. E. 1992. "Building an Information System Design Theory for Vigilant EIS," *Information Systems Research* (3:1), pp 36-59.

Wand, Y., and Wang, R. Y. 1996. "Anchoring data quality dimensions in ontological foundations," *Communications of ACM* (39:11) Nov., pp 86-95.

Wang, R. Y. 1998. "A Product Perspective on Total Data Quality Management," *Communications of the ACM* (41:2) Feb., pp 58-65.

Wang, R. Y., Reddy, M. P., and Kon, H. B. 1995. "Toward quality data: An attribute-based approach," *Decision Support Systems* (13:3-4), pp 349-372.

Wang, R. Y., and Strong, D. M. 1996. "Beyond Accuracy: What Data Quality Means to Data Consumers," *Journal of Management Information Systems* (12:4) Mar., pp 5-33.

Wei, C.-C., Chien, C.-F., and Wang, M.-J. J. 2005. "An AHP-based approach to ERP system selection," *International Journal of Production Economics* (96:1), pp 47-62.

Weick, K. E. 1979. *The social psychology of organizing*, (McGraw–Hill: New York.

Weistroffer, H., and Li, Y. 2014. "Multiple Criteria Decision Support Software," in *Multiple Criteria Decision Analysis: State of the Art Surveys - forthcoming*, Springer New York.

Weistroffer, H., Smith, C., and Narula, S. 2005. "Multiple Criteria Decision Support Software," in *Multiple Criteria Decision Analysis: State of the Art Surveys*, Springer New York, pp. 989-1009.

Wiederhold, G. 1996. "Foreword: on the barriers and future of knowledge discovery," in *Advances in Knowledge Discovery and Data Mining* U. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy (eds.), AAAI Press: Menlo Park, CA.

Will, H. J. 1975. "Model management systems," in *Information Systems and Organization Structure*, Walter de Gruyter,: Berlin, pp. 468-482.

Winkler, W. E. 2004. "Methods for evaluating and creating data quality," *Information Systems* (29:7), pp 531-550.

Wixom, B. H., and Watson, H. J. 2001. "An Empirical Investigation of the Factors Affecting Data Warehousing Success," *MIS Quarterly* (25:1) Mar., pp 17-41.

Yang, Q., and Wu, X. 2006. "10 challenging problems in data mining research," *International Journal of Information Technology & Decision Making* (5:04), pp 597-604.

Yin, R. K. 2003. *Case Study Research: Design and Methods*, (3rd ed.) Sage: Thousand Oats, CA.

Yu, J., Thom, J. A., and Tam, A. 2009. "Requirements-oriented methodology for evaluating ontologies," *Information Systems* (34:8), pp 766-791.

Zack, M., McKeen, J., and Singh, S. 2009. "Knowledge management and organizational performance: an exploratory analysis," *Journal of Knowledge Management* (13:6), pp 392-409.

Zorrilla, M., and García-Saiz, D. 2013. "A service oriented architecture to provide data mining services for non-expert data miners," *Decision Support Systems* (55:1), pp 399-411.

APPENDIX A: DATA QUALITY DIMENSION DEFINITIONS

DQ Quality Dimension Classification (Redman 1997)

Categories	Subcategories	Dimensions	Definitions
Conceptual Schema	Content	Relevance	The schema should provide data needed by the application.
		Obtainability	Data value should be easily obtained
		Clarity of definition	Each term in the definition of the schema should be clearly defined
	Scope	Comprehensiveness	Each needed data item should be included
		Essentialness	No unneeded data item should be included
	Level of detail	Attribute granularity	The attributes should be defined at the right level of detail to support applications
		Domain precision	The domain of possible values should be just large enough to support applications
	Composition	Naturalness	Each item in the schema should be just large enough to support applications
		Occurrence identifiability	The schema should make identification of individual entities easy
		Homogeneity	Entity types should be defined to minimize the occurrence of unnecessary attributes
		Minimum redundancy	Redundancy should be kept to a minimum
	Schema consistency	Semantic consistency	The schema should be clear and unambiguous and consistent
		Structural consistency	Entity types and attributes should have the same basic structure whenever possible
	Reaction to change	Robust	Be wide enough so it does require change over time application change
		Flexibility	When necessary the schema should be easily changed
Data	Data Values	Accuracy	A datum (e,a,v) has the accuracy as: the nearest of the value v to some v', which is considered as the correct one for the entity (e) and the attribute (a)
		Correctness	If the accuracy value v is the same as v', the datum is said to be correct
		Completeness	Refers to the degree to which values are

			presented in a data collection
		Currency	Refers to the degree a datum is up to date. A correct value v' may change over time. A datum is considered to be outdated if it is incorrect at time t, though it might be correct in previous times.
		Consistency	The same datum is represented coherently in different copies, or different data respect integrity constraints.
	Data Representation	Appropriateness	it is formatted appropriately for user needs.
		Interpretability	The user is able to interpret correctly from data format
		Portability	The format can be applied to as a wide set of situation as possible.
		Format precision	Ability to distinguish among elements in the domain that must be distinguished by users.
		Format flexibility	Changes in user needs and recoding medium can be easily accommodated
		Ability to represent null values	Ability to distinguish without ambiguities null as default values from applicable values of the domain
		Efficient use of memory	Efficiency in the physical representation.
		Coherence	Physical instances of data are formatted coherently

DQ Quality Dimension Classification (Jarke et al. 1999)

Categories (User)	Perspective - Object	Dimensions	Definitions
DW Design and Administration	Conceptual - Model	Correctness	Number of conflicts to other models/real worlds
		Completeness	Level of covering; level of represented business rules
		Minimality	Number of redundant entities/relationships in a model
		Traceability	Are the requirements and changes recorded?
		Interpretability	Quality of document

		Metadata Evolution	Is it (of model) documented?
	Conceptual - Concept	Correctness	Correct description wrt. real world entity
		Completeness	Number of missing attributes.
		Minimality	Equivalence of the description with that of other concepts in the same model
		Traceability	Are the requirements and changes recorded?
		Interpretability	Quality of document
		Metadata Evolution	Is it (of concept) documented?
	Logical - Schema	Correctness	Correctness of mapping of conceptual model to logical schema
		Completeness	Number of missing entities wrt. the conceptual model
		Minimality	Number of redundant relations
		Traceability	Are the requirements and changes recorded?
		Interpretability	Quality of document
		Metadata Evolution	Is it (schema)documented?
	Logical – Type (class in my view)	Correctness	Correctness of the mapping of concept to a type.
		Completeness	Number of missing concept wrt. the conceptual model
		Minimality	Number of redundant attributes
		Traceability	Are the requirements and changes recorded?
		Interpretability	Quality of document
		Metadata Evolution	Is it (type)documented?
Data Usage	Logical - Schema	Accessibility	Is the schema definition accessible?
		Availability	Frequency of updates
		Security	Access rights (level of security)
		Usefulness	Is the schema used by the user?
		Interpretability	Is the schema understandable?
	Logical - Type	Accessibility	Is the type visible and accessible?
		Availability	Frequency of updates
		Security	Access rights (level of security)
		Usefulness	Is the type used by the user?
		Interpretability	Is the type understandable?
	Physical - Agent	Accessibility	Is the network sufficient for delivered data?
		Availability	Response time
		Security	Physical access restriction?

		Usefulness	Is the data delivered by the agent really used in the destination store?
		Interpretability	Is the data delivered understandable?
	Physical – Data Store	Accessibility	Is the data source accessible?
		Availability	Uptime of data store; response time
		Security	Unauthorized access prevented?
		Usefulness	Is the data in this store queried?
		Interpretability	Is the data stored understandable?
Data	Physical – Agent	Completeness	Number of tuples delivered wrt. expected number
		Credibility	Believability in the process that delivers the values
		Accuracy	Number of delivered accurate tuples
		Consistency	Is delivered data consistent with other data?
		Data Interpretability	Number of tuples with interpretable data, documentation for key values, is the forma understandable?
	Physical – Data Store	Completeness	Number of store null values where there are not expected
		Credibility	Number of tuples with default values
		Accuracy	Level of preciseness; number of accurate tuples
		Consistency	Number of coding differences
		Data Interpretability	Number of tuples with interpretable data; documentation for key value, is the format understandable?

DQ Quality Dimension Classification in Web (Naumann 2002)

Categories	Dimensions	Definitions
Content	Accuracy (precision)	In web-context, the percentage of data without data errors
	Completeness (coverage)	The quotient of the number of non-null values in a source and the size of the universal relation.
	Customer support	Usefulness of human help via telephone or email
	Documentation	Amount and usefulness of documents with metadata.
	Interpretability	The degree to which the information conforms to the technical ability of the consumer.

	Relevance	The degree to which the provided information satisfies the users need. Important criteria in information retrieval.
	Value-added	Amount of monetary benefit the use of the data provided.
Technical	Availability	A probability that a feasible query is correctly answered in a given time range. A source either available or not available (no response).
	Latency	The amount of time in seconds from issuing the query till the first data item reaches the user. If the result of query only has one data item, it equals "response time" (see below)
	Price	The amount of money a user has to pay for a query (such as stock information)
	Quality of Service	Transmission and error rates of web sources.
	Response time	The delay in seconds between submission of a query by the user and reception of complete response from the source.
	Security	The degree to which data is passed privately from users to the data source and back.
	Timeliness	The average age of the data in a source. For example, a typical free stock quote service have a 15 minutes delay between occurrence of quote and its delivery to user.
Intellectual	Believability	Degree to which the data is accepted as correct by the user. It is "expected accuracy".
	Objectivity	The degree which data is unbiased and impartial.
	Reputation	Degree to which the data an its source is in high standing.
Instantiation	Amount of data	The size of query results.
	Representation Conciseness	The degree to which the structure of the data matches the data itself.
	Representation consistency	The degree to which the structure of the data conforms to previous returned data (could from different sources)
	Understandability	The degree to which the data can be easily comprehended by the user.
	Verifiability	The degree to the data can be checked for correctness.

APPENDIX B: DATA WAREHOUSE QUALITY CONCEPTS DEFINITIONS

Concept	Definition
Quality Goal	It is an abstract requirements, related to an object (see below measureable object), has a stakeholder (the viewpoint of a user), a quality dimension (focus), and purpose. The contextual factor is the DW environment. An example of quality goal can be "increase (purpose) the refreshment (quality dimension) of the data source (object)" from the DW administrator (viewpoint).
Measurable Object	It is an object in a DW. An object can be in conceptual, logic, and physical levels of the DW. All measurable objects should be represented as base classes in the architectural design. Examples of measureable objects were given previously (in DQ dimension review section).
Quality Dimension	This has been reviewed also in the DQ dimension section. Notably, we will classify quality dimensions using Jarke 1999 approach: define quality dimensions at different levels with regard to different types of users.
Quality Factor	A quality factor is a special characteristic of the related object wrt. to the quality dimension. It is a similar concept of attribute in Figure 6 wrt. to the quality categories and subcategories. As such, a quality factor can be either directly applied to the DW object by a metric function, or needs to be decomposed into derived attributes. Also similarly, a quality dimension may have several quality factors and derived quality factors, while a quality factor may relate to several different quality dimensions.
Quality Measurement	It is the documented activity to measure the quality of a measurable object, using a quality metric to get the actual quality value of the object. It is a goal-oriented measurement, which means the same object may have different measurement values for different quality goals.
Metric Unit	A quality value may have a metric unit, which is analog to physical units like "meter/second" for measuring speed. For example, the timeliness of a data source can be measured in "minutes". In case of multiple DQ factors are defined in one DQ goal, MCDM approach needs to be adopted to access the overall measurement. This will involve the standardized metric unit (e.g., all different metric units are transformed between [0, 1]).
Quality Query	It is issued to check if a quality goal is fulfilled, or if a measured quality has changed. In order for a quality query to wrong, a quality measure needs to have a quality domain and a quality range (see below).
Quality Domain	It specifies permissible quality values. It is important to define a quality domain for a quality query to be issued.
Quality Range	It is expected quality values (from the stakeholders), usually in the form of intervals.

APPENDIX C: INPUT CHARACTERISTICS AND DECISION PREFERENCES

Steps	Guideline	Choice	Description	Case study Example
Input (decision problem structuring)	Input Data Scales	Ordinal	The gap between two degrees does not have a clear meaning	Cardinal (price, number of bedrooms, etc.)
		Cardinal	The ratio between two degrees can receive a meaning	
		Mixed	Both ordinal and cardinal	
	Criteria*	True criteria	Either indifference or strict preferences	True Criteria
		Pre-criteria	Either strict or weak preference, no indifference	
		Pseudo criteria	A gradation of preference	
	Alternatives	Implicit	MOO problem	MADA (finite number of candidate houses)
		Explicit	MADA problem	
Preference Modeling	Elucidation Mode	Direct Rating	Directly assess the alternatives	Pairwise Comparison (The houses are compared in pairs)
		Tradeoffs	One criterion can substitute another	
		Lotteries	Assess by a draw	
		Pairwise comparison	Assess alternatives in pairs	
	Moments of Elucidation	A Priori, Progressive, A posterior,	The preference is elucidated either a priori, or progressively, or a posterior.	A Priori (for all MADA methods)
	Preference Articulation Categories (MOO)	No, A priori, A posterior, Interactive methods	Either the preference is not articulated, or it is articulated a priori, a posterior, or interactively.	Not Applicable
	Preference Structure**	Indifference (a ***I*** b)	A is indifference to alternative B.	***(P, I)***
		Preference (a ***P*** b)	A is strictly preferred to B.	
		Weak preference (a ***Q*** b)	Hesitation between the indifference and preference.	
		Incomparability (a **R** b)	Hesitation between "A is preferred to B" and "B is preferred to A".	
		Outranking	***S*** = (***P*** U ***Q*** U ***I***)	

		(a ***S*** b)		
	Alternative Ordering*** (Binary relations between alternatives)	Partial	Binary relation has reflexivity, transitivity, and antisymmetry.	Total Preorder (All houses are comparable, and there are no uncertain cases)
		Weak	Binary relation has irreflexivity, asymmetry, transitivity, and transitivity of incomparability.	
		Semi	A special case of partial ordering with alternatives can be incomparable if their scores are within a given margin of error.	
		Total Preorder	A strong case of partial ordering with totality.	
	Decision Problematic	Description, Choice, Sorting Ranking	The DMS is formulated by description, choice, sorting, or ranking.	Ranking or Choice (the houses would be ranked in the result)
Aggregation	Alternative Aggregation Evaluation	Compensatory	Absolute compensation.	Partially compensatory (There are some compensation accepted between the different criteria)
		Non-compensatory	No compensation is accepted.	
		Partially compensatory	Some kind of compensation is accepted.	

* Please refer to reference (Roy 1985) for detailed definition.

** A and B are two alternatives, and "a" and "b" are their respective values for the criterion considered.

*** Please refer to classic mathematic *order theory* for detailed definition.

APPENDIX D: SURVEY OF DM^3 ONTOLOGY USABILITY

Part I: General Information

1. Have you finished reviewing the KDDM ontology?

   □ Yes □ No

2. To what extent are you consider your knowledge of KDDM process?

Completely Un-experienced						Completely Experienced
1	2	3	4	5	6	7

Part II: Survey Questions:

Please answer the following questions using a 1 to 7 scale with 1-Strongly Disagree, and 7-Strong Agree. Please circle your answer to each question.

3.	It was easy for me to understand what the DM3 ontology was trying to do.	1 2 3 4 5 6 7
4.	Using the DM^3 ontology was often frustrating.	1 2 3 4 5 6 7
5.	Overall, the DM^3 ontology was easy to use.	1 2 3 4 5 6 7
6.	Learning how to use the DM^3 ontology was easy.	1 2 3 4 5 6 7
7.	Overall, I think the DM3 ontology would be an improvement to the KDDA process.	1 2 3 4 5 6 7
8.	Overall, I found the DM^3 ontology is useful for the KDDA process.	1 2 3 4 5 6 7
9.	Overall, I think the DM^3 ontology center improves my performance in the KDDA process.	1 2 3 4 5 6 7
10.	The DM^3 ontology adequately met the information needs that I was asked to support.	1 2 3 4 5 6 7
11.	The DM^3 ontology was not efficient in providing the information I needed.	1 2 3 4 5 6 7
12.	Overall, I am satisfied with the DM^3 ontology for providing the information I needed.	1 2 3 4 5 6 7
13.	The DM^3 ontology was effective in providing the information I needed.	1 2 3 4 5 6 7
14.	The DM^3 ontology represents the KDDA process correctly.	1 2 3 4 5 6 7
15.	The DM^3 ontology is a realistic representation of the KDDA process.	1 2 3 4 5 6 7
16.	The DM^3 ontology contains contradicting elements.	1 2 3 4 5 6 7
17.	All the elements in the DM^3 ontology are relevant for the representation of the KDDA process.	1 2 3 4 5 6 7

18. The DM^3 ontology gives a complete representation of the KDDA process. 1 2 3 4 5 6 7

Part III: About yourself and your organization

19. Your gender

□ Male □ Female

20. Your highest level of education

□ Less than high school □ Undergraduate degree

□ High school degree □ Graduate degree

□ College degree □ Other

21. Your age

□ 20–25 □ 56–65

□ 26–35 □ 66–75

□ 36–45 □ 76–85

22. How many years have you worked in KDDM domain? _____

23. How many years have you worked for your current organization?

□ less than 1 year □ 10 - 15 years

□ 1 – 5 years □ more than 15 years

□ 5 – 10 years

24. How many years have you worked in your current position in the organization?

□ less than 1 year □ 10 - 15 years

□ 1 – 5 years □ more than 15 years

□ 5 – 10 years

25. Your job title is _____

26. Number of employees in your organization

□ Fewer than 500 □ 5,000–10,000

□ 500–999 □ More than 10,000

□ 1,000–4,999

27. In which industry is your organization operating?

□ Education □ Real Estate

□ Financial Services □ Services

□ Government □ Information Technology

□ Food/Beverage/CPG □ Telecommunications

□ Health Care □ Travel

□ Manufacturing□ □ Wholesale/Retail

□ Nonprofit □ Other, please specify _____

□ Medical, Bio-Technology, Pharmacology

APPENDIX E: SURVEY OF MCDA SOFTWARE SELECTION FRAMEWORK USABILITY

Part I: General Information

1. Have you finished reviewing the MCDA Software Selection Framework?

   □ Yes □ No

2. To what extent are you consider your knowledge of MCDA Software?

Completely Un-experienced						Completely Experienced
1	2	3	4	5	6	7

Part II: Survey Questions:

Please answer the following questions using a 1 to 7 scale with 1-Strongly Disagree, and 7-Strong Agree. Please circle your answer to each question.

1.	It was easy for me to understand what the MCDA Software Selection Center was trying to do.	1 2 3 4 5 6 7
2.	Using the MCDA Software Selection Center was often frustrating.	1 2 3 4 5 6 7
3.	Overall, the MCDA Software Selection Center was easy to use.	1 2 3 4 5 6 7
4.	Learning how to use the MCDA Software Selection Center was easy.	1 2 3 4 5 6 7
5.	Overall, I think the MCDA Software Selection Center would be an improvement to the MCDA Software Selection process.	1 2 3 4 5 6 7
6.	Overall, I found the MCDA Software Selection Framework is useful for the MCDA Software Selection process.	1 2 3 4 5 6 7
7.	Overall, I think the MCDA Software Selection Framework center improves my performance in the MCDA Software Selection process.	1 2 3 4 5 6 7
8.	The MCDA Software Selection Framework adequately met the information needs that I was asked to support.	1 2 3 4 5 6 7
9.	The MCDA Software Selection Framework was not efficient in providing the information I needed.	1 2 3 4 5 6 7
10.	Overall, I am satisfied with the MCDA Software Selection Framework for providing the information I needed.	1 2 3 4 5 6 7
11.	The MCDA Software Selection Framework was effective in providing the information I needed.	1 2 3 4 5 6 7
12.	The MCDA Software Selection Framework represents the MCDA Software Selection process correctly.	1 2 3 4 5 6 7
13.	The MCDA Software Selection Framework is a realistic	1 2 3 4 5 6 7

representation of the MCDA Software Selection process.

14.	The MCDA Software Selection Framework contains contradicting elements.	1 2 3 4 5 6 7
15.	All the elements in the MCDA Software Selection Framework are relevant for the representation of the MCDA Software Selection process.	1 2 3 4 5 6 7
16.	The MCDA Software Selection Framework gives a complete representation of the MCDA Software Selection process.	1 2 3 4 5 6 7

Part III: About yourself and your organization

17. Your gender

□ Male □ Female

18. Your highest level of education

□ Less than high school □ Undergraduate degree

□ High school degree □ Graduate degree

□ College degree □ Other

19. Your age

□ 20–25 □ 56–65

□ 26–35 □ 66–75

□ 36–45 □ 76–85

20. How many years have you worked in KDDM domain? _____

21. How many years have you worked for your current organization?

□ less than 1 year □ 10 - 15 years

□ 1 – 5 years □ more than 15 years

□ 5 – 10 years

22. How many years have you worked in your current position in the organization?

□ less than 1 year □ 10 - 15 years

□ 1 – 5 years □ more than 15 years

□ 5 – 10 years

23. Your job title is _____

24. Number of employees in your organization

□ Fewer than 500 □ 5,000–10,000

□ 500–999 □ More than 10,000

□ 1,000–4,999

25. In which industry is your organization operating?

□ Education □ Real Estate

□ Financial Services □ Services

□ Government □ Information Technology

□ Food/Beverage/CPG □ Telecommunications

□ Health Care □ Travel

□ Manufacturing□ □ Wholesale/Retail

□ Nonprofit □ Other, please specify _____

□ Medical, Bio-Technology, Pharmacology

VITA

Yan Li is a Ph.D candidate in the Department of Information Systems at Virginia Commonwealth University. Her research focuses on knowledge and data management areas such as data mining and semantic technologies, data warehousing management and business intelligence, decision support systems, and multiple criteria decision analysis (MCDA), with an emphasis on exploring the synergies between information systems and data analytics. She has presented at international, national and regional information systems conferences on topics such as strategic information quality management, decision support systems, geographic information systems, data mining, information systems development, and information technology for developing economies. She enjoys teaching data warehousing, business intelligence and advanced analytics to undergraduate and graduate students. She also holds a Data Scientist role in the world's largest media and technology company with hands-on experience in advanced analytics, data mining, and big data platforms. She will be joining Claremont Graduate University as a tenure-track Assistant Professor with a specialization in teaching and research in Data Science.

Prior to her doctoral studies, Yan was trained to be a scientist at the best science and technology university in China with a major in Chemical Physics. Driven by her intellectual curiosity for data and emergent information technologies, and her passion for designing and building things, she has oriented her career in the direction that integrates research, teaching, and practice in the realm of information science.