A GOAL AND ONTOLOGY BASED APPROACH FOR GENERATING ETL PROCESS SPECIFICATIONS

AZMAN TA'A

DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2012

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUMCollege of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

Abstrak

Pembangunan sistem gudang data (DW) melibatkanbeberapatugassepertipenentuankeperluan, merekabentukskema DW danmenetapoperasitransformasi data.Sesungguhnya, kejayaansistem DW adalahbergantungkepadakesempurnaanrekabentuk proses penarikan,perubahan, dan pemuatan(ETL).Walaubagaimanapun, masalahbiasa yang berkaitandenganrekabentuk proses ETL sepertipenentuankeperluanpenggunadan spesifikasitransformasi data sukaruntukdiselesaikan.Masalahiniadalahberkaitandengankepelbagaian sumber data, kekaburandalamkeperluanpengguna, dankerumitandalamaktivititransformasi data.Pendekatan semasa mempunyaikekangandalammenyesuaikan semantik keperluan DW kearahrekabentuk proses ETL. Akibatnya, hal ini telah melewatkan proses penjanaan spesifikasi proses ETL. Kerangkakerja semantik sistem DW yang dihasilkandaripadakajian ini digunakan untuk membangunkankaedahanalisiskeperluanbagimerekabentuk proses ETL (RAMEPs) daripada aspek perbezaan perspektif organisasi, pembuat keputusan, danpembangunan sistem denganmengggunakanpendekatanmatlamat danontologi.Ketepatan pendekatan RAMEPs telahditentusahkan denganmenggunakanperisian yang barudibangunkan dandiubahsuai.RAMEPs juga telahdinilaidalamtigakajiankessebenariaituSistem Hal EhwalPelajar, SistemUtiliti Gas, danSistemUsahawanSiswazah.Kajiankesinitelahdigunakanuntukmenunjukkanbagai mana pendekatan RAMEPs bolehdilaksanakandalammerekabentukdanmenjanaspesifikasi prosesETL.Tambahan pula, pendekatan RAMEPs telahdisemakoleh pakar DW untuk mengenalpastikekuatandankelemahannya dan pendekatan baru tersebut telah diterima. Kaedah RAMEPs berjaya membuktikan spesifikasi proses ETL boleh dijana dari fasa awal pembangunan sistem DW dengan menggunakan pendekatan matlamat-ontologi.

Kata Kunci: Analisis keperluan, ProsesETL, Gudang data, Ontologi, Kepintaran Perniagaan

Abstract

Data warehouse (DW) systems development involves several tasks such as defining requirements, designing DW schemas, and specifying data transformation operations. Indeed, the success of DW systems is very much dependent on the proper design of the extracting, transforming, and loading (ETL) processes. However, the common design-related problems in the ETL processes such as defining user requirements and data transformation specifications are far from being resolved. These problems are due to data heterogeneity in data sources, ambiguity of user requirements, and the complexity of data transformation activities. Current approaches have limitations on the reconciliation of DW requirement semantics towards designing the ETL processes. As a result, this has prolonged the process of the ETL processes specifications generation. The semantic framework of DW systems established from this study is used to develop the requirement analysis method for designing the ETL processes (RAMEPs) from the different perspectives of organization, decision-maker, and developer by using goal and ontology approaches. The correctness of RAMEPs approach was validated by using modified and newly developed compliant tools. The RAMEPs was evaluated in three real case studies, i.e., Student Affairs System, Gas Utility System, and Graduate Entrepreneur System. These case studies were used to illustrate how the RAMEPs approach can be implemented for designing and generating the ETL processes specifications. Moreover, the RAMEPs approach was reviewed by the DW experts for assessing the strengths and weaknesses of this method, and the new approach is accepted. The RAMEPs method proves that the ETL processes specifications can be derived from the early phases of DW systems development by using the goal-ontology approach.

Keywords: Requirement analysis, ETL processes, Data warehouse, Ontology, Business Intelligence

Acknowledgement

Alhamdulillah, syukur ke hadrat Allah s.w.t kerana dengan pertolonganNya tesis ini telah dapat disiapkan. Setinggi penghargaan ditujukan kepada penyelia, Dr. Mohd Syazwan Abdullah dan Prof. Madya Dr. Norita Norwawi di atas segala tunjuk ajar, nasihat, dorongan, pengalaman dan ilmu yang dicurahkan semasa menjalankan kajian dan menyiapkan tesis ini. Ucapan terima kasih juga ditujukan kepada pihak sekolah pengajian pengkomputeran dan pihak universiti di atas keperihatinan dan sokongan dan seterusnya pihak Kementerian Pengajian Tinggi kerana membiayai pengajian ini.

Tesis ini juga telah disiapkan dengan bantuan dan sokongan daripada teman seperjuangan, rakan sepengajian, rakan sekerja, rakan penyelidik, Hal Ehwal Pelajar UUM, Perpustakaan dan Pusat KomputerUUM.Terima kasih yang tidak terhingga juga kepada Syarikat Gas Malaysia, Unit Usahawan Kementerian Pengajian Tinggi, dan Institut SAS Malaysia yang telah memberi ruang untuk melaksanakan kajian ini dan membuka peluang yang lebih signifikan kepada pembangunan aplikasi yang berkaitan dengan bidang kajian ini.

Akhirnya, kejayaan tesis ini dihadiahkan kepada seluruh anggota keluarga, khususnya kepada ibu, isteri dan anak-anak yang turut sama mengharungi cabaran serta memahami perjuangan pengajian ini seadanya. Kepada ayah, abah, dan emak mertua yang telah kembali kerahmatullah, semangat kalian akan terus menjadi inspirasi kepada kejayaan ini dan yang seterusnya.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Acronym	Meaning
AAD	Academic Affairs Department
DAAD	Director Academic Affairs Department
AI	Artificial Intelligence
AKEM	Application Knowledge Engineering Methodology
ASIS	Academic Student Information System
BI	Business Intelligence
BISE	Business Intelligence for Student Entrepreneur
CCS	Call Center System
CDLNR	Conceptual Data Language for N Repositories
CEDI	Co-operative and Entrepreneurship Development Institute
CIF	Corporate Information Factory
COG	Corporate Ontology Grid
CS	Computer Sciences
CWM	Common Warehouse Meta-model
DAML	DARPA Agent Mark-up Language
DDL	Data Definition Language
DFM	Dimensional-Fact Model
DL	Description Logic
DM	Dimensional Model
DMS	Document Management System
DOME	Domain Ontology Management Environment
DSA	Design Science Approach
DSS	Decision Support System
DW	Data Warehouse
DW-Tool	Data Warehouse Tool
EAI	Enterprise Application Integration
EII	Enterprise Information Integration
EIS	Enterprise Information System
EKP	Enterprise Knowledge Portal
EPC	Event-Driven Process Chain
ER	Entity Relationship
ETL	Extract-Transform-Loading
EVE	Evolvable View Environment
GAIS	Graduate Academic Information System
GAV	Global-as-View
GRL	Goal-oriented Requirements Language
HR	Human Resources

IaR	Information as Required
IHE	Institute of Higher Education
IRS	Internet Reasoning System
IS	Information System
JDBC	Java Database Connectivity
JDE	J.D. Edwards System
KAON	Kalrsruhe Semantic Web and Ontology Infrastructure
KM	Knowledge Management
KST	Knowledge Sharing Technology
LAV	Local-as-View
MAKMUM	Majlis Keusahawanan Universiti-Universiti Malaysia
MAS	Multi-agent system
MDC	Metadata Coalition
MDM	Multidimensional Modeling
MER	Multi-Entity Relationship
MOF	Meta-Object Facility
MoHE	Ministry of Higher Education
MOMIS	Mediator Environment for Multiple Information Sources
NGDS	Natural Gas Distribution System
NIAM	Natural Language Information Method
NLP	Natural Language Processing
OBSERVER	Vocabulary heterogeneity resolution
ODBC	Object Database Connectivity
ODS	Operational Data Store
OIL	Ontology Inference Layer
OIM	Open Information Model
OLAP	On-Line Analytical Processing
OMG	Object Management Group
ONION	Ontology Composition.
OO	Object-Oriented
OODB	Object-Oriented Databases
ORDB	Object-Relational Databases
ORM	Object Role Modeling
OWL	Web Ontology Language
P2P	Peer-to-Peer
PROMPT	Formalism-independent algorithm for ontology merging and alignment
RDF	Resource Description Framework
RDF-S	Resource Description Framework Schema
RFI	Request for Information
RO	Response Obtained
SA	Staging Area

SCD	Slowly Changing Dimension
SEAL	Semantic Portal
SIM	Semantic Information Management
SPARSQL	Protocol and RDF Query Language
SQL	Structured Query Language
STS	Socio-Technical System
TA-Tool	Transformation Analysis tool
UBIS	Utility Billing Information System
UCM	Use Case Maps
UML	Unified Modeling Language
UMIS	University Management Information System
UniMAP	Universiti Malaysia Perlis
URN	User Requirements Notation
UUM	Universiti Utara Malaysia
WFMS	Workflow Management Systems
WWW	World Wide Web
XMI	XML Metadata Interchange
XML	Extensible Markup Language
XOL	eXtended Ontology Language

CHAPTER ONE–INTRODUCTION

This chapter presents the background and motivation of this research. The chapter defines the research problems and the research gaps, as well as the research questions and research objectives. Then, the research strategy is discussed in three phases, followed by the scope and the research contributions. This chapter ends withan overview of the thesis organization and summary of the thesis.

1.1 Background

The trend of Business Intelligence (BI) system utilization for decision-making and monitoringperformance (e.g., Key Performance Indicator - KPI) has increased tremendously. The BI Verdict (formerly known as the OLAP Report) (2006)[1] reported that the On-Line Analytical Processing (OLAP) market grew from one billion US dollars in the year 1996 to 5.7 billion US dollars in the year 2006. The industry analyst firm, IDC, predicted that the business analytics software will grow by 10.3 percent annually through the year 2011[2]. This prediction is in line with the market survey conducted by BetterManagement[3], which showed that 84 percent of various organizations were using BI systems. Indeed, many studies conducted by researchers and practitioners have shown increasing use of the BI system by small, medium and larger organizations.

[1] http://www.bi-verdict.com/index.php?id=122 (Previously known as olapreport.com)
[2] http://www.oracle.com/corporate/analyst/reports/infrastructure/bi_dw/208699e.pdf
[3] http://www.bettermanagement.com/default.aspx

Most of the BI systems are developed based on the Data Warehouse (DW), which involves a labor-intensive workflow known as Extract-Transform-Loading (ETL). In other words, the success of a BI system is dependent on the performances of the DW and ETL processes. Indeed, the ETL processes are an important part of the DW systems components for gathering, modeling, storing, processing, and analyzing huge amounts of data using BI tools. These data are accessed, processed, and stored in centralized databases by the appropriate DW technology, and designed in order to provide the right information for the decision-makers. However, the DW systems is dependent on the ETL processes (previously known as DW operational processes) to process the data for the decision-makers. Therefore, the success of the DW systems relies heavily on the design of the DW structure and ETL processes.

ETL is a series of extracting, transforming and loading processes of data sources for the required DW modeling. There are several issues on modeling and designing the DW and ETL processes proposed by researchers or practitioners through their own methods and tools. Most of these issues are related to the difficulty in anticipating requirements for decision-makers, inefficiency of data flow and loading, and generating the data integration and transformation process. These are attributed to the DW systems, which are characterized by elements of complexity of requirements and heterogeneity of data stores. These challenges start from understanding the user needs, preparing the required data, and providing the information according to the format as specified. Hence, the relevant design tasks within the conceptual design framework are needed to tackle the challenges in a consistent, repeatable and systematic manner. Therefore, the design tasks should start from the early phases of

a requirement process until the final definition of ETL processes specifications. Therefore, the design tasks in the early phases of DW and ETL system development are important in ensuring the satisfaction of information delivery to the users.

1.2 Motivation

The ETL processes are one of the important components in DW systems development, which consumes 70 - 80 percent of the resources needed (Kimball & Caserta, 2004; Lujan-Mora, 2005). The success of ETL processes is very much dependent on the data integration and transformation process that deals with the semantics or terminology reconciliation of data sources and user requirements (Halevy, 2005; Schreiber, 2003; Skoutas & Simitsis, 2006). The executive survey on utilizing BI systems conducted in the United States and Europe in 2004 had identified that 49 percent of the executives claimed difficulties in getting relevant company data to make accurate decisions, and 77 percent of them were aware that the business managers have made bad decisions because of the insufficient information provided to them (Hammond, 2004). The majority of the business managers blamed the difficulties on the confusing file-name definitions and most of the information being scattered across large and decentralized locations.

The main motivation of the present study is driven by the problem of file-name definitions confusion, which normally refers to the conflict of semantics definition in the data store (i.e., data sources, or DW) (Goh, 1997; Kimball & Caserta, 2004). These types of problems can be identified as semantics conflicts, or heterogeneity problems within the data integration and transformation process. The definition of

semantics refers to the meaning of data in a business sense (An, 2007; Goh, 1997; Kimball & Caserta, 2004), whereas the heterogeneity refers to different systems that have presented the data (An, 2007; Goh, 1997). Thus, semantics heterogeneity refers to data representation in a different system that might have the same meaning, even when the sources of the data model are different. The ambiguous definition of semantics for each of the data sources schema (i.e., such as an attribute, table or constraints name) not only creates problems for end users, but also for the ETL designer in designing the ETL processes (Kimball & Caserta, 2004; Skoutas & Simitsis, 2006). Moreover, an acceptable decision-making process is driven by the reliability of the right meaning and definition of the business data (Kimball & Caserta, 2004).

The complexity of ETL processes always refers to the problem of generating the transformations of data sources toward the DW model. These transformations involve the semantic reconciliation of user requirements and data source schemas toward the predefined DW schemas (Alexiev et al., 2005; Kimball & Caserta, 2004). An ambiguous definition of user requirements occurs due to the inability of the users to define their requirements precisely and clearly (Inmon, 2002). Moreover, multiple meanings of data sources (i.e., attributes, tables) make it difficult to integrate, just because of the need to satisfy the user requirements. Thus, reconciliation of the appropriate semantics of user requirements and data sources is crucially important for generating the data transformations in an appropriate manner.

Generating the data transformations refers to the designing of the ETL processes from the early phases of DW systems development. This should be based on the systematic method of analyzing the user requirements towards generating the ETL processes specifications accordingly. Possibly, the generation of ETL processes specifications can be done automatically or at least in a semi-automatic manner. The systematic method for analyzing user requirements always emphasizes the determination of business goals by the stakeholders of the organization. Organization goals will produce meaningful requirements that can certainly determine the focus, scope, and alignment of the DW systems. This organization goals analysis initiative must be given the main priority in the design tasks. However, a suitable method is needed to represent the business semantics with the corresponding data sources that are derived from the requirement analysis tasks.

Recently, the emergence of ontology in dynamic knowledge representation has attracted researchers to explore the opportunities in providing solutions for enhancing and improving the DW systems. Some researchers have explored the application of ontology in BI system development, such as automated multi-dimensional design (Romero & Abelló, 2007), ontology-based BI systems (Cao, Zhang, & Liu, 2005), and ETL processes design (Skoutas & Simitsis, 2007; Tieniu, Jianhua, Haihe, Yinglin, & Tianrui, 2011). However, current DW systems are difficult to design due to several problems that are related to handling the user requirements and related data sources. The user requirements are so difficult to be specified, anticipated, and fulfilled (Berenbach, Paulish, Kazmeier, & Rudorfer, 2009; Winter & Strauch, 2004). Data sources are very much unstructured,

distributed, heterogeneous, dynamic, and have complex database structures that are provided by the client-based or web-based applications (Lenzerini, 2002; Ponniah, 2007). Most of these problems are related to the data integration and transformation processes that are orchestrated by well-defined user requirements. However, current methods in ETL design are incomplete due to the limitations and linkages in modeling and designing the DW systems. Most of the DW design methods focus on defining the DW modeling (Inmon, 2002; Kimball, 1996; Rizzi, 2007). Clearly, these limitations have contributed to the failure of most of the DW projects (Giorgini, Rizzi, & Garzetti, 2008; Hwang & Xu, 2007; Lujan-Mora, 2005).

1.3 Problem Statement

The modeling and designing of ETL processes are essential for developing DW systems successfully. Several traditional approaches have focused on the modeling of DW structure (Golfarelli, Maio, & Rizzi, 1998; Inmon, 2002; Kimball, 1996; Lujan-Mora, 2005; Rizzi, 2007; Sapia, Blaschka, Hofling, & Dinter, 1998). However, these methods do not address the problems of business semantics reconciliation and semantics heterogeneity problems during designing of the ETL processes. A few researchers have extended the design of ETL processes due to the difficulty and complex nature of data integration and transformation (Lujan-Mora, 2005; Papastefanatos, Vassiliadis, Simitsis, & Vassiliou, 2009; Simitsis, 2004). However, the business semantics were not reconciled with the related data sources at the design tasks, which have led to the confusion of the data transformation activities.

A detailed knowledge of data sources is needed in order to guarantee the data transformation process (Giorgini et al., 2008; Simitsis, 2004). The ETL processes specifications require mapping of the attributes of the data sources to the attributes of the DW structure. However, due to the business semantics reconciliation and the heterogeneity problems, the tasks to design and develop the ETL processes become difficult, tedious and complex. Business semantic requirements are derived from the analysis of user requirements. The traditional way to analyze user requirements is based on the principle of requirement engineering that is not efficient in identifying requirement definitions and tends to increase the possibility of misunderstanding the user requirements (Bresciani, Perini, Giorgini, Giunchiglia, & Mylopoulos, 2004; Giorgini et al., 2008). These problems require a solution to addresses the problems of semantic heterogeneity and possibly automate the design of the ETL processes. This could support the efforts in bridging the communication gap between DW developer and organizations (Stefanov & List, 2005).

Since most of the DW is based on the relational structure, this research focuses on three major data conflicts namely, syntactic, structural and semantic. These types of conflicts commonly occur in relational databases, and the research case study addresses this.Normally, the type of semantic heterogeneity is synonyms or homonyms. A synonym is about two pieces of information with the same meanings, but referring to the different name, while a homonym is the reverse. Clear definition and understanding of the semantic data sources are very essential for semantic reconciliation and avoiding terminology inconsistencies in information-sharing environments (Alexiev et al., 2005; Schreiber, 2003).

Current methods for defining and maintaining the ETL processes specifications are tool-driven, and rely on the proprietary functionalities of the tools. The problem with tool-driven is, when the user requirements or data sources change, the transformation activities, such as filtering and conversionneed to be updated accordingly. These tasks are error-prone and time-consuming (Alexiev et al., 2005; Halevy, 2005; Kimball & Caserta, 2004).These problems can be solved by developing a method for designing the ETL processes from the early phases of DW systems development. The proposed design method should be able to derive the ETL processes specifications and address the business semantics reconciliation and semantic heterogeneity problems during the ETL processes design. The primary components in this research are the user requirements, and the ETL activities that underlie the two problems to be solved. Other components are also important for the whole process of DW systems development. However, this research works focus on ETL processes design issues that are relevant to the activities of requirement engineering of DW systems. Therefore, this research fills the gap between data warehousing and model and design engineering by developing a method for designing the ETL processes beginning from the early phases of DW systems development. This gap, as represented by "X", is illustrated in Figure 1.1.

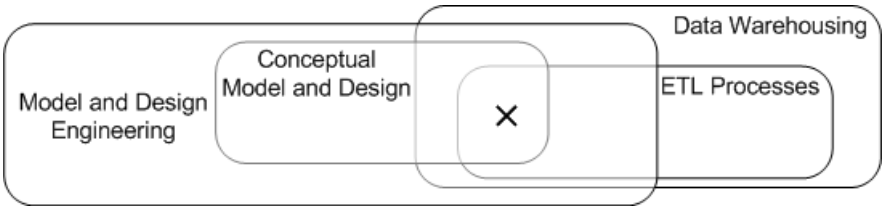


Figure 1.1: Research Gap

1.4 Research Questions

Due to the problems in DW systems development explained in Section 1.3, the process to design the ETL processes in the DW environment must be improved and enhanced. A systematic method for modeling and designing the ETL processes must be developed in order to address the design-related issues in ETL processes design. Therefore, the main research question is:

Can the ETL processes be designed from the early phases of DW systems development?

In detail, the main research question can be divided into four sub-questions as follows:

i) *Can the goaland ontology be utilized for analyzing the requirements of the ETL processes in the early phases of DW systems development?*

ii) *How can the ETL processes be designed by using goaland ontology within the semantic framework of DW systems development?*

iii) *How can the ETL processes specifications from the goaland ontologyapproach be generated?*

iv) *How can the goaland ontologyapproach be validated and evaluated?*

This research is an attempt to answer all the research questions by exploring the early phases of the DW requirement analysis method with the case studies

characterized by various kinds of heterogeneous problems in the data source environments. The objectives of this research are presented in Section 1.5.

1.5 Research Objectives

The goal of this research is to facilitate, manage, and enhance the design of the ETL processes from the early phases of requirements toward the ETL processes specifications generation and continuous evolution of the DW systems development. Thus, this research presents a method for designing the ETL processes from the early phases of DW systems development. The novelty of this method is in applying the goal and ontology approach during the early phases of DW systems development to generate the ETL processes specifications. Therefore, in particular, the research objectives can be divided into:

i) *To define the semantic framework of DW systems developmentfor guiding the ETL processes design.*

ii) *To develop the requirements analysis approachby using a goaland ontology for designing the ETL Processes.*

iii) *To develop an algorithm and demonstrating the process to generate the ETL processes specifications.*

iv) *To validate and evaluate the approach by using compliant toolsand applying to real case studies.*

Each of the research objectives is achieved through the research strategy as discussed in Section 1.6.

1.6 Research Strategy

The research strategy comprises three phases, where each phase achieves the research objectives and answers the related questions as illustrated in Figure 1.2. Phase I is to understand and identify the problems of the research domain by exploring the issues of DW and ETL processes design. The problems on data conflicts, semantic heterogeneity of DW requirements, and ETL processes generation are framed and focused and a suitable DW development framework is introduced. The DW components in the framework interact with each other in order to complete the lifecycle of ETL processes development.(Zhuolun & Sufen, 2008).

The design process is based on a generic ETL processes modeling (Kimball & Caserta, 2004; Lujan-Mora, 2005; Simitsis, 2004), where the proposed method focuses on the business requirement reconciliation and handling of the semantic heterogeneity of the data sources. As a result, Phase I determines the research problems and highlights the relating research works. Phase I has six activities labeled in sequence from 1.1 to 1.6, and delivery is a semantic framework of the DW systems development within the notion of semantic characteristics for modeling and designing the ETL processes successfully.

In Phase II, the significant problems in ETL processes design are further investigated by focusing on two major tasks: i) analysis DW requirements for semantic

reconciliation and data conflicts, and ii) data integration and transformation activities for generating the ETL processes specifications. Delving into a goal-oriented approach on requirement analysis and an ontology approach for modeling the ETL processes are carried out. The concepts of goal, actor, and plan in goal-oriented approaches and class, property, and axiom in the ontology approach are applied in the design method. To achieve the goal of the proposed design method, a workable research methodology is utilized. Phase II has four activities labeled in sequence from 2.1 to 2.4, and the requirement analysis method for ETL processes (RAMEPs) is the delivery in Phase II.

In Phase III, the focus is on validating and evaluating the RAMEPs for the ETL processes design. The compliant tools that comply with the goal (i.e., OME and DW-Tool) and ontology modeling (i.e., Protégé-OWL) are used to check and verify the ETL processes model. The new modeling approach of the ETL processes are used in various case studies. The entire process of the analysis tasks are carried out using RAMEPs and evaluating their results for each of the case studies. Phase III has four activities labeled in sequence from 3.1 to 3.4. The ETL processes designs and ETL processes specifications for each of the case studies are the delivery in Phase III.

A prototype of the ETL processes generation system is developed to generate the ETL processes specifications from the ETL processes model that has been designed. This prototype was developed using the Jena 2 framework, which runs on the Java Eclipse platform. The validation and evaluation results are supported by the

prototype of the ETL processes generator to generate the ETL processes specifications.

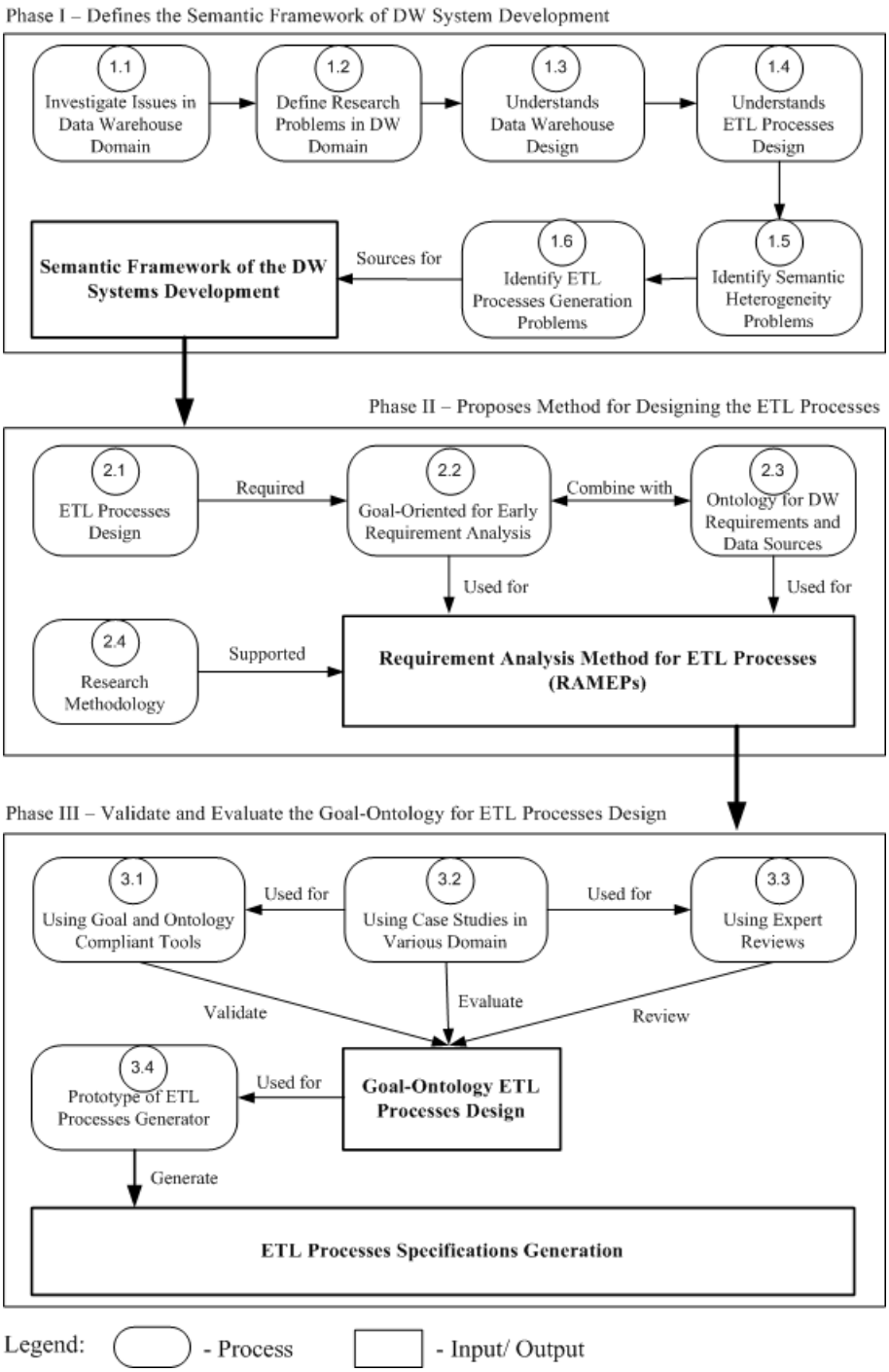


Figure 1.2: Research Strategy

1.7 Scope of the Research

The ETL processes design are organized in six components as presented in the semantic framework of the DW systems as depicted in Figure 4.8. In order to design the ETL processes, this research needed to understand all the component functions, and how the user requirements and data sources model are presented and related to each of the framework components. Furthermore, how the ontology model of data sources is mapped with the user requirements, and produces the ETL processes specifications that comply with the DW system requires further investigation.

The semantic framework for DW development is derived from the extensive review of literature in these areas. It is based on the relationships between six components: i) the business requirements, ii) the data sources, iii) the ontology sources, iv) the ETL processes, v) the staging area, and vi) the DW. All these components are covered in the design process. The design tasks are at the conceptual level and deal with the functional requirements (FR) of the DW system. The nonfunctional requirements (NFR) are not given attention in this study. The requirements analysis model is mapped to the ETL processes model for designing the ETL processes specifications. The ETL processes specifications are used to produce the DW for providing information to the end users.

Therefore, the scope of this research comprises analysis of user requirements from the early phases until deriving the ETL processes specifications for supporting the implementation of the DW systems. Nevertheless, the execution of the ETL

processes specifications is out of the research works because the requirement analysis only focuses at the data source schemas.

1.8 Research Contributions

The focuses of this research is mainly to develop an approach for designing the ETL processes, and finally helping the ETL developer to define the ETL processes specifications with the related DW structures, which is produced simultaneously. The proposed approach shows the integrated phases of ETL processes design methodology. Therefore, the summary of contribution of this research can be highlighted as follows:

i) The comparative analysis of the ETL processes design approaches that provides the summary of the research works in DW and ETL processes requirements analysis approaches.

ii) The provision of a new approach in designing the ETL processes from the early phases of DW systems development. The approach systematically reconciles the business semantic of DW requirements toward the data sources and resolves the semantic heterogeneity problems during the ETL processes design.

iii) The provision of ETL processes specifications from the ETL processes design automatically through ETL processes generation application to facilitate the implementation of ETL processes in the DW systems.

iv) Method for model checking with modified and newly developed compliant tools. The method is used to verify the correctness of the goal and ontology model at a design stage.

v) The bridging of the communication and knowledge gaps between ETL developers and business users by putting the right perspective of requirements in DW systems. On the other hand, the business and data sources semantics are closely intact.

vi) The building of ontology for DW requirements provides new centralized DW glossaries for ETL developer references. The diversification usage of ontology in various domains of software systems, especially in the DW systems will strengthens and matures the ontology utilization.

vii) The documentation of ETL processes and DW systems during the design tasks, which are systematically documented and organized. The used of newly extended concepts and notations of the i* framework will benefit the end-users, ETL developers, and organizations for future reference.

1.9 Thesis Organization

i) Chapter 1 – Introduction

This chapter provides an overview of the thesis. It explains the research background and motivation for conducting the research, the research problems, the research gaps,

the research questions, the research objectives, the research strategy, scope of research, and the research contributions.

ii) Chapter 2 – Data Warehouse and ETL Processes Design

This chapter explains the function of DW and ETL processes. The concepts of DW and ETL processes are elaborated with related works on modeling and designing approaches. The problems that occur in ETL processes design is highlighted together with the issues that need to be tackled. Moreover, the standard, modeling language and supporting ETL tools are discussed in order to comply with the current technology available.

iii) Chapter 3 – Requirement Analysis for ETL Processes

This chapter explains the requirement analysis process for the ETL processes design. The concepts of software requirement development are introduced and rationalized with the DW and ETL processes requirements. In detail, the ETL processes requirements from the early phases of DW systems development are explained. Moreover, the organization, decision, and developer perspectives that founded on the requirement analysis approach are introduced with supporting on the related theories.

iv) Chapter 4 – Ontology for the ETL Processes Model

This chapter explains the concept of ontology and how the ontology-based approach can be used in modeling and designing the ETL processes. The language and tools used in ontology development are highlighted with the concepts of ontology classification. Ontology approach in ETL processes modeling is presented and the related ontology-based works for modeling and designing the ETL processes are elaborated.

v) Chapter 5 – Research Methodology

This chapter describes two types of methodology used that complement each other for guiding the research tasks to be carried out systematically. These methodologies present the methods to be used in conducting the research to achieve the research objectives for designing the ETL processes in DW systems. Nevertheless, the validating and evaluation processwere introduced by using compliant tools and case studies respectively.

vi) Chapter 6 – Requirement Analysis Method for ETL Processes Design (RAMEPs)

This chapter discusses the RAMEPs for modeling and designing the ETL processes, which emphasize the goal-oriented approach for analyzing the user requirements, and the ontology approach for modeling the DW requirements and data sources. The requirement analysis process was explained in steps as RAMEPs tasks and the merging process for DW requirements and data sources are explained in detail.The ETL processes generation algorithms weredeveloped for generating the ETL processes specifications.

vii) Chapter 7 – Validation and Evaluation of RAMEPs

This chapter presents the validation and evaluation process of RAMEPs. The correctness of the RAMEPs model has been validated by using goal and ontology compliant tools. Then, the RAMEPs approach was evaluated in three different case studies and expert reviews. The evaluation findings were discussed in the context of ETL processes design for DW systems. Furthermore, the generation of the ETL processes specifications wasdemonstrated by application prototype.

viii) Chapter 8 – Conclusions and Future Works

This chapter reviews all the findings and concludes the research work by examining the research objectives. The summary of research works was discussed toward the contributions of the study. The main contributions are aligned with general contributions as presented in chapter 1. Finally, the limitations and future work for designing the DW and ETL processes are highlighted.

1.10 Summary

This chapter introduces the background and presents the motivation of the research, and outlines the problems that need to be investigated. It also rationalizes the research domain problem within the approach that needs to be used. The aim of this research is to enhance the design of ETL processes and help the ETL developer to produce the ETL processes specifications in an automated manner. The use of goal-oriented and ontology aims to resolve the semantic heterogeneity problems by reconciliation of the definition of business semantics underlying the user requirements, supported by the DW and data sources modeling. Chapter 2 discusses the literature related to DW and ETL processes design in detail.

CHAPTER TWO–DATA WAREHOUSE AND ETL PROCESSES DESIGN

This chapter presents the literature review related to DW and ETL processes design. The concepts of DW and ETL processes are discussed, which elaborate the research works on modeling and designing. Related works on ETL processes are highlighted that elaborate the ETL processes designing. The chapter ends with a discussion on the standard, modeling language, and ETL tools that are important in the ETL processes design.

2.1 Data Warehouses

DW is a special database that collects and manages the business transaction data into a high level of abstraction to provide information for decision making by the organizations. The most acceptable definition of DW is provided by Inmon (2002): *"A Data Warehouse issubject-oriented, integrated, time-variant, non-volatile"*, and Kimball and Caserta (2004): *"A Data Warehouse is a system that extracts, cleans, conforms, and delivers source data into a dimensional data store and then supports and implements querying and analysis for decision making".*

This definition has popularized the concept of a dimension model in DW design that was widely accepted by the DW community. In addition, a dimensional model has encouraged the approach to data analysis and supports the information neededfor decision making. With support from various vendors and technologies in data extraction, integration, and transformation, the organization will be able to develop the DW systems systematically.

Typically, the architecture of DW systems consists of five components: i) user requirements; ii) data sources; iii) ETL processes; iv) DW structure; and v) BI applications. These components are known as the *back room activities* (user requirements, data sources, ETL processes and DW structure) and *front room activities* (BI applications) (Kimball & Caserta, 2004; Simitsis, 2004). The phases between components show the transition of data throughout the workflows of DW processes as illustrated in Figure 2.1.

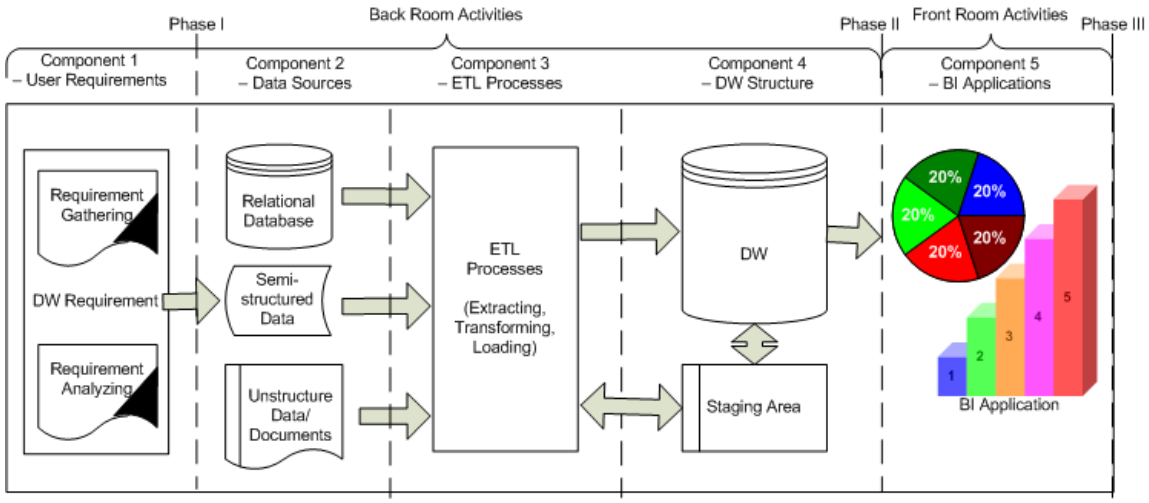


Figure 2.1: Typical Architecture of DW Systems (Kimball & Caserta, 2004)

The DW systems begin in phase I with the elicited and analyzed user requirements (component 1) toward the information as required. In the phase II, the data sources (component 2) [normally in heterogeneous environments] that comes from a relational (e.g., application system), unstructured (e.g., work documents), and semi-structured database (e.g., XML documents) are selected and extracted. The next component (component 3) is known as the ETL processes, which manipulate the

selected data sources through extracting, integrating, cleansing and transforming functions in a permanent storage of *staging area* (SA). When the data is ready for the process of manipulation, the data is transferred or loaded into a DW structure (component 4). All the ETL processes are done by the specific tools that have the capabilities to integrate the heterogeneous data sources and perform the processes of data manipulation as defined by the developer.

The DW becomes the repository of data and keeps the current and historical record in an organized manner. As mentioned earlier, DW is the high-level information that refers to the highly aggregated form of data which become sources of *data marts* or *data cubes* functionality. The process of DW systems ends with the delivery of the DW to the end users by the BI applications, shown as the next component (component 5). A BI application is a front-room activity and is basically based on OLAP or Data Mining techniques. These techniques are available in BI tools for reporting and data visualization. Clearly, the high-level of a data model that tightens up the user requirements and ETL processes initiatives are the appropriate communication tool for developing the DW systems.

2.2 Data Warehouse Design

Database (i.e., relational model) or information modeling is a method that captures the contents, relationships and constraints of data that reflect the user requirements at various description levels of the schemas. The modeling tasks are carried out by the modeler (i.e., DW developer) and domain expert (i.e., users in the organization) as a collaborative activity between them to produce a required database model (Halpin,

2001). Database model is an output from the modeling processes that can be defined as *"an integrated collection of concepts for describing and manipulating data, relationships between data, and constraints on the data in an organization"*(Connolly & Begg, 2005). A concept is a representation of organization activities and modeling in a way of communication among desired structure and behavior, visualizes and control, better understanding, and managing the risk of the system(Booch, Rumbaugh, & Jacobson, 1999). Therefore, it has a different approach in modeling and designing due to the hugeness and complexity of the DW structure and analysis. This approach leads the strategy to achieve the DW design goal, which is to provide the needed information for decision makers.

2.2.1 Modeling Approach

Normally, the structure of the database is defined during the design tasks of a database system. In practice, the design approach is carried out by user application needs in order to satisfy the organization's information needs. The application needs are derived from the business process that is implemented by individuals in each department. The database design focuses on complying data for the transactions of the business processes in three modeling approaches, namely conceptual, logical and physical. A similar approach is adopted in modeling the DW with visualizing the database as *a cube* that is surrounded by the *dimensions* and *measurements* views. These dimensions and measurements are created within the scoping of an information subject required by the users (Ponniah, 2007).

2.2.2 Dimension Modeling

In DW, the conceptual data model represents the important entities and the relationshipbetween the grounded *fact* and *dimension* structure is widely accepted in modeling and known as *dimensional modeling* (DM) or *multidimensional modeling* (MDM) (Kimball, 1996; Ponniah, 2007; Rizzi, 2007). The graphical view of DM is shown in Figure 2.2.

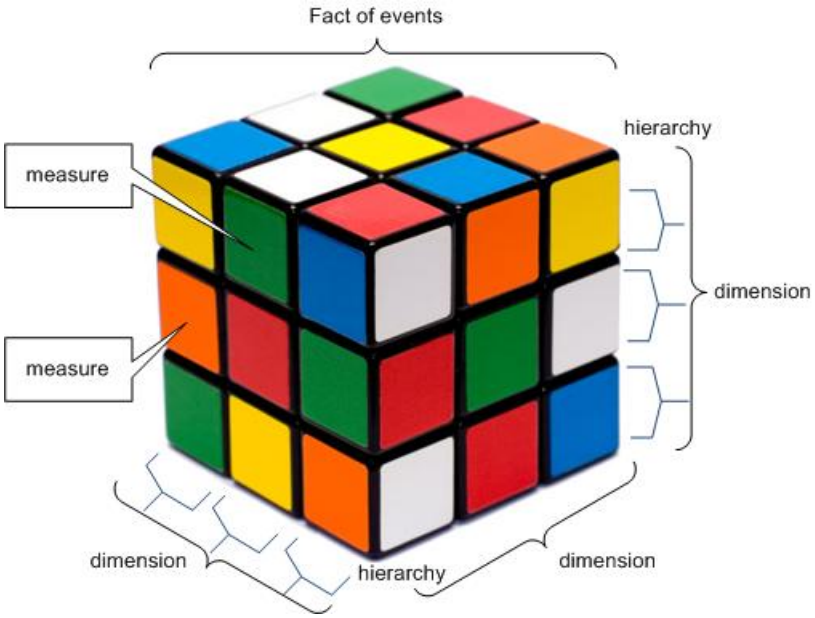


Figure 2.2: The Cube formed of DM

The concept of business dimension is a key definition for DW systemsdata modeling. Then, the business dimension can be understood as the required information that is derived from the sets of events in the real world (Rizzi, Abello, Lechtenborger, & Trujillo, 2006). This concept is known as *fact*, whichcontains the *measures* on specific requirements of the users and *dimensions* that provide the

description about the measurement in the modeling. The dimensions can be determined in the hierarchy that has a relationship among the attributes.

The DM is developed to support the conceptual and logical data model in order to describe the DW in as much detail as possible without considering how to implement it in physical DW systems. At this level, all the entities, relationships, attributes, data types, primary keys, foreign keys, and surrogate keys are captured. Finally, the whole specification of all entities based on the implementation platform (database, servers, etc.) is defined in the physical data model for the implementation of the DW systems. The main entities of DW modeling based on Dimensional Fact Model (DFM) (Golfarelli et al., 1998; Rizzi, 2007)are discussed in Section2.2.2.1.

2.2.2.1 Fact Definition

The *fact* is a focus on a set of events occurring in the real business world aimed to provide the information for the decision-maker. The fact is represented by the box with two sections, one for the fact name and one for the measures. The fact model is shown in Figure 2.3.

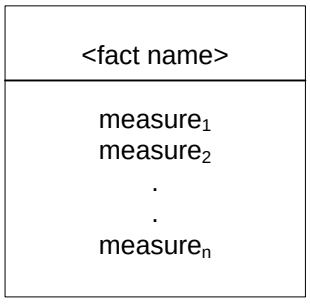


Figure 2.3: The fact model

2.2.2.2 Measure Definition

The *measure* is a non-alphanumeric property of fact that describes the amount determined by the aggregation functions of the analysis. The fact holds the measurements and it is modeled as a fact row as shown in Figure 2.3. However, if the measure cannot exist, then it is called a *factless fact*.

2.2.2.3 Dimension and Attribute Definition

The *dimension* or *attribute* is a fact property describing the context of fact and all the measurements defined in the DW domain. The set of dimensions determines the *grain* of a dimension. The grain of dimension describes the key of the dimension in business terms. In analyzing the task, the developer will ensure the set of data sources corresponds to the grain of dimension. The dimension or attribute model is represented as circles attached to the fact by lines as shown in Figure 2.4.

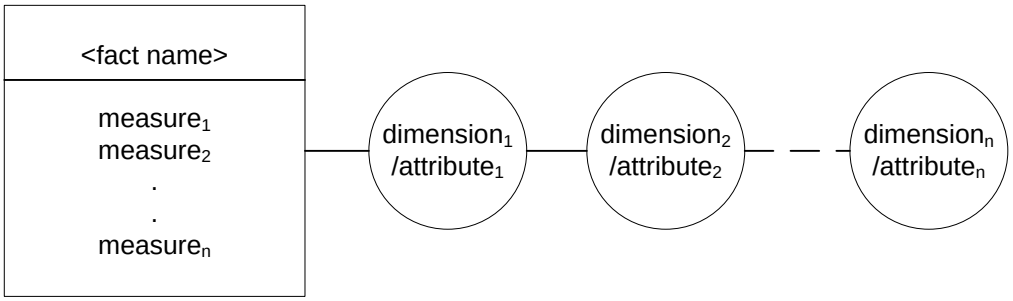


Figure 2.4: The dimension model

2.2.2.4 Hierarchy Definition

The hierarchy is a directed tree model that describes the dimension from the attributes of linked dimensions to give a meaning for the fact. The relationship of

dimensions can be in many-to-one or one-to-many. This gives a detailed explanation about the related fact. According to Kimball and Ross (2002), this type of hierarchy has produced a variant of DM to accommodate information as required by the users. The hierarchy attributes or dimensions are illustrated as straight lines in Figure 2.5.

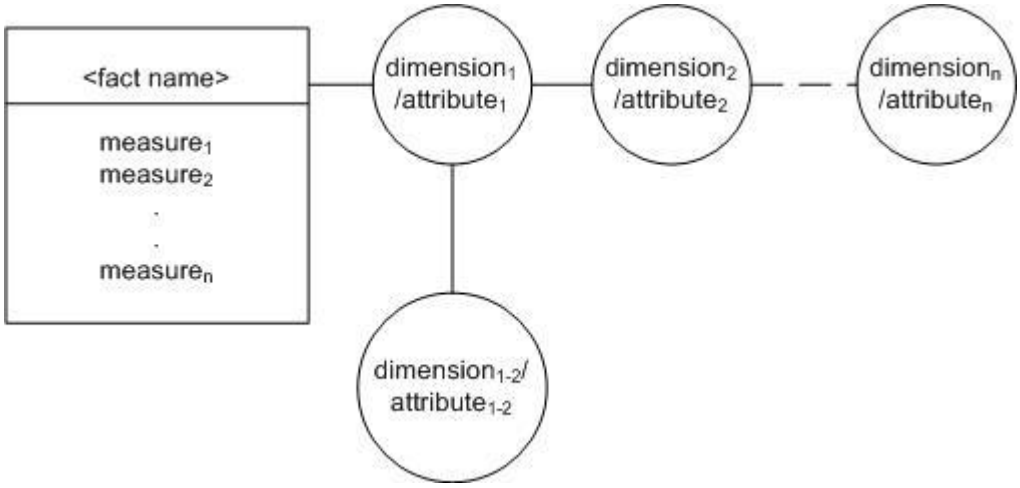


Figure 2.5: The hierarchy model

2.2.3 Research Works in DW Design

Currently, various data models for DW design have been proposed, whether in conceptual, logical or physical forms(Golfarelli et al., 1998; Husemann, Lechtenborger, & Vossen, 2000; Inmon, 2002; Kimball, 1996; Lujan-Mora, 2005; Rizzi, 2007). All the models were developed using various modeling languages such as Entity Relationship (ER) or UML that are implemented in a particular methodology. Until now, the DW design methods proposed by researchers or practitioners do not have a standard that is agreed uponby the community. Inmon (2002), as father of the DW, proposed the ER approach in DW design based on the

concept of a corporate data model for enterprise DW. However, this approach does not properly model the ETL processes as there are limitations of ER formalism in supporting the complexity of the corresponding DW models (Lujan-Mora, 2005).

Kimball and Ross (2002) presented the approach of DW design as a subject-area based model or *Data Mart*. This approach known as *star schema* identifies the fundamental principals in DM. Furthermore, with the architecture of *bus matrix* technique, the integration of data mart models was proposed towards the enterprise DW model. However, these works also have not focused on the ETL processes design, and only explains the ETL processes as *back room* activities. Golfarelli et al.(1998)proposed the DFM that focuses on the DW conceptual and logical design with their own notation. Again, his works also do not consider the design of ETL processes in DW design approach.

Jacky, Isabelle and Nicolas (2001)proposed the multidimensional approach as an aggregation and generalization hierarchies model by using UML for designing the DW systems. However, the research does not study in detail the design level, especially the part on ETL processes. Thus, there are no workable approaches that can be applied by DW developers for producing the ETL processes model by using this approach.

Lujan-Mora (2005) proposed almost a complete cycle of DW design based on UML profile. The data model (conceptual, logical and physical)was presented together with the model of ETL processes. However, this approach is still incomplete as it

does not consider the user semantic reconciliation in user requirements and semantics heterogeneity problems in data integration and transformation, which are essential issues that need to be tackled in the ETL processes design. Mazon, Trujillo, Serrano, & Piattini (2005)and Giorgini et al.(2008) treated the issues of DW requirements by proposing the methods for requirement analysis methods to support the design of DW systems. However, these methods focused on the dimension or multi-dimension schemas that modeled the DW structure. As such, not much attention was given to the ETL processes design phase.

Therefore, it can be concluded that some of these methods on DW modeling have been widely implemented by the DW community. Nevertheless, it is still plagued with problems. Part of the problems are related to not using the standard modeling language; other problems are related to not having clear explanation on the steps of the process; and the most significant weakness is that the whole processes involved in DW components, especially in ETL processes, are not addressed properly. As a consequence, most of the DW vendors developed their own proprietary methods to design the DW model together with ETL processes in order to integrate these methods with the associated operational system efficiently (Rudin & Cressy, 2003). However, this propriety method lacks the understanding of DW requirements, which creates difficulties to the DW developers to use it with the ETL tools that are implemented in various platforms (Rizzi et al., 2006). The research in ETL processes or theme of data integration, data cleaning, data workflow, and data transformation are discussed in Section2.3.

2.3 ETL Processes

ETL is a series of processes for the integration and transformation of data sources to the required data format (e.g., DW, Data Mart, OLAP, ODS) for a tactical or strategic information system. It is defined as the *inflow* data management (Connolly & Begg, 2005) and these processes are very important components in back room activity of DW systems development. It is highly recognized that well-designed and well-maintained ETL processes are key factors for accelerating a successful DW systems development (Hwang & Xu, 2007; Moss, 2005; Simitsis, 2004).

In general, the ETL processes transform and integrate the selected data sources into the DW schemas that fulfill the business requirements. Normally, the selected data sources are in a heterogeneous environment that consists of various schemas and structures. Based on the different schemas, the ETL processes execute the process of extracting data sources, transforming data sources to the DW structure and finally loading the transformed data sources into the DW. The transformation process can be defined as cleaning and conforming activities (Kimball & Caserta, 2004), and is presented as a data flow in Figure 2.6.

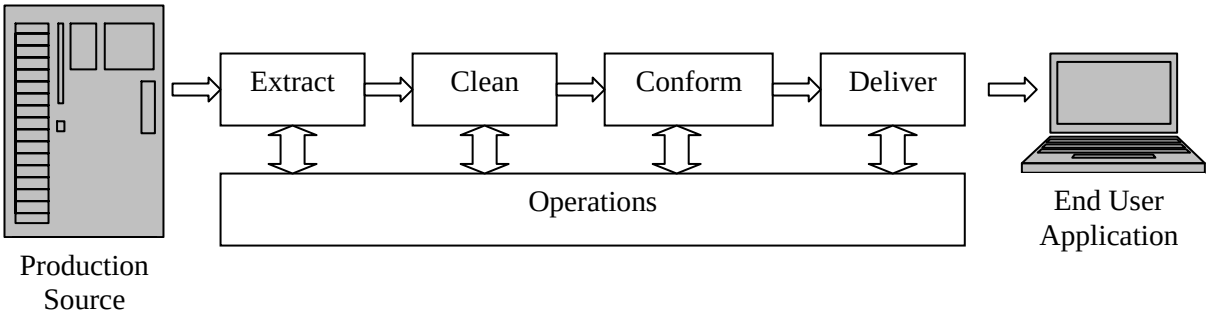


Figure 2.6: The Data Flow in ETL Processes (Kimball & Caserta, 2004)

According to Simitsis (2004), ETL is a tool responsible for the extraction of data from several sources, to do cleansing, customization, and transformation, and finally loading the data into the DW in order to fit the business needs. Furthermore, functions of the tool are mainly to: i) identify the relevant information at data sources; ii) extract the information; iii) transport the information to the data staging area; iv) transform the information into a common format; v) clean the result of data set based on database and business rules; and vi) propagate and load the data to the DW and refresh the data marts.

All these tasks can be viewed as general ETL framework defined by Simitsis (2004) as shown in Figure 2.7. The ETL processes start with a selection of the data sources and extracted to data staging area (DSA) where the process of transformation and cleaning the data sources take place as shown in left and middle part of Figure 2.7. Then, the final data will be loaded to DW as depicted in the right side with the loading mechanism provided by the tools

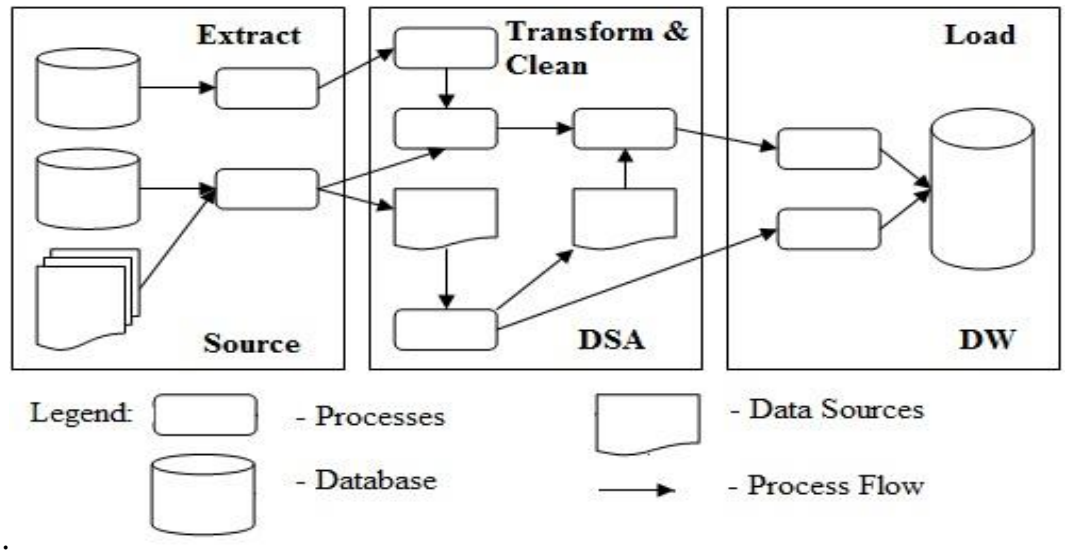


Figure 2.7: General Framework for ETL Processes (Simitis, 2004)

2.3.1 Problems of ETL

Generally, ETL processes face several issues, problems and constraints in design and implementation due to the hugeness of data, labor-intensiveness and complexity of tasks, lengthy procedure, constrained by unpredictable realities (Kimball & Caserta, 2004; Vassiliadis, Simitsis, Georgantas, & Terrovitis, 2003). First, the problems are related to the inefficiency of data loading, which normally runs as nightly batch during off-line operational system (Chaudhuri & Dayal, 1997; Kimball & Ross, 2002). Therefore, an up-to-date data cannot be loaded into the DW because of the latency in extracting the data sources. However, this problem can be solved by extracting and updating the data sources at any time needed by the users.

The efforts to resolve this problem was proposed by Garcia-Molina, Labio and Yang(1998) with an efficient algorithm to identify the updated data and Rundensteiner, Koeller and Zhang (2000)with the extension of *Structured Query Language* (SQL) called evolvable-SQL (E-SQL) for identifying and updating the data. In addition, the final loading of DW, any changes and how the data is changed must be known. Howthe data changes will be reflected by the responses taken by the ETL processes. According to Kimball and Caserta (2004), these responses are defined as *Slowly Changing Dimension* (SCD) that supports the identifying and updating of changes of data sources toward the DW structure.

Secondly, the problem occurs in the data integration and transformation process that refers to the transforming or conforming and cleaning activities. Various tasks on these activities have been detailed out by researchers and practitioners. Rahm and Do

(2000)determine the ETL processes as data cleaning activities that comprise data analysis, definition of workflow and mapping rules, verification, transformation and backflow of cleaned data. Meanwhile, Lujan-Mora (2005) defined six steps of ETL processes: i) select the sources for extraction; ii) transform the sources; iii) join the sources; iv) select the target to load; v) map source attributes to target attributes; and vi) load the data. However, the main problem in these activities is related to the integration of disparate data sources that can be identified as data conflicts or semantic heterogeneity problems in the data sources (Halevy, 2005; Lujan-Mora, 2005).

The structural heterogeneity problem is a situation where the identical information stores as different structure in disparate data sources. The semantic heterogeneity always refers to the conflict of meaning between information items, whether in attribute names or instances (data value). These scenarios can be shown in the following example as illustrated in Figure 2.8:

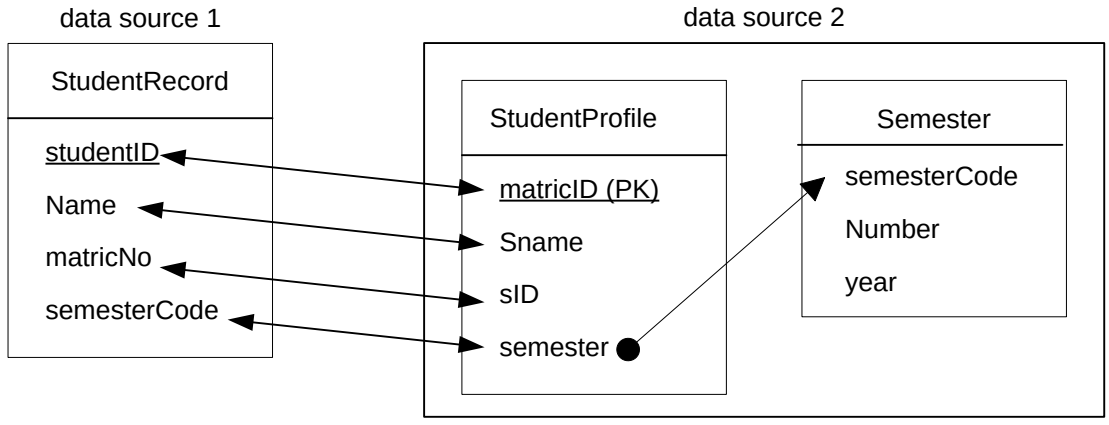


Figure 2.8: Data Structure for Student Record

Figure 2.8 shows the student records from two different data sources (data source 1 and data source 2). Data source 2 presents a different structure compared to data source 1. Furthermore, the differences also exist in tables, and attribute names that are subjected to different interpretation or meaning. This type of semantic (meaning) heterogeneity can be classified as *synonyms* or *homonyms*(Buccella, Cechich, & Brisaboa, 2003; Skoutas & Simitsis, 2006). A synonym is about two datawith the same meanings referred to the different name; conversely a homonym is about two data having asame name but referred to thedifferent meaning. The well-defined and concise semanticdata sources are essential for successful data integration and transformation in DW systems. This will avoid the confusion of understanding the data sources (i.e., tables, attributes) name or definition and preserve the autonomy of data owners. Furthermore, the meaning of data items should be well-accepted in data interchanged across the data integration and transformation processes.

Lastly, the problems are related to generating the data transformation specifications. According to Alexiev et al.(2005), generating the data transformation is a big challenge in EAI, EII or DW environments. This is because the nature of data transformation is application-driven, which requires high maintenance when the requirements or data sources are changed. The transformation mechanism (e.g., filtering, conversion, aggregation, merging) needs to be updated according to the changes.

Indeed, current approaches need to write a program for each of the transformation processes, and this is prone to errors and consumes need more time for writing and

maintaining the program codes (Kimball and Caserta, 2004; Alexiev et al., 2005). Thus, the possible way to automate the generation of data transformation specifications and to ensure the applicability of the solution needs to be explored further. The solution to this problem will reduce the burden of ETL developers in completing the back room activities.

Summing up, the main factors causing problems in ETL processes are: i) complexity and hugeness of DW; ii) latency on data extracting and loading; iii) business requirement reconciliation; iv) data sources heterogeneity; and v) generating of data transformation specifications. In the development of ETL processes, the ETL developers need to understand the requirement of business users, the whole process of ETL workflow, the building-up of ETL processes operations, and the structure of DW.

Therefore, a good and appropriate ETL processes modeling is needed in order to visualize the whole processes of DW operations and facilitate the communication with the business users for designing their requirements. Moreover, the mapping of user requirements and data sources schemas need to be agreed to and confirmed for reconciling the business semantics and resolving the data heterogeneity problems. This two-fold solution is important for designing the ETL processes successfully. The workable modeling of ETL processes is significantly important for the success of DW systems development. The next Section2.3.2 presents the work of modeling and designing of ETL processes in DW systems.

2.3.2 The Modeling

The modeling of ETL processes is required for helping the developer to design and maintain the ETL processes from the early phases of DW systems development. Due to the characteristics of DW, the tasks to design and develop the ETL processes are also difficult, tedious and complex. The creation of ETL scripts and managing their changes in ETL tools are very important for implementing and maintaining the DW systems(Sen & Sinha, 2007). Moreover, an effort to organize these tasks through the ETL tools will reduce the burden of the DW developer (Friedman & Gassman, 2005). However, without a proper modeling and systematic method for designing the ETL processes, the ETL processes specifications will be unmanageable and would worsen the implementation of DW systems.

2.3.2.1 Conceptual Modeling

Conceptual modeling is a high level abstraction of a domain problem without defining solution for the problem and uses terms, concepts and their relationship that is familiar to the application of users (Halpin, 2001; Olivé, 2007). Thus, conceptual modeling is the earliest model for DW design that captures the general specification of user requirements, data sources schemas, data transformations, and mapping of data sources to the DW schemas. The goal of ETL processes is to perform the integration and transformation of data sources toward the structure of DW. Therefore, the modeling artifacts should be able to document and formalize the whole process of ETL and help the DW developer to capture the right semantics

ofbusiness requirements and resolve the semantic heterogeneity problems during the designing stages.

Many efforts have been taken for modeling the DW at a conceptual level with some of them being acclaimed as novel approaches. However, none of the approaches was accepted as a standard approach by the community because of the proprietary elements in the design methods (Rizzi et al., 2006). The DM is popular DW model developed by Kimball (1996) composed of conceptual and logical design technique often used in the DW design. It gives the ability to visualize an abstract set of data in a concrete and tangible way and purposely for end-user delivery, which stresses on the data summarizing and aggregation (Kimball 1996; Kimball & Ross, 2002). The DM structure, often called *star schema* or *star joins,* diversifies the model in various scenarios. Other approaches such as ER (Franconi & Kamble, 2004; Inmon, 2002; Sapia et al., 1998), UML (Abello, Samos, & Saltor, 2002; Lujan-Mora, Trujillo, & Song, 2006) and ad-hoc models (Golfarelli et al., 1998; Rizzi, 2007) have been proposed to conceptualize the DW and ETL model. ER approach is considered very useful for capturing the data transaction but not data aggregation.

Simitsis (2004) and Lujan-Mora (2005) classified the conceptual modeling of ETL processes in three different angles, namely, functional, dynamic, and static. The functional model focuses on the functionality of the ETL processes that intentionally describes the mapping between data sources and DW structure. Both researchers acclaimed novelty in their modeling approaches by proposing new notation for ETL mechanism. However, not many explanations on resolving the semantic

heterogeneity problems in a conceptual model within the scoping of business requirements for deriving the ETL processes specifications were given.

Froma dynamic point of view, the modeling is focused on the refreshment process of DW presentation (previously known as *materialized* view) according to changes on the data sources (Bouzeghoub, Fabret, & Matulovic-Broqué, 1999; Rundensteiner et al., 2000). Froma static point of view, the modeling is described as a formal description of concepts, relationships and information requirements for application integration (Calvanese, Giacomo, Lenzerini, Nardi, & Rosati, 1998; Husemann et al., 2000). Then, the term 'domain model' has been used to denote the union of an enterprise model and data sources model that act as *an inter-model relationship* to capture the mapping between data sources and the DW. Although the user requirements have been captured and analyzed, no explanation has been given to the integration and transformation of data sources for completing the DW and ETL processes design.

The efforts in modeling the relationship between business processes with DW systemswere proposed by Stefanov and List(2005) and Akkaoui, Mazón, Vaisman, & Zimányi (2012). The modeling approach bridged the gap between DW systems and business processes in the organization. The BI perspective has been embedded in the business processes via business processes modeling language called Event-Driven Process Chain (EPC). Indeed, the business processes that contain the business requirements can be used to acquire the specific information in DW

systems. However, obviously, the proposed model does not consider the ETL processes' components in the modeling activities.

For practitioners, the approaches implemented by Kimball and Caserta (2004) would be a practical step to follow in modeling and designing the ETL processes. The ETL process is defined as *back room* activities comprising data extraction, integration and transformation mechanisms. However, certain things are missing in explaining the conceptual level of the ETL processes. The high abstraction of the ETL scenarios is not clearly defined especially in mapping up the semantics of business requirements to the data sources' schemas. This makes the ETL model not being able to fully specify the ETL specifications to produce the DW as required.

As mentioned earlier, the conceptual model of ETL processes is a high level abstraction of modeling that aims to understand the general mapping between data sources to the DW within the scoping of user requirements. In order to proceed for implementation, the design efforts need to be detailed into logical modeling. The ETL processes logical modeling that focuses on specific database model is discussed in Section2.3.2.3.

2.3.2.2 Logical Modeling

ETL logical modeling details the ETL conceptual model specification to the model, which is toward the implementation of the entire ETL processes. According to Kimball and Caserta (2004), the logical modeling of ETL processes can be viewed in ETL processes specifications and meta-data perspectives. The ETL processes

specifications contain logical mapping of data sources to the DW, whereas metadata perspective consists of the source-to-target mapping that explains logically what happen to the data, originally from the data sources until it is loaded to the DW. Furthermore, the ETL processes logical modeling is a formal logical meta-model, where the data stores, activities and their rules, relationships and constraints that are formally defined (Simitsis, 2004).

Some researchers focused on the ETL logical modeling issues related to the data quality and cleaning mechanism. The proposed solutions are implemented through AJAX tool and Potter's Wheel that mainly tackle the basic issues in data integration and transformations (Galhardas, Florescu, Shasha, & Simon, 2000; Raman & Hellerstein, 2001). Basic tasks of the data transformation mechanisms were supported by both tools such as mapping, matching, clustering and merging. In addition, Potter's Wheel provides an iterative and interactive way of data cleaning procedures and automatically infers the data value in terms of user-defined domain and accordingly examines the constraint violations. In data quality issues, Vassiliadis (2000) extensively reviewed and presented the quality meta-model for quality management of DW systems.

Simitsis (2004) presented a framework for modeling the ETL processes and optimization of ETL workflows. In order to realize the framework, the researcher defined the formal logical meta-model containing the information about data stores, activities, rules, relationships and constraints. The proposed approach was established with notation of modeling language and provides the developer with a

methodology on how to design the ETL processes at the conceptual and logical levels. Furthermore, the issues of ETL processes' workflows optimization were given attention with the set-up of the theoretical framework for the problem and modeling of the workflows as a *state space search* problem.

However, no effort was made for analyzing the user requirements to support the designing or optimization of the ETL processes. The model represented by the various notations that symbolized the data stores (i.e., data sources, DW, ETL activities) is emphasized on the logical workflows of data sources to the DW. The ETL activities are centered on the workflows and identified as data staging area (DSA) in most of the ETL tools. However, the complexity of DSA is dependent on the detail digging of the data that is based on the structure of data sources.

Lujan-Mora (2005) presented a DW and ETL processes modeling by using the UML-based approach. This modeling provided a common ETL processes operations such as filtering, aggregating, joining, conversion and loading by its own notation. The DW modeling comprises conceptual, logical, and physical design, but no conceptual model of ETL processes was given. In order to fulfill specific DW requirements, the UML approach was extended through a stereotype to build the UML Profile for modeling the ETL processes. However, no consideration on user requirements was made to support the modeling of DW and ETL processes. The user requirements were assumed to be readily available to be used for designing the DW systems.

Summing up all the possible issues of research in modeling the ETL processes, the approaches for modeling technique are presented, where some approaches have followed the ER and UML modeling languages. However, the semantic heterogeneity problems in data conflict were not systematically tackled in modeling and designing the ETL processes. The available approaches do not assist the developers to understand and capture the business requirements accordingly in order to define the ETL processes for propagating the data sources into the DW. This will increase the semantics heterogeneity problems, especially in heterogeneous data sources that are located in many places with autonomous environments.

Clearly, the problems begin with the design of conceptual ETL processes, which emphasize capturing and analyzing the user requirements. Thus, the modeling and designing the ETL processes should start by analyzing the user requirements from the perspective of individual and organization. The analyzing process will determine the common definition of the DW structure and guide in building the ETL processes specifications. As a result, the ETL specifications will be much clearer and easily understood by the users and developers. This will help the developers to design the entire DW components in a controlled and systematic way. Therefore, it can be concluded that these issues and problems in ETL processes design is centered on the data integration and transformation activities. The discussion on data integration and transformation in DW systems environments are discussed further in Section2.3.3.

2.3.3 Data Integration and Transformation

Data integration and transformation are mechanisms to manipulate the data sources toward the format provided by the DW structure. Normally, the mechanismsare executed in a heterogeneous data sources environment. Both integration and transformation are performed together for complementing the ETL processes cycle. These mechanisms attempt to provide the users with a unified view of accessed, distributed and autonomous data sources (Calvanese et al., 1998; Kimball, 2006; Lenzerini, 2002). Ultimately, the aim of data integration is to enable all the information systems to work together seamlessly. The problems of data integration and transformation occur during the process of turning the data sources into the DW.

2.3.3.1 Semantic Heterogeneity Problem

These are many problems in heterogeneous data sources due to various data format, convention, quality, incomplete lineage, difficulty of data access, and unavailable current and complete data. The detailed explanation about these problems was classified in taxonomy structure by Kim, Hong, Hong, Kim, and Lee(2003). However, the most obvious problem in data integration can be summarized as data conflicts in a structural and semantic manner (Halevy, 2005; Simitsis, 2004; Ziegler & Dittrich, 2004). This causes the problems in implementing the ETL processes, whenever the data acquisition (prior data integration) and data transformation (after data integration) need to accommodate the user requirements. Moreover, the heterogeneous data sources are physically extracted from the disparate autonomy of organization that is legally bound by procedures and security issues.

Comparing with other environments, DW is the only architecture that emphasizes on data transformation and cleansing for producing the quality information from the heterogeneous data sources (Skoutas & Simitsis, 2007; Stylianou & Kuman, 2000; Vassiliadis, 2000; Wand & Wang, 1996). The quality of data sources is partly improved by the ETL processes, which implement the transformation and cleaning mechanism according to the requirements of business users. All the approaches apply a different mechanism of data integration aimed to provide integrated information to the users in homogeneous and unified view of information from the heterogeneous data sources (Ziegler & Dittrich, 2004). However, the problems of semantic heterogeneity in data integration, transformation, and cleansing mechanism such as schema resolution, data mapping, data cleansing, and data transformation, still remain as valid issues for research. Thus, this work explores the solution of semantics heterogeneity problems in data integration and transformation mechanism through the use of ontology approach.

2.3.3.2 Related Works

The previous work on data integration method was proposed by Levy (1999)called *local-as-view* (LAV) and Ullman (2000)called *global-as-view* (GAV). The problem in both methods was further investigated as a theoretical foundation in modeling of data integration, queries processing, data sources inconsistency and reasoning on queries by Lenzerini (2002). Basically, the LAV method emphasizes on inter-schema mapping among the data sources to a global schema for querying and producing the information to the users. This method is also known as a *source centric approach*

and has been used for querying in DW systems (Calvanese, Giacomo, Lenzerini, Nardi, & Rosati, 2001). Meanwhile, the GAV method integrates the data sources as *global schemas* and creates mapping to the data sources before querying. Both methods have been applied in a multi-database system (Wang & Murphy, 2006).

Abiteboul et al.(1999) discussed heterogeneous data integration issues in a web-based environment. The researchers proposed the solutions known as YAT data models and TranScm, which tackle the problems of data conflicts. Furthermore, the YAT solutions are based on capabilities of mapping between different schemata; meanwhile TranScm automatically finds the mapping between different schemata. However, no example has been given for integrating the relational model of data sources and a model proposed for resolving the data conflicts in the heterogeneous web data.

Haas et al.(1999)discussed the general setting of data schema and data integration. The prototype system called GARLIC is used as a mediator or wrapper for data transformation, view definition and data integration. These approaches provide the foundation of mediator system architecture for data integration. In the issues of ETL for DW systems, Hellerstein, Stonebraker and Caccia(1999) presented the definition of ETL processes in logical and physical independence which need to be defined by a specific language and extend the utilization for querying DW system. As a result, they proposed the SQL99 as a language and introduced the COHERA system to represent the mapping between tables (i.e., relational database) in federated database

environments. However, these approaches neglected the proper process to define thedata integration and transformation activities.

Several data integration approaches have been suggested by researchers, but most of these methods focus on the structural kind of integration. Technically, structural integration approach can be tackled by tools via data sources connectivity such as object database connectivity(ODBC), java database connectivity(JDBC) or database native connection, whether from the structured or unstructured data sources. Then, the collection of data sources are organized in the form of local or global schema for establishing the integration tasks. However, the integration approaches did not focus on resolving the data heterogeneity problems at the design stages. In DW systems, business requirements did not properly guide the developer to design the DW and ETL processes. This was due to the ambiguous meaning of data sources that had not been properly analyzed and defined. This then contributed to the problems of semantics heterogeneity underlying the requirement of the users.

Recently, an effort for resolving the semantics heterogeneity problems was done through the semantic web technology. The emergence of ontology as the main artifacts of semantic web technology in recent years has been used as a solution in semantics heterogeneity problems (Alexiev et al., 2005; Buccella et al., 2003; Firat, Madnick, & Grosof, 2002; Guo et al., 2003; Maedche, Staab, Studer, Sure, & Volz, 2002). It has been claimed that ontology can resolve the semantics heterogeneity problems in database integration, especially in DW systems(Cao et al., 2005; Priebe & Pernul, 2003; Romero & Abelló, 2007; Sell, Cabral, Motta, Domingue, & Pacheco,

2005; Skoutas & Simitsis, 2007; Toivonen & Niemi, 2004). This is because database schemas can be modeled as an ontology structure with the in-depth definition of data sources semantics (Alexiev et al., 2005; Fonseca & Martin, 2007; Leuf, 2006).

The solution of this problem should aim to reconcile the various meaning of data sources that obviously depend on the user's knowledge of business requirements, and developer knowledge in data values or schemas (Doan & Halevy, 2005; Giorgini et al., 2008). Therefore, an acceptable common semantics of business requirements and data sources are important for ensuring the correctness of data loading into the DW after implementing the ETL processes. This is the agreed-upon model of the entire business information that actually emerged from the ETL processes' principles (Schreiber, 2003).

Few works have focused on resolving the heterogeneity problems in DW systems, especially in the ETL processes. It is important to provide the systematic method for designing the ETL processes with consideration of user requirements and data heterogeneity problems. An outstanding work on this problem was carried out by Simitsis (2004) and further enhancement on ETL processes design by using ontology by Skoutas and Simitsis (2007). However, the approach did not focus on analyzing user requirements that obviously reshape the design of ETL processes, in particular, and DW in general. Therefore, this research defines the systematic process for modeling and designing the ETL processes, which is then the method for analyzing user requirements toward the design of conceptual ETL processes.

The functions of modeling languages and tools that support the current designing the DW and ETL processes are important. This will help the proposed method to comply as much as possible with the current technology for accelerating the learning process of a developer. The next Section2.4 highlights the technology used for the ETL processes development.

2.4 Standards, Modeling Language and ETL Tools

Several tools for modeling and designing DW components such as dimension schema, OLAP schema or ETL processes specifications have been developed and commercialized. Nevertheless, none of these has been accepted as a standard and agreed upon method by the DW community as the *de facto* for the DW systems interoperability (Hwang & Xu, 2007; Rizzi et al., 2006). The efforts to achieve a standard methodology for modeling and designing the ETL processes are crucial tasks for interoperability of DW systems in various platforms. This effort is difficult to realize because the DW solutions are always based on technology used by the organization. However, the standard DW metadata has been proposed as guidelines for developing the DW systems.

2.4.1 Standards

Standard terms in DW development are explained in two points of view. Firstly, a standard definition for DW or enterprise meta-data is needed in order to facilitate the reusability, portability and interoperability of object-based software in distributed and heterogeneous environments (OMG, 2003). The DW meta-data standard has

been provided by Object Management Group (OMG), which previously developed by Metadata Coalition (MDC). The meta-data is known as Common Warehouse Meta-model (CWM) and is shown in Figure 2.9. The CWM contains guidelines for metadata interchanges in various components during the DW systems development, including the ETL processes. However, the complexity of the guidelines has created difficulties for the DW provider and developer to adopt the standardization (Rizzi et al., 2006).

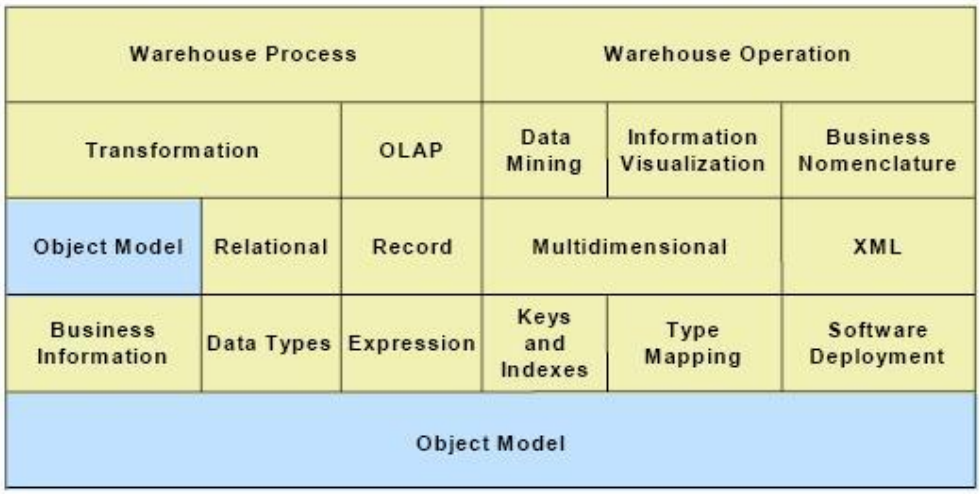


Figure 2.9: Common Warehouse Meta-model (OMG, 2003)[4]

Secondly, standardized development techniques are needed for ETL developer to provide a consistent and maintainable ETL specifications environment (Kimball & Caserta, 2004). Some of the areas that need to be standardized are naming conventions and the best practices' design methodology, which documents and follows the best approach in developing the ETL processes.CWM consists of: i) a standard language for defining the structure and semantics of metadata using *Meta-*

[4] http://www.omg.org/

Object Facility (MOF), and UML; ii) a standard interchange mechanism for sharing metadata defined in XML and *XML Metadata Interchange* (XMI); and iii) a standard specification for metadata access and discovery. Both UML and XML are standard modeling languages for modeling and designing the ETL processes. As far as modeling is concerned, the proposed ETL processes design in this thesis complies with the CWM specifications.

2.4.2 Modeling Language

Entity Relationship (ER) is a traditional way for modeling the database system (Chen, 1976). In DW systems, ER was used to model the DW components and their related relationships (Golfarelli et al., 1998). However, the ER was not enough to represent the functionalities of DW, which main objective was to provide the detailed level of information for decision making purposes (Kimball, 1996). Therefore, an enhanced of ER-based DM was proposed for modeling the DW by Golfarelli et al. (1998) and Rizzi (2007).

Unified Modeling Language (UML) is widely used as a modeling language that unifies the methods mostly used by the software developer, especially in an object-oriented paradigm. The UML is defined as a rich set of graphical diagrams consisting of *use cases, class diagram, activity diagram* and others. Lujan-Mora (2005) used the UML as the modeling language for modeling the ETL processes, which are defined as the UML profile for ETL processes. Some ad-hoc notations have been proposed to support the modeling of the DW components, such as dimension, measure, and ETL activities.

Goal model is widely used for modeling the agent-based software system, which is emphasized software requirements analysis in the early phases of software system development (Ali, Dalpiaz, & Giorgini, 2010; Bresciani et al., 2004). The goal model was used in the modeling of DW requirements and supporting the DW systems design by using the well-known approach of software system development such as, i* and Tropos methodology (Bresciani et al., 2004; Giorgini et al., 2008; Yu, 1995). The goal model was given a proper treatment in the early phases of user requirements by using an international standard User Requirements Notation (URN).

Many ad-hoc models have been proposed by the researchers and new modeling language has been proposed that support the modeling of ETL processes. Some of the researchers like Simitsis (2004) proposed new graphical notations for data sources entity and ETL operations;and Lujan-Mora (2005) proposed new graphical notations for ETL operations. Although the proposed model supports the entire ETL modeling, the application of the approach is not widely used. This is due to the difficulty of the language to model the ETL processes from the analysis of user requirements. Therefore, this research adapts the goal model for designing the ETL processes and is further discussed in Chapter 3, Chapter 5, and Chapter 6.

2.4.3 ETL Tools

The market for ETL tools (part of the Business Analytic Software - BAS) has grown over the years since it was introduced in 1996. The BAS market has reached 667 million US dollars since 2001 with a growth rate of approximately 11-16 percent(Agosta, 2002). In the year 2006, the market reached $19.3 billion US

dollars[5] and this shows the interests of people or organizations in developing the DW systems.Some of the vendors who provide the ETL solutions are Oracle with Warehouse Builder, Informatica with Powercenter, Microsoft with Data Transformation Services, SAS Institute with Data Integration Suite, IBM with Warehouse Center and many more. Currently, most of the ETL tools are *engine-based* or *code-generation based* (Simitsis, 2004) that run the implementation of data flow in DW systems.

To date, there are no tools available that incorporate the ontology-based approach in any of the BSA tools for designing the DW and ETL processes. Therefore, this research finding contributes to the development of the ETL tools that support the ontology and goal approaches in modeling and designing the ETL processes in DW systems development. Possibly, the current tools used for this research can be upgraded into a workable tool for implementation in the real DW environment. However, this research does not focus on developing the tool.

2.5 Conclusion

This chapter presents the literature on DW and ETL processes designing. The concepts of DW and research works on modeling and designing the data sources and DW are explained. Furthermore, literature on ETL processes are emphasized and research issues in ETL processes and DW design in general are highlighted. In summary, most of the design works for DW systems have not focused on the ETL processes activities. Since there are limited researches on this area, this research

[5] http://www.idc.com/

explores and conducts in-depth study on the ETL processes design for the DW systems. The standard, modeling language and ETL tools are also discussed with the various approaches of the ETL processes modeling. Chapter 3 discusses the requirement analysis for DW.

CHAPTER THREE– REQUIREMENT ANALYSIS FOR DATA WAREHOUSE

This chapter presents the approach for requirement analysis process in DW and ETL processes design.The detail explanation on the organizational, decisional, and developer modeling that isemphasized on the early phases of DW requirement is provided. Anagent-based approach in the requirement analysis method is presented, and the related theories are elaborated. This chapter concludes by relating the requirement analysis method with ontology.

3.1 Introduction

This chapter elaborates on the early phase of requirement analysis in DW systems development, which involves the reconciliation of business semantics with the relevant data sources. This is due to the fact that the business semantic reconciliation is crucial for supporting the design of the ETL processes.

3.2 Requirement in Software Development

A software requirement can be defined as *"a property which must be exhibited by software developed or adapted to solve a particular problem"*(IEEE, 2004). The problem is about the issues that need to be tackled by the software being developed. In another definition, Thayer and Dorfman (1990)defined *"Software requirement is a software capability needed by the user to solve a problem to achieve an objective. The software capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documentation"*.

The definitions are about the capability set of requirements to build a software system that is organized in a requirement development and management system. Leffingwell and Widrig(2003)stated the requirement development as *"a systematic approach to eliciting, organizing, and documenting the requirements of the system, and a process that establishes and maintains an agreement between the customer and the project team on the changing requirements of the system"*. In the entire software development life cycle, this is known as requirement engineering that also includes activities related to management, such as identification, traceability, and change management (Sommerville, 2007).

Software requirement focuses on fulfilling the user requirements that is also applicable in the DW systems scenario. This requires a proper requirement analysis method to be part of the systematic software development process. Traditional methods were performed in the beginning of the requirement process, without relooking into the requirement changes throughout the development process of requirement analysis (Thayer & Dorfman, 1990). However, this approach was unsuitable for a large or complex software system like DW that requires an incremental and iterative method in defining the DW requirements. The requirement analysis is very important in the beginning of a requirement process for resolving the business problems of users by a software system. Therefore, DW problems should be able to address by proper analysis of user requirements at the early phase of DW systems development. This is discussed further in the next Section3.3.

3.3 Requirement for the DW system

A requirement for the DW systems is about the decision-analysis kind of information required by the business users in an organization. The business users are stakeholders involved in any business transactions and decision-making process. The ETL processes requirements should be enhanced as what was defined in DW model for completing and implementing the DW systems development. In practice, understanding the business needs is important in determining the mapping between data sources and the DW (Simitsis, 2004; Kimball & Caserta, 2004). This task needs to be detailed and focused for modeling the ETL processes.

The task for capturing and analyzing the DW requirements is not an easy task because it involves various levels of business users or stakeholders, departments and organizations in different perspectives of DW requirements. The requirements from the management and user perspective represent the high-level goal of the organization and the tasks that the users must be able to work through. The requirements from an implementation perspective represent the requirements on a detailed level of DW (Bruckner, List, & Schiefer, 2002) that refers to DW structure and ETL processes specifications. The requirement perspectives of DW are shown in Figure 3.1.

The tasks refer to gathering the requirements, realities, business rules and constraints affecting the ETL processes in one place (Kimball & Caserta, 2004). It aims to identify the decisional information to be stored in DW. According to Prakash and Gosain (2003), the organization needs to identify the organization's goals, and define the related information for decision-making. Thus, both data-driven and demand-driven approaches are applied in the requirements analysis of DW systems. The final requirement of DW will be mapped to the conceptual and logical modeling of ETL processes and will support the implementation of back-end activities of DW development.

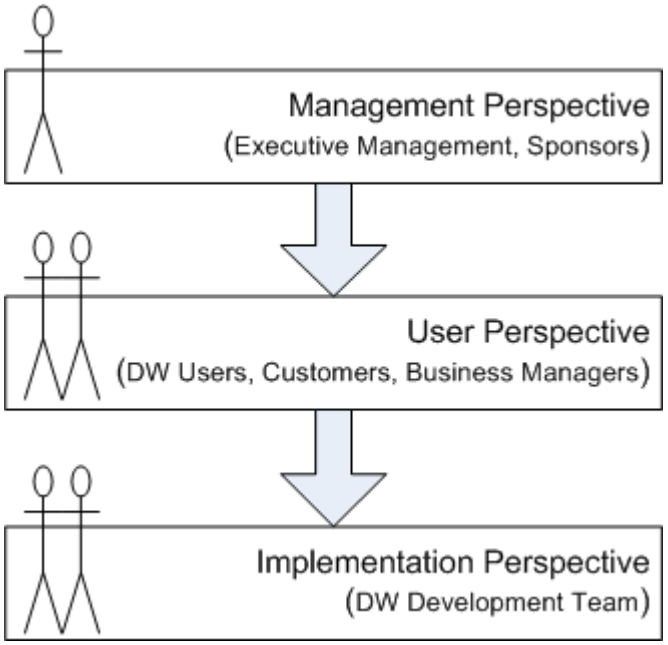


Figure 3.1: Requirement Perspectives of DW Systems

The DW requirements should maintain the synchronization with ETL processes to avoid obsolescence of DW systems. In other words, any changes on DW

requirements should refresh the ETL processes specifications in order to maintain the freshness of DW systems. In addition to defining and maintaining the ETL processes specifications, the DW requirement process should tackle the data heterogeneity problems. These works are not well-explored from the early phase of DW development and the solution to these is proposed in the analysis process of DW requirements of this research. Some methods in organizing the business semantic and propagating the data sources to the DW structure through ETL processes are explored and presented in the proposed requirement analysis method for ETL processes in Section3.4.

3.4 Requirement Analysis for ETL processes

Requirement analysis of ETL processes focus on the transformation of informal statements of user requirements into a formal expression of ETL processes specifications. The informal statements of user requirements can be derived from two main approaches, namely, supply or data-driven and demand or user-driven. These approaches are practical in the real implementation of DW, where the user requirements are elicited and analyzed from the organization and decision-maker's perspective (Prakash & Gosain, 2008; Giorgini et al., 2008). These requirements will be mapped with the available data sources through appropriate ETL processes, which should be derived from the data integration and transformation analysis.

Arguably, efforts to analyze the DW requirement from the high-level abstract of user requirements (e.g., goal, sub-goal, stakeholder) toward the detailed specification of ETL processes (e.g., extracting, filtering, conversion) are important in order to

handle the complexity of ETL processes design and ensure the successof DW systems development. The analysis is crucial to obtain a consistent reconciliation of business requirement toward the data sources that provide the data (Mazon et al., 2007). Therefore, a proper and systematic transformation analysis method is required in the early phase of requirement analysis to overcome these problems and provide a set of functional requirements for the ETL processes specifications.

3.4.1 Early Phase Requirement Analysis

The early phase of requirements analysis is about identifying the stakeholders and their intentions on information needs. Stakeholders are modeled as business actors who depend on other actors for goals to be fulfilled, plans to be executed, and resources to be utilized (Bresciani et al., 2004). In software engineering literature, it is widely accepted that the early requirement analysis will significantly reduce the possibility of misunderstanding user requirements (Mazon et al., 2005; Yu, Giorgini, Maiden, & Mylopoulos, 2011). The higher understanding amongst stakeholders will possibly increase the agreed terms and definitions to be used during the ETL processes execution.

Therefore, the requirement analysis approach is centered on the organizational and decisional modeling, and focuses on the transformation model from the perspective of a developer for defining the ETL processes specifications. By revisiting the approaches of Prakash and Gosain (2008) and Giorgini et al. (2008) on requirement analysis method of DW systems, the general flow of information perspectives can be viewed in Figure 3.2.

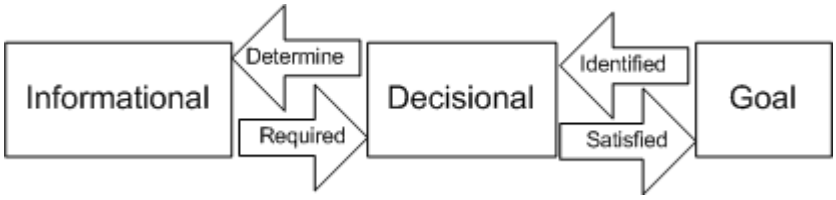


Figure 3.2: General Flow of Information Perspectives in DW systems

The model presented in Figure 3.2 is built from two different perspectives of requirement analysis: i) organizational modeling that centers on stakeholders; and ii) decisional modeling that centers on the decision makers. Nevertheless, the model does not include the perspective of data integration and transformation analysis that describes the ETL processes specifications. In the early phase of requirement engineering, the analysis that fits the organizational, decision-maker, and developer contexts is clearly important. Thus, the general flow of information that fits these perspectives is shown in Figure 3.3.

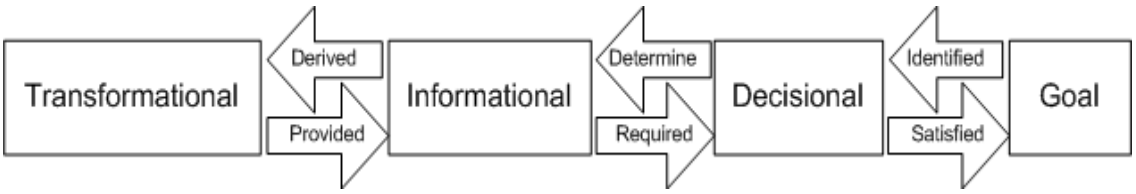


Figure 3.3: General Flow of Information Perspectives in DW systems

The flow of information begins from the organizational perspective that was identified from the organization goals. Then, the information is determined by decision-maker in order to satisfy the organization goals. Finally, the informational determined by the decision-maker derives the data integration and transformation process for providing the data. Details on these perspectives are presented in the next Sections3.4.1.1, 3.4.1.2, and 3.4.1.3.

3.4.1.1 Organizational Modeling

Organizational modeling is used to identify organization goals, which the DW components (i.e., facts, dimension, measures) must satisfy and this is asan analysis of DW systems. The basic goal model is illustrated in Figure 3.4 (Bresciani et al., 2004; Yu et al., 2011).

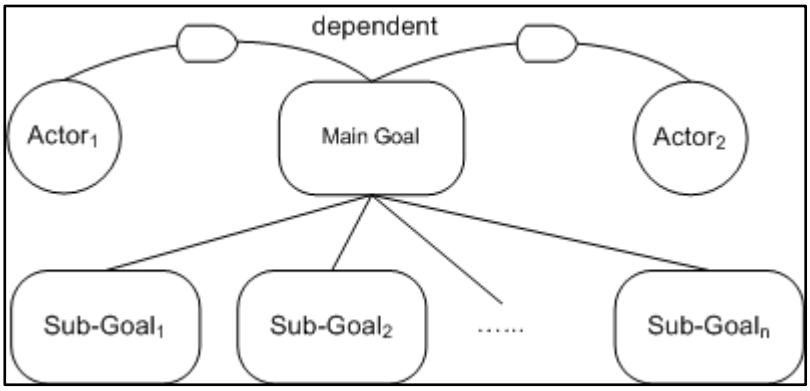


Figure 3.4: Basic Goal Model

It consists of three different analyses, which are produced in the iterative process. The analyses are: i) goal analysis, in which the actor diagrams and rationale diagrams are produced; ii) fact analysis, in which the goal rationale diagrams are extended with facts; and iii) attributes analysis, in which the fact rationale diagrams are extended with attributes. All goals, facts, and attributes are defined in the context of individual and organization views. The details of this model are explained in Chapter 5.

3.4.1.2 Decisional Modeling

The decisional modeling directly focuses on the information needs by decision makers and refers to the analysis of DW systems(Winter & Strauch, 2004). Thus,

the modeling needs to be established by information, which is provided by the transformation analysis activities. Furthermore, the transformation analysis defines the set of activities that directly produced the information needed by the DW. Basic decision-goal model is illustrated in Figure 3.5 (Bresciani et al., 2004; Giorgini et al., 2008):

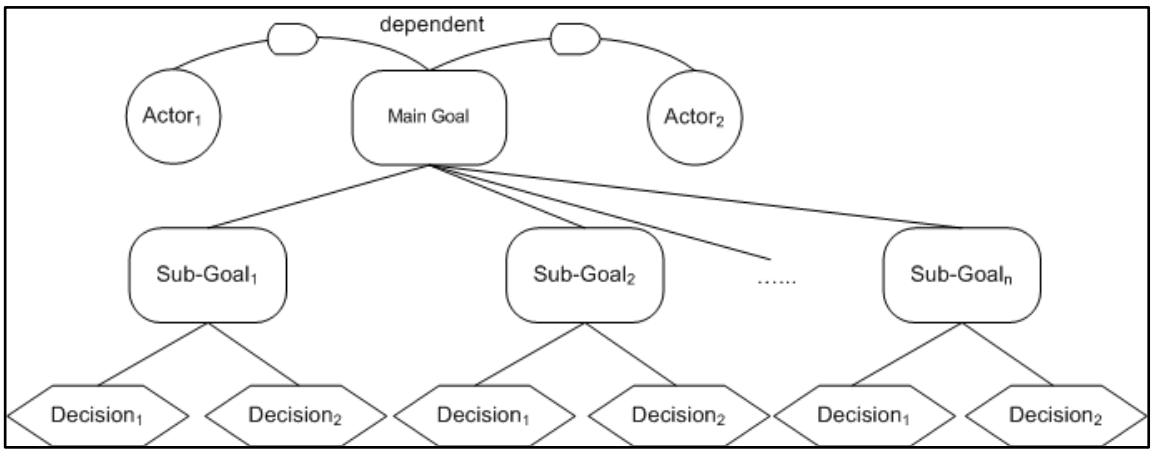


Figure 3.5: Basic Decision-Goal Model

The decisional modeling consists of four different analyses, which are produced in the iterative process. However, these analyses focused on the goal of a decision maker, which are represented by the actors as defined in the organizational model. The analyses are: i) goal analysis, which produces the rationale diagrams of decision-goal; ii) fact analysis, which extends the decision-goal diagrams with facts; iii) dimension analysis, which extends the fact diagrams with dimensions; and iv) measure analysis, which further extends dimension diagrams with measures. Finally, the decision modeling analysis will produce the informational model that is required in supporting the decision making. The details of this model are explained in Chapter 5.

3.4.1.3 Developer Modeling

As previously highlighted, to model and design the ETL processes, the requirement analysis needs to consider the goals of a developer in achieving each of the ETL processes' tasks. Thus, the aim of an analysis should identify the intentions of a developer for ETL tasks to be achieved, ETL plans to be performed, and user requirements and data sources to be reconciled. Developer modeling consists of three different analyses, which is also produced in the iterative process. These analyses are focused on the goal of a developer, which are represented by the actors as defined in the decisional model. The analyses are: i) data sources analysis, which produces the lists of data sources related to the goals, facts, dimensions and measures; ii) business rule analysis, which produces the lists of business rules and constraints for related facts; and iii) transformation analysis, which extends decision-goal diagram with transformation activities and rules involved. The basic developer-goal model is illustrated in Figure 3.6.

The transformation analysis explains the facts about actions and rules applied to the data sources in order to achieve the developer's goals for implementing the ETL processes tasks. The developer modeling will complete the goal-driven analysis of user requirements in order to produce the final DW requirement model of DW systems. However, data sources analysis will be conducted separately with business rules and transformations analysis. The data source's analysis will be used for mapping with DW requirement during the conceptual design of ETL processes. The details of this model are explained in Chapter 5.

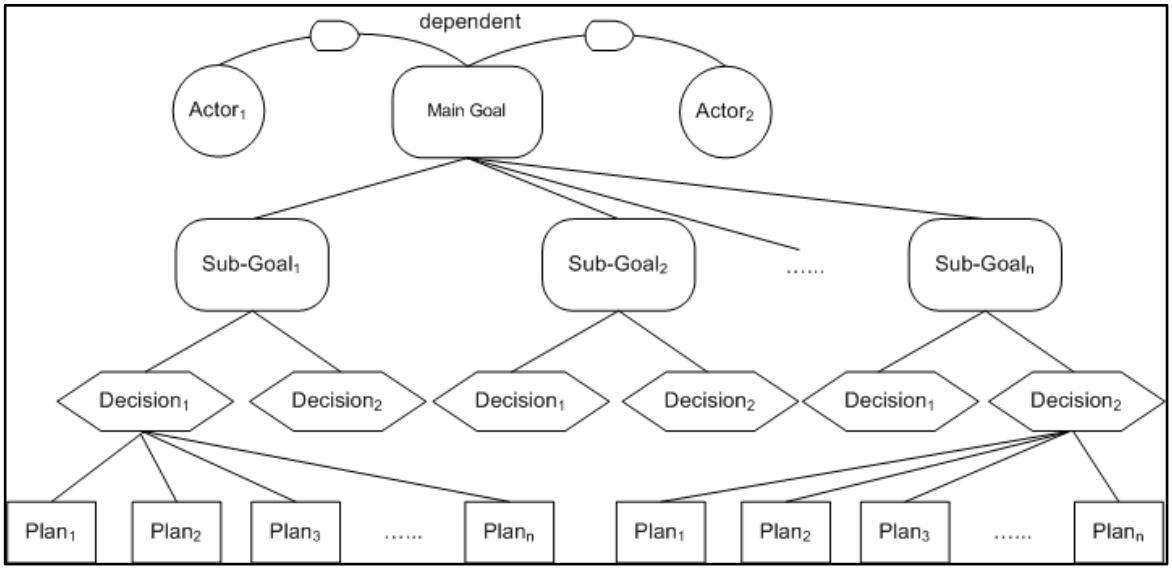


Figure 3.6: Basic Developer-Goal Model

3.4.2 Late Phase Requirement Analysis

In the late phase requirements analysis, the conceptual model of ETL processes is extended involving new actors and dependencies of ETL processes environment. These dependencies link the actors and goals for defining functional requirements of the system *to be*(Bresciani et al., 2004), i.e., the ETL processes. The analysis is conducted to provide the context of DW systems to be designed. The ETL processes model details about plans involved in executing the ETL operations and how the ETL processes specifications are derived. Clearly, this analysis will complete the requirement analysis process and reconcile the misunderstood DW requirements.

In summary, the general requirement analysis method of ETL processes encompasses three modeling perspectives (i.e., organizational, decisional, developer). These perspectives are used for analyzing phase by phase to presents the DW requirements (Giorgini et al., 2008). Moreover, the new concept of developer

modeling has been used for modeling the ETL processes that continues from the previous modeling tasks. Again, from the goal-oriented approach of DW requirement analysis, this research extends the existing approach to the design of the ETL processes model from the agent-based software development methodology. However, the proposed method is developed according to the object-oriented paradigm, which adopts the principles of agent-based approach in applying a high-level model of user requirements. This approach is useful because agent-based approach provides a way to reason the flow of control in a highly heterogeneous system (Booch et al., 1999).

3.5 Agent-Based Approach for Requirement Engineering

An agent-based approach has been used as an alternative in software engineering for quite some time. This approach is an extension to the requirement of engineering software system development. The notion of agent in requirement engineering is used to support the elicitation and analysis of software due to unique characteristics, such as intentionality, sociality, autonomy, reactivity, and proactivity (Nwana & Ndumu, 1999). Software system that has been developed using the agent-based approach can provide greater functionality and quality, especially on the *early-phase* requirement of engineering activities (Antonio, Ramırez, Imbert, & Mendez, 2005).

Several outstanding agent-based modeling had been proposed to support the software development methodology. The agent-models and languages are used to represent an abstract computational behavior of a software program for the software to be developed. Standard model on software development phases is defined as: i) early

requirements; ii)late requirements; iii)architectural design; andiv)detaileddesign. A comparison amongst the agent-based methodologies based on observation of Bresciani et al.(2004) and Antonio et al.(2005)are shown in Table 3.1.

Table 3.1: Comparison of Agent-Based Methodology

Methodology	Early Requirements	Late Requirements	Architectural Design	Detailed Design
I* (I-Star) (1995)	Full Support	Support	Not Support	Not Support
Tropos (2003)	Full Support	Full Support	Full Support	Full Support
Kaos	Support	Full Support	Support	Not Support
Gaia (2000)	Not Support	Full Support	Full Support	Support
MaSE	Not Support	Support	Full Support	Full Support
AUML (2001)	Not Support	Not Support	Support	Full Support
Prometheus (2002)	Full Support	Support	Not Support	Not Support
Cassiopeia (1996)	Full Support	Support	Not Support	Not Support
Message/UML (2000)	Support	Full Support	Support	Support

Based on the comparison, the Tropos methodology (shaded row) was selected to be applied in this research methodology because of its strength in methodological approach and it fully supports all phases of software development. Additionally, Tropos methodology is founded on the strong and well-accepted i* methodology that presents the semi-formal framework of agent-oriented modeling of a software system (Yu et al., 2011). This approach has been accepted for modeling in the *early-phase* requirement analysis and widely adopted for designing a software system in various

domains (Bresciani et al., 2004; Giorgini et al., 2008). Details on this methodology are explained in Chapter 5.

So far, the requirement analysis in DW systems and extending the reviewed system in current analysis approach for the ETL processes has been discussed. This research emphasizes the early phase requirement analysis as a very important task for reconciling the user requirements and organizing the mapping for data sources to the DW. The requirement analysis approach is conducted from three different perspectives that come from *a social reality* viewpoint of DW system environment. The theories of these perspectives that are related to the research problems are discussed further in Section 3.6. The early phase requirement analysis using the agent-based approach will be elaborated in Chapter 5.

3.6 Underlying Theories in Requirement Analysis Approach

The viewpoint of social reality in a DW system environment describes the strong interaction between organizations and users (i.e., end users and developers) for DW usage (Stefanov & List, 2007). This scenario shows the roles played by organization and users in realizing the usage of DW system. Therefore, the usage of DW system is derived from three perspectives of social reality: organization, user, and developer. This research adapts an agent-based method that utilizes these three perspectives. However, current methods do not consider the developer perspectives. The underlying theories of these perspectives are discussed to give reasons and understanding on the requirement analysis approach.

3.6.1 Organizational Theory

Organizational theory comprises theories and models that attempt to describe the relationship between organizations function and the environment. The purpose to understand the theories is to allow us to design and manage the organization in an efficient, effective, and responsive way (Hatch & Cunliffe, 2006). It deals with structures, systems, and processes that are coordinated to each other to achieve organization goals. The coordinating mechanism creates dependencies among entities formed together to operate the organization. Underlying this theory, a requirement analysis approach models the coordinated entities as social actors, who depend on each other for goals to be fulfilled, tasks to be performed, and resources to be utilized. Additionally, the theory provides better understanding for DW developer on why DW requirements are unveiled.

In another aspect, the *Institutional theory* of organization is also applied together with *Decisional theory* in order to establish requirements for users or stakeholders. According to *Institutional theory,* organizations can better survive if they focus on meeting stakeholder requirements and needs (Daft, 2008; Hatch & Cunliffe, 2006). Theorists suggested an organization is successful if they can satisfy the user demands, and establish their legitimacy (i.e., accepted as doing the right thing in the eyes of their stakeholders). The legitimacy is related to mission and vision of the organization. This is where the goal analysis started and further refined through the requirement analysis process.

3.6.2 Decisional Theory

Decision theory is about choice between alternatives. Drucker (1974) defined *"A decision is a judgment … a choice between alternatives"*. The management in an organization uses their judgment power to take decisions about business operations. In the same situation, the requirements of a decision-maker in DW systems are defined within the scope of decision making. However, the decision theory about the option to choose between alternatives is not directly applied. The concern with *goal-director behavior in the presence of options* is rightly supporting the requirement determination by decision-makers (Hansson, 1994).

There are two prominent approaches in a decision-making process, either by individual (e.g., staff, customer, and stakeholder) or by organization (i.e., management of the organization). Requirement analysis method proposed in this thesis applies the theory as presented in *rational* modelof an individual and organizational decision-making process for formulating the decisional modeling. The rational model consists of eight steps in making the decision (Archer & Tritter, 2000), and this research tailors these steps with the proposed solutions (explained in Chapter 6) as shown in Table 3.2.

Table 3.2: The Rational Model of Decision-making Process

Steps	Rational Model	Our Proposed Solutions
1.	Monitor the decision environment	Monitor organization model for setting the decision model
2.	Define the problem about which a decision has to be made	Define the problem about which a decision has to be made
3.	Diagnose the problem	Diagnose the problem into four specific analyses: goal analysis, fact

		analysis, dimension analysis, measure analysis.
4.	Identify decision alternatives	Identify decision alternatives by broaden the possibility of information granularity.
5.	Analyze alternatives	Analyze alternatives for complying with data sources available.
6.	Select best alternatives	Provide information for all alternatives or selective.
7.	Implement the alternatives	Design the DW structure and related ETL processes.
8.	Evaluate the decision	Refinement the DW and ETL processes design.

Based on Table 3.2, not all steps in the proposed solutions are implemented in decisional modeling of requirement analysis. Steps seven and eight are involved in the conceptual design of the ETL processes that also contain the DW models. Details on how to implement the decisional model analysis are explained in Chapter 6.

3.6.3 Socio-Technical Theory

Socio-technical system (STS) concept is not new, and it has been studied for long time to account for the organizational and social context in which a software system is designed and operated. The STS is regulated by organization rules, business process, and laws by the authority (Sommerville, 2007). Since STS comprises software, hardware, people, environment, etc., then the requirements and process of gathering the requirements are also tedious and complex. Therefore, a developer perspective in requirement analysis method according to STS phenomenon as a new definition of modern information systems is proposed. This phenomenon is complying with the nature of DW systems that was noticed earlier and

requiressupport from organization, decision-maker, and developer to produce the successful DW systems.

The notion of STS is viewed from two different perspectives: social sciences (Ropohl, 1999; Walker, Stanton, Salmon, & Jenkins, 2008) and engineering sciences (Simon, 1996; Sommerville, 2007). Researchers with background in the behavioral sciences (i.e., sociology, psychology, anthropology) suggested fitting a technical sub-system and a social sub-system to operationalize the organization(Mumford, 2000, 2003). This researchis interested in focusingon the perspective of engineering sciences, which is more appropriate in software and requirement engineering. To visualize the technical and social components in STS, the STS theory is reviewed as illustrated in Figure 3.7 (Bostrom & Heinen, 1977).

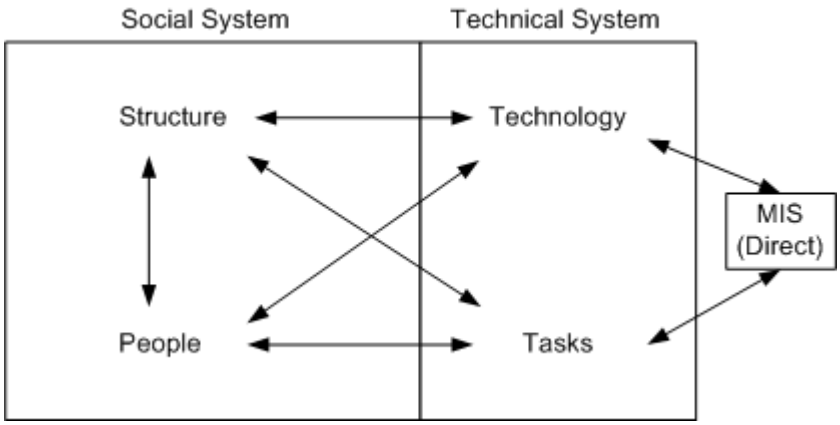


Figure 3.7: The STS Diagram Theory (Bostrom & Heinen, 1977)

The STS theory model in Figure 3.7 explains the relationship between technical sub-system and social sub-system actors, which are harmonized to produce the management information system. With respect to the theory, this research enhances the elements in the technical sub-system to consider the tasks played by developer in

harmonizing the software system, especially in DW systems development. The new model is based on the structure of STS approach while considering the relevance of developer roles as depicted in Figure 3.8. The three elements (i.e., technology, developer, and task) in a component technical sub-system are shown in Figure 3.8.

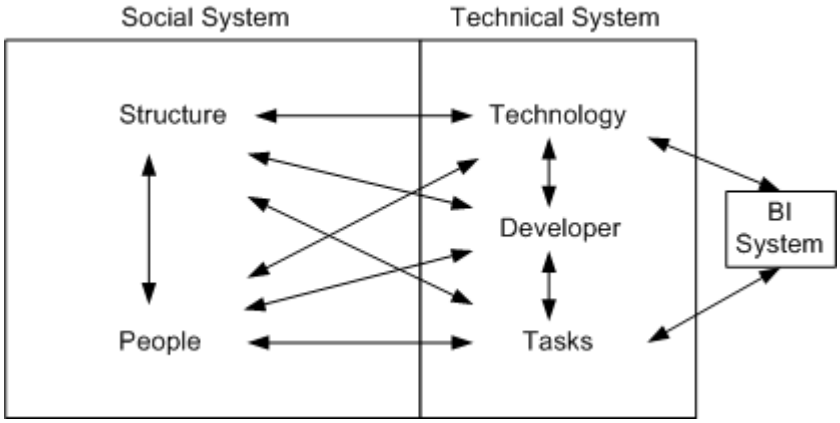


Figure 3.8: New STS Diagram Theory with Developer element

3.7 Related Works

The research efforts on developing software requirements (Nuseibeh & Easterbrook, 2000; Parviainen, Tihinen, Lormans, & Solingen, 2005) and DW requirements (Bruckner, List, & Schiefer, 2001; Giorgini et al., 2008; Winter & Strauch, 2004) according to the requirements engineering guidelines have been carried out. In short, the approach on DW requirements can be classified into *process-driven* (Kimball, 1996), *supply-driven*/*data-driven*(Inmon, 2002; Winter & Strauch, 2004) and *demand-driven*/*requirement-driven* (Winter & Strauch, 2004) approaches. The summary of the research works is shown in Table 2.1.

Table 3.3: The DW and ETL Processes Requirements Analysis Approaches

Researchers	**Approaches**
Kimball (1996)	Process-driven
Inmon (2002), Winter and Strauch (2004)	Supply-driven/Data-driven
Winter and Strauch (2004)	Demand-driven/Requirement-driven
Niedrite, Solodovnikova, Treimanis, and Niedritis (2007), Giorgini et al. (2008)	Goal-driven
Mazon, Pardillo, and Trujillo (2007), Farhan, Marie, El-Fangary, & Helmy (2012)	Model-driven
Romero and Abello (2007), Skoutas and Simitsis (2007)	Ontology-driven

Since DW requirements are toward information-centric, both approaches on supply and demand-driven are relevant for analyzing the user requirements. Moreover, supply, demand, and process-driven are to complement each other to support the complex requirements of DW systems as part of a socio-technical system (Golfarelli, 2010). Generally, the socio-technical system deals with a complex process of DW systems, which combines the philosophy of humans and machines to foster the requirement analysis process (Parviainen et al., 2005).

Understanding the DW requirements to model the DW and ETL processes is essential in building DW systems. Having a method to analyze the user requirements according to an appropriate model will help developers to design the ETL processes, and finally perform the ETL processes for implementing the DW systems. Performing the requirements to conceptual design of ETL processes is about

mapping the DW schemas to the data sources' schemas according to the analyzed user requirements. The analyzed user requirements have to reconcile according to data sources definition by using a method that is capable of modeling and structuring these terms in a unified and consistent way.

3.8 Conclusion

This chapter presents the functions of a requirement process in DW systems. The concepts of the requirement process are explained, emphasizing on the requirement analysis phase in DW systems development. The early phase of the requirement analysis process is elaborated in detail and their importance in modeling the ETL processes in DW system is highlighted. The research works on DW requirement analysis process are discussed and the issues related to this research problem are highlighted. Furthermore, the design-related issues in ETL processes design are explained and the proposed solution is briefly discussed.

In order to strengthen the direction of the solution with the research problems, this research explores the related theories about the subject matter. These theories are carefully explained and describe how these supportthe proposed DW requirement analysis solutions. Specifically, the organization, decision and socio-technical system theories are used in an integrated manner to frame the proposed solution. The proposed solution requires ontology to be utilized with the requirement analysis process. Chapter 4 discusses the ontology approach in detail.

CHAPTER FOUR– ONTOLOGY FOR ETL PROCESSES MODEL

This chapter explains the concept of ontology and how the ontology-based approach can be used in modeling and designing the ETL processes. The ontology approach in ETL processes modeling is highlighted and the related ontology-based research works for modeling and designing the ETL processes are provided. This chapter ends by summarizing the importance of ontology in the ETL processes design.

4.1 Introduction

The traditional ways of viewing data or information sources are through the three-schema level architecture: external, conceptual, and internal. Based on three-schema perspectives, the data integration as per specifications defined by business requirements can be modeled. Normally, data integration in DW is based on *global schema* that becomes a joint model, formally known as a dimensional/multidimensional model. However, the dimensional model is designed not to manipulate the data, rather to aggregate the data for further analysis and final delivery to the DW systems (Kimball & Ross, 2002; Alexiev et al., 2005).

Thus, current data modeling approach does not capture the concepts, definitions and relationships regarding the data aggregation well for smooth ETL processes implementation. This is where ontology becomes a suitable solution that systematically captures the activities and specifications in modeling the ETL processes with the capabilities to define a shareable concept that arises from the data integration and transformation mechanism (Skoutas & Simitsis, 2004). In web

applications, ontology provides the ways to retrieve and extract information based on the actual content of a web page and helps to navigate the information space based on semantic concepts (Sure, Angele, & Staab, 2002).

4.2 Ontology Concepts

In the fieldof information technology, ontology is used to define vocabularies or thesaurus and their meaning (semantic). The ability to define systematically the semantics explicitly and expressively in computer application has given more understanding not only to humans, but also can be interpretable by machine.Thus, ontology is used for formal representation of the ETL processes activities.

4.2.1 Definitions

There are several definitions of ontology given towards tightening the understanding of ontology in the real world. Meriam-Webster Online[6] defines ontology as: i) *"a branch of metaphysics concerned with nature and relations of being"*;and ii) *"a particular theory about the nature of being or the kinds of existents"*. However, most influential definition that describes the essence of ontology in Computer System (CS) was defined by Gruber (1993)as *"A formal, explicit specification of a shared conceptualization"*.

A *conceptualization* refers to an abstract model of some phenomenon in the real world, which identifies the relevant concepts of that phenomenon. *Explicit* means that the type of concepts and the constraints used are explicitly defined. *Formal* refers to

[6] http://www.m-w.com/

the fact that the ontology should be machine-readable and *shared* means that the ontology arises from a consensus between the several sources(Daconta, Obrst, & Smith, 2003; Shibaoka, Kaiya, & Saeki, 2007).In knowledge sharing, ontology is determined by the form of vocabulary definitions that are represented by type of hierarchies, classes and their sub-sumption relationships (Gruber, 1994). Indeed, relational database schemata can be seen as ontology by specifying their relationships and constraints among shared databases (Alexiev et al., 2005; Gruber, 1994; Meersman, 2001).

In general, ontology can be in a variety of forms and emphasis on vocabulary of terms and specification of their meaning. This includes definitions and specifications of inter-related concepts in the domain and constrains that affect the possible interpretations of the terms (Uschold, King, Moralee, & Zorgios, 1998). Guarino(1998)defined ontology as *"A set of logical axioms designed for the intended meaning of a vocabulary"*. Ontology is defined by logical characterization through axioms, where axioms are referred to as sentence or proposition that is not proven and is considered as self-evident. Jarrar(2005)defined ontology as a shared understanding (semantics) of a certain domain, axiomatized and represented formally as logical theory in computer resources. The essences of sharing semantics in inter-related concepts were inspired by the notion of information sharing in various approaches.

The term *ontology* is rooted in Greek history and has a long history in philosophy that refers to the subject of nature or existence. It has been adopted in various fields of research such as philosophy (Smith, 2003), linguistics (Kerremans, Temmerman, &

Tummers, 2003), logic (Baader, Horrocks, & Sattler, 2005), and computer science. In computer science, the research communities can be divided into two interest groups: i) Artificial Intelligence (AI) group that is committed to building shared knowledge bases; and ii) database group that is committed to building conceptual data schemas called semantic data model (Bekke, 1992; Fonseca & Martin, 2007).

Both disciplines contribute to the development of computer system application in various areas such as e-commerce, bioinformatics, e-learning, database design, software engineering, information access and retrieval, and semantic web (Jarrar, 2005; Meersman, 2001). In particular, the importance of ontology has been recognized in knowledge representation, natural language processing (NLP), knowledge management (KM), multi-agent system (MAS), intelligent integration of web resources and databases, as well as cooperation of distributed enterprise application and web services.

Ontology as knowledge representation is highly suitable for representing the ETL processes operations that are organized data in various sources. Without depending on the data sources structures and implementation strategies, the ontology is used to integrate heterogeneous databases. Therefore, the applicability of ontology plays essential roles in modeling the DW systems, especially in the ETL processes design.

4.2.2 Languages and Tools

The ontology tools facilitate the storing and accessing the content of ontology documents. This is an important step towards offering an efficient resource discovery

of ontology contents (Guarino, 1998). The ontology can be generic like WordNet[7] or very broad scope like Cyc[8] or can be a domain dependent covering the concepts related to a particular domain such as academic ontology (covering the concepts related to academia) or financial ontology (covering the concept and relation about financial) and legislation against financial fraud (Zhao, Gao, & Meersman, 2004).

Many tools or ontology servers for ontology building have been developed by the user communities (Ahmad & Colomb, 2007; Denny, 2004). It significantly contributes to the benefits of a web-based system for adding meaning to web documents and enabling the meaning to be used by the applications, agents and intelligent system. The use of ontology in this context requires a well-designed syntax compatible with web language technologies such as eXtensible Markup Language (XML) and Resource Description Framework (RDF). XML is a tag-based language for describing document structures, whereas RDF is an XML application that is customized for adding meta-information to web documents (Fensel, 2004).

The ontology language is presented in the layered stack of the semantic web tower that was envisioned by Tim Benners-Lee who developed the World Wide Web (WWW) technology and promoted the idea of semantic web technology (Patel-Schneider & Fensel, 2002). The semantic web technology is developed with technological protocols and social convention values for a universal internet platform to the users (Berners-Lee, Hall, Hendler, Shadbolt, & Weitzner, 2006). This semantic web tower is shown in Figure 4.1.

[7] http://wordnet.princeton.edu/
[8] http://www.cyc.org/

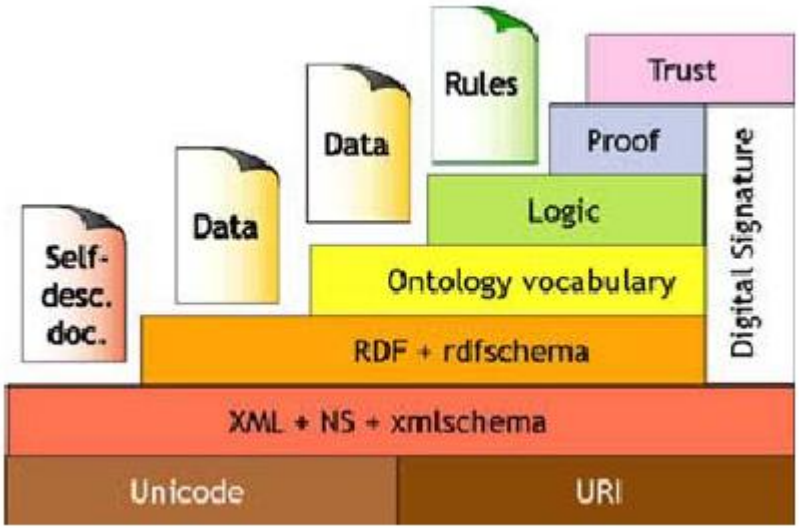


Figure 4.1: The Semantic Web Layer Tower

The semantic web tower shows the ontology vocabulary or termsas positioned in the semantic web architecture and provides the meaning by the RDF or RDF Schema (RDF-S). One can examine the connection between different terms in more advanced ways by definitions in the ontology other than RDF-S. An integrated heterogeneous data term can be easily connected in the information sharing architectures by utilizing the ontology technology in a practical way.

Several web ontology languages have been developed for the past ten years. For example, in Europe, they have developed Ontology Inference Layer (OIL) and the United States also has developed a similar project called Distributed Agent Markup Language (DAML). These two projects have been combined into a merged ontology language known as DAML+OIL. The work on syntactic standardization has already approved an ontology language based on DAML+OIL, and is known as Ontology Web Language (OWL) (Antoniou & Harmelen, 2003). Another language that enables

the exchange of ontologies for molecular biology was developed and is known an eXtended Ontology Language (XOL) (Karp, Chaudhri, & Thomere, 2000). However, OWL is a standard ontology language that is capable of expressing and representing ontologies in any domain and almost supported by the ontology tools nowadays.

4.2.3 Classification of Ontologies

Type or classification of ontologies is explained by the degree of generality of the shared conceptualization (Guarino, 1998; Jasper & Uschold, 1999). This is because of the difficulty to obtain consensus among the domain experts, users and developers to formalize the concepts or terms of ontology. Generally, three types of ontologies based on their generality have been classified (Guarino, 1998):

- **Top-level ontologies** or foundational ontologies present the general concepts such as a space, time, event, universe, which are not dependent on a particular problem domain or task. This ontology aims to support the large communities of users and application such as web-based surfers. The examples for these ontologies are Cyc[9](Lenat, 1995), Wordnet[10](Fellbaum, 1998), DOLCE(Gangemi, Guarino, Masolo, Oltramari, & Schneider, 2002), DMOZ – Open Directory Project[11] and so forth.
- **Domain ontologies or Task ontologies** present the generic concepts related to a specific domain or task such as university, students or staff. Furthermore, it can be specializing in a concept introduced in a top-level ontology. Example for this ontology is Enterprise Ontology for Business Enterprise (Uschold et al., 1998).

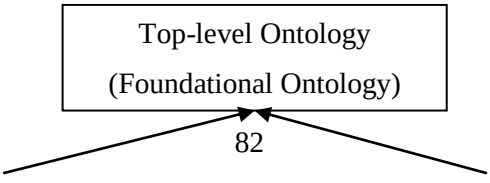
[9] http://www.cyc.com/
[10] http://wordnet.princeton.edu/
[11] http://www.dmoz.org

- **Application ontologies** are the most focused and specific ontologies. The concepts described correspond and depend on the domain entities or on any particular tasks. In other words, application ontologies are specialization of domain or task ontologies underlying on the implementation of their application. Examples for these ontologies are Legal Decision Support System (Zeleznikow & Stranieri, 2001) and Government Budgetary System (Graciela, Ma. Laura, & Omar, 2006).

The generality of ontology classification explains the *expressiveness* of the ontology spectrum that can be categorized into (Noy & McGuinness, 2001):

- Controlled vocabulary – a list of terms;
- Thesaurus – provide relations between terms (e.g., synonyms, homonyms)
- Taxonomy – explicit hierarchy of relationships (e.g., generalization)
- Frames – a class containing properties, subclasses and instances.
- Value restrictions – values of properties are restricted.
- General logic constraints – values are a constraint by the logic formula.
- First-order logic constraints – values are a constraint by a first-order logic formula.

The classification of ontology and their relationships can be viewed as hierarchy structure in Figure 4.2 (Guarino, 1998).



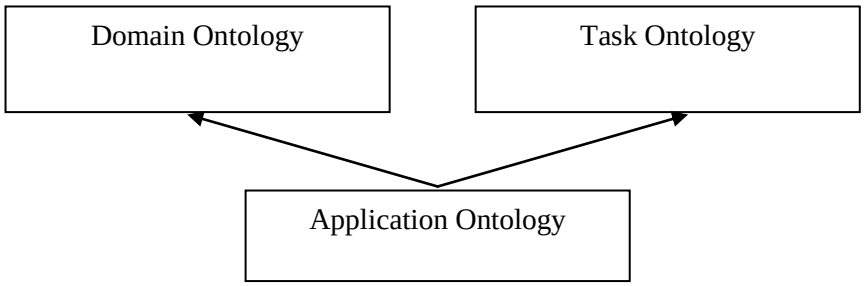


Figure 4.2: Classification of Ontologies and Their Relationships

4.2.4 Ontology as Data and Process Modeling

The efficacy of ontology in modeling the data in static or dynamic mode has been studied progressively since the use of ontology in developing information system. The dynamic mode of a data model can be referred as ETL processes in DWsystems. The data modeling is an activity to construct data or process specification in a particular domain by using specific data modeling language. Section 4.2.4.1reviews the roles of ontology in modeling the database and their process in integration and transformation activities.

4.2.4.1 Relational Data Modeling

The aim of database design is to capture as much as possible the contents, relationships, and constraints of the data that implies the user requirements in the real world. In order to facilitate the design process, the developers need to model the data structure closely to the user requirements so that it can be meaningful and understood by the users and developers (Bekke, 1992; Halpin, 2001; Ponniah, 2007). Nowadays, the relational database system has become the dominant data-processing software and

de-facto standard for structured information storage technologies (Alexiev et al., 2005; Connolly & Begg, 2005) and traditionally as a foundation for database modeling.

The emergence of object-oriented database system and capabilities of database systems has been continuously enhancing the functionality of database systems. However, the database modeling approach based on the three-schema architecture consisting of external layer, conceptual layer and internal layer for supporting the different views of users is not capable of capturing the true semantics of user requirements. Overcoming this incapability requires accommodating more knowledge of the real world to better understand the user requirements(Bekke, 1992; Fonseca & Martin, 2007; Storey, 1993). Indeed, the conceptual layer or conceptual schema is the main modeling technique of a database system that captures the concepts and their relationships to present the semantics in the real world of requirements.

The semantics presented in relational database modeling is based on mathematical algebra that is complied with relational database theory (Codd, 1979). Several methods such as ER modeling techniques (Chen, 1976) and UML (Booch et al., 1999) have been developed to model the database system. However, the modeling artifacts are not sufficient to express all the requirements of a database model.Therefore, additional constraints called business rules are applied. These business rules specifications define the restrictions of the data model and can be stated formally as predicates of the entities and relations. However, the nature of

conceptualization and the vocabulary of a data model are not the priority to be shared by other applications (Spyns, Meersman, & Jarrar, 2002).

In information sharing environments, the data model is based on data integration and transformation that is developed based on various approaches. Based on the survey by Rahm and Bernstein (2001), most of these approaches are based on schema mapping on a different data model. A schema mapping or matching refers to the integration of database conceptual schema that is derived from the external schema as presented by the user requirements. The integrated model concept is based on LAV or GAV approaches that depend on the information sharing architecture (e.g., EAI, EII, or DW).

In DW architecture, the schema matching approach is useful to design the data integration and transformation mechanism (Rahm & Do, 2000). Since the conceptualization and the vocabulary of a data model are not shared by heterogeneous data sources, then it is difficult to model the integrated data, especially on handling the semantics heterogeneity problems. Therefore, a modeling approach that considers the semantics issues in the artifacts of DW design should be the suitable solution for these problems.

4.2.4.2 Ontology as Data Modeling

Ontology is used to describe the concept and the semantics of a certain domain and express them formally as logical theory in computer resources, which enable them to model the data in database environments. A database schema can be seen as ontology

and integrated model of a database system (Alexiev et al., 2005; Skoutas & Simitsis, 2006). However, the capability of ontology to model a database for knowledge sharing is questionable, since typically database schemas produceda limited set of knowledge (Alexiev et al., 2005). In case of DW systems, schemas are developed from various types of a data sources model. Thus, shared knowledge representation through DW systems is possible, and modeling the DW using ontology is considered the most applicable in the context of information sharing.

Based on the definitions, ontology model deals with the *concepts* that are understood by human in a particular body of knowledge or subject or domain area such as a human resources (HR) domain. For example, the ontology model for HR domain is presented in a graphical form as shown in Figure 4.3. The ontology is graphically presented as relationships among the main concepts in HR domain (Daconta et al., 2003). These concepts and their relationships are usually implemented as classes, relations, properties, attributes, and values. Person, Employee and Organization are implemented as classes, whereas *is a employee_of*, *managed_by*, and *manages* are defined as relations. Each of the classes contains properties or attributes such as address, name, birthday, and ssn in Person class. Furthermore, each attributes has values or ranges of value.

Essentially, the ontology mechanism is capturing the meaning of a particular domain that corresponds from the human knowledge such as expert users in the HR system. This meaning is also commonly referred to as *semantic*and supposed to be understood by users and machines. Moreover, the ontology structure can be developed and

formalized by the particular ontology languages and tools (as explained in Section4.2.2). Some of the ontology tools have been commercialized and used for ontology-based application system. However, there is no commercial tool that embeds the ontology-based approach for designing and developing the ETL processes or DW systems.

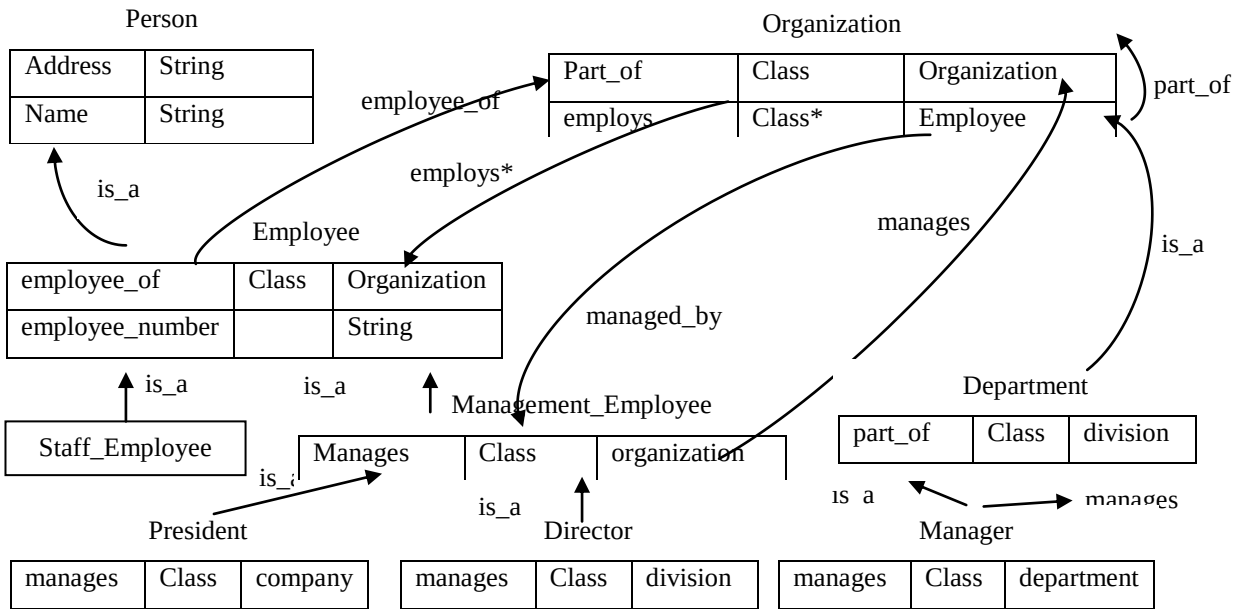
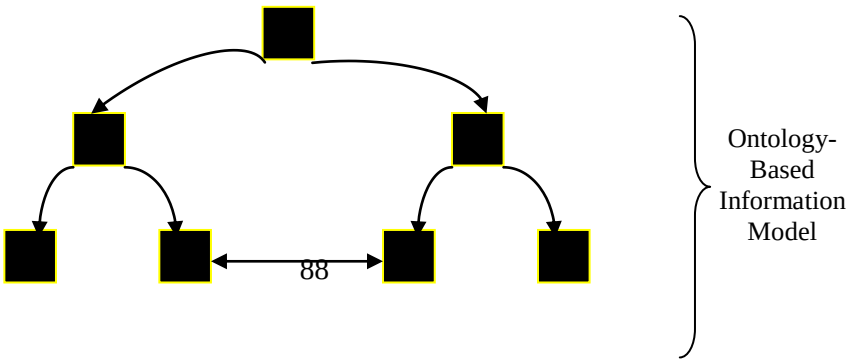


Figure 4.3: Graphical Ontology examplefor HR

As mentioned previously in Section 4.2.4.1, typical data model is not sufficient to express all the user requirements because the data schemas are based on lexical form (relating to the words or vocabulary) that is obtained from the conceptual schemas (normally expressed in ER, UML, ORM). Some of the information about roles, and concepts involved are missing because of the flattening of information sources, which are described from a table structure that basically used standard algorithms (Halpin, 2001; Ponniah, 2007; Sumathi & Esakkirajan, 2007).

Thus, using ontology to model the data, especially integrated data, is much closer to the user requirements. Furthermore, ontology is suitable for modeling the conceptual database system because (Alexiev et al., 2005): i) their intended use is not limited to some particular applications; ii) their expressiveness becomes high because it captures knowledge domain in more detail; and iii) their ability to be machineprocessable for database implementation. Therefore, suitable ontology-based model that integrates various data models need to be built in order to support the requirements of information sharing environments (Ta'a, Abdullah, & Norwawi, 2008). This is shown in Figure 4.4.

In general, Figure 4.4 depicts the relationship between three main components: i) ontology model, ii) data model, and iii) database schemas in ontology-based information sharing environments. In order to establish the linkages between the components, a mapping mechanism needs to be developed. The mapping mechanism should be able to synchronize the integration of database instances with ontology instances in order to maintain the integrity of information provided (Buccella et al., 2003; Cui & O'Brien, 2000). Detailed explanation on these mapping mechanisms is discussed in the next Section 4.2.5.



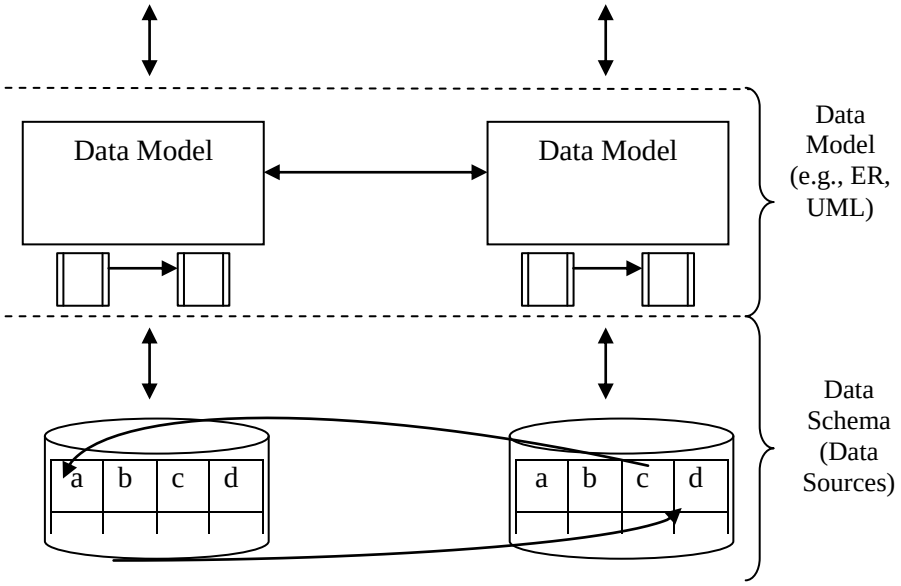


Figure 4.4: Ontology-based model for Information Sharing

4.2.5 Ontologyfor Data Integration and Transformation

Referring to Section2.3.3, the mechanism of data integration and transformation in information sharing architecture is based on LAV and GAV theories. The LAV and GAV approaches have been used to integrate the heterogeneous data using ontology. In the new millennium,this integration approach, known as *semantic information integration* or *ontology-based data integration,* has gained more attention by researchers such as Cui and O'Brien(2000),Brisaboa, Penabad, Places, and Rodriguez (2002), Buccella et al.(2003), Dou and LePendu(2006), Alexiev et al. (2005),Aparício, Farias, and Santos (2005),and Cure and Jablonski(2007).

Ontology-based approach is based on the semantic data model and aims to integrate the data according to the semantic level of the data involved. Two main tasks involved in the processes are: i) building the ontology; and ii) establishing the

mapping between ontologies. According to Alexiev et al. (2005), ontology building can be based on a local model (local ontology based LAV theory) and global model (global ontology based on GAV theory). Meanwhile, a mapping approach can be established based on: i) one-to-one mapping, ii) single-shared, and iii) clustering. One-to-one mapping is created based on pairs of ontologies, while a single-shared ontology is created as central ontology and other ontologies are mapped with them. Ontology clustering is created on similarity of concepts in different agents, which are organized in hierarchical fashion.

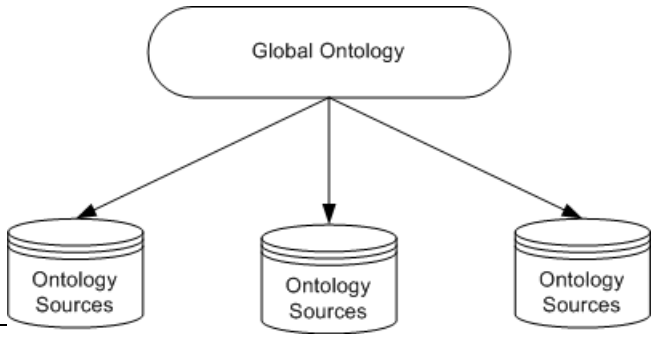
In another approach, Buccella et al.(2003)presented global and local ontology and defined the mappings between them. The data integration takes place when the mappings between the concepts defined in global ontology, and local ontology are established. Cui and O'Brien(2000) developed the ontology-based techniques called Domain Ontology Management Environment (DOME) in order to support the development of a *one stop knowledge shop* for enterprise information. Furthermore, ontology for integration of heterogeneous database was proposed by Dou and LePendu (2006), and Aparicio et al. (2005).

Gardner(2005) proposed the ontology and semantic integration in pharmaceutical industry. In web-based application, Maedche et al.(2002) proposed a comprehensive framework for web information integration called Semantic Portal (SEAL). Technical architecture of SEAL is derived from the Kalrsruhe Semantic Web and

Ontology Infrastructure (KAON)[12], which promotes the building of ontology for business application.

In DW environment, ontology has been used in DW, OLAP and Data Mart application in the notion of data sharing from the heterogeneous data sources by Priebe and Pernul(2003), Sell et al.(2005), Cao et al.(2005), Toivonen and Niemi(2004),and Skoutas and Simitsis(2006). Researcherssuch asNimmagadda, Dreher, and Rudra(2005)developed Petroleum DW systems by using ontology for knowledge mining process. An effort to automate the DW design process by using ontology was proposed by Romero and Abello (2007). However, only Skoutas and Simitsis (2007) works have extended the DW design to the ETL processes. Nevertheless, this only concentrates on ETL processes model and optimization, but is not related to ETL requirements analysis.

Based on the ontology-based information integration survey by Wache et al.(2001), there are three kinds of ontology architecture used: i) single ontology; ii) multiple ontology; and iii) hybrid ontology. These architectures are depicted as in Figure 4.5, 4.6 and 4.7, respectively.



[12] http://kaon.semanticweb,org

Figure 4.5: Single Ontology Approach

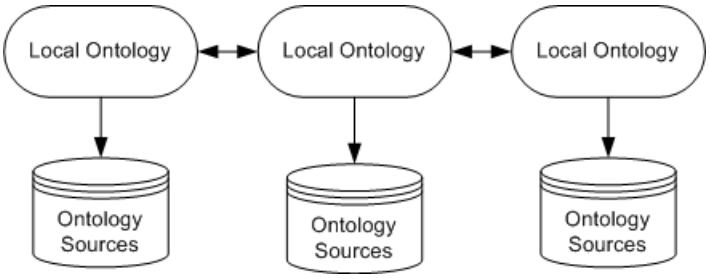


Figure 4.6: Multiple Ontology Approach

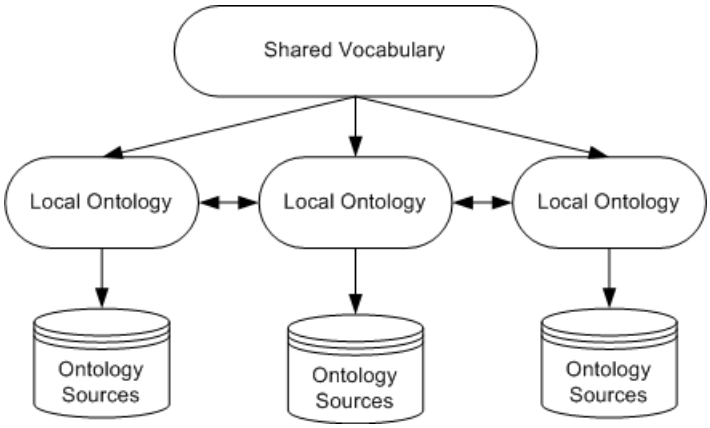


Figure 4.7: Hybrid Ontology Approach

From these integration approaches, many tools have been developed by the researchers and practitioners with different purposes of information integration and underlying different ontology languages used in the application. The comparison between these tools is presented in Table 4.1, which summarizes the survey works of Alexiev et al. (2005).

Many tools have been developed for facilitating the integration of data using an ontology-based approach. However, no specific tools or framework can support the

integration of heterogeneous data sources for the DW systems. Indeed, the semantic data model for DW systems comprising ETL processes functionality is not yet available. To implement the ETL processes with semantic consideration, an understanding on how the semantic data is being modeled in the DW systems and explaining the movement of the data model in ETL processes with the underlying mapping of data sources to DW is required.

Table 4.1: Comparison of Ontology-based Data Integration Tools

Criteria	COG	MOMIS	ONION	OBSERVER	KRAFT	PROMPT	Chimaera
Mapping Pattern	Single-Shared	Single-Shared	Hybrid	Multiple (one-to-one)	Hybrid	Single-Shared	Multiple
User Model	Global	Global	Global	Local	Local	Global	Local
Mapping Support	Class, Property, Value transformation	Class, Property, Constraints	Class, Property	Class, Property, Value transformation	Class, Property, Constraints, instances	-	-
Inter-operability	RDBMS, XML, COBOL, Wrappers	Custom Wrappers	n/a	Custom Wrappers	XML, Custom Wrappers	Any language supported by Protégé-OWL	Any language supported by Ontolingua

Notes: COG – Corporate Ontology Grid, MOMIS – Mediator envirOnment for Multiple Information Sources, OBSERVER – Ontology-based system enhanced with relationships for vocabulary heterogeneity resolution, KRAFT – Knowledge reuse and fusion or transformation, PROMPT – Formalism-independent algorithm for ontology merging and alignment, ONION – Ontology Composition.

Thus, the modeling of ETL processes aims to reconcile the business requirements with available data sources toward the final DW structure in order to provide better knowledge representation in DW application for decision-making process. The approach for designing the ETL processes and how the ontology-based approach was applied is discussed in Chapters 5 and 6.

4.3 Ontology Approach for Modeling the ETL Processes

ETL processes are the main activity to integrate and transform the required data sources to the intended DW. The integration and transformation processes are mechanisms that deal with the mapping of data sources attributes to the DW attributes within the scope of ETL processes operations (Kimball & Caserta, 2004; Patil, Rao, & Patil, 2011; Simitsis, 2004). In order to define the DW schemas and specifications of the mapping mechanism, the ETL developer needs to understand the user requirements and schemas of data sources. In the current approach used, the ETL developer needs to collect and analyze the user requirements together with the data sources to identify the DW structure and their intentional mapping by using techniques such as *Logical Data Map* (Kimball & Caserta, 2004) as shown in Table 4.2.

Table 4.2: Logical Data Map Template

Data Target (DW)					**Data Sources (Heterogeneous)**				**Transformation (ETL activities)**
Table Name	Column Name	Data Type	Table Type	SCD Type	Data-base Name	Table Name	Column Name	Data Type	
Dim_tb1	Tb1_f1	Int	Dim	1	DB1	Tb1	f1	Int	Select * from tb1
Dim_tb1	Tb1_f2	Char	Dim	1	DB1	Tb1	f2	Char	-
Dim_tb2	Tb1_f1	Date	Dim	1	DB2	Tb1	f1	Date	Conversion (f1,f2)
Dim_tb3	Tb1_f1	Char	Dim	2	DB2	Tb2	f2	Char	-
Fact_tb1	Tb1_f2	Int	Fact	-	DB2	Tb3	f3	Int	Count (tb3.f3)

It can be argued that, a good and practical approach in designing the ETL processes need to build on the user requirements, while reconciling the semantics of business terms with heterogeneous data sources within the setting of ETL processes operations. However, it is not easy to resolve the semantic heterogeneity problems, since the data sources schemas were developed independently and therefore, various data models need to be used to present the same overlapping concepts (Halevy,

2005). Clearly, the design of ETL processes is mainly driven by the semantics of data sources and DW, which is always derived from the user requirements of DW systems (Skoutas & Simitsis, 2006). This requires a proper semantic framework to guide the design of ETL processes.

4.3.1 Semantic Framework of DW System

The ETL processes are not just a process of transforming the data for DW systems, but in many cases, it supports the operational processes or keeps operational systems working synchronously (White, 2006). Therefore, the modeling and designing of the ontology-based ETL processes is a solution for resolving the semantic heterogeneity problems and close the gaps between business requirements and DW for providing better quality information. In order to develop the ontology-based ETL processes model, a suitable framework that supports the semantic component of the ETL processes is required, comprising the early phases of the DW systems development.

As depicted in Figure 2.1, the typical framework for DW development contains five main components: i) user requirements, ii) data sources,iii) ETL processes,iv) DW and data staging area,v) DW or BI application. Occasionally, the DW and data staging area are referring to the same component. The framework proposed by Inmon (2002) haslarger scope as it comprises ODS, DW, data marts, DSS applications, exploration warehouses, data mining warehouses, and alternate storage. This framework is known as Corporate Information Factory (CIF). However, all these frameworks do not consider the semantic elements in rationalizing the differences of data structures. Thus, a semantic-based framework of DW development is necessary

in order to bridge the terminological inconsistencies underlying business semantics in DW model (e.g., dimension and fact attributes) for smoothing the design of ETL processes.

The semantic framework for DW systems development is derived from the extensive review of literature in this area (Kimball & Caserta, 2004; Lujan-Mora, 2005; Schreiber, 2003; Simitsis, 2004). It was developed based on the notion of Semantic Information Management (SIM) proposed by Schreiber and Gonchar(2004). The SIM is inspired by the semantic web vision, which aims to address the core problem in modern information enterprise by capturing the precise meaning of data in common agreed-upon business terms. Furthermore, ontology as part of semantic web components, will tackle the major problems of data integration and transformation in typical ETL processes by following the semantic framework of DW systems development. This semantic framework is shown in Figure 4.8.

The semantic framework of DW systems refers to the capabilities of DW systems to overcome the semantic heterogeneity problems in the implementation of ETL processes and bring the business requirements closer toward the intended DW. This is important due to the current scenario of DW systems that support the tactical and strategic users (White, 2006). The process for integrating and transforming the data sources to the intended DW can be systematically modeled and designed. Possibly, the generation of ETL processes specifications can be automated by a programming language (e.g., JAVA, RDF, OWL) that permits reasoning to be inferred to the ontology (Fonseca & Martin, 2007; Skoutas & Simitsis, 2007).

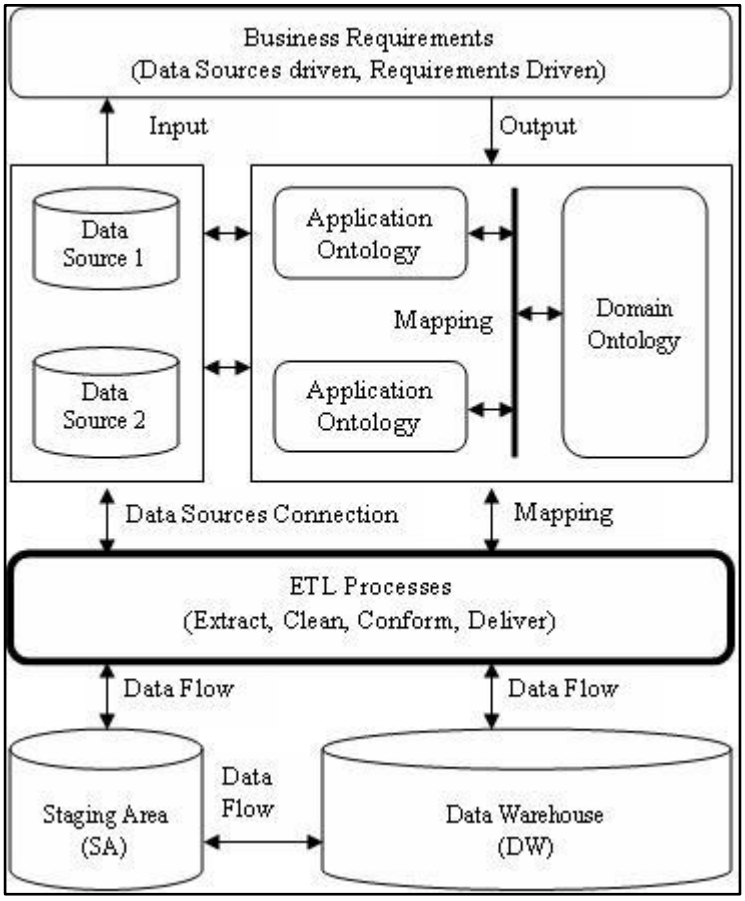


Figure 4.8: Semantic Framework for DW Systems Development

4.3.2 Business Semantic for ETL Processes

According to Kimball (2006), data integration means reaching an agreement on the meaning of data from the perspective of two or more data sources (heterogeneous). With this agreement, the results from two data sources can be combined into a target database (e.g., DW or OLAP) for further analysis. Reaching an agreement is all about resolving the problems of semantic heterogeneity and at the same time, facilitating the transformation challenges in DW systems. To overcome these problems, the explicit meaning for each of the semantic data sources should be

defined clearly and the definition of business needs must be presented accordingly through ontology.

The capability of ontology to provide name, description and relationship of the entities for specific domain will enable the data sources to be accepted and understood by various users and applications. Thus, the ontology approach can resolve the semantic heterogeneity problems by providing the explicit description of the data sources in DW systems.

4.3.3 Ontology-Based Conceptual Modeling of ETL Process

In designing the conceptual modeling of ETL processes, this research needs to understand the aims of DW by analyzing the user requirements and data sources schemas that provide the data for the DW (Lujan-Mora, 2005). The conceptual modeling portrays a DW system domain at a high level abstraction using the terms and concepts that are familiar to the business users (Halpin, 2001; Olivé, 2007). The logical and physical model aspects such as database or programming structure need to be ignored. Therefore, the conceptual modeling of ETL processes can be simplified as depicted in Figure 4.9 (Ta'a, Abdullah, & Norwawi, 2008).

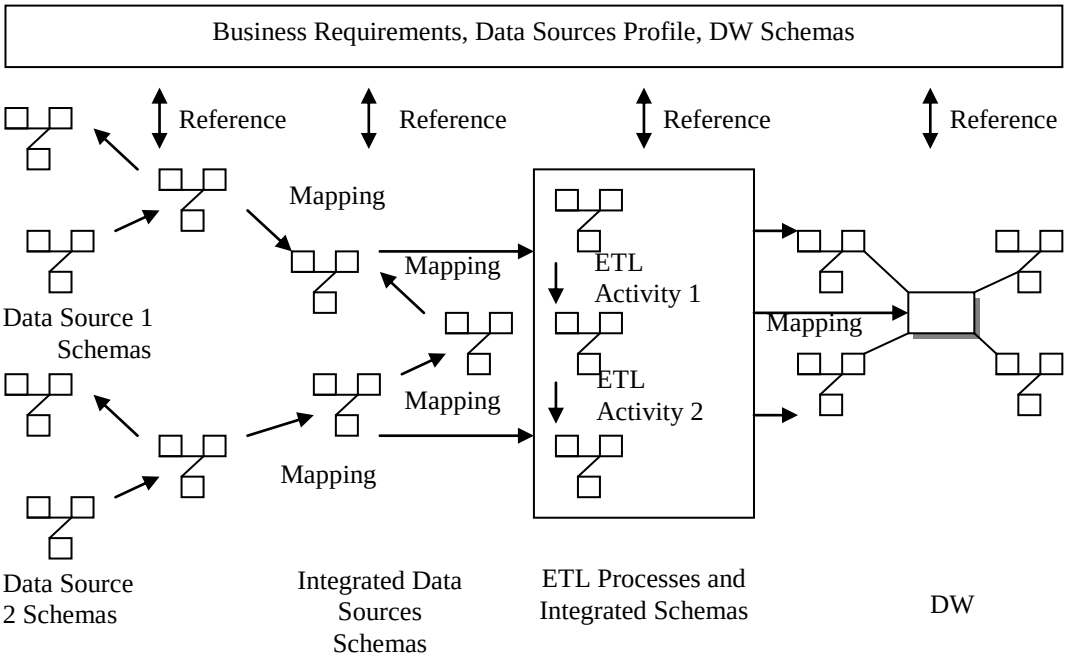


Figure 4.9: Conceptual Modeling of ETL Processes

Conceptual modeling of ETL processes (as shown in Figure 4.9) is the abstract view of the ETL processes that start with collecting and analyzing business requirements until the DW schemas. As mentioned earlier, the aim of modeling the ETL processes is to understand the mapping between heterogeneous data sources to the intended DW underlying the user requirements for decision-making. Thus, the right understanding and interpretation of user requirements in various concepts and terms need to be analyzed to obtain the correct specifications of the ETL processes.

The correct specifications of ETL processes enable the true mapping between data sources schemas to the DW schemas (Kimball & Caserta, 2004; Simitsis, 2004). However, the current approaches are not helping the designer to resolve the semantic heterogeneity problems during the modeling phase, and this creates difficulties in

designing the ETL processes. Therefore, this research proposes two-fold ontology and goal-oriented approach for a developer to model and design the ETL processes and tackle the design-related problems mentioned earlier. In Figure 4.9, the conceptual ETL processes is modeled with the assertion of goal-oriented and ontology as depicted in Figure 4.10.

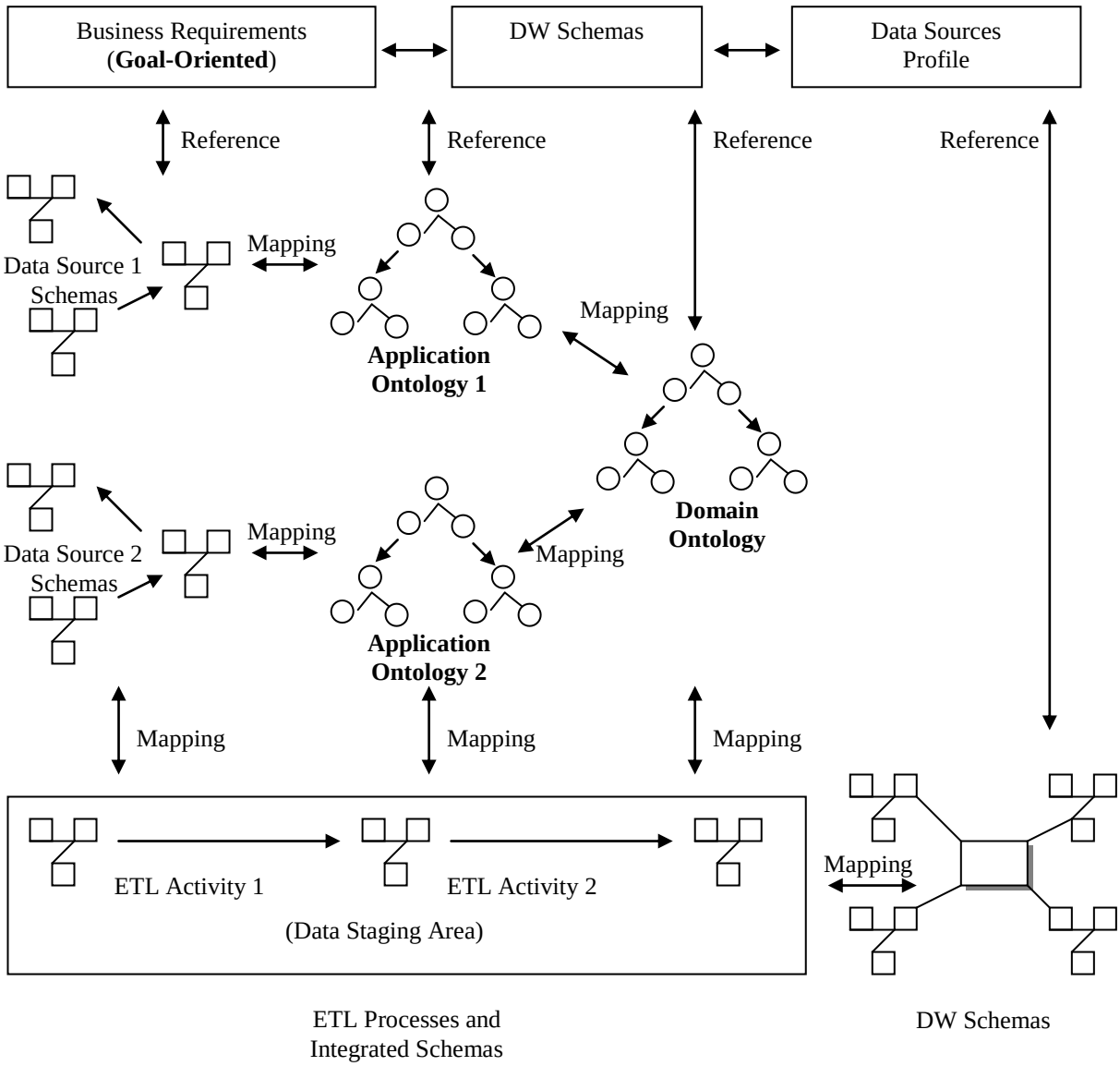


Figure 4.10: Ontology-based Conceptual Model of ETL Processes

Figure 4.10 shows the usage of ontology sources in defining the appropriate data sources from the references of user requirements underlying the various data sources schemas. The ontology for each of data sources (assume that the data sources are derived from the particular application such as Student Record System or HR System) is developed. The defined concepts for user requirements are mapped with the underlying data sources (classes, attributes and relationships). Then, all the application ontologies are mapped to the domain ontology, which contains more general concepts that refer to the applications involved. As a result, application and domain ontology are used to define the appropriate data sources, their related structures and format for defining the ETL specifications.

In the flow of ETL processes, the data sources schemas (from left to right) are changed over the ETL processes activities (e.g., extract, filter, conversion, and join) until the instances are loaded into the DW. The data sources that are treated by the ETL processes are mapped to the ontology and all these activities are taking place in the data staging area. Data staging area is not required if the data sources are directly mapped to the DW and vice versa (Kimball & Caserta, 2004). However, the ETL processes need to be maintained in the staging area for further implementation of the DW systems.

4.3.4 Ontology-Based Logical Modeling of ETL Process

In conceptual modeling, it is an abstraction of what are the components involved in the ETL processes (business requirements, data sources schemas, ontology sources, ETL activities and DW schemas) and the positioning of ontology in the modeling.

However, in logical modeling, the ETL processes activities are defined in detail, and how it works towards the implementation of the DW systems. Specifically, it describes how each of the components is mapped and guides the transition of required data sources to the intended DW. The activities of ETL processes need to be identified within the semantic framework as presented in Figure 4.8, which are specified as 'extract, clean, conform and deliver' (Simitsis, 2004; Kimball & Caserta, 2004; Lujan-Mora, 2005). The typical activities here can be summarized as follows:

i) **Identify and extract the data sources**. The data from the various sources are identified and defined for extraction. These data can be in various formats such as relational, XML, flat-file or plain-text. Most ETL tools today have already a built-in kind of *wrapper* that allows these data to be extracted by the *extracting* activity function. However, this research only focuses on the relational model because of their stability and practicality compared to other data sources such as flat files, unstructured XML files, and others.

ii) **Clean the data sources**. Normally, data cleaning process refers to data quality at the data sources system. Most organizations treat the data quality process as a different process with the ETL activities, since it involves discrete steps such as checking for valid values, ensuring consistency, removing duplicates and many more (Kim et al., 2003; Kimball & Caserta, 2004). However, some of these activities can be implemented by the ETL functions such as *filtering* and *conversion*.

iii) **Conform the data sources**. The conformation of data sources refer to the merging of two or more data sources. This is a significant step because it requires

agreement among enterprise in order to use the concept and measures in the problem domain. This is where ontology plays the roles in merging the data sources under the concept and business rules that have been defined by and agreed to by the owner of data sources. Some of the activities here are*aggregating, joining, merging* and others.

iv) **Deliver to the DW**. The delivery process refers to the final step in the ETL processes that involves loading the data into the structure schemas known as dimensional models or star schemas (Kimball, 1996; Kimball & Caserta, 2004). DM structure is design, mainly for the querying purposes and becomes the basis for OLAP cubes (provided by commercial ETL tools). Some of the activities involved here are *loading* and *creating surrogate keys*.

By including the ontology sources, this research proposes two additional steps to be implemented in supporting the ETL processes design as follows:

i) **Map the DW requirement with the ontology**. The concepts (i.e., facts, dimensions, measures, attributes) of DW requirements are mapped to relevant ontology sources. The possible activity here is *ontology mapping* or *ontology matching* with the relational database schemas.

ii) **Merge the DW concepts with the ETL processes**. The concepts of DW are mapped with the ETL processes activities (e.g., filtering, merging, converting, joining) for data sources to be transformed to the intended DW. With the composition of the ETL processes activity, the *ontology mapping* between DW

requirements and data sources possibly automates the ETL processes specifications generation.

Typically, the ontology-based ETL processes is defined as activities of extracting, filtering, converting, aggregating, joining, merging, surrogate keys, loading, ontology construction, ontology mapping and merging.In typical view of the ETL processes design, theETL activities can be implemented in series without necessarily following the ETL processes execution order. Based on the conceptual model in Figure 4.10, the logical model is presented by the ETL activities, which is illustrated the ontology-based ETL processes in detailin Figure 4.11(Ta'a, Abdullah, & Norwawi, 2008).

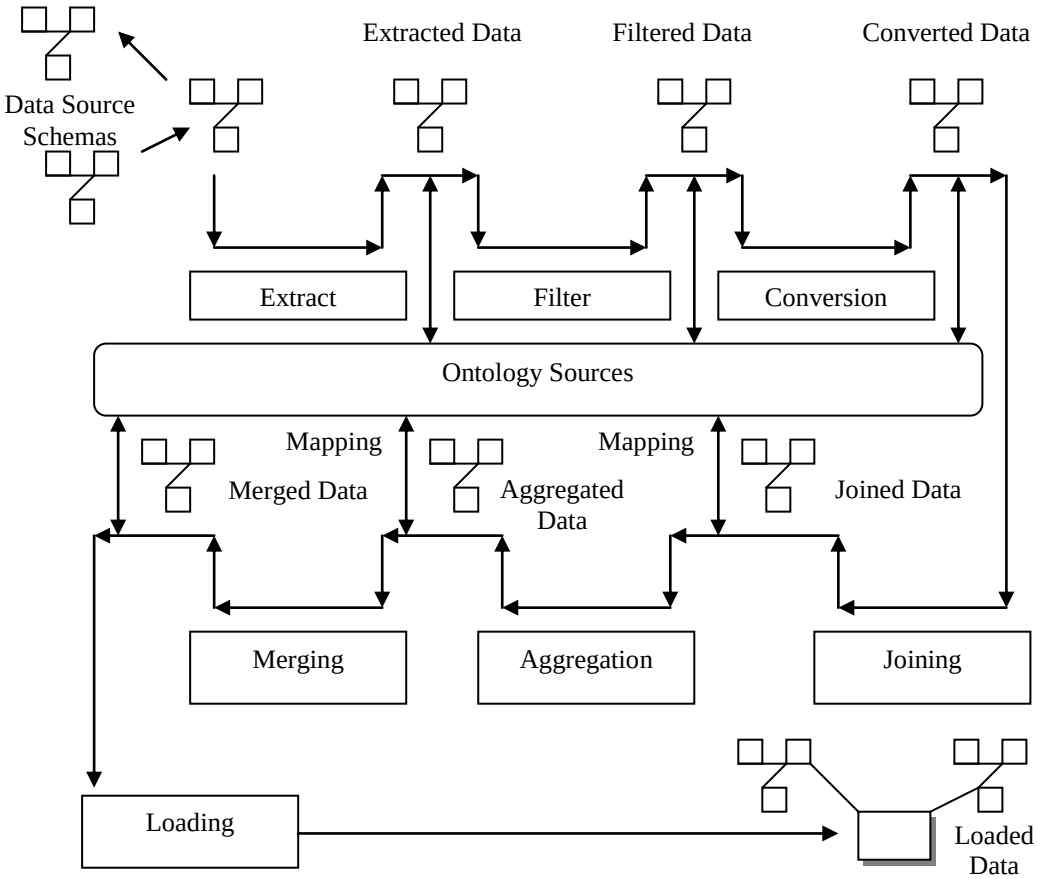


Figure 4.11: Ontology-based Logical Model of ETL Processes

Figure 4.11 explains the logical view of how the ETL processes are implemented from the earliest points of the process (data sources schemas) to final points of the process (DW schemas) with the typical ETL activities involved and mapping them with the ontology sources. Normally, the series of ETL mechanisms are not required to follow the typical process flow because it depends on the DW model (Simitsis, 2004). However, the ETL processes development is still in the semantic framework that was discussed in Section 4.3.1.

4.3.5 Ontology Development

Building the ontology becomes a necessary task in ontology-based applications. Until today, many approaches and tools were developed in order to support the development and maintenance of ontology. However, none of the approaches or tools really covers all aspects of ontology needs such as an integration paradigm, mapping mechanism, degree of automation, interoperability, visualization, evaluation, user model and versioning (Alexiev et al., 2005). This is because most of the approaches were mainly developed for particular projects or as academic exercises. Additionally, no methods have been accepted as a standard in developing the ontology sources (Noy & McGuinness, 2001). Nevertheless, some development methodology with tools helps a developer to develop and maintain the ontology accordingly.

4.3.5.1 Development Methodology

A well-known ontology development methodology is widely supported by different communities such as Application Knowledge Engineering Methodology (AKEM)

and METHONTOLOGY (Corcho, Fernandez-Lopez, Gomez-Perez, & Lopez-Cima, 2005). These methodologies are supported by ontology tools such as WebODE (Arpírez, Corcho, Fernández-López, & Gómez-Pérez, 2003), Protégé-2000 (Noy et al., 2001), and Developing Ontology-Guided Mediation for Agents (DOGMA) (Jarrar, 2005; Tang & Meersman, 2005). For the specific use of ontology, an ad-hoc method that is based on a specific model (i.e., ontology for data stores) is developed directly by ontology language model (e.g., RDF, OWL) (Skoutas & Simitsis, 2007).

Since this research is using ontology for modeling the ETL processes, the ad-hoc methodology supported by the Protégé-OWL tool (current version for Protégé-2000) is adapted that provides excellent ontology construction and maintenance. This methodology supports the evolution of ontology that allows refinements of ontology along the life cycle of the DW systems development. This is suitable for the nature of DW systems, which has to deal with the changes of user requirements that also reflect the ontology sources. Additionally, the ontology development guidelines from Noy and McGuinness(2001)are used to guide the developer because it is compatible with the Protégé-OWL tool.

The modeling of ETL processes is within the semantic framework of DW system, which requires methodological approaches in iterative processes. An iterative approach is important in order to comply with the nature of DW systems environment. Therefore, an efficient and easyto use ontology tool is highly required. The Protégé-OWL tool is considered suitable for this task. Details on the methodology and tools usedarediscussed in Chapter 5.

4.3.5.2 Ontology Construction

The ontology construction is a process to develop ontology (i.e., semantics capturing) based on application and domain of the DW systems. Specifically, the construction processes involve acquiring and structuring the knowledge domain (Corcho et al., 2005; Noy & McGuinness, 2001). Moreover, domain ontology contains the concepts in domain level (e.g., faculty, university) that implies the application ontology (e.g., student profile, CGPA). Thus, the construction of ontology can be divided into two types of ontology: i) application ontology; and ii) domain ontology.

Application ontology refers to the concepts defined from the underlying data sources that provide the raw data to the DW systems, whereas the domain ontology refers to the concepts defined from the underlying application ontology that provide the domain concepts for the information system.Moreover, the involvement of domain experts, application developer and business users is essential in order to develop the ontology (application and domain) for ontology-based application. Indeed, this task is an important step in development methodology for developing the ontology.

4.3.5.3 Ontology Mapping

Modeling the ETL processes will help the developer to develop, manage and maintain the complexity of back room activities. Besides understanding the data involved in the integration and transformation process, the developer should understand the connections between data sources and the intended DW, known as data or ontology mapping. The ontology mapping is an approach to specifying the translation between ontology and gives the meaning to the data residing in ontology sources (Cui

&O'Brien, 2000). It can be classified into three approaches: i) local ontology and global ontology (Beneventano, Bergamaschi, Guerra, & Vincini, 2003; Calvanese et al., 2001); ii) local ontology and local ontology (Maedche et al., 2002); and iii) ontology merging and alignment (Noy & Musen, 2000).

Many tools have been developed to support all the mapping categories as presented in Table 4.1. Thus, the same scenario takes place when using the ontology in the ETL processes, in which the ontology sources (application and domain) are mapped to the data sources, DW, and ETL processes. The data mapping needs to cater for the schemas level only.Schema level refers to the structure of ontology sources, whereas the instance level refers to the actual data or values for the particular schemas (Halpin, 2001; Ponniah, 2007). Using university student record system as an example, the StudentRecord(matricID, sname, sID, semester) is at the schema level, whereas StudentRecord("80000", "Azman Taa", "650901085529", "A0107") is at the data level. An example of this ontology mapping model is in Figure 4.12.

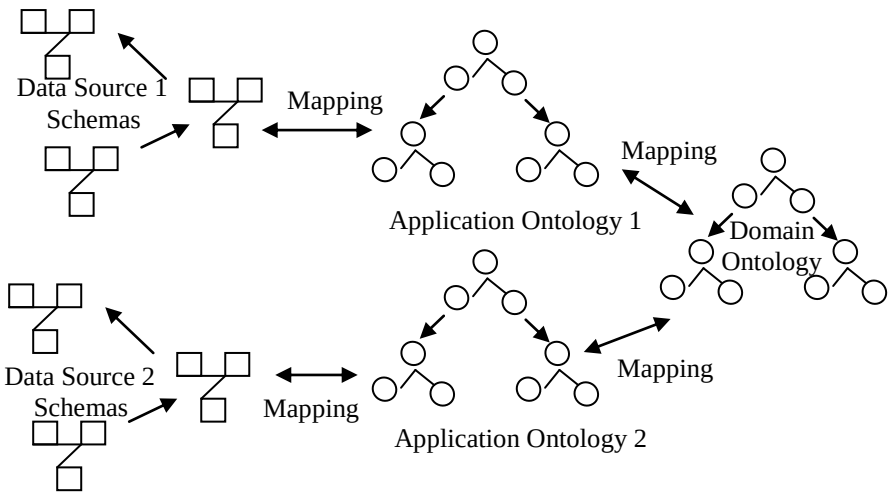


Figure 4.12: Ontology Mapping with Data Sources

The development of ontology for the specific application and the construction of domain ontology have created a concrete foundation of concepts that belong to the business requirements and data sources. Thus, the ontology model underlying the data sources schemas is mapped to the intended DW schemas underlying the composition of the ETL processes activities. These mapping specifications can be described by using *Logical Data Map* techniques (Kimball & Caserta, 2004)as shown in Table 4.3. The data sources element consists of database name, table name, column name, and data type. The DW element consists of table name, column name, data type, table type (fact or dimension) and SCD type (1, 2 or 3).

Table 4.3: Logical Data Map – An Example

Data Target (DW)					**Data Sources (Heterogeneous)**				**Transformation (ETL activities)**
Table Name	Column Name	Data Type	Table Type	SCD Type	DB Name	Table Name	Col. Name	Data Type	
FactRegister	TotalRegister	Number	Fact	n/a	Student	Student Record	matricID	Int	SUM(StudRecord)
DimProfile	Name	Vchar	Dim	2	Student	Student Record	sname	Char	Name=sname
DimProfile	IC_No	Char	Dim	2	Student	Student Record	sID	Char	IC_No=sID
DimProfile	Semester	Char	Dim	2	Student	Student Record	Semes-ter	Char	Semester=semester
DimProfile	TotalRegister	Number	Fact	n/a	Pelajar	Student Profile	Student ID	Char	SUM(StudRecord)

These elements were created as classes, properties, and relationships, which used the ontology mapping to integrate and transform the relevant data sources toward the DW. The ontology sources (application and domain) containing the data sources mapped with the DW schemas generate the ETL processes specifications. Indeed, the mapping approach aims to: i) determine the data in the heterogeneous sources, which belong to the concepts needed by DW; ii) implement the data integration and

transformation activities; and iii) determine the DW structure, which is ready to receive the data sources from the ETL processes implementation. This mapping model is shown in Figure 4.13.

To define and develop the mapping specification is a difficult and crucial task in ontology engineering works (Cui & O'Brien, 2000; Sung & McLeod, 2006). Detailed and accurate specifications are required to avoid losses of the information and increase the degree of schemas matching (Giunchiglia & Shvaiko, 2004). Thus, this will be the challenges since the semantic integration issues are still a long way from being resolved (Doan & Halevy, 2005; Halevy, 2005; Ziegler & Dittrich, 2004). Moreover, the other challenges are also in clarifying and positioning the mapping tasks in modeling the ETL processes.

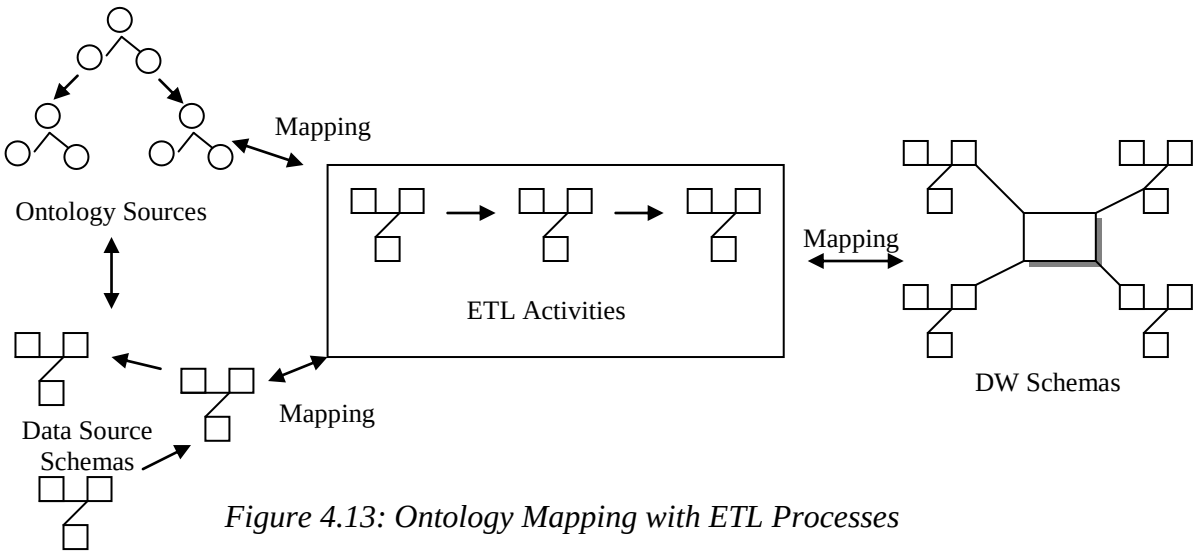


Figure 4.13: Ontology Mapping with ETL Processes

4.4 Related Works for Ontology-Based Approach

Several efforts have been proposed by researchers for applying the ontology in supporting and enhancing the typical process of DW components (i.e., data sources,

ETL, DW, OLAP, BI application). Priebe and Pernul(2003) developed the ontology-based approach for integrating three information sources from OLAP, DW and Document Management System (DMS) to provide the unified access on organization contents through an enterprise knowledge portal (EKP). However, the approach did not cover the ETL processes and only described the general architecture of EKP.

Sell et al.(2005) utilized the ontology to leverage the semantic web services based on the Internet Reasoning System (IRS-III) framework to support the semantic extension of BI analytical tools, where no ETL processes were involved in the proposed semantic web services. Cao et al. (2005) defined the hybrid ontology architecture to integrate the user profile, DW, OLAP, Data Mining and Enterprise Information System (EIS).

Each of the ontology sources captures the particular concepts, entities and business rules from the business perspectives. Then, an algorithm was developed for smoothing the mapping mechanism from one level to another which ultimately enhanced the BI application functions for the end users. However, this work did not describe the modeling process of such approaches, but essentially focused on defining the integration mechanisms for ontology service-based DW, OLAP, DM and EIS.

Furthermore, the modeling of ETL processes was not given much attention; rather it focused on the management of ontology-based services. Toivonen and Niemi (2004) described the semantics of data sources in the ontology model for allowing the data

integration from several data sources into an OLAP cube, and again no model elements were described for the DW or ETL processes. Moreover, no explanations were given on how the ontology structure for data sources can be transformed into the OLAP cubes.The only outstanding work on modeling and designing the ETL processes with ontology was carried out by Skoutas and Simitsis (2006, 2007). They used the ontology to facilitate the process of selecting relevant information from the available data sources and transform it to populate the DW. Moreover, an algorithm was developed to construct the domain ontology of discourse and determine the attributes mapping and ETL transformation for conceptual design of ETL processes.

Essentially, this researchaims to generate the ETL processes specifications by using ontology and hopes to facilitate the mapping process between the data sources and DW schemas. However, the method to model the ETL processes with ontology does not consider the important element of DW components (i.e., fact, dimension, measure), which begin with reconciliation of user requirementsand the data sources. Therefore, an approach to model and design the ETL processes with ontology is clearly reasonable to overcome this research problem.

4.5 Conclusion

This chapter explains the concept of ontology and how the ontology-based approach can be used in modeling and designing the ETL processes. The ontology roles in semantic representation are defined and elaborated. The chapter highlights the ontology solution in conceptual and logical ETL processes modeling and discusses the development of ontology data sources and DW requirements. To carry out the

design process, the semantic framework of DW system development is explained in order to scope the ETL processes modeling and presented as a unified view of ETL processes modeling by using ontology. Finally, the research works on ontology-based approach for modeling the DW and ETL processes are presented, highlighting the issues that were uncovered by the previous researchers. The next Chapter 5 discusses the methodology used in this research.

CHAPTER FIVE– RESEARCH METHODOLOGY

This chapter describes goal-oriented and ontology-based methodologies for guiding the design of the ETL processes. These methodologies present the methods to be used in developing the approach for achieving the research objectives of designing the ETL processes. The introduction of validation and evaluation process for modeling and designing the ETL processes areended the chapter.

5.1 Introduction

Previous chapters discussed thoroughly the issues and problems in modeling and designing the ETL processes modeling in the context of typical DW systems development. Based on the problems highlighted in data integration and transformation of ETL processes, the goal of this research is to facilitate the design of the ETL processes using goal-oriented and ontology-based approach. The goal-oriented approach is chosen for eliciting and analyzing user requirements because of its capability to understand the current organizational situation and relate the business goals to the functional and non-functional software components.

Although many requirement analysis methods support goal-oriented approach, this research adapts the Goal-Oriented Approach to Requirement Analysis in Data Warehouses (GRAnD) that was developed from well-accepted Tropos methodology and i* framework for software development (Yu et al., 2011; Bresciani et al., 2004; Giorginiet al., 2008). Tropos methodology has been applied for requirement analysis approach in the DW systems development. Based on the i* framework, the concepts

of agents/actors and social dependencies among agents such as goal, softgoal, task, resource, and other mentalistic notions on software development phases are utilized.

The ontology-based approachwas chosen because it is a more logical and practical solution for the semantic heterogeneity problems in information sharing environments. Moreover, ontology is used to reconcile the semantics and can be represented in modeling language that enables human and machine agents to understand it. Although there are many methods for developing and maintaining ontology, there is no specific method for modeling the DW and ETL processes by using ontology. However, this research has adapted an ad-hoc ontology model proposed by Skoutas and Simitsis (2006, 2007) for proposing ontology model for the ETL processes. Thus, this chapter elaborates the research methodology used for the ontology-based and goal-oriented approach for developing the requirement analysis method of ETL processes and achieving the research goal in general, as well as the specific objectives outlined in the research strategy in Section 1.6.

5.2 Goal-Oriented Approach

As described in Section3.4,the goal-oriented approach is centered on the organization and individual goals that are proposed for the representation and reasoning about a software system's usage (Jureta, Faulkner, & Schobbens, 2007; Lamsweerde, 2009). Since a goal-oriented approach is not new in software engineering research, this research selects a suitable framework and methodology that have already given a big impact in software system development and goal oriented paradigm. The *i** framework and Tropos methodology that had showed a

significant contribution in goal-oriented approach are highlighted. Particularly, the features for requirement analysis method used in this research are elaborated upon.

5.2.1 i* Framework for Software Development

The *i** (pronounced *eye-star*) modeling framework[13] is a modeling approach, which represents the software artifacts by using a semi-formal notation that is centered on intention of agent-based characteristics. The agent-based approach offers an interesting way to model the early phase of a software requirement process. This is the reason why *i** framework is suitable for any requirement analysis method that focuses on the goals achievement of individual and organization.

The main concept in *i** is an actor model, which has intentional properties of software agents such as goals, beliefs, desire, abilities, and commitments that are used in modeling the requirements. The organizational actors are identified and their intentional characters are used to establish dependencies among them for goals to be achieved, tasks to be performed, and resources to be furnished. On the non-functional requirements, the intention is softgoals to be satisfied. Basically, *i** framework consists of two modeling components: i) strategic dependency model (SD); and ii) strategic rationale model (SR).

The SD model consists of a set of nodes and links that represents an actor depending on each other to attain some goals. A pair of connecting actors is known as *depender* and *dependee*, while the object positioned between the actors is known as *dependum*.

[13]http://www.cs.toronto.edu/km/istar/

The depender always depends on the dependee to present some state in the *real world*. The SD model represents the goals, tasks, resources, and dependencies between actors by using the *i** URN, which has been accepted as international standard for requirement engineering[14].

Three types of dependencies are: i) goal-dependencies – the depender depends on the dependee for goal to be achieved; ii) task-dependency – the depender depends on the dependee to perform the task, and it sometimes looks similar with goal-dependency; and iii) resource-dependency – the depender depends on the dependee to achieve a goal or to perform a task based on the availability of resources. The depender becomes vulnerable if the dependee fails to achieve a goal, perform a task, and/or make a resource available.

A SR model is about the internal intention of actors, where the task is decomposed through using MEAN-END analysis and represented by MEAN-END links. The task decomposition link explains about the tasks and sub-tasks to be performed by each actor and relate goals to be achieved with the tasks or resources. As defined by Yu (1995), the SR model is derived by asking *why* questions,for examples: i) why is it necessary to schedule the meetings ahead of time?ii) is confirmation via the computer-based scheduler sufficient? If not, why not? Having answers to these *why* questions will help develop successful systems and facilitate the development and evolution of the enterprise system. The goal, softgoal, task, and resource involved in the system are represented according to the i* notations, which explain the

[14]http://jucmnav.softwareengineering.ca/ucm/bin/view/UCM/DraftZ151Standard

relationships between the SR components. These models were used to build the Tropos methodology for detailing the process of software system development.

5.2.2 Tropos Methodology

Tropos is a software development methodology that is based on agent-oriented architectures (Bresciani et al., 2004). The main concepts in Tropos such as actor, goal, and dependency are derived from the *i** framework. New concepts have been introduced such as resource, plan, capability, and belief. With these concepts, Tropos is suitable for designing the agent-oriented, distributed, and open application. The software application based on Tropos methodology is aimed to carry out the requirement process in as a detail as possible.

5.2.2.1 The Key Concepts

The instances of a conceptual meta-model of Tropos methodology is conceptualized from a number of concepts. The main concepts applicable in this research method are actor, goal, resource, plan, and dependency. Detailed explanation about the conceptsmeta-model can be found in Bresciani et al. (2004).

5.2.2.2 The Development Phases

Tropos methodology contains five main development phases: *early requirements, late requirements, architectural design, detailed design, and implementation*. It intends to support all the analysis and design activities of a software development process for a deeper understanding of the software *to-be* within the social and

environmental context. Therefore, all phases of the development process are pressing on a mentalistic notion of agents such as belief, desire, and intention (Bresciani et al., 2004; John, Lin, & James, 2002).

i) **Early requirement** is about understanding the business problems by studying the context of existing organizational setting. The result of this analysis is an organizational model that covers relevant goals, actors, and their respective dependencies of *as-is* of a system.

ii) **Late requirement** share the same conceptual and methodological approach with early requirement. It concerns an operational environment along with relevant functions and qualities of the system *to-be*. Precisely, domain stakeholders are identified and modeled as social actors, who depend on each other for goals to be achieved, plan to be performed, and resources to be furnished.

iii) **Architectural design** is defined as an interconnected sub-system for producing a global architecture of a system *to-be*. The global architecture presents the flow of data and control through sub-systems, which specifies the interconnection between actors and data or control by dependencies. Specifically, the actors can be mapped to a set of software agents by defining agent capabilities.

iv) **Detailed design** is defined as specification of software agent capabilities and interactivities. Normally, the capability and interactivity of agent are based on the chosen platform that possibly is mapped to the codes to be constructed.

v) **Implementation** is the actual execution of the system that is finally revealed from the methodological phases and implemented on chosen platform.

5.2.2.3 The Modeling Activities

The modeling is a series of activities for acquiring as much information as possible about the system from an early requirement toward its refinement and evolution of the modeling process. The main modeling is:

i) **Actor modeling** is identifying and analyzing the actors of a system and its environment. In particular, the modeling work focuses on modeling the application domain and their intentions as social actors to achieve the goals. In each phase of development, the modeling focuses will be changed according to the aims of the development phase.

ii) **Dependency modeling** is identifying dependencies between two actors, where one actor depends on another actor for goals to be achieved, plans to be performed, and resources to be furnished. In particular, the modeling work focuses on the goal dependencies between social actors within the environment setting. Like actor modeling, the modeling focus will be changed according to the aims of the development phase.

iii) **Goal modeling** is identifying goals for actor, and conducting the analysis of goal from actor views. Basically, the goal analysis is performed by using reasoning techniques such as MEANS-END analysis, Contribution Analysis, and AND/OR decomposition. The goal modeling is applied in the early and late requirement model for refining these to elicit new actors and dependencies.

iv) **Plan modeling** is considered as an analysis task to support the goal modeling. All the reasoning techniques can be applied to analyze the plan and sub-plan for achieving the goals.

5.2.2.4 The Reasoning Techniques

The reasoning techniques are applied for analyzing the goal or plan for identifying the sub-goal or sub-plan for each modeling. These techniques have their own purposes and are used for different aims as follows:

i) MEANS-END analysis aims to identify plans, resources, and softgoals to provide means for achieving a goal.

ii) Contribution analysis aims to identify goals that can contribute positively (+ encourage) or negatively (- discourage) in fulfillment of the goal to be analyzed. Diagram for this technique is shown in Figure 5.1.

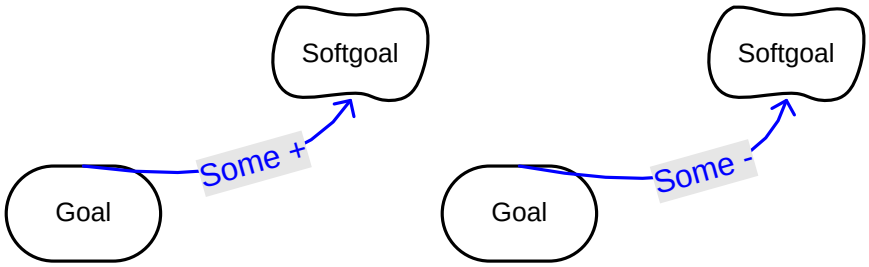


Figure 5.1: Contribution Analysis

iii) AND/OR decomposition is a combination of AND and OR for decomposing the goal to determine whether the sub-goalone AND sub-goaltwo or sub-goalone OR sub-goaltwo can achieve the goal. Precisely, decomposition

technique is to refine the goal structure. Diagram for this technique is shown in Figure 5.2.

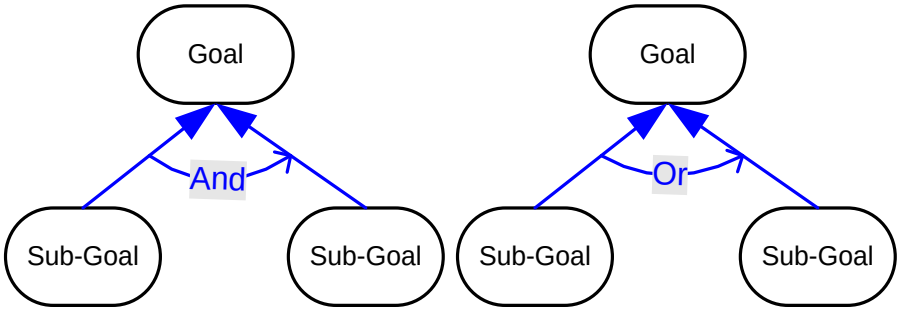


Figure 5.2: AND/OR decomposition

Based on the Tropos methodology, GRAnD is developed and used as a foundation of this research solution.

5.2.3 GRAnD for Requirement Analysis Approach

GRAnD aims to offer an alternative for analyzing user requirements in DW systems as the current requirement analysis approach is always causing failures to the DW systems development (Giorgini et al., 2008). The GRAnD can be employed within the demand-driven, supply-driven, and mixed framework of DW design. The analysis approach focuses on early requirement that deals with the high-level goals of the stakeholders and decision makers(Horkoff, 2012). Stakeholders and decision-makers have created two different perspectives of analysis that need to be modeled accordingly. Figure 5.3 show an overview the analysis phases that is implemented on the organizational and decisional perspectives. Both perspectives are derived from

the theory of organization and decision, which compliments each other for building the GRAnD approach.

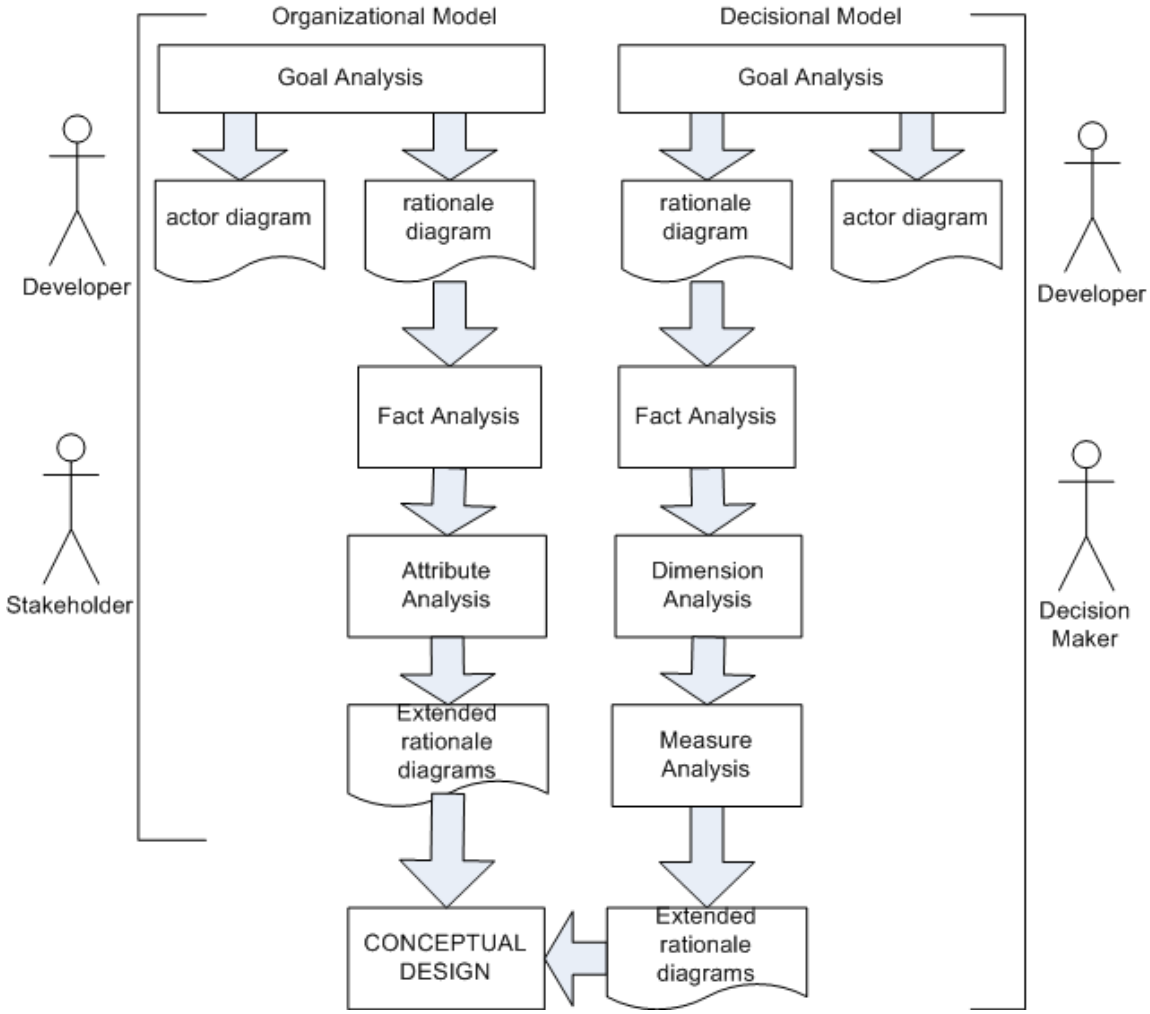


Figure 5.3: The GRAnD Approach (Giorgini et al., 2008)

5.2.3.1 Key Concepts

The main concepts used in the DW domain were successfully adapted from the Tropos methodology. New concepts in ETLprocesses context were introduced to

comply with the requirement analysis model, where these are no related notation in the Tropos methodology. The notations used for GRAnD approach are:

i) **Actor** is representing an enterprise stakeholder. Precisely, an actor can model a physical or software agent, and a role or position of a role. Like notation in Tropos, an actor is symbolized as a circle in Figure 5.4(i).

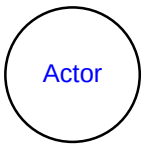


Figure 5.4(i): An Actor for GRAnD

ii) **Goal** is representing stakeholder strategic interests. In DW systems, the goal concept is pursuing *an achievement* for information of a decision-making process. Therefore, the concept of goals is defined within the organizational and decisional setting. Like notation in Tropos, a goal is symbolized as an oval in Figure 5.4(ii).

Figure 5.4(ii): A Goal for GRAnD

iii) **Dependency** is representing a relationship between two actors. The dependency explained about an actor depends on the other for attaining some goal, execute some plan, and deliver some resource. Like notation in Tropos, a dependency is symbolized as a line with an arrow in between in Figure 5.7(iii).

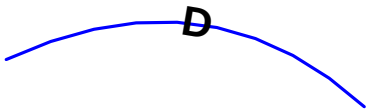


Figure 5.4(iii): A Dependency for GRAnD

iv) **Fact** is determined in both analysis models. In organizational modeling, a fact is representing a set of events that happen when a goal is achieved. In decisional modeling, a fact represents a set of analysis for goals to be achieved. A fact is symbolized as a rectangle in Figure 5.4(iv).

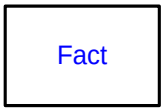


Figure 5.4(iv): A Fact for GRAnD

v) **Attribute** is representing a field, which value is provided when a fact is recorded to achieve a goal. An attribute is connected to goals and symbolized as small diamond in Figure 5.4(v).

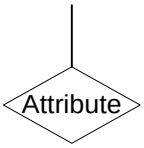


Figure 5.4(v): An Attribute for GRAnD

vi) **A dimension** is a fact property that represents a possible perspective of analysis for goal to be achieved by a fact. A dimension is connected to goals and symbolized as a small circle in Figure 5.4(vi).

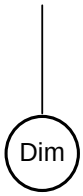


Figure 5.4(vi): A Dimension for GRAnD

vii) **Measure** is a numerical property that represents an aggregation aspect of analysis for goal to be achieved by the fact of a decision maker. A measure is connected to goals and symbolized as a small square in Figure 5.4(vii).

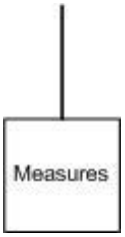


Figure 5.4(vii): A Measure for GRAnD

All the key concepts applied in GRAnD are formally specified syntactically in the language meta-model of Tropos. Although new concepts of Tropos for DW context were not specified in the meta-model, the aims of software modeling are not interrupted.Nevertheless, it fulfills the purpose of a DW requirement model.

5.2.3.2 The Modeling Activities

The modeling activities for GRAnD are performed in two different perspectives, but related on each other. These perspectives are: i) organizational modeling that is centered on the organizational setting in which the DW is operated, and ii) decisional modeling that is centered on the decision maker setting in which the functional and

non-functional requirements are captured. As illustrated in Figure 5.3, both perspectives are modeled based on an actor and rationale diagram. The actor diagram is a graph of actors related by dependencies that explain why and how the actors are related. By using the key concepts, the actor diagram is illustrated in Figure 5.5.

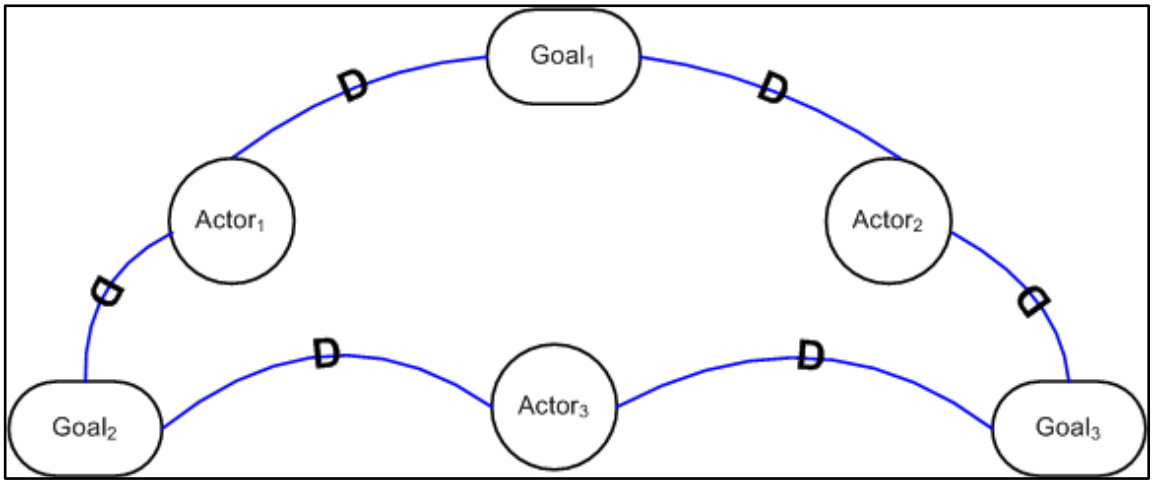


Figure 5.5: The Actor Diagram

Actor diagram in Figure 5.5shows the dependencies between the actors, where actor one is depending on actortwo for achieving goalone, and is also dependent on actorthree for achieving goaltwo.

Another diagram is a rationale diagram. This diagram is used to represent the logical foundations for rules applied in an actor diagram. The rules are applied for decomposing the goals into sub-goals by several reasoning techniques (i.e., MEANS-END, AND/OR, Contribution). It appears as a balloon (boundary), where goals of a specific actor are analyzed and dependencies with other actors are established. The rationale diagram is illustrated in Figure 5.6.

The rationale diagram in Figure 5.6 explains the decomposition of goal, where actorone is decomposed into sub-goalone or sub-goal twofor achieving goalone and also the rest of the goals' hierarchy. This basic diagram drives the building of organizational and decisional modeling. Each modeling activity contains several analyses that need to be conducted in iterative sequence before the DW requirement model can be finalized. These analyses are explained in the next Section 5.2.3.3.

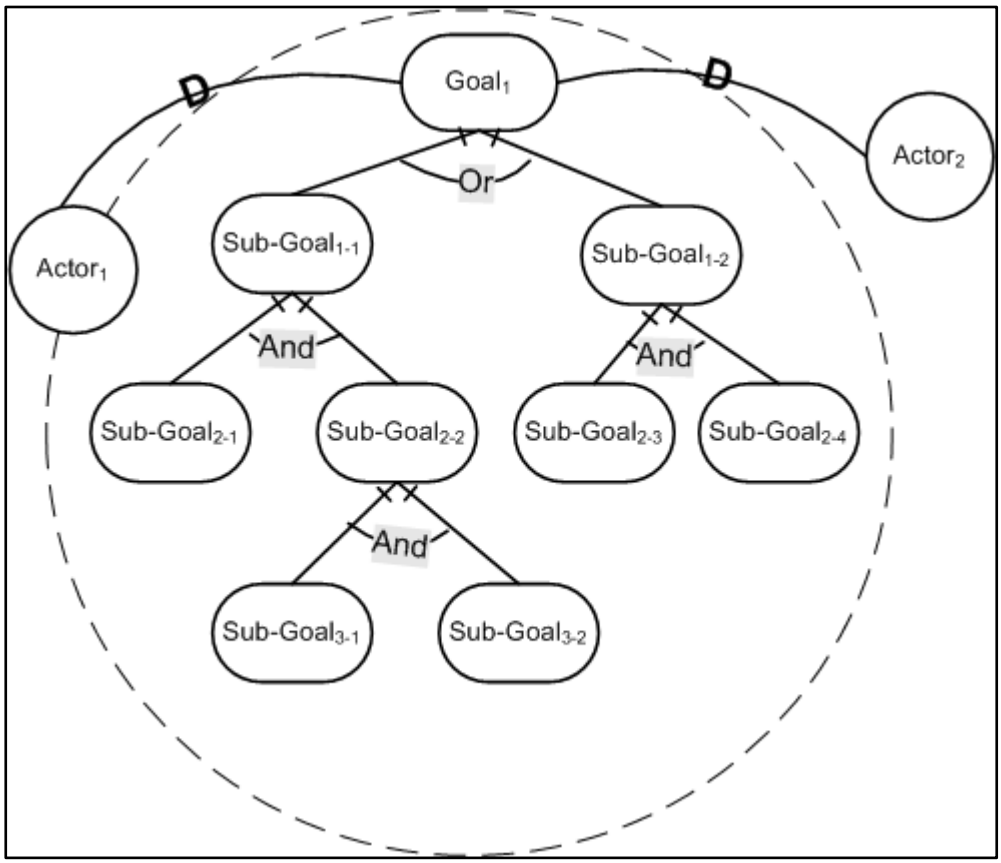


Figure 5.6: The Rationale Diagram

5.2.3.3 Organizational Modeling

Organizational modeling consists of three different analyses, which are produced in the iterative process. These analyses are: i) *goal analysis*, in which the actor

diagrams and rationale diagrams are produced according to a goal that needs to be achieved by the stakeholders. The information is collected by using template form of (*actor, objectives*), (*sub-actor, type, goals*), and (*depender, dependee, goal*); ii) *fact analysis*, in which the rationale diagrams are extended with facts. The information is collected by using template form of (*fact, description*) and (*goal, fact*); and iii) *attributes analysis*, in which the rationale diagrams with facts are extended by connecting attributes to the goals. The information is collected by using template form of (*attribute, goal, fact*).All goals, facts, and attributes are defined in the context of organization setting.

5.2.3.4 Decisional Modeling

Decisional modeling consists of four different analyses, which are produced in the iterative process after completing the organizational modeling. The analysis focuses on the goals of a decision maker and the requirement model is about the DW *to-be*, which emphasizes how the DW can support the decisional process of the organization. After the decision makers are identified, the following analysis are carried out: i) *goal analysis*, where the rationale diagrams are produced according to the decision maker's goals; ii) *fact analysis*, where the rationale diagrams are extended with facts; and iii) *dimension analysis*, where the rationale diagrams with facts is extended by connecting dimensions to the goals. The information is collected by using template form of (*goal, fact, dimension*) and (*dimension, description*); and iv) *measure analysis*, where the rationale diagram with facts is extended by connecting measures to the goals. The information is collected by using template

form of (*goal, fact, measure*) and (*measure, description*).All goals, facts, dimension, and measures are defined in the context of decision-maker setting.

In DW systems, initial requirements need to be gathered from the stakeholder's viewpoint, which requires different views of information. These need the developer to identify the stakeholders and model them together as social actors for goals to be fulfilled, tasks to be performed, and resources to be furnished. Moreover, through these actors' models and their dependencies, developer can understand why and how the DW functionalities link objectives, user requirements, preferences, and processes (i.e., ETL processes). Therefore, this approach extends the GRAnD by exploiting the Tropos methodology to enhance the DW requirement analysis method.

In the proposed approach, goal-orientedis used to cater for the problems of user requirement reconciliation and ontology which is utilized for resolving the semantic heterogeneity problems. The methodology used in construction and manipulation of ontology is explained in the next Section 5.3.

5.3 Ontology-Oriented Approach

Ontology is used to model the DW requirements and data sources in a unified manner. The unified models are used to design the ETL processes according to user requirements within the setting of DW environment. Moreover, by applying programming into ontology language, the ETL processes specifications can be generated and be ready to implement the DW system. In order to utilize the ontology, it needs to construct, map, and maintain according to the

proposedrequirement analysis method. The appropriate methodology should be able to capture, construct, map, publish, and utilize the ontology for designing the ETL processes. The selected methodology is explained in the Section 5.3.2.

5.3.1 Ontology Classification

Based on the concept of ontology classification, the modeling of ontology for the proposed solution is classified into three types of ontologies: DW requirement ontology, data source ontology, and merging ontology. These ontologies are developed interactivelyto ensure the purpose of modeling is achieved. The modeling of DW requirements and data sources as ontology enables the ETL processes design to be conducted in the unify views. This helps the developer and users to interact more efficiently in conceptualizing the information as required.

5.3.1.1 DW Requirements Ontology

DW requirement ontology represents the requirement glossaries produced from the analysis process such as facts, attributes, dimensions, measures, business rules, and actions. This domain ontology should provide the ability to describe the semantics of user requirements, so that the mapping to the data sources can be accomplished by using an appropriate mapping mechanism. These tasks comprise the following main steps: i) *ontology construction*, to develop the rationale diagram of ontology for requirement glossaries based on a rationale diagram of decisional modeling; ii) *ontology mapping*, which establishes the rationale linking diagrams between requirement glossaries and data sources; and iii) *data integration and*

transformation, which determine the rationale diagram of ETL processes for the sake of data propagation and aggregation.

5.3.1.2 Data Sources Ontology

The construction of ontology for data sources is necessary to enable the mapping between DW requirements and the data sources. The data sources provide the necessary data for the information required by the user requirements. The ontology is constructed according to data source schemas that represent the relationship between defined concepts and the related tables in a heterogeneous environment. The concepts are established to present the appropriate tables, fields and attributes that have been agreed upon by involved stakeholders. The agreeable concepts or glossaries in data source schemas are important for constructing the ontology and resolving the semantics heterogeneity problems during the implementation of the ETL processes.

5.3.1.3 Merging Ontology

Merging ontology is about the semantic mapping of the DW requirements with the data sources. By mapping the requirements glossaries with the data sources, the necessary requirements for the ETL processes can be derived. Furthermore, based on the mapping of a set of defined classes representing the data sources, DW and ETL processes are generated and added to the merging ontology. The merging ontology is a confirmed ontology to reveal the semantics of the elements contained in the data sources. Since the ontology is represented in OWL language, thus the reasoning

process can be applied for identifying necessary ETL processes toward the propagation and aggregation of the DW.

5.3.2 Methodology

Identifying and understanding the concepts of DW requirements and data sources schemas are vital for developing the ontology. A suitable model for ontology structure is important to ensure the developed ontology is correctly presenting the DW requirements and enable the generation of the ETL processes specifications. The ontology model nurtures the building of ETL process specificationsthat constitute the data sources model, data integration and transformation model, and DW model.

The entire model is viewed as a combined or unified model that presents the references for designing the ETL processes. Therefore, the development of ontology is part of the modeling process and choosing the right and good methodology is very important for developing the ontology system. Section 5.3.2.1 discusses the models and methods applied to construct, map, utilize, and maintain the ontology for this research.

5.3.2.1 Ontology Development Process

A methodology that encompasses semantic characteristics is used to develop the ontology and establish mapping of the heterogeneous data sources to this ontology. Therefore, this research adapts the Semantic Information Management (SIM), which consists of semantic elements in the ontology development process (Schreiber, 2003). The SIM methodology consists of six steps as shown in Figure 5.7:

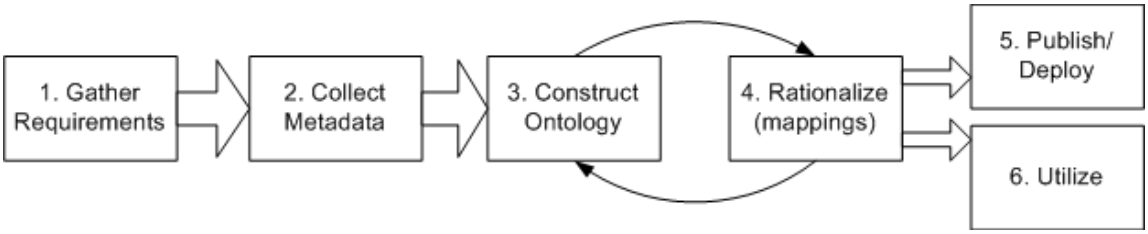


Figure 5.7: Semantic Information Management methodology ((Schreiber, 2003)

i) **Requirement gathering** is a process for collecting the information architecture and defining the scope of the ontology project.

ii) **Meta-data collection** is a process for cataloguing all the relevant data assets and data profiling (i.e., data sources schemas and conceptual model).

iii) **Ontology construction** is a process for creating the ontology from the collected metadata through a reverse engineering or manual process.

iv) **Rationalization** is a process for mapping the ontology with the data sources' schemas. The process is done iteratively for refinement of the ontology until completing the rationalization process.

v) **Publishing or deployment** is a process for transferring the information model along with the mappings to the relevant stakeholders.

vi) **Utilization** is a process for maintaining the information architecture due to the changes of data sources and user requirements.

Most of the steps in SIM methodology are supported by the Unicorn Workbench tool[15]. However, in the proposed methodology, this research utilizes various tools to accommodate the implementation of these steps. This is explained in the Chapter 6.

5.3.2.2 Ontology Modeling

Based on the SIM methodology, an appropriate model to develop the ontology is required. The appropriate modeling should enable the modeling of the information as required by the domain and application ontology as follows (Skoutas & Simitsis, 2007): i) the concepts of the domain, ii) the relationships between those concepts, iii) the attributes characterizing each concept; and iv) the different representation format's values for each attribute.

In OWL ontology language, the concepts of the domain are represented by *classes*, and the relationships between concepts, as well as the attributes of the concepts are represented by *properties*. However, the properties are categorized into *object property* and *data type property*. The different type of values (i.e., data type property) for each attribute is represented by classes that are organized in a *hierarchy*. Importantly, due to the significance of *aggregate operations* in DW systems, specific elements are introduced to explicitly specify such operations.

Formally, the ontology model (O) is defined as**O = (C, P, A),** where:

- $\mathbf{C = C_c \cup C_t \cup C_g}$

  C_c is a set of classes of concepts in the domain, C_g is set of aggregate operation class (i.e., AVG, SUM, and COUNT), and $C_t = \{C_{tp} \cup C_{tf} \cup C_{tr} \cup C_{tg}\}$ is a union

[15] http://www.yehuditcohen.com/

of a set of classes that is used to present different kinds of values for a property that corresponds to an attribute of a concept.

For $\mathbf{C_t = \{C_{tp} \cup C_{tf} \cup C_{tr} \cup C_{tg}\}}$, C_{tp} is a class declared to be a range of property. C_{tf} is a class denoted by different representation format. C_{tr} is a class denoting different ranges of value for a property, and C_{tg} is a class representing values from the aggregate operations.

- **$P = P_p \cup$ {convertsTo, aggregates, groups)**
  P_p is a set of properties that represents attributes of the concepts or relationships.

- **A** is a set of axioms used to assert sub-sumption relationships between classes, specify domain and range properties, specify cardinality constraints, assert disjointness classes, and define new classes.

In ontology development, a set of classes, properties and relationships to specify the domain and range of each property and to arrange the classes in hierarchical kind of structure is a fundamental task. This task can be done by using ontology management tools (e.g., Protégé-OWL) with basic understanding of ontology structure and language (e.g., RDF and OWL). However, an understanding of user requirement and data sources semantics are imperatives for constructing the correct ontology and annotate the ETL processes specifications.

5.3.2.3 Ontology Mapping

Ontology mapping is required for linking the DW and data sources schemas. In mapping process, the DW requirement glossaries are merged with the data sources schemas to provide a combined view of a data and process model for the developer

to proceed with the design and maintenance of ETL processes from the early phases of a DW systems development. The method used for establishing the ontology mapping is defined as:

i) Specify a pair of concepts from DW requirements and data sources for possibly establishing the relationship for giving meaning to the ETL processes.
ii) Establish the interrelationship of concepts between DW requirements and the data sources through matching mechanism.

However, there is no complete and suitable method for implementing the ontology mapping, especially in a DW domain. Based on various methods in ontology mapping available (Aleksovski, 2008; Alexiev et al., 2005; Noy & Musen, 2000), this research adapts the method used by An (2007) that supports the roles of merging ontology task. The merging ontology emphasizes the combination of similar concepts, and separation of the concepts dissimilarity. Particularly, this task deals with the merging of data warehouse requirement ontology and data source ontology for producing the merging requirement ontology.

5.3.2.4 Ontology Language

As explained in Chapter 3, there are many ontology languages available to represent the ontology model. One of the most recent and expressive language is the OWL. OWL is a XML-based markup language for defining and instantiating ontologies to

enable machine-processable semantics (Geroimenko, 2004). This research has chosen OWL to be used for ontology model representation for several reasons.

First, OWL (i.e., OWL-DL) is based on description logics (DL) that provide a formal and explicit method for knowledge representation. It allow the reasoners (e.g., Pellet or Fact) to be invoked for automating several tasks such as checking correctness of the classes and sub-sumption relationships between classes. Second, OWL is feature rich that enables the creation of class expressiveness using Boolean operators such as *union*, *intersection*, etc., which are very useful in data integration and transformation definitions. Third, an OWL language is easily programed by using object programming languages (e.g., Java or Prolog) and supported by testable semantic-based classes that were developed by researchers (Hutter, Stephan, Baader, Horrocks, & Sattler, 2005).

OWL features are used to define the classes in ontology, as listed in Table 5.1.However, only a subset of the features in Table 5.1 is applied in the approach. This is explained in Chapter 6.

Table 5.1: OWL Language Features

Type	**Name**
RDFS features	Class, rdfs: subClassOf, rdf: Property, rdfs: subPropertyOf, rdfs: domain, rdfs: range, Individual
(In)Equality	equivalentClass, equivalentProperty, sameAs, differentFrom, AllDifferent, distintMembers
Property Characteristics	ObjectProperty, DataTypeProperty, TransitiveProperty, SymmetricProperty, FunctionalProperty, Inverse

Restricted Cardinality	FunctionalProperty minCardinality, maxCardinality, cardinality

5.4 Development Process of the RAMEPs

Based on the GRAnD approach, this research has established the RAMEPs approach for supporting the ETL processes design. The RAMEPs approach defines the conceptual and logical ETL processes model by applying the goal-oriented and ontology approach. This approach deals with the semantic requirements that are needed to be reconciled with the relevant data sources in order to analyze and model the ETL processes specifications. The RAMEPs was developed by combining the Tropos methodology used in goal-oriented analysis, and SIM methodology used in ontology analysis. The main components of RAMEPs are explained on the next Sections5.4.1 to 5.4.3.

5.4.1 Component 1 – DW Requirement Management

In DW requirement management, the DW and ETL processes requirements are elicited and analyzed according to goal-oriented approach to ensure the requirements in the domain are completely captured. This component contains two significant tasks: i) requirement elicitation, and ii) requirement analysis. Both tasks are complementing each other for organizing the requirements as needed by user, organization and developer. Detailsof these tasks are discussed in Section 6.2.4 (Step 1-4), Chapter 6.

5.4.2 Component 2 – Ontology Management

In ontology management, the DW requirements and data sources are modeled as ontology structure. Then, both ontologies are mapped accordingly to become a merging ontology. The merging ontology is refined until all the requirements have been mapped to the data sources completely. This component contains three significant tasks: ontology construction (i.e., DW requirement ontology, data sources ontology), ontology mapping (i.e., DW requirement ontology and data sources ontology), and ontology refinement (i.e., merging ontology). All these tasks are implemented iteratively until the complete ontology is ready for deriving the ETL processes specifications. Detailsof these tasks are discussed in Section 6.2.4 (Step 5-8), Chapter 6.

5.4.3 Component 3 – ETL Processes Generation

This is the third and last component of the RAMEPs development process. This component contains two significant tasks: develop algorithms, and build-up semantic-based application. The complete merging ontology is manipulated by using semantic-based programming to derive the ETL processes specifications. The algorithms for reading and manipulating the ontology are developed. The semantic-based programming uses the algorithms for generating the ETL processes specifications and produce the ETL processes specifications list accordingly. At the end, the ETL processes specifications can be used to implement the DW systems in providing information to the users. Detailsof these tasks are discussed in Section 6.2.4 (Step 9), Chapter 6.

5.5 Validation and Evaluation Process

The RAMEPs is validated to ensure its correctness by using tools that are compliant with the models produced from the analysis process. Then, the RAMEPs are evaluated to ensure its usability in designing and developing the ETL processes by conducting the real-world case studies. SinceRAMEPs produce the list of user requirements for DW systems, thus the requirement's list can be verified on their correctness, completeness, consistency, and unambiguity. These quality characteristics of software requirements can be measured and estimated (IEEE, 2004; Kaiya & Saeki, 2006). However, there are no outstanding techniques for validating and evaluating the whole process of requirement analysis, especially in the ETL processes requirements. Therefore, the techniques for validation and verification are explained in Sections 5.5.1 and 5.5.2.

5.5.1 Goal-Oriented Compliant Tools

The goal-oriented compliant tools aretools to be used for modeling and analyzing the DW and ETL processes requirements. The compliant tools should be able to model and analyze high-level user requirements toward the detail of ETL processes. However, few available tools can support these functionalities. This research utilizes two types of tools, which can support both the high-level requirements and ETL processes artifacts. However, the issues of model transition between both toolshave to be compromised due to the use of different tools.

5.5.1.1 Organization Modeling Environment (OME) Tool

OME[16] is a goal-oriented modeling tool that provides the developer with a graphical user interface to develop models, and supports access to a knowledge base that allows for advanced model analysis. This tool is based on *i** modeling framework that intends to provide a clear link between the requirements, specification and architectural design phases of software development. The developer can also use this tool for business reengineering to understand how the business process is operated. The RAMEPs utilizes this tool for modeling the DW requirements at the early-phase of ETL processes.

5.5.1.2 Data Warehouse Design Tool (DW-Tool)

The DW-Tool[4] is a DW modeling tool used for modeling and designing the DW systems from the organizational modeling toward the decisional modeling. The modeling and analysis tasks produce the DW schemata used for designing the DW system accordingly (Giorgini et al., 2008). This research utilizes this tool in combination with OME tool for completing the entire model of the ETL processes. However, some adjustments on the DW-Tool are needed to accommodate the modeling of the ETL processes. The adjustment is required in order to allow the *action* and *business rule* notation to be captured and presented in the DW-Tool.

[16] http://www.cs.toronto.edu/km/ome/index.html

5.5.2 Ontology Compliant Tool (Protégé-OWL)

Protégé-OWL is a tool originally developed for assisting the users in developing large computerized knowledge bases (Gennari et al., 2003). The emergence of semantic web technologies enhanced the functionalities of Protégé-OWL to build, store, visualize and maintain the ontologies in many different formats such as relational databases, UML, XML, RDF and OWL. The developed ontology can support the domain model and knowledge-based applications. The ontology structure is based on *frame-based* approach that is applied in a knowledge representation system (Chaudhri, Farquhar, Fikes, Karp, & Rice, 1998).

In this structure, ontology is defined by a set of classes representing the domain concepts, a set of properties and relationships and a set of instances for the classes. Recently, Protégé-OWL Version 4.0[17] supports the OWL language that is designed for complex model of ontology structure. Ontology can be presented by OWL and implemented as semantic-based architecture (Alexiev et al., 2005). The graphical-based and rich set of functions of a tool will assist the building and maintaining of the ontology sources. This is an important factor for ontology development in iterative environments and is surely applicable in a DW systems environment.

5.5.3 Case Studies

Case study is important for evaluating the usability of the RAMEPs in modeling and designing the ETL processes. Most of the DW systems are implemented in a specific domain of the enterprise. However, various kinds of information system architecture

[17] http://protege.stanford.edu/

in the organization provide a different type of heterogeneous environment. This research employed three different types of heterogeneous environment, which implemented the information system in a different application domain. These case studies are:

i) **Academic Affairs Department** in Universiti Utara Malaysia (UUM), where the core business is supported by Academic Student Information System (ASIS), and Graduate Academic Information System (GAIS).
ii) **Billing Department** in the Utility Company of Gas Malaysia Sdn. Bhd., whereits businesses are supported by Utility Billing Information System (UBIS), J.D. Edwards System (JDE), and Call Center System (CCS).
iii) **Entrepreneur Department** in the Ministry of Higher Education, whereits decision-making process for university entrepreneur program is supported by Business Intelligence for Student Entrepreneurs (BISE) that contains data from the IHLs.

These case studies are discussed comprehensively in Chapter 7.

5.5.4 ETL Processes Specifications Construction

ETL processes specifications contain a list of data integration and transformation proposed from the implementation of RAMEPs. In order to generate the ETL processes specifications from DW requirement ontology, a prototype application wasdeveloped by using the Java programming language based on the Eclipse

platform[18]. Java is the only language available to manipulate the OWL language within the Jena 2 framework[19]. The Jena 2 framework is one of the testable programming environments for building ontology-based applications. With Jena, the functions for accessing, reading, updating, and writing the semantic web language (e.g., RDF, RDFS, OWL, SPARQL) can be done successfully.

5.6 Conclusion

This research focuseson modeling and designing the ETL processes. Throughout this process, the DW structure as a target data store has been defined. Most of the DW modeling and designing research only focuses on the modeling and defining the DW schemata. However, this research is tackling the crucial part of the DW systems development,i.e., the ETL processes. Therefore, the main interest of the research methodology is focused on the goal-oriented and ontology approach for developing the RAMEPs. The goal-oriented and ontology approach is applicable in RAMEPs for modeling and designing the ETL processes specifications.

The validation and evaluation process is implemented by using compliant tools and the ETL processes specifications in the real environments. The elements of easy to use and understandable are important in order to adopt the goal-oriented and ontology-based approach for designing the ETL processes. Furthermore, the ontology developed in the design tasks must be corrected and accepted by the business users to guarantee semantic heterogeneity problems are resolved during the

[18]http://www.eclipse.org/
[19]http://www.openjena.org/

ETL processes implementation. The successful validation and evaluation process will achieve the research objectives as defined in phase III of Figure 1.2, discussed in Chapter 1, Section 1.6. Chapter 6 discusses in detail about the RAMEPs approach and explains its capabilities for modeling and designing the ETL processes in DW systems development.

CHAPTER SIX–REQUIREMENT ANALYSIS METHOD FOR ETL PROCESSES (RAMEPS)

This chapter explains the RAMEPs approach for modeling and designing the ETL processes. The goal-oriented approach for analyzing the user requirements is highlighted, while the ontology approach for modeling the DW requirements, data sources, and merging of requirements and data sources are discussed. The ETL processes generation algorithm is developed and demonstrated the generation of the ETL processes specifications ended the chapter.

6.1 Introduction

Requirement analysis method for ETL processes - RAMEPs is a solution for resolving the problems of business semantics reconciliation and semantics heterogeneity in designing the ETL processes. The RAMEPs is used in analyzing user requirements and reconciling the business semantics toward the available data sources by using a goal-oriented approach. Here, the DW requirements and data sources schemas are defined as ontology for tackling the semantics heterogeneity problems and unifying the DW requirements and data sources model for designing the ETL processes. All these tasks are supported by RAMEPs, which was developed and guided by the goal-oriented and ontology discussed in Chapter 5.

6.2 The RAMEPs

Requirement analysis of ETL processes focuses on the transformation of informal statements of user requirements into a formal expression of ETL processes

specifications. The informal statements are derived from the requirement of stakeholders and analyzed from the organization and decision-maker perspectives (Giorgini et al., 2008). It argues that an analysis of the DW requirements from high-level user requirements toward the detail of ETL processes are important in tackling the complexity of DW systems design.

It is widely accepted that the early requirement analysis significantly reduces the possibility of misunderstanding users' requirements (Horkoff, 2012; Yu, 1995). The high-level understanding among stakeholders possibly increases the agreeable terms and definitions used during the ETL processes execution. Therefore, the RAMEPs is centered on organizational and decisional modeling and focuses on the data transformation model from the perspective of a developer as emphasized by socio-technical system theory.

6.2.1 Requirement Analysis Method

The initial requirements are gathered from the stakeholder's viewpoint, which are elicited from different views of information required. This requires the developers to identify the stakeholders and model them together as social actors for goals to be fulfilled, tasks to be performed, and resources to be furnished. Moreover, the developer can understand *why* and *how* the DW functionalities are linked with the objectives, user requirements, preferences, and processes (i.e., ETL processes) to implement the DW systems. This can be achieved through actors and their dependencies in a DW requirement model.

The focus of analysis is to define the decisional information from the perspective of organizational and decision-makers. Thus, the components of DW structure need to be defined in the analysis diagrams. The components of DW structure are represented in specific symbols explaining their roles and analysis activities respectively. These analysis activities are implemented sequentially, since the outputs will be the inputs for the next analysis. At the end of these analyses, the glossaries of facts, dimensions, and measures are used to design the conceptual of DW structure.

However, these tasks are not enough to implement the DW systems since the detailed activities of ETL processes are not defined yet. Further analysis on data transformation activities needs to be carried out in order to determine the ETL processes specifications, and completing the development of DW systems. Therefore, the whole process of RAMEPs is also developed from the concepts of *Information as Required* model, which is compliant with the ETL processes scenario.

6.2.2 The Information as Required

The RAMEPs comprises three main perspectives: i) organization views, ii) decision-maker views, and iii) developer views. These perspectives address the information and transaction scenario information system, which complies with the scenario of DW systems. Moreover, the iteration of an information requests is sequence pairs of Request for Information (*RFI*) and Response Obtained (*RO*) (Prakash & Gosain,

2008). Thus, an ETL scenario can be referred to as Information as Required (*IaR*) and the entire scenario of IaRs refers to RFI and RO as illustrated in Figure 6.1.

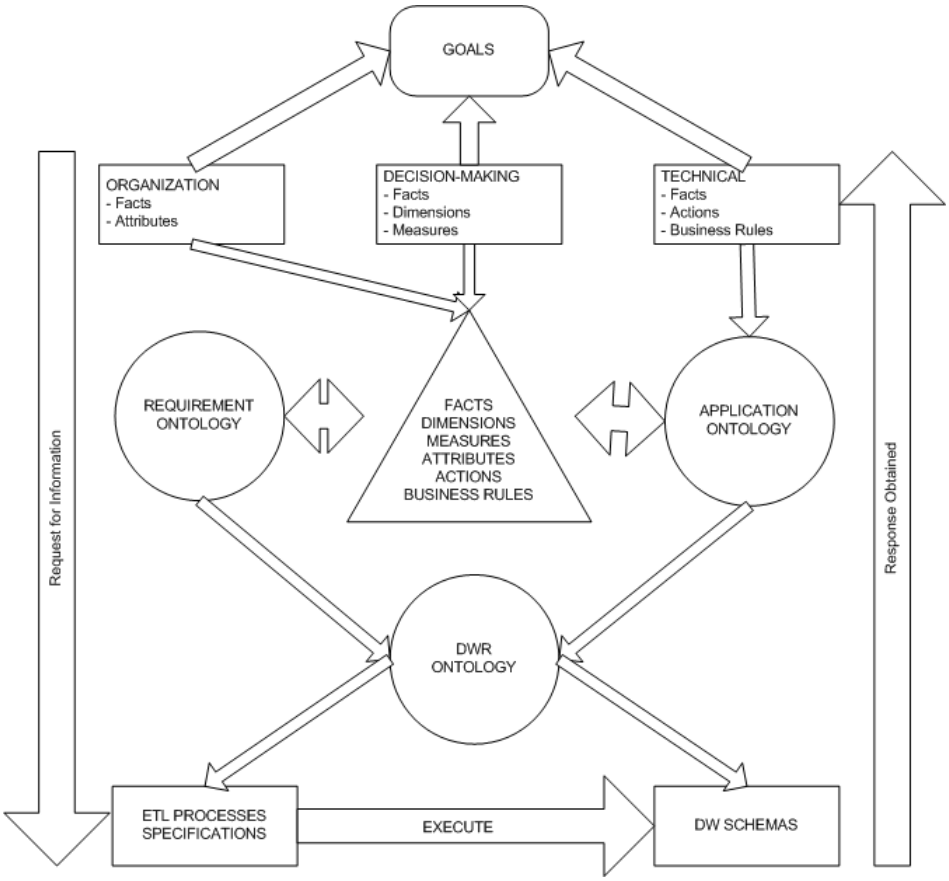


Figure 6.1: Information as Required Model (Prakash & Gosain, 2008)

Based on Figure 6.1, the entire scenario of IaR can be fitted in the RAMEPs method in order to implement the DW requirement analysis toward the ETL processes scenario. Collectively, these tasks are played by the stakeholders, decision-makers and developers. Thus, a unified view of requirement representation is important to ensure the accuracy and consistency of user requirements. This can be achieved by analyzing the users' requirements on these perspectives and conducting the analysis according to steps in the RAMEPs model.

6.2.3 RAMEPs Model

By adapting the GRAnD approach (Giorgini et al., 2008), the model of RAMEPs is presented in Figure 6.2, and the extended works are highlighted on the right hand in the highlighted developer model. Figure 6.2 shows the detailed approach in the requirement analysis process of the ETL processes from elicitation and understanding of organization goal toward defining the data transformation activities for implementing the DW systems. The RAMEPs approach contains analysis phases, which are divided into three perspectives: organization modeling, decisional modeling, and developer modeling(Ta'a, Abdullah, & Norwawi, 2010). This three modeling will be used to produce the specifications for designing the ETL processes.

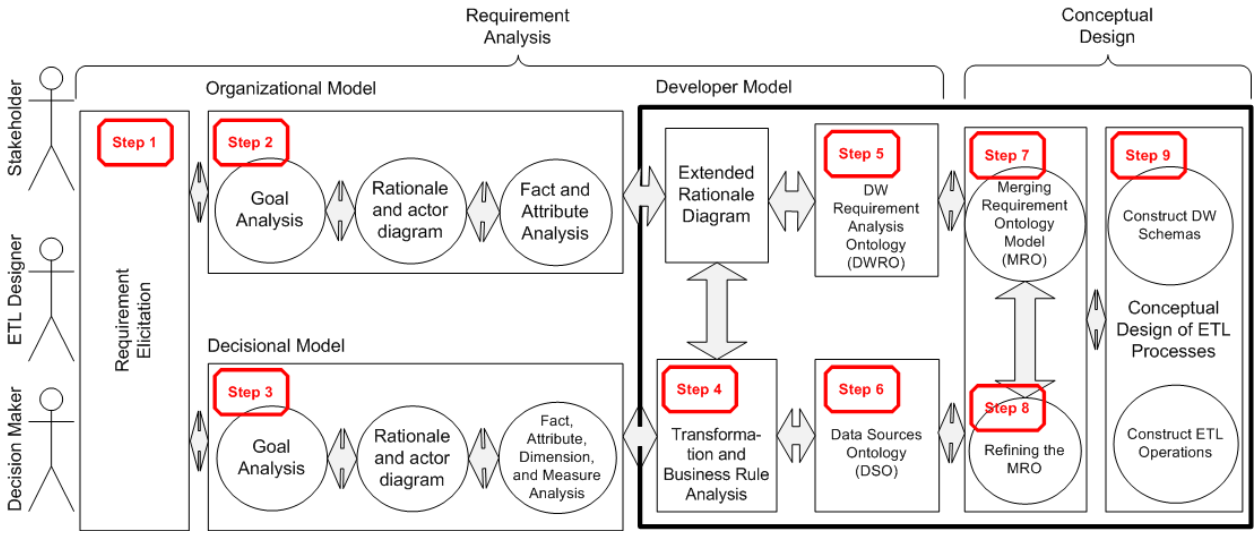


Figure 6.2: The RAMEPs

Finally, the conceptual design of ETL processes is produced based on the specifications given before the implementation of DW systems takes place. The conceptual design of ETL processes consists of DW schemas, and ETL processes

specifications, which complement each other for enabling the implementation of DW systems. To produce the conceptual design of the ETL processes, the analysis process of DW requirements needs to be conducted according to steps as shown in Figure 6.2 and elaborated in the RAMEPs tasks.

6.2.4 RAMEPs Tasks

Based on RAMEPs model presented in Figure 6.2, Table 6.1 highlights the implementation steps of RAMEPs.

Table 6.1:Steps in RAMEPs Approach

Steps	**Activities**	**Stages of RAMEPs**	**Level of ETL Processes**	**Method**	**Output**
1.	Gather and elicit DW requirements through common methods with stakeholders.	Requirement Gathering and elicitation.	Requirement gathering and elicitation.	Interview, presentation, discussion, and document analysis	Requirement Documents
2.	Analyze DW requirements based on the organization perspective by using goal-oriented approach.	Organizational-based analysis on facts, and attributes.	Requirement Analysis	Tropos – Goal-Oriented	Diagram on Organization Model
3.	Analyze DW requirements on the decision-maker perspective based on the organization model by using goal-oriented approach.	Decisional-based analysis on facts, dimensions, and measures.	Requirement Analysis	Tropos – Goal-Oriented	Diagram on Decisional Model
4.	Analyze DW requirements on the developer perspective based on decisional model by using goal-oriented approach.	Data sources, Business rules, and transformation analysis.	Requirement Analysis	Tropos – Goal-Oriented	Diagram on Developer Model

5.	Ontology construction on the requirement analysis glossaries	Ontology model of requirements analysis.	Requirement Analysis	SIM with RDF/OWL	Ontology for Requirement Glossaries
6.	Ontology construction on the data sources schemas	Ontology model of data sources schemas.	Requirement Analysis	SIM with RDF/OWL	Ontology for Data Sources
7.	Mapping and merging the requirements ontology with the data sources ontology.	Conceptual model of ETL processes.	Design	SIM with RDF/OWL	Merging Ontology
8.	Refine the structure of merging ontology and make adjustment to fully satisfy the user requirements.	Conceptual model of ETL processes.	Design	SIM with RDF/OWL	Refine Merging Ontology
9.	Constructing the required ETL processes specifications from the merging ontology for constructing the ETL processes design.	Conceptual model of ETL processes.	Design	RDF/OWL, Java and Jena 2 Framework	List of ETL Processes Specifications

The main steps in RAMEPs tasks are centered on the three types of modeling. The organizational modeling is used to identify the goals that are related to facts, and attributes. Then, the decisional modeling focuses on the information needs by decision makers and is related to facts, dimension, and measures. Finally, the developer modeling defines the actions for the data sources with the related business rules. All these steps are explained as follows:

i) Step 1 – Elicit DW Requirements

Initially, this step starts early in any software development task. In this model, understanding DW requirements begins by gathering the user requirements through

traditional and commonly used elicitation techniques such as interviews, discussion, and document analysis. Structured interview techniques, where the questions for the interview are available in the form of templates are conducted. The templates used are based on the general questions for DW requirements by Kimball (1996) and Goal-oriented templates by Giorgini et al. (2008).

From the interviews, information about goals, sub-goals, actors, business processes, business rules, and others are elicited. Analysis on organization profiles and documentationsprovides detailed facts about the user requirements and business terms. Moreover, a presentation on DW and BI systems philosophy are given in order to provide clear understanding on what users need in the information system. However, this research does not focus on the issues of requirement elicitation as it onlyapplies the existing methods available (e.g., interview, presentation, discussion, and document analysis) for carrying out this task.

ii) Step 2 – Analyze DW Requirements Based on Organization Perspectives

Organizational modeling consists of three different analyses conducted iteratively. The analyses are: i) *goal analysis*, where actor diagrams and rationale diagrams are produced; ii) *fact analysis*, where the goal rationale diagrams are extended with facts; and iii) *attributes analysis*, where the fact rationale diagrams are extended with attributes. All goals, facts, and attributes are defined in the context of the organization setting. The approach to analyst goal, fact, and attribute is conducted in sequence and the information about goal, fact, and attribute is captured in specific templates.

iii) Step 3 – Analyze DW Requirements Based on Decision Maker Perspective

Decisional modeling consists of four different analyses, which is also performed iteratively. However, this analysis focuses on the goal of a decision maker, which is represented by the actors as defined in an organizational model. These analyses are: i) *goal analysis*, which produces the rationale diagrams of decision-goal; ii) *fact analysis*, which extends decision-goal diagrams with facts; iii) *dimension analysis*, which extends fact diagrams with dimensions; and iv) *measure analysis*, which further extends dimension diagrams with measures.

iv) Step 4 – Analyze DW Requirements Based on Developer Perspective

Developer modeling consists of three different analyses, which is also performed iteratively. These analyses are focused on the goal of a developer, which is represented by the actor's diagram as defined in the decisional model: i) *data sources analysis*, which produces the list of data sources related to the goals and facts; ii) *business rules analysis*, which produces the list of business rules and constraint for related facts; and iii) *transformation analysis*, which extends decision-goal diagram and produces the list of actions for data transformation activities with related business rules.

The developer modeling explains the facts about rules and actions applied from the perspective of ETL developers. The information provided is used to complete the data transformation analysis tasks. The transformation analysis is based on *plan modeling*of Tropos methodology that is applied in detailed design of a software system. The plan modeling captures the suitable actions with relevant business rules

for each data transformation. Finally, the decisional modeling analysis produces the informational model that is required in supporting decision making. Furthermore, the developer model supports the decisional model analysis to produce the information as required by the decision maker.

v) Step 5 – Ontology Construction on Requirements Glossaries

In designing the ETL processes,addressing data semantic problems require adequate understanding of user requirements in order to ensure appropriate mapping between data sources to targets (i.e., DW). This can be done through the ontology model, which is the intermediatemodel between the requirement analysis process and conceptual design task. The requirement analysis process produces the glossaries (i.e., facts, attributes, dimensions, measures, business rules, actions) and is linked to the appropriate data sources through the ontology mapping mechanism.

These tasks comprise three main steps: i) *ontology construction*, which produces the rationale diagram of ontology for requirement glossaries based on the rationale diagram of decisional modeling; ii) *ontology mapping*, which establishes the rationale linking diagrams between requirement glossaries and data sources; iii) *data transformation specifications construction*, which produces the rationale diagram of ETL processes for data propagation and aggregation. However, the data sources need to be modeled into ontology schemas in order to enable the mapping between DW requirements and data sources.

vi) Step 6 – Ontology Construction on Data Sources Schemas

The construction of ontology for data sources is necessary to enable the mapping between DW requirements and the data sources. The data sources involved provides the necessary data for the information required by the user requirements. The ontology is constructed according to data sources schemas that represent the relationship between concept data and the related tables in a heterogeneous environment. The concepts are established to present the appropriate tables, fields and attributes that have been agreed by the stakeholders involved. The agreeable concepts or glossaries in data sources schemas are important for constructing the ontology and resolving the semantics heterogeneity problems during the implementation of the ETL processes.

vii) Step 7 – Mapping the DW Requirements with the Data Sources

In this task, a method for the semantic mapping for the requirements with data sources is introduced. By mapping the data sources with the requirements' glossaries, the necessary ETL processes can be derived systematically. Furthermore, based on these mappings, a set of defined classes representing the data sources, DW and ETL processes is generated and added to the DW requirements ontology(DWRO). The DWRO is a merging ontology that reveals the semantics elements contained in the data sources. Since the DWRO is represented in OWL language, thus the reasoning process can be applied for identifying necessary ETL processes toward the propagation and aggregation of the DW.

viii) Step 8 – Refinement the Merging Ontology Structure

In this task, a DW schemas structure is constructed and the possibility of hierarchy existence in the DW schemas is explored. The schemas that represent the DW is derived by navigating the merging ontology structure starting from the *fact* toward the related dimension and measured to build the attribute hierarchies and create conceptual schemas of DW. Based on the many-to-one relationship between data sources and the DW, the navigation mechanism can easily define the DW schemas in the form of dimension modeling (Kimball & Ross, 2002). As a result, the DW schemas are established according to merging ontology, which supports the design of ETL processes specifications.

ix) Step 9 – Constructing the ETL Processes Specifications

In this task, a method to construct the ETL processes specifications from the merging ontology is proposed. By imposing the appropriate reasoning to manipulate the merging ontology, the set of conceptual ETL processes for transforming data sources to the DW schemas are defined. Importantly, the population of data sources toward the DW must satisfy the business rules, constraints and formats of DW schemas. In order to visualize the transformation processes, some generic types of conceptual ETL processes (Skoutas & Simitsis, 2007)are proposed in the modeling. Then, the new type of conceptual ETL processes is introduced to support the modeling of DW requirements. Finally, the list of ETL processes specifications is generated. This supports the ETL developer to design the ETL processes for implementing the DW systems.

6.3 Goal-Oriented Approach

The RAMEPs method is an approach for analyzing user requirements toward the design of ETL processes from an early phase of DW systems development. It covers two main phases of DW engineering: *requirement analysis* and *conceptual design*. As defined in research methodology, the requirement analysis phase involves the utilization of goal-oriented approach. The goal-oriented approach is adapted from GRAnD that was developed in the Tropos methodology, which uses the goal, actor, dependent, dependee, dependum, resources, and other concepts to present the DW requirements from the perspective of organization, decision modeling, and developer modeling.

6.3.1 Organizational and Decisional Modeling

As explained in Section6.2.4 (ii), organizational modeling consists of three different analyses, which are produced iterativelyin the requirement analysis based on the organization settings. This research uses the analysis methods such as goal analysis, fact analysis, and attributes analysis by applying the actor and rationale diagram for modeling the organization perspective. In decisional modeling, this research uses the analysis method such as goal analysis, fact analysis, dimension analysis, and measure analysis. The notation and templates for illustrating the model is used based on Tropos methodology and is explained in Section6.3.2 and 6.3.3.

The Tropos methodology is adopted in DW requirements analysis approach from the perspectives of decision-maker and organization. However, the approach does not cover the analysis on data transformation that belongs to the intention of the ETL

developers. The intention of the ETL developer is about determiningthe roles played by ETL developer for fulfilling the DW requirements. Section 6.3.2explains how the Tropos methodology can be used in analyzing the user requirements for the data transformations needed by the ETL processes as stated in step 4 of RAMEPs.

6.3.2 Developer Perspective Modeling

Developer perspectives are required in addition to organization and decision-maker perspectives to compliment the need of requirements analysis for ETL processes. These perspectives comply with to socio-technical theory that requires the social and technical components complimenting each other to achieve a workable information system. The outcome for each of the perspectives is presented in Table 6.2.

Table 6.2: Outcome of the Analysis Perspectives

Perspective	Outcome	Notes
Organization	• List of Facts • List of Attributes	Represent the main data in an organization and comprise of the most relevant attributes that exist in data sources.
Decision-Maker	• List of Facts • List of Dimensions • List of Measures	Represent decision-maker needs, summarizing the role played in glossary-based requirements (i.e., facts, dimensions, measures).
Developer	• List of Actions • List of Business Rules • List of Tables, Attributes	Represent the required information for a developer to define the data transformations.

The technical components determined by the developer are modeled in a developer setting. This modeling is matched with the organizational and decisional modeling for presenting the DW structure and data transformations specifications. Based on the general principles of ETL processes functionality, the transformation analysis answers these questions: i) whichdata sources to be selected for the DW, ii) how the DW and data transformation is prepared, iii) what are the actions required for data transformation, and iv) which data to be used for viewing the information.

Referring to these questions, the analysis task can be divided into three phases: i) data sources analysis –answers question number one, ii) business rules analysis – answers question number two, and iii) transformation analysis –answers question number three and four. These analyses are not necessarily executed in sequences. However, the data sources' analysis is carried out separately as *data profiling* tasks (Kimball & Caserta, 2004; Schreiber, 2004). The data profiling is the information about data (i.e., table names, field names, data types) that is captured is based on templates discussed in Section 6.3.3.

By using modeling software tools (e.g., PowerDesigner), the data profiling can easily capture, store, and manage information about data according to the provided template. The data profiling is used to model the data sources as ontology structure during the design of ETL processes. Although the data sources' analysis can be carried out separately with another analysis, the scope of required information as defined in decisional modeling is important to ensure the relevant data sources are

analyzed. Section 6.3.2.1 discusses in detail how the business analysis and transformation analyses are implemented.

6.3.2.1 Transformation Analysis

The transformation analysis deals with the specification of actors and goals at high-level of requirement analysis. The actors and goals that are already defined in the organizational and decisional modeling are further analyzed for defining actions and related business rules to produce the transformations that fulfill the goals. Action is represented in literal statements that explain the aggregation operations to achieve the goals. Based on the Tropos methodology, a Plan approach (Bresciani et al., 2004) is used to present the modeling of the analysis. The meta-model of the plan approach is presented in Figure 6.3.

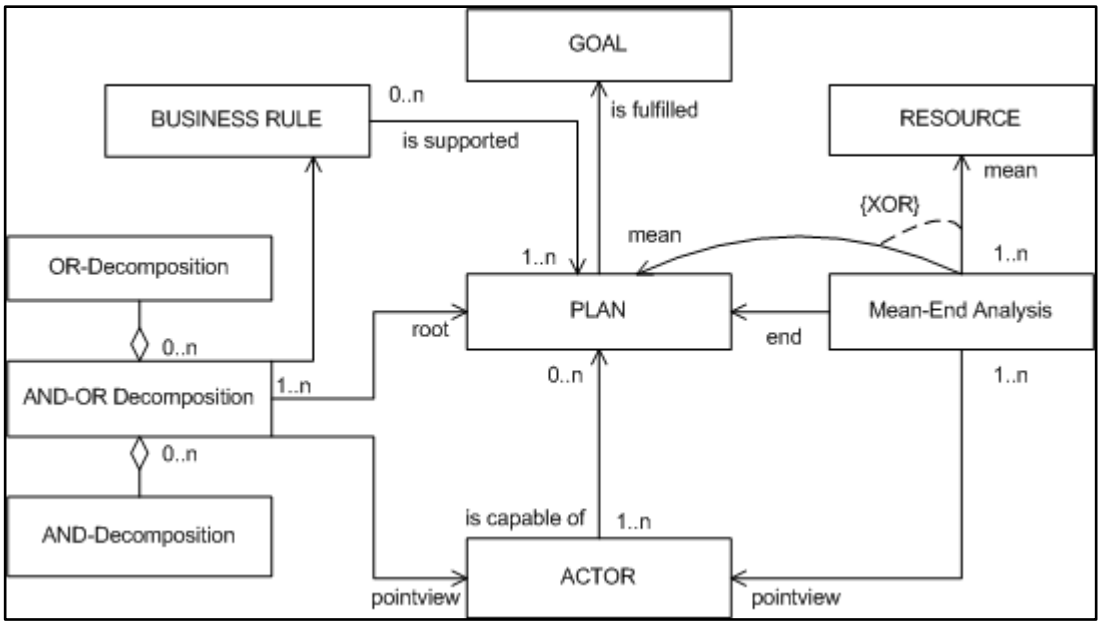


Figure 6.3: Plan Modeling Meta-model

In Figure 6.3, for each actor and goal, the plan and task should be determined to achieve the goals. The same analysis techniques such as MEAN-END and AND-OR are used to produce and extend the developer actor diagram. This task introduces new actors, actions, and resources that complement the goals setting. Inclusion of new actors contributes positively to fulfill some of the functional or non-functional requirements. A developer diagram of new actors, actions, and resources that support the goals of each fact is complied with the DW requirement components (i.e., fact, dimension, measure).

These new actors can be classified into three common types of transformations namely extract, transform, and loading (ETL). These actors are known as software agents in a DW systems(Antonio et al., 2005). In short, the plan modeling contains a number of actions that contribute to the fulfilling of the goal. This contribution is also supported by a list of business rules that are related to the action.

To begin the transformation analysis, the final goals of facts, supported by the related dimensions and measures as defined in the rationale diagram of a decision-maker are selected. Then, tasks are executed by answering these questions: i) what actions are needed to achieve the goals, ii) how the actions can be executed, and which actors will execute these actions. In order to answer these questions, an understanding of the knowledge domain is important to analyze the goals with the related dimensions and measures. By using MEAN-END and AND-OR analysis techniques, the plans needed to fulfill the goals are defined as $Action_1$, $Action_2$,

…Action$_n$. To illustrate the plan modeling, the plan notation is symbolized by the hexagon as shown in Figure 6.4.

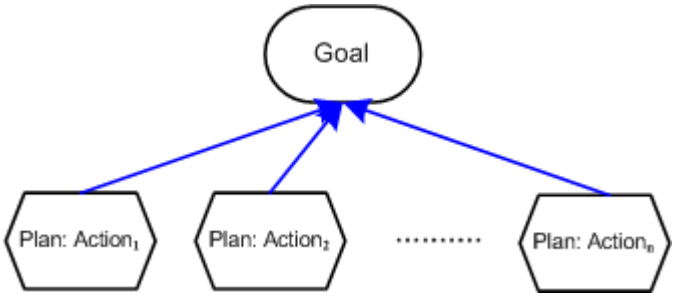


Figure 6.4: Plan Modeling

Further analysis determines the actor who will execute the plan as defined in plan modeling. The actor does not refer to the actors who were defined in organization and decision modeling. Here, the actors mean**E**xtract, **T**ransform, or **L**oading that characterize the actions on each plan. This analysis is focused on the **T**ransform, and actor called *Transformer* because the action for **E**xtract and **L**oading is defined in a straight forward way. Indeed, a transformation could comprisea number of actions (e.g., filtering, merging, joining, converting) in completing the cycle of a transformation process (Kimball and Caserta, 2004; Simitsis, 2004).

The plan modelingexplains the dependency between *Transformer$_1$* to *Transformer$_2$*, and so forth. The actions on data transformation are the main issues that need to be tackled to address the heterogeneity problems. Moreover, the transformation mechanism deals with various data sources that need to be represented by a uniform

model of the data model. The *Transformer*(actor) is symbolized by circle notation as presented in Figure 6.5.

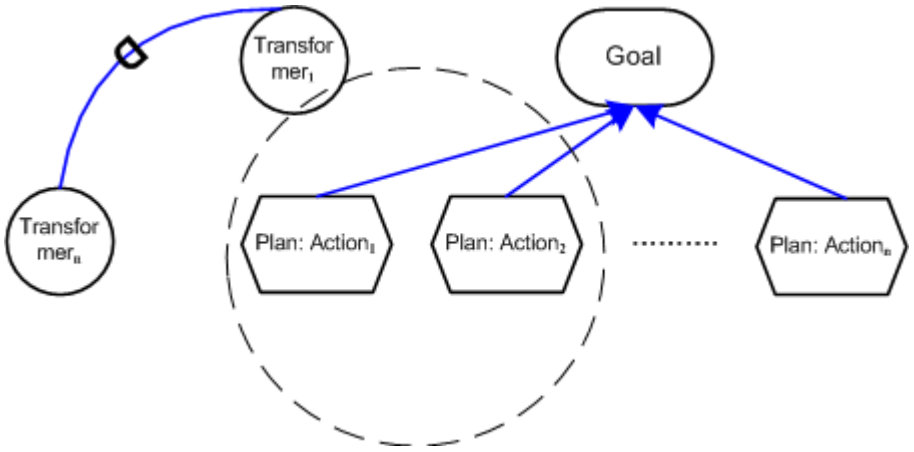


Figure 6.5: Plan Modeling with Transformers (Actors)

Based on the plan approach, a part of the developer perspective can be modeled by analyzing what actions are needed to achieve the goals, how these actions can be executed, and who are the transformer(actors) to execute the actions. Next,the business rule that requires elements for completing the analysis in developer modeling is discussed.

6.3.2.2 Business Rule Analysis

The business rule (Br) is provided by the users to support the execution of a plan. The plans can have more than one rule and are translated to plan modeling by the ETL developer. It is not easy to analyze and define the business rules required by the ETL processes. This is because the business rules for ETL processes are more technical and complex as comparedto the business rules as defined in data modeling

(Kimball and Caserta, 2004). Moreover, most of the business rules given by the business usersand the ETL developer need to be translated into relevant meaning for purposes of data integration and transformation.

However, implementation of data integration and transformation need not necessarily contain the business rules; it depends on the plan goals that need to be fulfilled. There is a plan which needs many business rules to complete the plan process and finally fulfill the plan goals. This explains the importanceof actions and business rules to be documented systematically for further refinement by both user and developer. The business rules are represented by a rectangle symbol as shown in Figure 6.6.

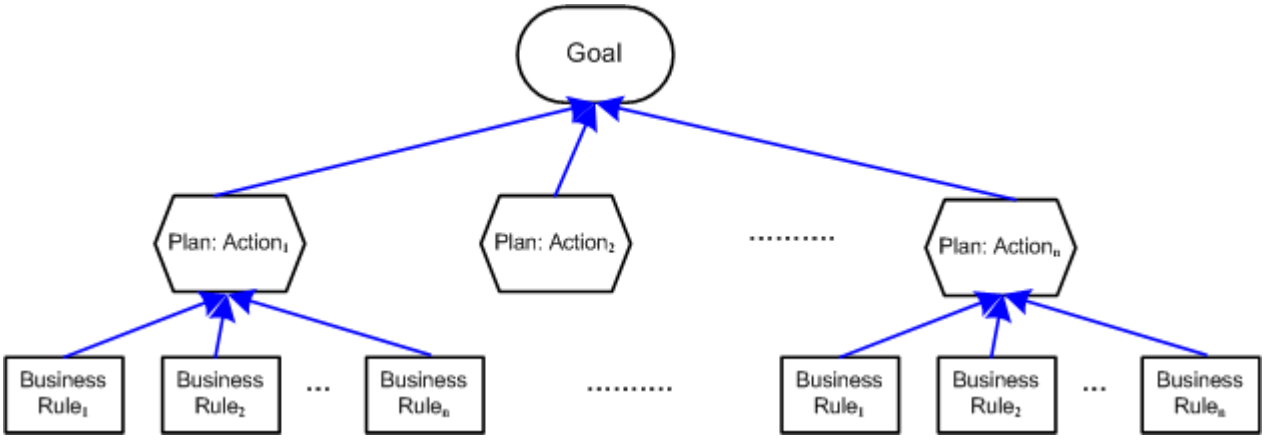


Figure 6.6: Plan Modeling with Business Rules

The initial facts about business rules and related actions are gathered and determined during requirement gathering to produce documentation that is organized in templates. The template is defined in a table form and used to record the information

about business rules (seeSection6.3.3). The business rules are used to determine the boundary of the data sources to be loaded into the DW. This guaranteesonlythe relevant data is used in the integration and transformation process.

6.3.2.3 Suggestion for Aggregation Operators

Based on the *measures*as defined in decisional modeling, and supported by the plans modeling with business rules, the developer can determine the appropriate aggregation and propagation operators for executing the plan actions. For example, if the measure is AMOUNT, then the appropriate operator might be SUM or AVERAGE. The business rules provide controls for executing the actions with aggregation operators for achieving the plan goals.

This entire task is performed within the general concept of transformations that is used by the current ETL tools. As mentioned earlier, the transformation activities are laid within the **E**xtract, **T**ransform, and **L**oading actors. Each of the activities contains their own specific operations that are represented by the aggregation operators. Therefore, understanding the aggregation operators leads the requirement analysis toward the appropriate conceptual design of ETL processes. Clearly, the ETL processes design is used to derive the ETL processes specifications.

6.3.3 Templates for Collecting Requirements

The analysis process is conducted by various methods such as interviews, presentations, discussion, and document analysis. All the information about requirements is documented in an organized manner by using templates. These

templates are used for analyzing user requirements and can be itemized as shown in Table 6.3.

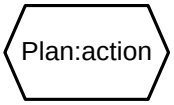
Table 6.3: Templates for Collecting Requirements

Analysis Types	Templates	Modeling Applied
Goal	• (actor, objectives) • (sub-actor, type, goals) • (depender, dependee, goal)	Organizational, Decisional, Developer
Fact	• (fact, description) • (goal, fact)	Organizational, Decisional
Attribute	• (attribute, goal, fact)	Organizational
Dimension	• (goal, fact, dimension) • (dimension, description)	Decisional
Measure	• (goal, fact, measure) • (measure, description)	Decisional
Data Profiling	• (table names, field names, data types, descriptions).	Developer
Transformation	• (goal, fact, action) • (fact, action, description)	Developer
Business Rule	• (fact, action, business rule)	Developer

6.3.4 Notation for Diagram Modeling

The notation used in Tropos and GRAnD was adapted for modeling the diagram produced in RAMEPs. However, several notations are new and introduced in the context of an actor and rationale diagram of developer modeling. These notations are used and complied with the requirement analysis model for the ETL processes as illustrated in Table 6.4.

Table 6.4:Newly developed notation for actor and rationale diagrams

Notation	Symbol	Modeling Applied
Measure – represents an aggregation aspect of analysis for goal to be achieved		Decisional
Transformation – represent actions for plan to be executed for goal to be achieved by fact		Developer
Business Rule – represent business rule to be applied on plan for goal to be achieved by fact		Developer

The notations for presenting the analysis techniques (e.g., AND/OR decomposition, MEAN-END analysis) are adopted from the Tropos methodology. This research is interested in the schemas of the data sources and not the specific notation for representing the data profiling. Therefore, the graphical representation of data sources schemas is not required for this modeling because it is used for constructing the ontology model. The construction of an ontology model is the next task for merging the DW requirements and data sources in a single representation.

6.4 Ontology-Oriented Approach

In RAMEPs, the ontology-oriented approach is used for modeling the DW requirements and their related data sources. As defined in research methodology, the construction of ontology is based on the SIM methodology, which covers the entire process of ontology development. Although the construction for two types of

ontologies (i.e., domain and application ontology) is similar, however applying the ontology model is different. This is due to the different level of knowledge to be captured, i.e., DW requirements and data sources schemas. The way to construct both ontologies and their related ontology model is explained in Section 6.4.1.

6.4.1 Ontology Development Process

As described in Chapter 5, SIM methodology was adapted for constructing the ontology, which consists of steps required by the ontology model of this research. For this research, the ontology development process has five steps as shown in Figure 6.7:

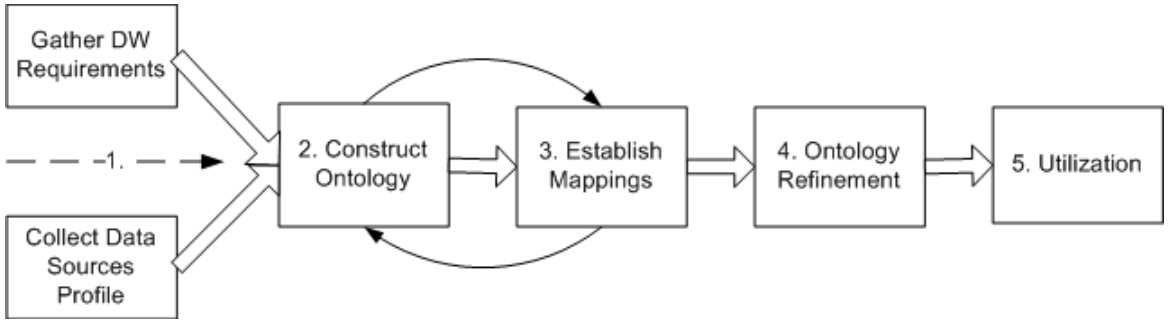


Figure 6.7: The Ontology Development Process for RAMEPs

Briefly, the steps used in developing the DW requirements and data sources ontology areas follows:

i) **Elicit DW Requirements** and **Collect Data Sources Profile** is a process for collecting the information to be captured and defining the scope of the ontology structure. The knowledge to be captured is either DW requirements or data sources schemas.

ii) **Ontology Construction** is a process for manually creating the ontology from the collected knowledge or information. However, data sources ontology can be created through reverse engineering or manually.

iii) **Establish Mapping** is a process for mapping both the ontology for producing the merging ontology. The process is done iteratively for ensuring the required ontology is completed.

iv) **Ontology Refinement** is a process for rechecking the developed ontology is structured accordingly. Several steps are carried out for ensuring the correctness of ontology before allowing for utilization.

v) **Utilization** is a process for using the ontology for application development and maintaining the information architecture due to the changes of data sources and user requirements.

Most of the steps in this methodology can be supported by Protégé-OWL tool. However, some of the steps need to be conducted manuallybecause no functions are available to support the process. The step for ontology construction is done manually because of the need to carefully define the ontology structures (e.g., concept, class, property). The step for ontology mappings is established in an automatic manner, however the final ontology needs to be adjusted manually. The refinement process is done automatically in order to ensure the ontology is structured accordingly.

Since the DW requirements and data sources profile are already available after the requirement analysis process, the next Section 6.4.2 discusses in detail step 2 to 5 for

constructing and utilizing the ontology for designing the conceptual of ETL processes.

6.4.2 Ontology for DW Requirements

The organizational, decisional, and developer models determine the DW glossaries (i.e., facts, dimensions, measures, attributes, actions) through goal-driven diagrams. The glossaries for facts, dimensions, attributes, measures, and actions must be agreed to by the users. This is used for building the conceptual design of ETL processes according to the design framework available (e.g., supply-driven, requirement-driven, hybrid-driven, model-driven).

Since these agreeable glossaries are mapped to the data sources in the heterogeneous environments, the semantic heterogeneity problems still occur in the implementation of ETL processes. Importantly, the agreeable glossaries should be able to present the semantics of user requirements accordingly. Thus, the semantic heterogeneity problems in the data sources can be resolved by using an ontology model. The same approach was successfully applied to resolve the data integration problems from the various data sharing systems as presented by Alexiev et al. (2005).

6.4.2.1 Process of Ontology Construction

This section explains the process for constructing the DW requirements' ontology (DWRO) for semantically describing the requirement glossaries. This ontology should be able to describe the semantics of the DW requirements in high-level meaning, so that the DW requirements can be possibly mapped to the data sources'

ontology for accomplishing the integration and transformation process. The strong linkages between requirement glossaries and appropriates data sources through ontology model produce the ETL processes specifications automatically. This can be done by invoking an appropriate algorithm and reasoning to the application ontology.

In particular, the ontology used mustbe based on description logic (DL), which constitutes most of the commonly used knowledge representation formalism (Baader et al., 2005). This research uses OWL language for knowledge representation that adopts the DL formalism. Furthermore, the Resource Description Framework (RDF) language is used together with OWL in presenting the ontology structure, especially when involving the schemas of the data sources.

The DWRO should be capable to model the following type of information: i) the concepts of the domain, ii) the relationships between concepts and attributes, iii) the attributes and relationship that belongs to each concept, iv) the different formats and values that belong to each attribute, and v) the axioms that belong to attribute and relationship. The concepts refer to the *facts*, whereas the *dimensions*, *measures*, *business rules*, and *actions* refer to the attributes. The relationship between concepts and attributes are:*hasDimension, hasMeasure, hasAction, and hasBusinessRules*. The axiom is used to apply restrictions for the attributes and their relationships.

6.4.2.2 RDF/OWL Features

In OWL-based ontology, the concepts of the domain are represented by *classes,* while the relationships and attributes are represented by *properties*. Due to the specialty of aggregation and population operation in DW systems, specific representation classes must be specified. However, the RDF/OWL features need to be suited for the high-level presentation since all the terms defined are in an abstract form. Therefore, this research uses the standard RDF/OWL[20] features and ontology notation (Skoutas & Simitsis, 2007) as shown in Table 6.5.

Table 6.5: RDF/OWL features

Notation	Name	Description
C	Class	Classes represent the concepts of the domain.
$C_1 \equiv C_2$	Equivalent	To state that two classes are equivalent
$C_1 \sqsubseteq C_2$	subClasssof	To create class hierarchies
$C_1 \sqcap C_2 = \emptyset$	disjointWith	State that two classes two $C_1$and C_2 are disjoints
$C_1 \sqcup C_2$	unionOf	The union of two classes
$C_1 \sqcap C_2$	intersectionOf	To state that two classes are intersected
P	Property (ObjectProperty, DataTypeProperty)	To represent attributes of concepts and relationships between concepts.
dom(P)	Domain	Specifies the class (-es) to which the property belong to.
rang(P)	Range	Specifies the class (-es) to which the value of the property belong to.
$\forall P.C$	allValuesFrom	To restrict the range of property when apply to specific class – *universal restrictions*.
$\exists P.C$	hasValue	To restrict the set of individuals those have at least one relationship along a specific property.
$\forall P(x,y).C$	someValuesFrom	To restrict at least one relationship along a specific property to an individual - *existential restrictions*.
$\geq nP$, $\leq nP$	mix/max cardinality	Specifies the min/max cardinality of a property
RDF (S) features	rdfs: subClassOf rdf: Property rdfs: subPropertyOf rdfs: domain rdfs: range Individual	Specifies the RDF(S) features to present the classes, property, and relationships.

[20]http://www.w3.org/TR/owl-ref/

6.4.2.3 The DWRO Model

The DW requirements contain facts (F), dimensions (D), measures (M), business rules (Br), and Actions (Ac). The DW requirements are modeling the Fact with a set of dimension, a set of measures, a set of business rules, and a set of actions. In the ontology structure, facts (F) are defined as set of classes, whereas the dimensions, measures, business rules, actions, and the relationships among them are defined as set of properties. The relationships refer to the link between class to class, classes to property, or property to property.

As described in ontology definition, set of axioms are used to assert sub-sumptions or restrictions between classes that are defined from the business rules and actions. The business rules specify the domain and range properties, cardinality constraints, disjointness class, whereas the actions are defined as new classes for aggregation functions used for each fact. However, the relationship between fact classes to another fact class is not allowed according to the principles used in DW systems modeling (Kimball, 1996; Rizzi et al., 2006).

Formally, the DWRO can be modeled:

DWRO = (F, D, M, Br, Ac) ……. (6.1)

Where: F = Facts

D = Set of dimensions (D_1, D_2, D_3, …. D_n)

M = Set of measures (M_1, M_2, M_3, …… M_n)

Br = Set of business Rules (Br_1, Br_2, Br_3, ….. Br_n)

Ac = Set of actions (Ac_1, Ac_2, Ac_3, …. Ac_n)

The DWRO is modeled according to the glossaries defined in the requirement analysis process. Clearly, data sources are not included in the DW requirements analysis. Therefore, the type of class values is not defined in the DWRO because the relevant data values are not required to be determined at this level.

6.4.3 Ontology for Data Sources

This section presents the construction process of the data sources' ontology (DSO) for enabling the mapping to the DWRO. The construction process is to transform data sources profile to the ontology structure. The data sources' profile is prepared based on the template (*table names, field names, data types, descriptions*). Practically, this process can be automatically implemented by using *reverse engineering* functionality in any software modeling tools (e.g., PowerDesigner). For this research, the data profiling is prepared manually from the documentation because of difficulty to obtain such a tool. However, there is no significant implication to this research, unless more time is needed to define the data profiles manually.

The challenges here are to construct the data sources' ontology by establishing a semantic mapping from a relational data sources model to OWL ontology model. The idea is to maintain the instance's data sources in a persistent way, while the ontology definition and the corresponding data sources remain in the semantic mapping structure. The mapping rules applied must be able to build DSOthat maintains the semantics of data sources as defined by the database model. This is explained in next Section 6.4.3.1 and 6.4.3.2.

6.4.3.1 Process of Ontology Construction

The data sources' ontology is constructed based on the mapping between data sources schemas and OWL definitions (OMG, 2007; Sane & Shirke, 2009; Shen, Huang, Zhu, & Zhao, 2006). The data sources' schemas for OWL is developed based on data sources profile that were collected by using the template (*table names, field names, data types, descriptions*). The mapping of UML to OWL that is published in Ontology Definition Meta-model (ODM)is used as the guideline for ontology construction. However, this research does not present the data sources' schemas in UML, rather it applies the relation schemas definition in plain statements. The task to map the data source schemas to the ontology structure is also called *semantic reengineering* of the legacy information system. These tasks are as follows:

i) **Apply the *reverse-engineering* approach** to define the conceptual model of existing data sources system. This can be done through any modeling tools such as PowerDesigner. The conceptual model of existing data sources also can be visualized in UML class diagram. By using a tool like PowerDesigner, the task can be easily implemented.

ii) **Define the ontology specifications** by restructuring the UML diagram of data sources profile toward the UML-OWL by following some mapping rules shown by Shen et al.(2006). The rules are also based on the UML to OWL mapping principles, which consist of *owlClass, objectProperty, objectTypeProperty*, and others to present the data sources in the ontology representation (OMG, 2007). These basic rules are:

a) One relation R_i is mapped to one *Concept* C_i

b) Dependency of each foreign key (FK) in one relation R_i on the primary key (PK) in another relation R_j is mapped to an *ObjectProperty* OP_i

c) Each property (exclude FK) of a relation R_i is mapped to a *DataTypeProperty* DP_i

d) Each *tuple* of a relation R_i is mapped to an individual I_i

e) The data type corresponding relationships between relational model and OWL is similar to one given in Table 6.5.

iii) **Construct the ontology by using Protégé-OWL** manually, since there is a limitation on existing tools to construct the ontology from UML-OWL diagram.The ontology specifications are used to construct the ontology and produce the OWL/RDF representation, supporting the recent reasoning techniques (i.e., Pellet) in manipulating the ontology contents.

Generally, the mapping is focused at the schemas level and overall workflow of the mapping process is shown in Figure 6.8.

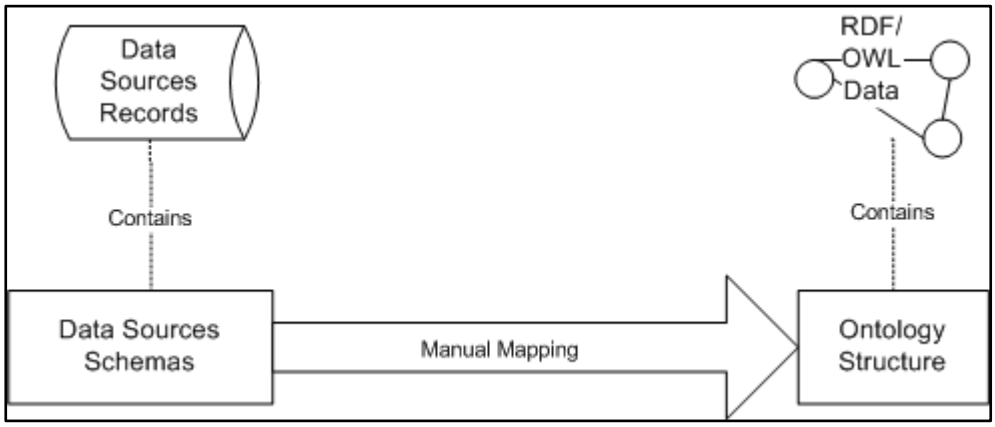


Figure 6.8: The Mapping of Data Sources to RDF/OWL

As far as this research is concerned, no automatic procedure is available to transform the relational database schemas to the ontology structure. However, some methods are proposed to facilitate the mapping and generating the ontology from the database (Barzdins, Barzdins, & Cerans, 2009; Sane & Shirke, 2009). Thus, the manual mapping of data sources schemas to the OWL structure is enough for this research. Importantly, the RDF/OWL coding that corresponds to the data sources ontology can be easily generated by the Protégé-OWL.

6.4.3.2 The DSO Model

Formally, the DSO model is constructed according to generic ontology model. The tuple of DSO is defined as:

DSO = (C, R, A, I) ……. (6.2)

Where: C = a finite set of concepts in the domain

R = a set of relations between concepts.

A = a set of axioms imply in property of concepts.

I = an instance that presents the values of the ontology tuple.

As mentioned in Section 6.4.2.2, the RDF/OWL features in Table 6.5 were also adopted for defining and instantiating DSO. Several RDF/OWL features are used to represent the semantics of data sources schemas. These features were utilized as recommended by W3C in ODM document (OMG, 2007). Importantly, the proposed DSO model should be able to be merged for unifying view of DW requirements.

6.4.3.3 The Mapping Rules

In order to translate the data sources' schemas toward the ontological structure as defined, this research adopts the mapping rules by Shen et al. (2006)with the assumption of data sources' schemas is in the third normal form (3NF). These rules are organized in four groups: i) rule for concepts, ii) rule for properties, iii) rule for restrictions, and iv) rule for instances. The explanation on these rules is as follows:

i) Rules for Identifying Concepts

Rule 1 – For relations R_i in data sources that contain a same primary key, then the information across all the relations can be integrated into one agreeable ontological class. Formally, relations in data sources: R_1, R_2, R_3, …..R_i. Primary Key: P_1=pkey(R_1), P_2=pkey(R_2), ….. P_i=pkey(P_i). If $R_1(P_1)=R_2(P_2)$, ….. $R_i(P_i)$, then set of R can be mapped $\rightarrow C_i$.

Rule 2 – If pkey($R_{i)}$ = pkey(R_j), and ((R_i), pkey(R_i)), ((R_j), pkey(R_j)) $\in I_{c,}$ that means R_i and R_j have the same primary key, then both relations can be mapped to the same concept C_i.

Rule 3 – If rule 2 is satisfied, and concepts for both relations exist, $R_i$and R_j can be mapped to the concept C_i and C_j respectively, but C_i should be a sub-concept of C_j.

ii) Rules for Identifying Properties

Rule 1 – For relation R_i (entity table), if |pkey(R_j)| $\geq$ 1, and then R_i has a primary key, and associates A_i = ($A_i \in$ attr(R_i)) is mapped to the

property of concept C_i. If this is satisfied, then the $| fkey(R_k)| \geq 1$ is satisfied and the foreign key(s) can be mapped to the object property OP_i of concept C_i.

Rule 2 – For relation R_i, R_j, R_k, if $pkey(R_i) \cup pkey(R_j) = fkey(R_k)$. If $pkey(R_i) \cap pkey(R_j) = \emptyset$, $| pkey(R_i)| = |pkey(R_j)|=1$, and R_k is related with R_j and R_j, thus $pkey(R_i)$ and $pkey(R_j)$ can be mapped to the objectproperty OP_i and OP_j respectively. The domain of OP_i is C_i and range is C_j, whereas the domain of OP_j is C_j and range is C_i. OP_i and OP_j are inverseOf relationship and C_i, C_j are corresponding concepts of R_i, R_j respectively.

iii) Rules for Restrictions or Constraints

Rule 1 – If rule 1 in (ii) is satisfied, this means a foreign key exists and object property OP_i has a restriction *allValueFrom*, which have a dependency restriction concept. Related to this rule, other rules are also related to cardinality constraints, which are not necessary to be explained here.

iv) Rules for Instances

Rule 1 – The tuples of the relation R_i can be transformed to the instances for ontology data by mapping R_i to the concept C_i, where one tuple $R_i.t$ becomes instances of C_i, and each $t[A_i = A_i \in attr(R_i)]$ can be transformed to the properties of instance.

Rule 2 – If all the tuples of the relation $R_i$is distinct, then the instances can be asserted to be *allDifferent*.

All the mapping rules proposed are used in this research for transformation process of data sources to the DSO.

6.4.3.4 Merging the DW Requirements with the Data Sources

The need to map and merge the DW requirements toward the associated data sources is important in order to construct a single view of ontology for conceptual representation of ETL processes. The different view of the ontology model (i.e., DWRO and DSO) is a phenomenon of generating different models for a single domain known as heterogeneity in the ontologies (Aleksovski, 2008). Since the heterogeneity problems in data sources have been tackled via ontology representation of data sources, thus the same approach is used in the mapping and merging process. Indeed, the merging ontologies are performed from the domain knowledge of user requirements, and domain and application knowledge of existing application system.

The DWRO should be able to describe the semantics of the DW requirements toward the semantics of data sources in order to establish the mapping between both ontologies. Furthermore, the process of mapping is possibly implemented by using appropriate software and tools with reasoning functionality. As defined previously, the DWRO models the user requirements according to the following elements: i) the concept of the domain, ii) the relationship between the concepts, iii) the attributes characterizing the concepts, iv) the different representation format or value for each of the attributes, and v) the restriction imposed by attributes or relationships.

These elements can be represented in the ontology structure as follows:

i) Concept is represented by Classes (e.g., Student Register, Student Examination)

ii) Relationship is represented by Properties (e.g., hasDimension, hasMeasure)

iii) Type of format or value is represented by new classes in the hierarchy (e.g., currency – RM, Dollar)

iv) Specific elements in DW setting are represented by new aggregate classes (e.g., SUM, COUNT, AVERAGE)

v) Restriction is represented by Axioms (e.g., "Student must be Malaysian")

Based on the DWRO and DSO definitions, the characteristics of both ontologies can be mapped as shown in Table 6.6.

Table 6.6: DWRO and DSO elements mapping

DWRO elements	**DSO elements**	**Ontology mapping elements**
Fact	Concept	Concept ↔ Fact
Dimension = (Dim_1, Dim_2, Dim_3, … Dim_n)	Table:ConceptName (tbl_1, tbl_2, … tbl_n)	Class: ConceptName ↔Dim_1, Dim_2, Dim_3, …Dim_n
Measure = (M_1, M_2, M_3, … M_n)	Attribute: M_1 = $Action_1(attr_1, attr_2, \dots attr_n)$, M_2 = $Action_2(attr_1, attr_2, \dots attr_n)$ M_n = $Action_n(attr_1, attr_2, \dots attr_n)$	Property: ConceptName ↔ [M_1 = $Action_1$ ($attr_1$, $attr_2$, … $attr_n$)], [M_2 = $Action_2$ ($attr_1$, $attr_2$, … $attr_n$)], [M_n = $Action_n$ ($attr_1$, $attr_2$, … $attr_n$)
Business Rule = (Br_1, Br_2, Br_3, … Br_n)	Attribute/Relationship	Property: M_1↔ [$attr_1$ (Br_1), $attr_2(Br_2)$, … $attr_n(Br_n)$], M_2↔ [$attr_1$ (Br_1), $attr_2(Br_2)$, … $attr_n(Br_n)$], …
Action = (Ac_1, Ac_2, Ac_3, … Ac_n)	Behavior/Constraint	Axiom: $Ac_1 \dots Ac_n$↔ [ConceptName ↔$M_1 \dots M_n$]
-	Data	Instance/Individual

In Table 6.6, the ontology elements can be described as follows: i) fact is defined as a concept, ii) concept refers to class, iii) attribute and relationship refer to

property,iv) constraint or restriction refers to axiom, and v) individual refers to instance. Based on the mapping results, new classes and properties pertaining to the merging ontology (i.e., DWRO and DSO) are produced. These new classes and properties present the elements of ETL processes. These elements areshown in Table 6.7.

Table 6.7: Description of New Classes

Type of Elements	Classes: Example	Description
Concept	Student Register	Represent the concept of Student Register
Aggregated	Total student registered	Represent the measure of Student Register
Range	Student must be Malaysian	Represent the business rule for the measure
Aggregation	COUNT, SUM, AVERAGE	Represent the calculation for the measure
Table	RETRIEVE, LOADING	Represent the getting and pushing of the data
Formation	CONVERSION	Represent the transformation of one format to another
	FILTERING	Represent the transformation of one set of data to another set of data

These new classes need to be organized accordingly into the merging ontology during the mapping and merging process. This ontology is called Merging Requirement Ontology (MRO) and is defined systematically through Protégé-OWL. This process ends when the MRO structure is reconstructed and rechecked for consistency and correctness by using the *Pellet* reasoner. A new RDF/OWL document is produced to represent the entire specification of the ETL processes. The RDF/OWL codes are manipulated to determine the appropriate ETL processes specifications. However, before this can be done, some refinement on the MRO

structure needs to be carried out in order to ensure the ETL processes fully satisfy the DW formats and constraints.

6.4.4 Refinement of the Merging Requirement Ontology

The MRO is produced by the DWRO and DSO that represent the entire schemas of DW and ETL processes specification being developed. The ETL processes specification contains knowledge about data sources to be extracted, transformed, and loaded into DW schemas. However, all the elements defined from the goal-oriented process are not straight forwardly accepted without further investigation into their correctness and consistency. Therefore, the refinement process needs to be done in order to ensure the MRO represents the ETL processes operations accordingly. The refinement process focuses on five elements: facts, dimensions, measures, business rules, and actions.

6.4.4.1 Refinement on Facts

Facts that determine the information required can be merged or split according to their similarity or differences of goal to be achieved. Two or more facts can be merged if they have a common goal as defined in decisional modeling. One single fact can be produced with the union or intersection of dimensions, measures, business rules, and actions. The dimensions that are not involved in the new fact can be set as an option for further consideration. In addition, the facts can also be split into another fact if the related dimensions and measures can better characterize the

fact. All these works need to be carried out by ETL developer who has the proper knowledge and experience.

6.4.4.2 Refinement on Dimensions

Dimensions provide an explanation on facts. Refinementof the dimensions needs to determine the detail of information required and their granularity. Thus, attributes or hierarchies that are not relevant in the information required need to be removed, otherwise these will be added in new association to other tables. All these processes can be defined as attributes prune and graft (Giorgini et al., 2008). Finally, the relevant and complete attributes with hierarchies are determined according to the fact definition. However, at times, dimension can be defined as an attribute, or otherwise. This can be refined by rechecking the applicability of dimension in supporting the fact definition.

6.4.4.3 Refinement on Measures

When a *fact* can be merged or split, then the measures also can be merged or split. The refinement starts by identifying the similar or different meaning for measures that can be merged and yield the same values by using single operator to operate. Sometimes, the measure does not require a specific field or attribute, but the measure values can be defined by calculating the number of records. This is identified as *factless* measure (Kimball & Caserta, 2004).

6.4.4.4 Refinement on Business Rules

In addition to the measure, the business rule determines some of the actions for data transformation in ETL processes. Business rules determine the scope and constraints of data that need to be populated into the DW. In the refinement process, the business rules are rechecked and realigned with actions implicit to them. Since the actual attributes from the data sources are already known, some of the business rules can be changed in order to fulfill the process of calculating the measures.

6.4.4.5 Refinement on Actions

Action determines the data integration and transformation for generating the ETL processes specifications in a fact definition. The refining of action involves the process to reorganize the operations required in ETL processes. Each set of actions is responsible for calculating and producing a measure. Several actions need to be changed in order to satisfy the operations to produce the measure. In real practice of ETL implementation, this process refers to the staging area of DW. Many actions are organized and rearranged properly to populate the data sources into the DW in an optimum manner.

6.5 Generating the ETL Processes Specifications

Producing the ETL processes specifications is the main aim for modeling and designing the ETL processes. Using ontology as knowledge representation of DW structure and ETL operations can create the possibility for producing the ETL processes specifications within the scope of user requirements. These tasks can be

realized through manipulation of the semantic annotation of user requirements and data sources. However, this research work anticipatedthe early tasks of DW systems development by setting the data stores (i.e., DW and data sources) and data process from the analysis of user requirements. Thus, the method proposes the set of ETL processes specifications for transforming the data sources to DW, which determines the user requirements through goal-oriented analysis approach.

6.5.1 The ETL Processes Operations

The ETL processes operations comprise the process of extract, transform, and loading. These extract, transform, and loading processes are implemented in sequence and in parallel according to the optimization of process flow as defined by the developer. Most of the generic ETL processes that are frequently used in ETL processes design are shown in Table 6.8.

Table 6.8: The Generic ETL Processes Operations (Skoutas & Simitsis, 2007)

Operations	Actions
RETRIEVE(n)	Retrieve the data from data sources
EXTRACT(c)	Extract the data from retrieving data sources
FILTER(c)	Filters the data from retrieving data sources
MERGE	Merge two or more set of data sources
CONVERT(c_1,c_2)	Convert set of data sources to another format or type
AGGREGATE(f_g, a_1 .. a_2)	Aggregate the data sources into some criteria via some functions
JOIN	Join two data sources related to each other by some attributes
UNION	Unites recordsets from two or more sources
MIN_CARD(p, min)	Filters incoming recordsets having cardinality less than

	min on property p
MAX_CARD(p, max)	Filters incoming recordsets having cardinality more than min on property p
STORE	Loads or store data sources into the DW

Table 6.8 presents the generic types of ETL processes operations that are commonly used in ETL tools nowadays. These operations are the main tasks in ETL processes and are frequently used for integrating and transforming the data sources. Mostly, the ETL tools apply:

i) SQL-based or MDX query for RETRIEVE(n) operation.
ii) A simple conversion functionality for CONVERT (c_1,c_2) between one type units (e.g., EUR for Euro) to another type of units (e.g., RM for Ringgit Malaysia).
iii) Calculate the value of measure by using simple aggregation operators (e.g., SUM, AMOUNT) or complex operator (e.g., Analytic operations – regression, pivoting).

6.5.2 Algorithms for ETL Processes Generation

As stated in MRO, the information as required and their related data sources have been defined in RDF/OWL based language. The design of ETL processes is represented by MRO, which is processed according to the appropriate ontology reasoning mechanism to identify and propose the ETL processes specifications. The functions of the reasoning are based on the inference mechanism for ontology structure that deals with the wide range of information processing in ontology

representation. The inference mechanism glues the semantics of ETL processes for generating the ETL processes specifications.

The algorithm is based on the RDF/OWL model that contains nodes/classes (represent *subject* and *object*) and arcs/properties (represent *predicate/links* between nodes). The nodes and arcs form a statement comprising subject, predicate and object that are always known as triples as explained in research methodology (Allemang & Hendler, 2008). The MRO contains set of RDF/OWL triples, which can be read and manipulated. The procedure to read and manipulate the RDF/OWL statements is developed to achieve the following objectives:

i) Identify nodes/classes and arcs/properties and list their purposes in a tabular form, which is contained in triples (*subject*, *predicate* and *object*).
ii) Recheck the mapping nodes that represent the MRO (i.e., classes for dimensions and measure) and nodes that represent the data sources. These nodes need to hold the following conditions in order to remain applicable:
    - Classes in DWRO and DSO must have a common superclass – explanation about the particular records or data is referred to the right concept of the domain. Therefore, semantics of both classes is truly related.
    - Classes in DSO and DWRO are not disjoint – explanation about the constraints of both classes does not contradict each other.
iii) Examine the pair of nodes/classes and their related arcs/properties. This process identifies each class and their respective properties.

iv) Reasoning is used on classes and their related properties to ensure the correctness and consistency of structure for deriving the ETL processes specifications.

v) The ETL processes specifications are rearranged according to generic ETL workflow model for completing the ETL processes cycle tasks.

Based on these objectives, the formal algorithm is developed for deriving the ETL processes specifications from the MRO. The MRO structure is represented by RDF/OWL language and becomes an input for the algorithm. The algorithm works are based on nodes/classes that present data sources (C_s) and DW (C_{dw}) as defined in MRO. Formally, the algorithm is presented in Figure 6.9. By reading on each node/class, the algorithm rechecks and executes the following tasks:

i) If $C_s \subseteq C_{dw}$ is true, then no transformation activities are required. However, if false and $C_{dw} \subset C_s$ is true, then subset of data sources are relevant to the DW.

ii) If the subset of data sources is relevant to the DW, appropriate operations of the data sources are suggested. The type of operation can be defined as RETRIEVE, FILTER, or EXTRACT. Otherwise, aggregate operations are suitable for DW classes that represent aggregation type.

iii) Recordsets from nodes/classes that are related by the property are combined by several operations such as MERGE or JOIN. If the related classes have a common superclass, then the recordsets are combined by UNION operation.

iv) Finally, the ETL processes specifications end up with the STORE operation to load the transformed data sources to the DW.

6.5.3 Generationof the ETL Processes Specifications

To generate the ETL processes specifications, a prototype application for reading and manipulating the MRO needs to be developed. The MRO is manipulated through Jena 2 Framework that runs Java's programming on Eclipse platform. By using the algorithm developed by the researcher in Figure 6.9, the ETL processes specifications can be generated.The algorithm appropriately extracts, transforms, and loads the data sources to the DW based on corresponding data sources and DW classes' position that related each other in the MRO.

The prototype application is divided into two main modules. The first module is to read the MRO source files through URL functionality. The Java program for this module is shown in Figure 6.10. When the MRO source file is successfully loaded, the second module identifies the ETL classes and produces the ETL processes specification according to ontology definition by applying appropriate reasoning that are defined in algorithm. Part of the Java program for this module is shown in Figure 6.11.

Input: MRO

Output: A List of ETL Processes Specifications (ListOfETL)

Begin

$C_s \leftarrow$ Class corresponding to MRO nodes sources

$C_{dw} \leftarrow$ Class corresponding to MRO nodes DW

IF ($C_s \subseteq C_{dw}$)

{ ListOfETL $\leftarrow \emptyset$ }

ELSE {

IF ($C_{dw} \subset C_s$) {

For each class C_i in the path from C_s to C_{dw} {

IF ($\exists C_g$: Aggregate (C_g, C_i)) {

$C' \leftarrow$ one or more classes C_i or groups (C_i, C)

ListOfETL $\leftarrow$ add AGGREGATE FUNCTIONS (C_g, C') }

ELSE

{

IF ($\exists C_m$: MergeSource (C_m, C_i))

{$C' \leftarrow$ one or more classes C_i or groups (C_i, C)

ListOfETL $\leftarrow$ add MERGE (C_m, C') }

ELSE

{ ListOfETL $\leftarrow$ add FILTER (C_i) }

}

ELSE

IF ($\exists(C_1, C_2)$: $C_s \subseteq C_1$ AND $C_{dw} \subseteq C_2$ AND ConvertTo (C_1, C_2)

{ ListOfETL $\leftarrow$ add CONVERSION (C_1, C_2); $C_s \leftarrow C_2$

$C_i \leftarrow$ i+1 (Repeat for each class in the path from C_s to C_{dw})}

ELSE

{ $C_s \leftarrow$ classes C_0

$C_i \leftarrow$ i+1 (Repeat for each class in the path from C_s to C_{dw})}

}

}

End.

Figure 6.9: Algorithm for Deriving the ETL Processes Specifications

The result of the prototype application is a list of ETL processes specifications, which is explains the population of the data sources to the DWthat can be implemented in real DW systems. The ETL processes specifications can be executed by the ETL tools that support the script-based or SQL-based functions in ETL processes implementation. Moreover, the ETL processes specification needs to be translated into SQL-based coding prior to the execution of ETL processes that are supported by the tools. Probably, some adjustment needs to be added to the ETL tools for complying DW environment and platforms. Nevertheless, the implementation of the ETL processes specifications is out of this research scope.

```
//package etl.specification.owl;
package etlSpecification;
import java.util.Iterator;
import com.hp.hpl.jena.ontology.*;
import com.hp.hpl.jena.rdf.model.ModelFactory;

public class ProgramStudentAffairs {
   // Constants, Static variables, Instance variables, Constructors
public static void main( String[] args ) {
      // read the DWR Ontology file.
      //String source = (args.length == 0) ?
"http://192.168.1.100/phdproject/MergeOntology_Registration.owl" : args[0];
      String source = (args.length == 0) ? "http://localhost/phdproject/gasmalaysia_MRO.owl" : args[0];
      OntModel m = ModelFactory.createOntologyModel( OntModelSpec.OWL_MEM, null );
      // read the MRO
m.read( source );
      ETLClass dc = new ETLClass();
      //DescribeClass dc = new DescribeClass();
          //JOptionPane.showMessageDialog(null, "ETL PROCESSES GENERATION", "Power by
GOFED", JOptionPane.WARNING_MESSAGE);
System.out.println("ETL PROCESSES SPECIFICATIONS FOR STUDENT AFFAIRS DOMAIN");
System.out.println();

if (args.length >= 2) {
         // we have a named class to describe
         OntClass c = m.getOntClass( args[1] );
dc.describeClass( System.out, c );
      }
else {
```

Figure 6.10: Module for Reading MRO source file

```
package etlSpecification;

// Imports
import java.io.PrintStream;
import java.util.*;
import com.hp.hpl.jena.ontology.*;
import com.hp.hpl.jena.rdf.model.*;
import com.hp.hpl.jena.shared.PrefixMapping;

public class ETLClass {
   // Constants, Static variables, Instance variables

private Map<AnonId,String> m_anonIDs = new HashMap<AnonId,String>();
private int m_anonCount = 0;

   // Constructors, External signature methods
public void describeClass( PrintStream out, OntClass cls ) {

        String[] classETL = {"Sum", "Count"};
        renderClassDescription( out, cls );

        for (int s=0; s<classETL.length; s++) {
                if (cls.getLocalName() == classETL[s])
                {
                        renderClassDescription( out, cls );
                }

if (cls.isUnionClass()) {
        renderBooleanClass( out, "MERGE", cls.asUnionClass() );}
```

Figure 6.11:Part of the Module for Generating ETL Processes Specifications

6.6 Discussion

This chapter focuses on the problem of modeling and designing the ETL processes that contributeto addressing the issues of semantic heterogeneity problems and generating the ETL processes specifications. More specifically, the research question (ii) is answered by proposing a requirements analysis method for designing and deriving the ETL processes specifications from early phases of DW system development named RAMEPs. The RAMEPs model is constructed within the scope of organization, decision-maker and developer for ensuring the deliverables of DW requirements are properly analyzed.

The RAMEPs has shown that the ETL processes specifications can be designed from the early phases of DW systems development by utilizing the goal-oriented and ontology-based approach. The methodology used in analyzing the user requirements is supported by DW-Tool and Protégé-OWL, which is widely used by researchers in this domain. The ETL processes model that is part of DW modeling is properly presented through the transformation analysis in developer perspective. This gives new enhancement for the existing approach, which covers the very important aspects of a DW systems model.

RAMEPs is the design approach that focuses on the requirements analysis method from the higher level toward the low level of DW operation abstractions. Thus, the difficulty of mapping between DW concepts and the relevant data sourcescannot be avoided during designing the ETL processes. However, the abstract concepts of ETL processes can be detailed into pieces of generic ETL operations for mapping the appropriate data sources according to the proposed mapping mechanism. This can be done by using ontology, which highly structures the ETL processes, data sources, and DW schemas. Moreover, the reasoning capabilities can possibly automate the generation of ETL processes specifications instantly.

Importantly, the RAMEPs propose a language and methodology that define the conceptual schema of DW and ETL processes directly from the analysis of organization, decision-maker, and developer perspectives. These perspectives are important for ensuring the ambiguity of the DW requirements can overcome the design-related problems in modeling and designing the ETL processes. Additionally,

the designer can fully utilize the supply and demand driven approach for DW requirements that complies with the analysis perspectives.

6.7 Conclusion

The RAMEPs can help the developers to design ETL processes systematically prior to the construction of the DW systems. An application for generating the ETL processes specifications needs to be developed for ensuring the usefulness of RAMEPs and accelerate the implementation of ETL processes. The ontology model helps the developer to resolve semantic heterogeneity problems during data integration and transformation. Moreover, the RDF/OWL language canbe easily used and maintained by tools (e.g., Protégé-OWL),which make the design of ETL processes specifications manageable and controllable, although the changes in user requirements frequently occur. Chapter 7 presents the validation and evaluation process of RAMEPs. The evaluation is carried out by implementing the RAMEPs in various domains of case studies.

CHAPTER SEVEN–VALIDATION AND EVALUATION OF RAMEPS

This chapter presents the validation and evaluation process for RAMEPs by using goal and ontology compliant tools in three domains of case studies. The validationprocess is presented in the context of model correctness of the ETL processes design. The evaluation process is to ensure the implementation of RAMEPs and the expert reviews are used to identify strengthen and weaknesses of the RAMEPs. The discussion about general findings on validation and evaluation process ends with the conclusion of the chapter.

7.1 Introduction

The aim of RAMEPs is to support the design of ETL processes by analyzing and producing DW requirements as required by the decision-maker and organization. Through RAMEPs, the ETL processes is modeled and designed by capturing important information in the DW systems development: i) DW schemas/structure, and ii) data sources integration and transformation. Since the RAMEPs isbased on goal-oriented and ontology approach, the validation process emphasizes the correctness of both approaches. Consequently, the correctness of RAMEPs is not enough until it can be evaluated in the real design of ETL processes.

To validate the correctness and ensuring the consistency of the RAMEPs, the appropriate goal-oriented and ontology compliant tools are required for capturingand analyzing the DW requirements. The compliant goal-oriented tools must be able to

accommodate the organizational, decisional, and developer elements into the modeling functionalities. Furthermore, the compliant ontology tools should be able to capture and present the requirements and data sources into the ontology according to the model as defined in RAMEPs.

The evaluation is conducted for ensuring the usefulness of RAMEPs for designing the ETL processes. This evaluation process is implemented in real DW projects development, which dealswith various domains and different kinds of heterogeneity setting in database architecture. The different setting of data sources architecture requires different approach to elicit user requirements, which requiresthe RAMEPs to be adaptable accordingly. Finally, the RAMEPs are reviewed by DW developers for obtaining their feedback. Evaluation by DW experts and their comments are used to strengthen the RAMEPs weaknesses.

The validation process of RAMEPs by the compliant tools is discussed in Sections 7.2 and 7.3. This is followed by the evaluation process of RAMEPs that was done by conducting three case studies for DW systems development in Sections 7.4, 7.5, and 7.6. Finally, the discussion on expert reviews is discussed in Section7.7.

7.2 Model Checking Process

Generally, model checkers are used to verify the correctness of software systems at design stage (Ogawa, Kumeno, & Honiden, 2008). The correctness of a software system is verified according to their system's properties that must be model-checked. System properties in RAMEPs are DW components (i.e., facts, dimensions,

measures, business rules, measures) as defined from the goal-oriented analysis. The method proposed by Ogawa et al. (2008)was adopted to validate the DW components by using compliant tools (i.e., DW-Tool and Protégé-OWL).

This method was chosen because it uses goal oriented requirement analysis for formal presentation of the software properties. Moreover, the validation of properties focuses on the sufficiency of design against requirements, which is similar to the research objectives. However, this research approach is based on the Tropos model that emphasizes the goal and resources that describe the DW characteristics. The model checking process by using compliant tools are illustrated in Figure 7.1.

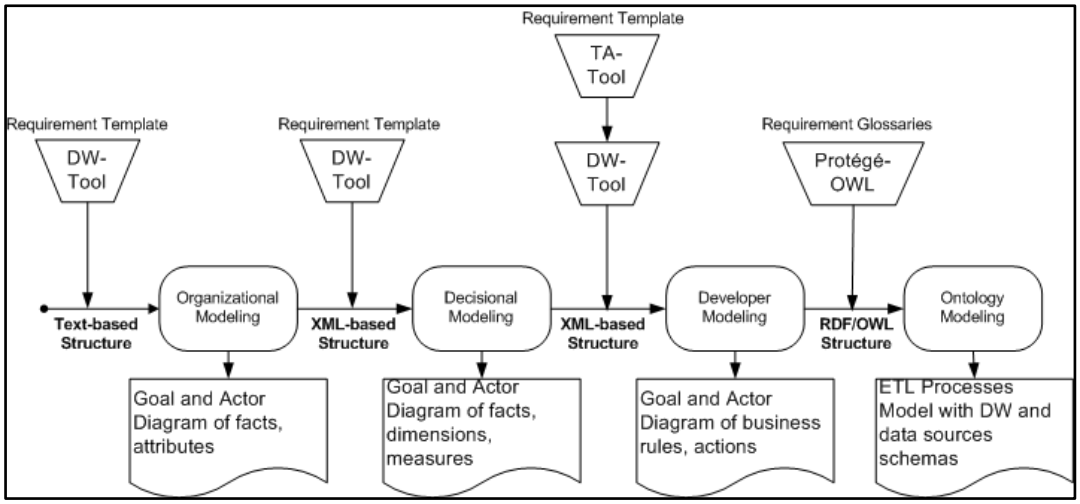


Figure 7.1: Model Checking Process and Compliant Tools

As illustrated in Figure 7.1, the compliant tools are used to ensure the DW components are properly captured from one model to the next model. For example, the goals, facts, and attributes in organizational modeling are correctly supporting the goals, facts, dimensions, and measures in the decisional modeling. These DW components in decisional modeling must correctly support the data transformation

components in developer modeling. Finally, the complete DW requirements are modeled as ontology and rechecked for their correctness as ontology structure by using the *Pallet* reasoner.

Since the DW-Tool does not supported the data transformation analysis as required for the ETL processes, a transformation analysis tool called *TA-Tool* was developed which provides the data transformation diagram in developer modeling. The TA-Tool was developed and integrated with the DW-Tool because the model produced in the DW-Tool is based on XML representation. By reading and manipulating the XML-based model, the data transformation diagram (i.e., contains actions and business rules) is easy to be modeled and integrated with the existing decisional modeling.

7.3 Tools for Validation

The approach to validate the RAMEPs by using tools is guided by the snapshot generation method used to validate UML and OCL model by Gogolla, Bohling, & Richters (2005) and testing the validity of UML profile model by Abdullah (2006).Tools used for validating the RAMEPs are chosen from the existing outstanding research tools. These tools contain features that are capable of validating the correctness of goal and ontology design models. The limitations of these tools are acceptable because these are used for research purposes and are not yet utilized as commercial tool.

However, in the transformation analysis, new functions are required and thereforenew applications are developed to support thesefunctionalities and be a part of the validation tools. The application is used to complete the entire model of ETL processes that are accomplished throughout the RAMEPs. The roles of compliant tools in a validation process are explained in the Section 7.3.1, 7.3.2, and 7.3.3.

7.3.1 DW-Tool for Organizational and Decisional Analysis

The DW-Tool is a tool for a developer to design DW schemas by using the goal-oriented approach. This tool was developed in Tropos project[21] and successfully applied in GRAnD approach (Giorgini et al., 2008). The DW-Tool functionalities are:

i) **Storing the requirements** – information gathered during the interviews with stakeholders are captured and recorded into templates as discussed in Chapter 6.
ii) **Managing the dictionary** – the requirements are organized as a dictionary, which can be used in other models during the design.
iii) **Modeling the organizational perspective** – the requirements are modeled as organizational modeling based on the given templates.
iv) **Modeling the decisional perspective** – the requirements are modeled as decisional modeling based on the given templates.

[21] http://troposproject.org/tools/dwtool/index.htm

v) **Establish conceptual design** – support the producing of DW conceptual design (i.e., dimension modeling) with various design frameworks (i.e., supply, demand, and mixed). The dimension modeling can be refined later.

The DW-Tool is developed in Java and represents the modeling diagram in XML-based structure. This research applies the controlling mechanisms in DW-Tool to validate the correctness of a model on each modeling perspective.

7.3.2 TA-Tool for Transformation Analysis

The transformation analysis is not supported by the current DW-Tool. In order to make the transformation analysis model available in DW-Tool, and connected to the organizational and decisional model, the intermediate tool is required. These limitations were addressed by developing the supporting tool -*Transformation Analysis tool* (TA-Tool). The functionality for inserting or updating actions and business rules was fully supported and workable. Then, these new transformation analysis diagrams were shown in the DW-Tool as developer modeling. Furthermore, the diagrams in developer modeling were modeled as DWRO.

The TA-Tool uses the XML-basedDW-Tool goal diagram to be read and manipulated for inserting and updating existing transformation diagrams. New transformation diagram becomes part of the completed DW requirement's diagram, which represents the design of the ETL processes. The transformation diagram containing actions (represented by hexagon symbol) and business rules (represented by rectangle symbol) are fully supported by DW-Tool. By defining the actions and

related business rules in TA-Tool, the complete diagram of a developer model can be visualized in DW-Tool. This helps a developer to design the ETL processes in a complete and systematic manner.

This research overcame the existing constraint of DW-Tool by enhancing the functionalities with the TA-Tool that supports the building of a transformation analysis diagram. The completion of the transformation diagram completesall the components of ETL processes, which comprises facts, dimensions, measures, actions, and business rules.

7.3.3 Protégé-OWL for Ontology model

Protégé-OWL is a tool for building domain models and knowledge-based applications with OWL-based ontology. At its core, Protégé-OWL provides a rich set of knowledge modeling structure and functions that support the creation, visualization, and manipulation of ontologies in various ontology languages. The ontology-based applications make useof the Protégé-OWL to share, reuse, and process domain knowledge in many applications such as electronic commerce, information management, scientific knowledge portal, and semantic web services.

As described in Chapter 4, the purpose of ontology is to describe the concepts, properties, and relationships in a particular domain that provides a vocabulary of the domain for human and computerized system. The ontologies can range from taxonomies, classifications, database schemas and others. These ontologies are developed by two ways of modeling namely: *Protégé-Frames*, and *Protégé-OWL*.

Current Protégé tool supports the Protégé-OWL that is widely used in the semantic web community.

This research uses the Protégé-OWL because it is free, developed using open-source platform, and is tightly integrated with Jena 2 Framework so that it can be used for developing the application for generating the ETL processes specifications. Compared to other ontology tools (e.g., Oiled, Apollo, OntoLingua, OntoEdit, WebODE, KAON, etc.) available in a market, the Protégé-OWL is the only research-based tool that has been commercially used by the community. Moreover, the core-based functions are written in Java, which gives strong extensibility and reliability for adding new functions to satisfy user requirements. This research has benefits from the Protégé-OWL ontology to be manipulated by Java's program by using the Jena 2 framework, which is difficult to be realizedin other ontology tools.

7.4 Model Checking Examples

In the model checking examples, the process shows how the DW components are captured from the templates and model the requirements from one modeling to another modeling. The compliant tools show the continuity of the model to ensure the consistency and the correctness of DW components until the model is ready to be transformed to the ontology model. Again, in the ontology model, the DW requirements with data sources are rechecked to ensure the correctness of DW representation in RDF/OWL ontology structure. All the checking examples are based on the Gas Malaysia (M) Sdn. Bhd case study. This case study focuses on the utility

billing area, and the DW information is provided for the Billing Manager as the decision maker.

7.4.1 Organizational Modeling

In organizational modeling phase, the goal is created in the goal analysis tab. Then, the created goals are used for defining facts in the fact analysis tab. To ensure the consistency of goals for the next analysis, these goals cannot be updated in the fact definition and in the dimension analysis. This scenario is shown in Figure 7.2.The DW-Tool ensures the goals can only be inserted or updated within the goal analysis area. The gray area of goal description explains the checking mechanism of the model correctness.

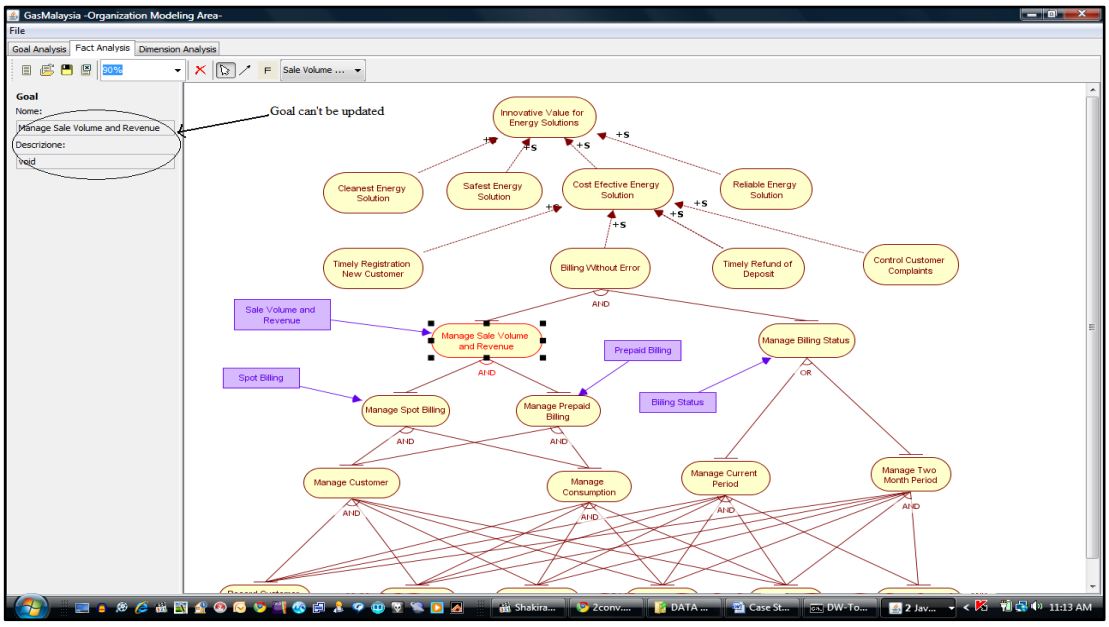


Figure 7.2: Goal Model Checking for consistency

This principle is applicable for all phases of modeling to avoid inconsistency among the goal diagrams produced by the DW-Tool. The fact is created in the fact analysis

tab and cannot be inserted in the dimension analysis. However, the fact can be updated in dimension analysis for ensuring its correctness in the organizational modeling by rechecking the dimension and goals. This scenario is shown in Figure 7.3.

Validation of DW requirements in organizational modeling ends when the goals, facts, and attributes/dimensions are completely created. The linkage between goals, facts, and attributes/dimensions must be consistent in order to ensure the organizational modeling correctly captures the DW requirements from the perspective of organization. Incorrect organizational model will discard the creation of the decisional model and therefore it is unable to proceed to the next modeling perspective (i.e., decisional and developer modeling).

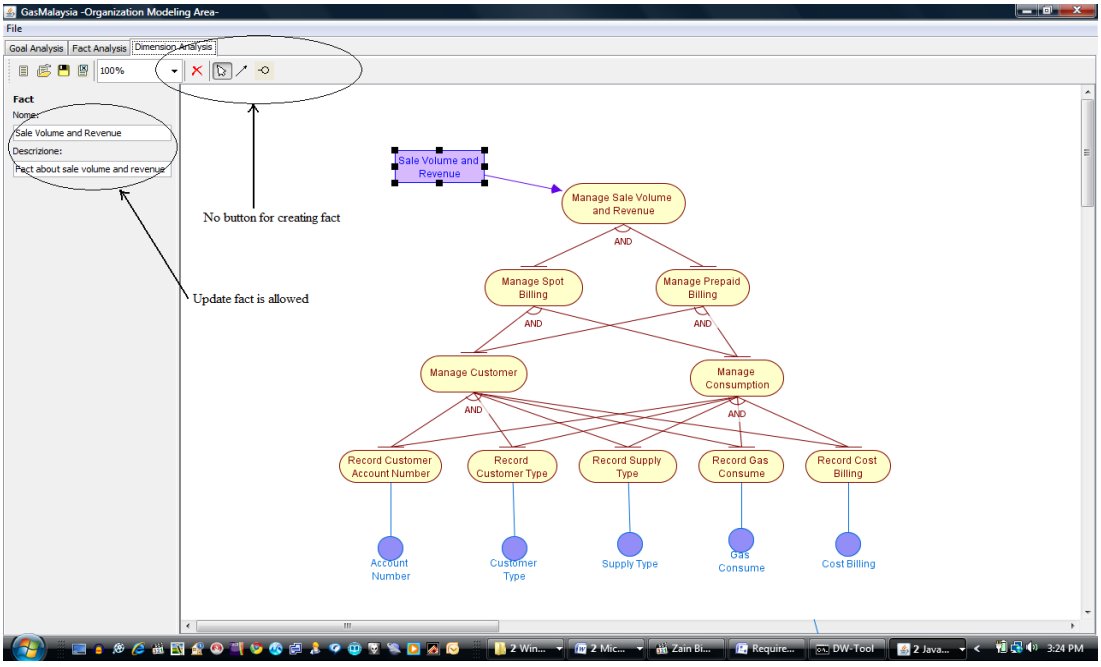


Figure 7.3: Fact Model Checking for consistency

7.4.2 Decisional Modeling

In decisional modeling phase, the goal is created by loading the organizational model into the goal analysis tab. Then, the organization goals are used to define goals for decision makers. To ensure the consistency of decision maker goals for the next analysis, these goals cannot be updated in the fact, dimension, and measure analysis tab. Any changes in decision goals need to be done in the goal analysis tab. For example, the decision goals cannot be updated in the fact analysis tab as shown in Figure 7.4.

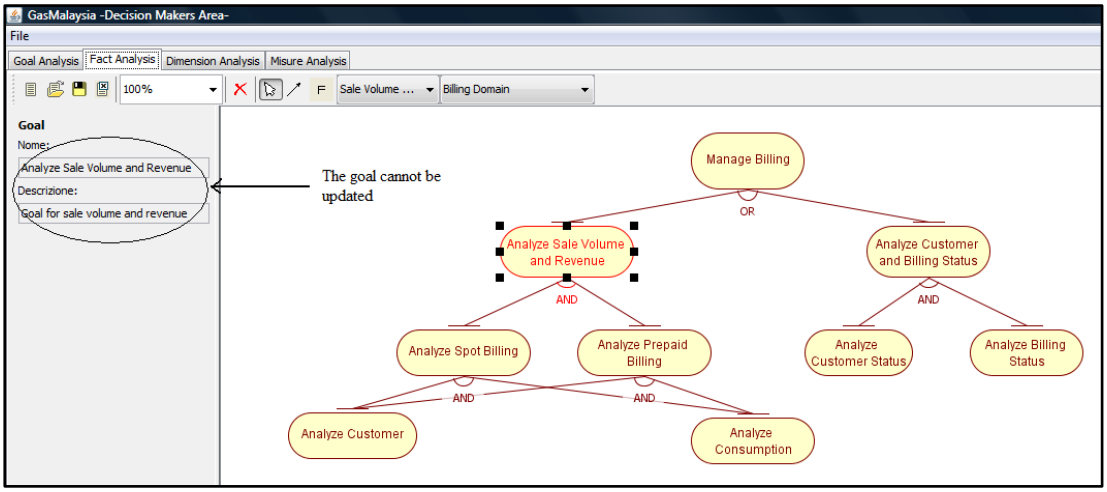


Figure 7.4: Fact Components in Decisional Modeling

As in decision goals, the correctness of the fact, dimension and measure components are guaranteed by allowing these components to be updated in their appropriate analysis tab respectively. The DW-Tool ensures the appropriate analysis is done according to the sequence of processes and supports the requirements determined in the decision-maker perspective. The example of this scenario is shown in Figure 7.5.

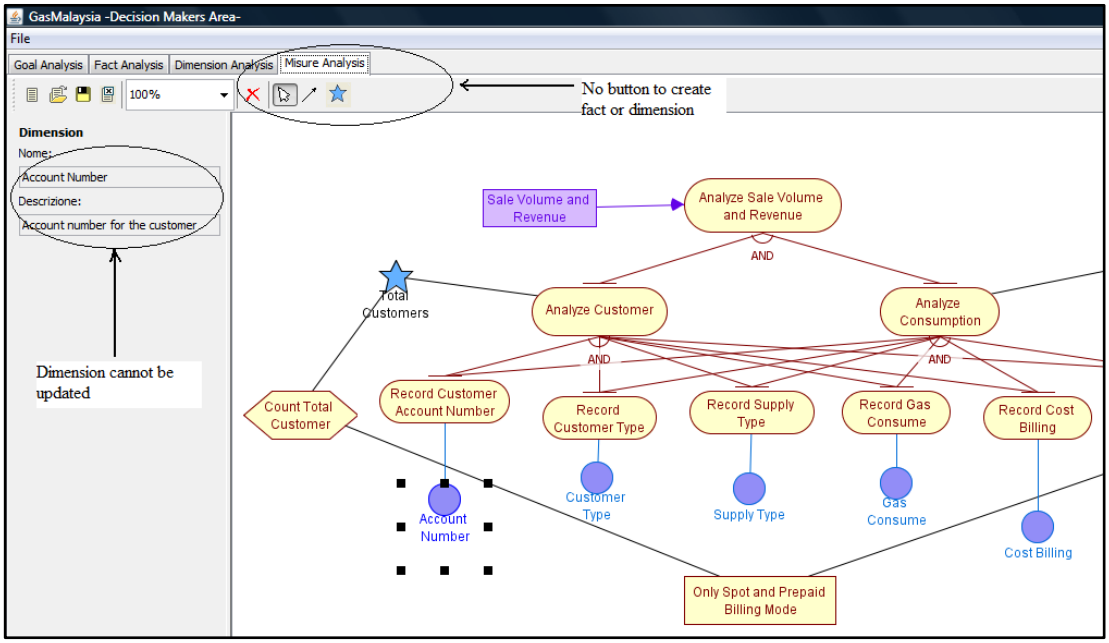


Figure 7.5: Dimension and Measure components in Decisional Modeling

7.4.3 Developer Modeling

In developer modeling, two important components for transformation analysis are captured: plan of actions and its related business rules. These components are not supported by the current DW-Tool. This limitation is addressed by using the TA-Tool. The inserting or updating of new actions and business rules for data transformation analysis is done through TA-Tool and the results viewed in DW-Tool. The diagrams in DW-Tool are stored as XML structure and helptoadd new diagram for the data transformation analysis into the DW-Tool through TA-Tool.

The TA-Tool captures the plan of actions and related business rules for achieving the measure and fulfills the goal of the decision maker. The screen to capture the transformation analysis components is shown in Figure 7.6.

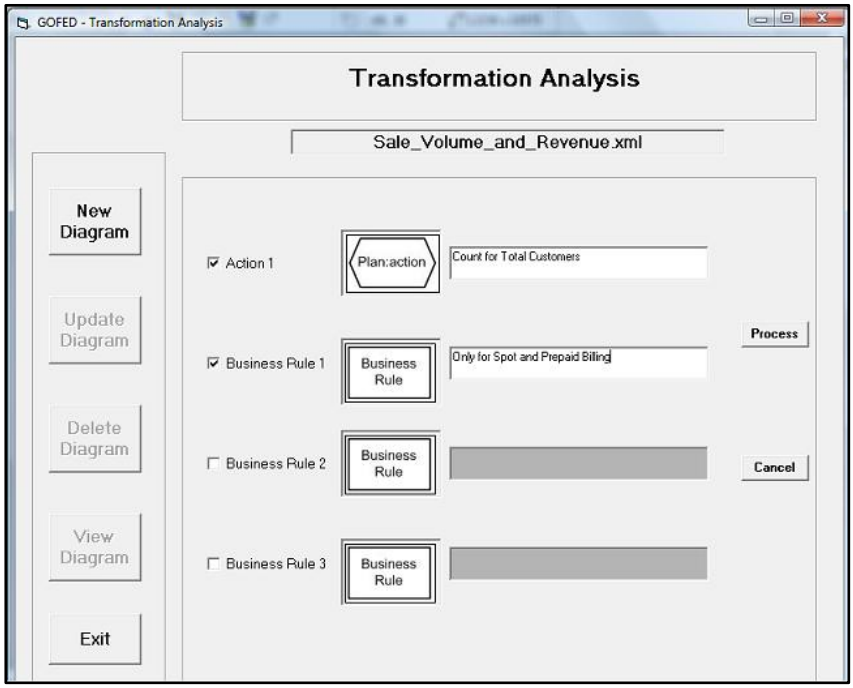


Figure 7.6: Transformation Analysis Tool for Developer Modeling

When the actions and business rules are captured, the transformation analysis in developer modeling is established and the rational goal diagram of the DW requirements is reorganized properly for better views in DW-Tool. This scenario is presented in DW-Tool as shown in Figure 7.7.

The goal diagram as shown in Figure 7.7 is a final goal-oriented model of the DW requirements. Successfully capturing all the DW components (i.e., fact, dimension, measure, action, and business rule) and the consistent use of concepts, properties, and relationship among the components, will ensure the correctness of the RAMEPs approach in designing the ETL processes.

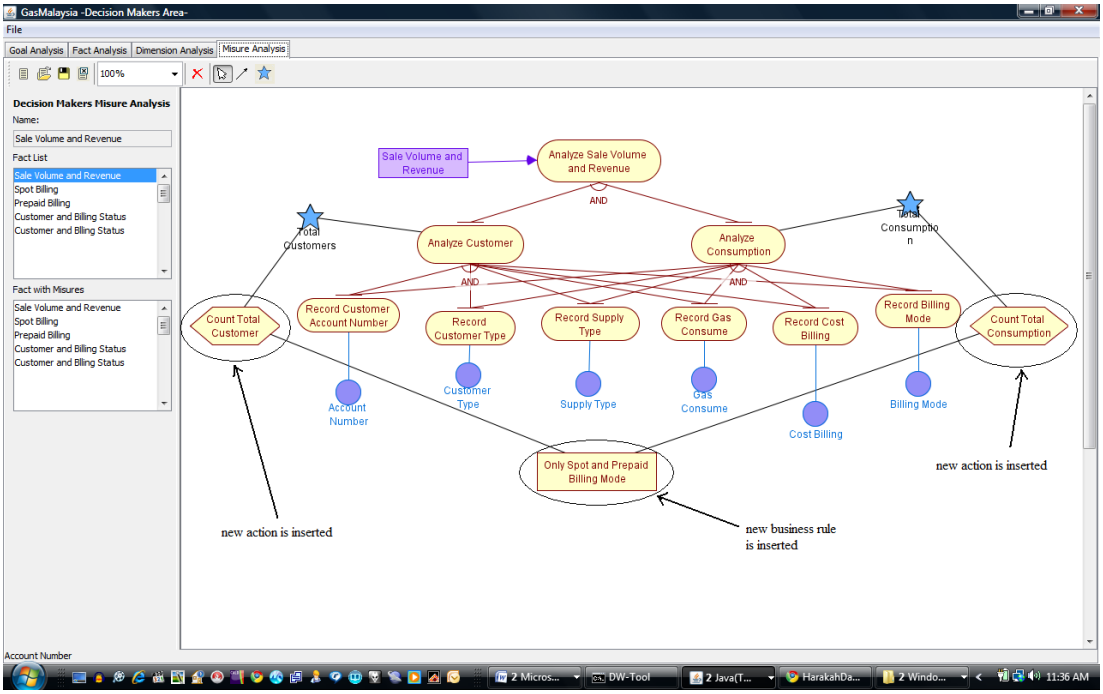


Figure 7.7: Transformation Analysis Diagram in Developer Modeling

The validation process through DW-Tool and TA-Tools shows that the modeling of DW requirements on each modeling phase has been successfully maintained. The next step is to transform the DW requirement's model into the ontology and merge it with the data sources' ontology for completing the ETL processes design. The DW requirement ontology needs to be validated to ensure the DW semantic requirements are represented accordingly.

7.4.4 Ontology Modeling

The ontology model is validated by using *Pellet* reasoner. *Pellet* reasoner is a complete protégé-OWL checker that is based on DL. The current *Pellet* reasoner, which comes together with the Protégé-OWL has gthe ability to validate the

RDF/OWL-based ontology model (Sirin, Parsia, Grau, Kalyanpur, & Katz, 2007)by executing the following functions:

i) **Consistency checking** – ensures an ontology is free from any contradictory facts such as type, property-value, equality and inequality assertion.
ii) **Concept satisfiability** – checks whether the classes should have any instance or not.
iii) **Classification** – computes the subclasses' relations between every named class to create the complete class hierarchy.
iv) **Realization** – finds the most specific classes belonging to specific individual.

In Protégé-OWL, the *Pellet* reasoner is integrated with the OWL editor to be easily used by the developer. By running the *Pellet* reasoner, the ontology model is checked for correctness and consistency. The representation of ETL processes semantics through ontology is validated by the *Pellet* reasoner. Therefore, the usefulness of RAMEPs approach depends on the correctness of ETL processes specifications that are produced from the execution of ontology.

Consistency checking is used to ensure ontology class, properties, relationships, and formal ontology definitions are free from any contradictory facts. In Protégé-OWL editor, the checking for existing class name during the creation of a new class “CUSTOMER” is shown in Figure 7.8.

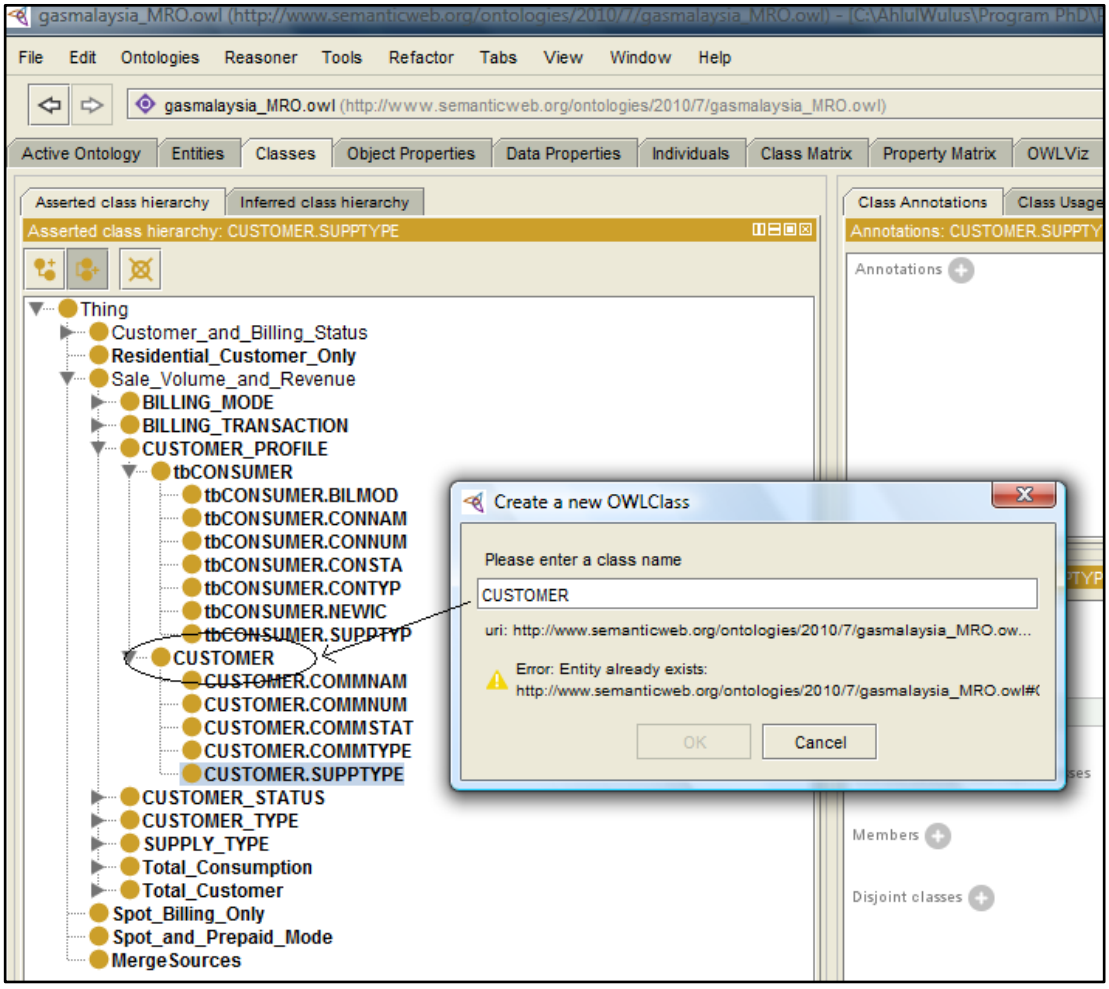


Figure 7.8: Consistency checking for new class name

The formal definition used in ontology is an abstract syntax and semantic document of OWL, which was developed and defined by W3C[22]. The example for this scenario is shown in Figure 7.9, where the formal definition of restriction "*any*" is not supported for semantic of *Total_Customer* and should use "*some*". The "*some*" restriction is known as *existential restrictions* that describe the existence of *at least one* (some) individual who has a relationship with the member of the class. In other words, the relationship should have some values from the restrictions.

[22]http://www.w3.org/TR/owl-semantics/syntax.html

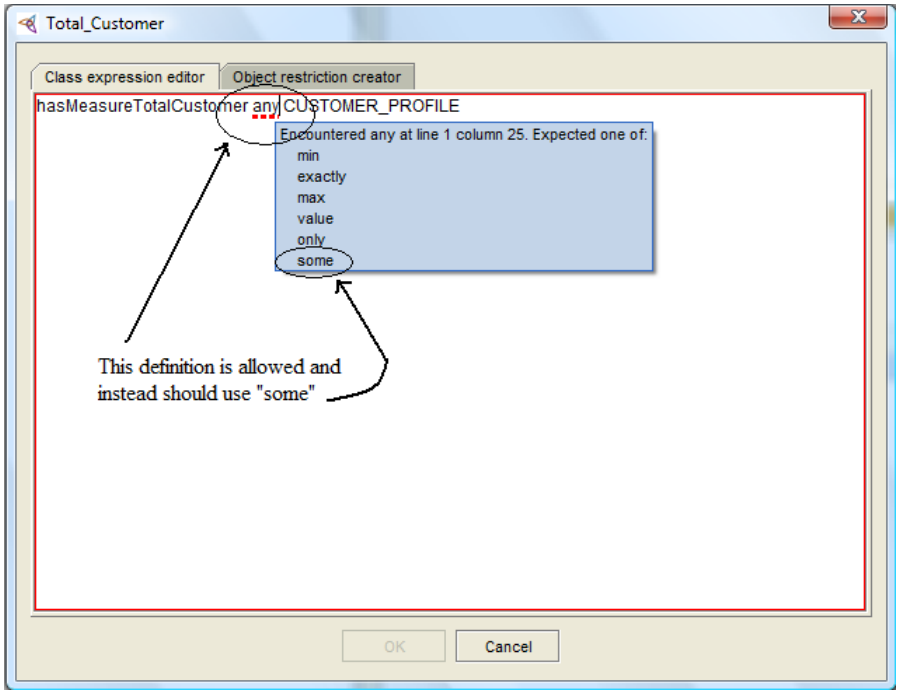


Figure 7.9: The use of formal definitions in ontology

The *Pellet* uses this formal definition for checking the correctness of the DW requirements' ontology of Gas Malaysia. The *Pellet* reasoner is invoked in Reasoner menu by clicking the *Pellet* reasoner. When selected, the Protégé-OWL editor processes the ontology and produces the *inferred* ontology in new tab area as shown in Figure 7.10. The inferred ontology tab area is shown in color, and highlights the *error* in red color if an error exists. In this scenario, no errors are found and therefore the DW requirements are correctly represented in ontology.

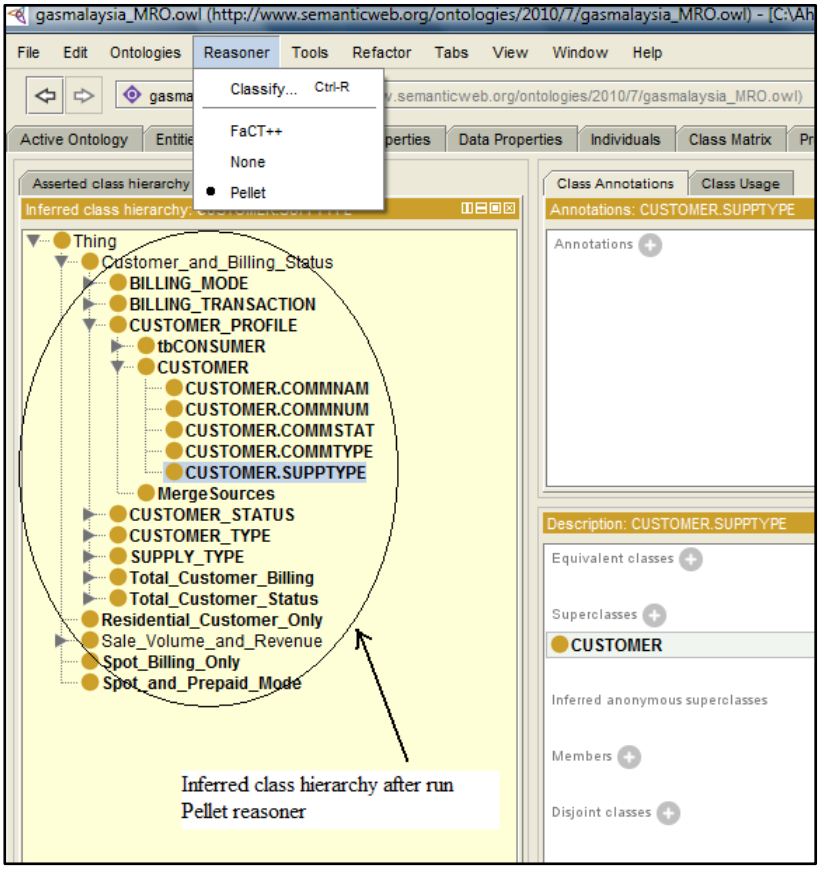


Figure 7.10: Invoked Pellet reasoner to infer the ontology

7.5 Evaluation Using Case Studies

The purpose of these case studies is to evaluate the usefulness of the RAMEPs approach in designing and developing the ETL processes for DW systems. The evaluation is conducted on real world case study. The results of the evaluation give significant impact on the usage of the RAMEPs approach to be implemented in large scale DW systems development. The number of case study is three, which is adequate for the purpose of this research. These case studies are chosen in order to strengthen the evaluation results, which are developed from diverse scenarios of heterogeneous environments. Moreover, the case studies only focus on the particular DW area or domain.

7.5.1 Case Study 1 – Student Affair Area in University Domain

This case study is based on theresearchof modeling business intelligence model in an academic domain(Ta'a, Bakar, & Saleh, 2008). However, requirement analysis work was not properly tackled and disregarded the goal-oriented paradigm. The requirement's elicitation process is based on structured interviews with the Universiti Utara Malaysia (UUM) stakeholder (i.e., Director of Academic Affairs Department (AAD) and System Analyst) and study on current system documentations, which focuses on goal-oriented business processes.

7.5.1.1 DW System Environment

UUM has developed a University Management Information System (UMIS) to support the university functions as required by the users such as students, operational staff, management staff, MoHE, and the public. UMIS comprises several main applications that are implemented in different databases. These applications such as Academic Student Information System (ASIS), Graduate Academic Information System (GAIS), Personal Information System (PERSIS), Integrated Financial and Accounts System (IFAS), and others are integrated as shown in Figure 7.11.

This case study focuses on the DW system development for producing information of the student affairs that comes from ASIS and GAIS. Originally, these systems were designed by different departmentsand are entirely managed by the AAD. However, these systems are still implemented in different databases, and therefore still face the heterogeneity problems during the data integration and transformation.

Based on this environment, this research has designed the ETL processes according to RAMEPs approach.

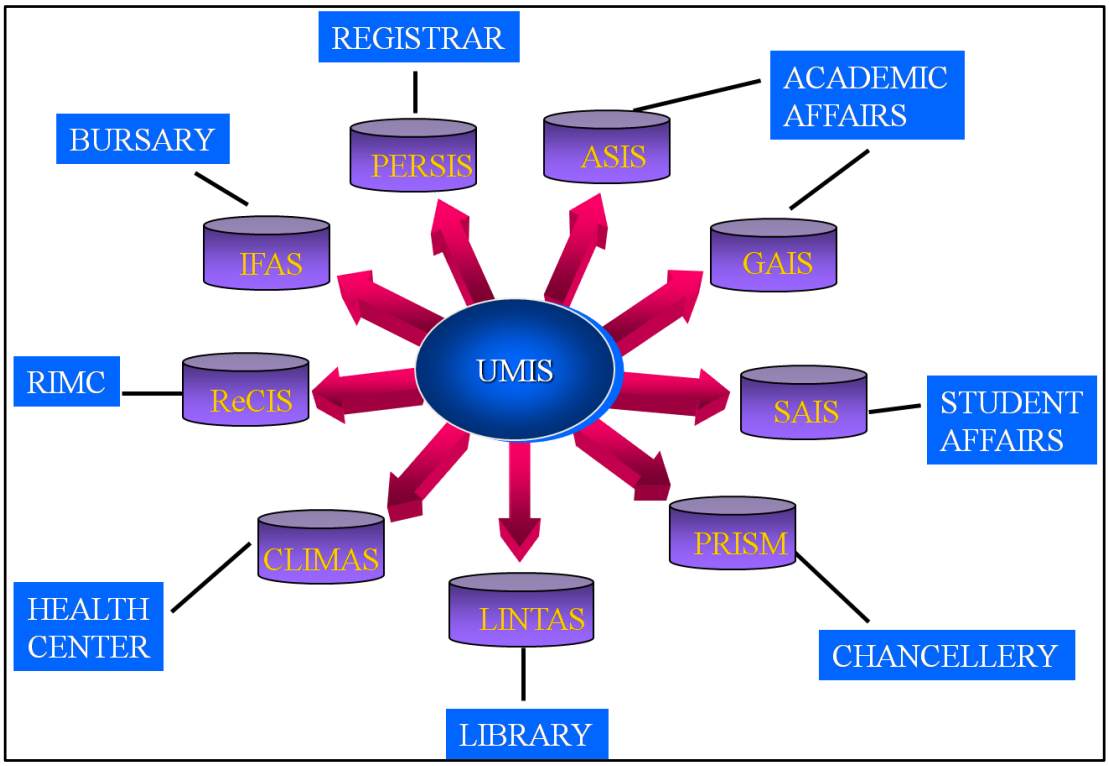


Figure 7.11: University Management Information System (UMIS)

7.5.1.2 Goal-Oriented Requirement Analysis

Based on the results of the interview, the university goals are identified and details of the AAD goals are explored in supporting the university's main goals. The university goals are shown in Figure 7.12.To simplify the process, the case study is focused on the subject area of student affairs. The sub-goal *to be the center of excellence for management education* is relevant with the business tasks of AAD. Thus, the next task of requirement analysis is focused on this sub-goal. The scenario of student affairs that needs the information from the DW system to support the goals can be described as follows:

"The AAD depends on the student for achieving the excellent student and depends on the lecturer for the goal of creating a culture of academic excellence. Moreover, the lecturer depends on the student for the goal of providing excellent teaching and learning."

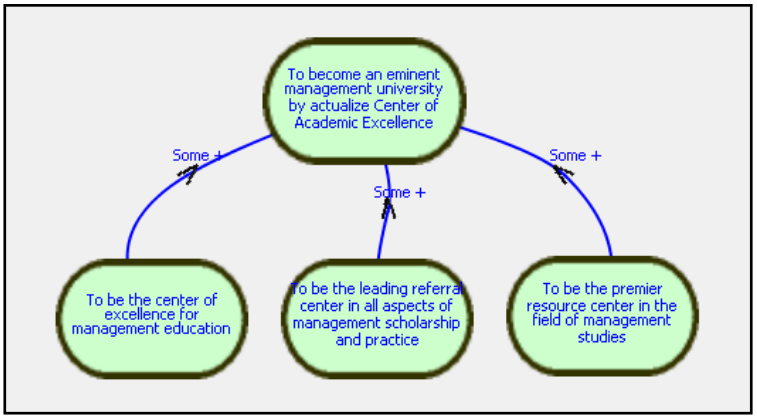


Figure 7.12: University Goals

The analysis task commencesby modeling the requirements in the perspective of organization (i.e., the AAD). In organization modeling, each phase of analysis is implementing iteratively. In goal analysis, the stakeholders involved in student affairs are identified and are represented by using the actor diagrams. An Actor diagram explains about dependencies among actors (i.e., stakeholders such as AAD, student, and lecturer) in university and is presented in Figure 7.13.

The analysis on the actor diagram produced the requirement's documentation that are organized in three difference templates namely: main actor (actor, objectives), sub-actor (sub-actor, type, goals), and dependencies (depender, dependee, goal). The scenario of student affairs in supporting the AAD and university goal is applied for both under-graduate and post-graduate students. Even though both business

processes are not similar, it must support the information required by the AAD and university. In other words, both need to be integrated for producing the information as required.

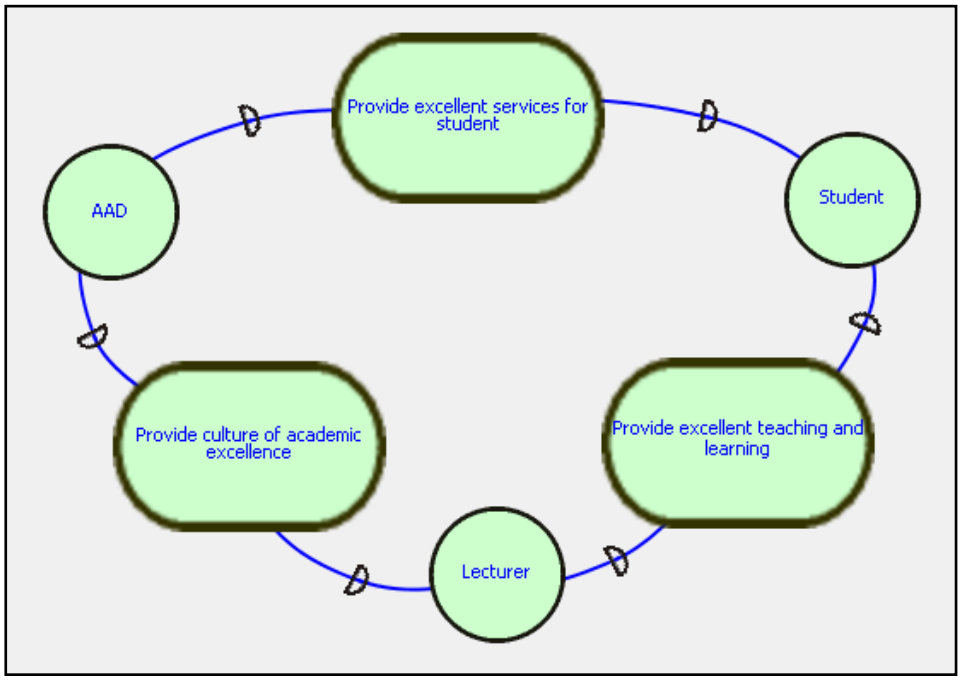


Figure 7.13: Actor Diagram for University

The next tasks are to analyze the DW requirements within the perspectives of organization, decision-maker, and developer. All these works are presented in Appendix A. Section 7.5.1.3 discusses the final results in goal and ontology modeling, and the ETL processes specifications generation.

7.5.1.3 Result for Goal-Oriented Requirement Analysis

After the transformation analysis is completed, the information about facts, dimensions, attributes, measures, actions, and business rules are presented in the DW requirement's diagram. The diagram for *Student Registration* and *Student*

Performances is shown in Figure 7.14 and Figure 7.15 respectively. These represent the final DW requirements to proceed to the ontology model. Based on this diagram, the DW schemas (i.e., dimensions and measures), and ETL activities (e.g., count student registered) are suggested.

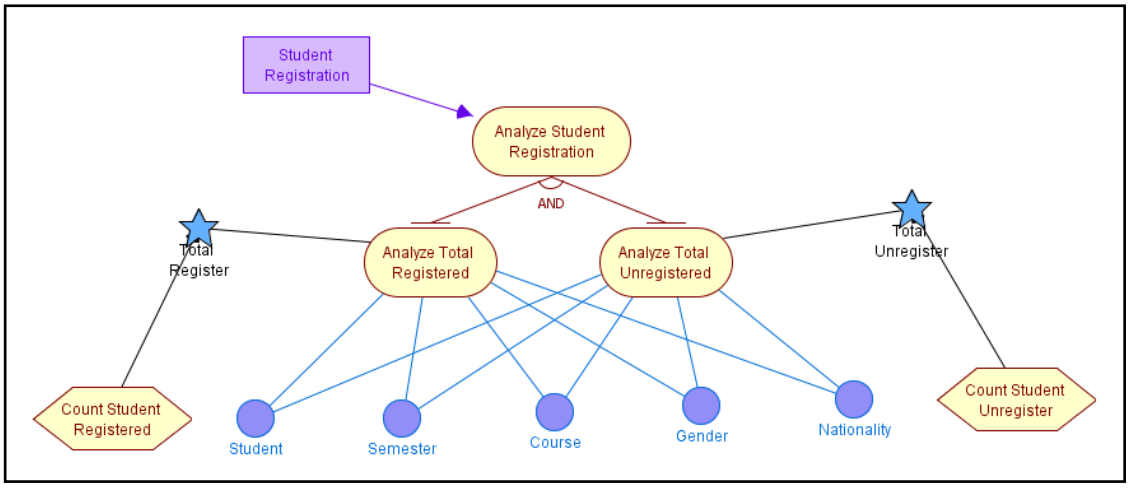


Figure 7.14: Student Registration Goal Diagram

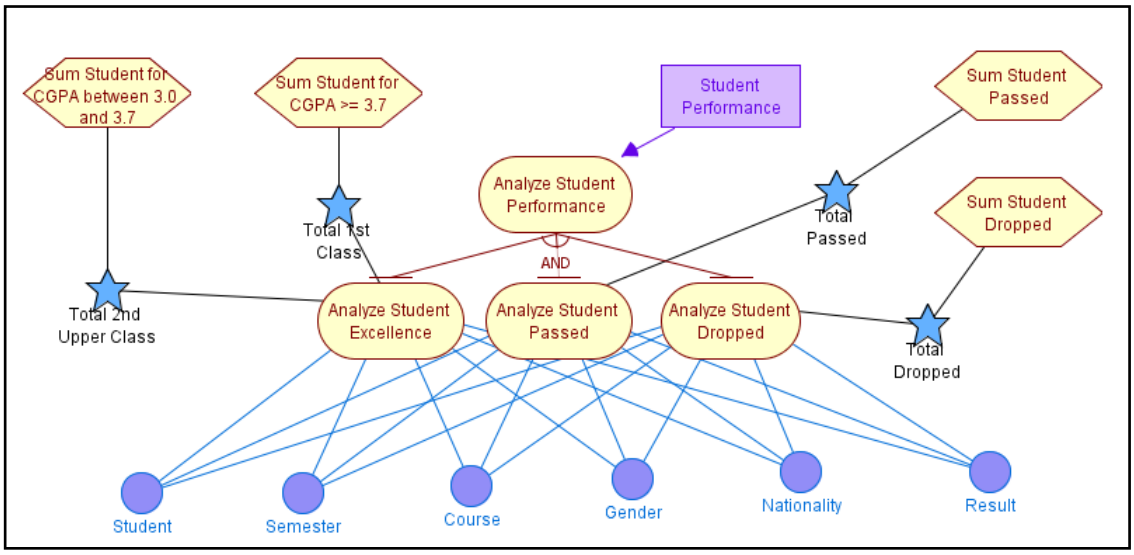


Figure 7.15: Student Performances Goal Diagram

In Figure 7.14, the *Analyze Student Registration* diagram (refer to Section 6.3.4 for details on notation used) proposed the DW schemas as follows:

Fact (Student Registration)

Dimension (Student, Semester, Course, Gender, Nationality)

Measure (Total Registered, Total Unregistered)

Action (Count Student Registered, Count Student Unregistered)

Business Rule ("Student must be Malaysian")

In Figure 7.15, the *Analyze Student Performance* diagram proposed the DW schemas as follows:

Fact (Student Performances)

Dimension (Student, Semester, Course, Gender, Nationality, Result)

Measure (Total 1st Class, Total 2nd Class, Total Passed, Total Dropped)

Action (Sum Student for CGPA between 3.0 and 3.7, Sum Student for CGPA greater or equal to 3.7, Sum Student Passed, Sum Student Dropped)

Business Rule ("Student must be Malaysian")

The DW components produced from the requirement analysis process are modeled as ontology structure for generating the ETL processes specifications. The result for this is discussed in the next Section 7.5.1.4.

7.5.1.4 Results for Ontology Modeling

The DW components such as facts, dimensions, attributes, actions, and business rules are modeled in ontology as a conceptual design of ETL processes. The ontology is constructed based on the defined model O = (F, D, M, Br, Ac). Set of classes representing the concepts of the facts, dimensions, and measures, set of properties representing relationships between facts, dimensions, and measure, and set of axioms used in defining the business rules, actions, and relationship between classes are given. All these definitions are translated into the ontology model (i.e., DWRO) as presented in Appendix A. The ontology model includes the ontology for data sources ASIS and GAIS (i.e., DSO).

The mapping process involves the identification of similarity and dissimilarity of concepts and their associate attributes toward the data sources. These elements are represented in the ontology structure as follows:

- Concept is represented by classes such as Student Registered, Student Performance.
- Relationship is represented by properties such as hasDimensionStudent, hasMeasureTotalRegister.
- Specific element in DW is represented by new classes such as SUM, COUNT
- Restriction is represented by axioms such as “Student must be Malaysian”

Based on the mapping definition as described in Table 6.6 in Section 6.4.3.4, the ontology mapping between DWRO and DSO is shown in Table 7.1. These mapping should not change the semantics of user requirements as presented in DWRO.

Table 7.1: DWRO and DSO mapping for Student Registration

DWRO	**DSO**	**The mapping elements (DWRO ↔ DSO)**
Fact (Student Register)	-	Concept: Student Registration
Dimension (Student, Semester, Course, Gender, Nationality, Result)	Concept: Student Profile (t210student, t801studmas) Concept: Sex (t012jantina, t801jantina) Concept: Session (t005term, t005termx) Concept: Program (t006program, t808kursus) Concept: Race (t013bangsa, t801ras)	Student ↔ Student Profile Semester ↔ Session Course ↔ Program Gender ↔ Sex Nationality ↔ Race * Result is not applicable in this Fact. Thus, no mapping is established.
Measure (Total student register, Total student Unregister)	- Concept: Student Profile for status active - Concept: Student Profile for status inactive	[Total student register] ↔ Student (Active) [Total student unregister] ↔ Student (Not active)
Business Rule ("Student must be Malaysian")	Concept: Race (t013bangsa, t801ras)	[Student must be Malaysian] ↔ [Race]
Action (COUNT for Student Register, COUNT for Student Unregister, FILTER for Student must be Malaysian)	Concept: Student Profile (t210student, t801studmas), Concept: Race (t013bangsa, t801ras)	[COUNT for Student Register] ↔ [Student Profile is active] [COUNT for Student Unregister ↔ [Student Profile is inactive] [FILTER Student must be Malaysian] ↔ [Student Profile JOIN Race is Malaysian]

Table 7.1 present the mapping elements of DWRO and DSO that were derived from the analysis process of user requirements and supported by the related data sources. However, to complete the entire cycle of ETL processes design, the tasks must have actions for *extract* and *loading* functionalities. These functionalities are the generic

activities for extracting and loading data sources to the DW after transformation activities are completed. The actions for extract and loading can be added as shown in Table 7.2.

Table 7.2: The Actions for Extract and Loading Activities

DWRO	**DSO**	**The mapping elements (DWRO ↔ DSO)**
Action (RETRIEVE for Student, Semester, Course, Gender, Nationality)	Concept: Student Profile (t210student, t801studmas) Concept: Sex (t012jantina, t801jantina) Concept: Session (t005term, t005termx) Concept: Program (t006program, t808kursus) Concept: Race (t013bangsa, t801ras)	[RETRIEVE Student] ↔ [Student Profile] [RETRIEVE Gender] ↔ [Sex] [RETRIEVE Semester] ↔ [Session] [RETRIEVE Course] ↔ [Program] [RETRIEVE Nationality] ↔ [Race]
Action (LOADING for Fact into the DW)	Concept: Student Profile (t210student, t801studmas) Concept: Sex (t012jantina, t801jantina) Concept: Session (t005term, t005termx) Concept: Program (t006program, t808kursus) Concept: Race (t013bangsa, t801ras)	[LOADING Student] ↔ DW_Student [LOADING Gender] ↔ DW_Gender [LOADING Session] ↔ DW_Session [LOADING Course] ↔ DW_Course [LOADING Race ↔ DW_Race [LOADING Total student register] ↔ DW_StudentRegister [LOADING Total student unregister] ↔ DW_StudentUnregister

Based on the mapping results, new classes and properties pertaining to the merging ontology (i.e., DWRO and DSO) are produced. These new classes and properties are listed in Table 7.3.

Table 7.3: The New Classes and Properties for Student Registration

New Class	**Class Type**
Total student registered	Aggregated
Total student unregister	Aggregated
Student must be Malaysian	Ranged
MERGE	Merging
COUNT	Aggregation
FILTER	Ranged
RETRIEVE	Table
LOADING	Table

These new classes are reorganized accordingly into the merging ontology through Protégé-OWL. The merging process is done through the ontology setting as defined in Table 7.4. This setting example is for *Student Registration* merging ontology.

Table 7.4: Setting for Ontology Merging of Student Registration

Mapping List	**Ontology Setting**
Merge ASIS, GAIS	Classes Student : t210student ∪ t801studmas Gender : t012jantina ∪ t801jantina Session : t005term ∪ t005termx Course : t006program ∪ t808kursus Race : t013bangsa ∪ t801ras MergeSources: hasMergeStudent *some* Student, hasMergeGender *some* Gender … Properties hasMergeStudent(Domain:Student, Range:t210student, t801studmas) hasMergeGender(Domain:Gender, Range:t012jantina, t801jantina) …
FILTER Race for "Malaysian"	∃hasMalaysian ← Total_Registered, Total_Unregistred hasMalaysian *some* Total_registered hasMalaysian *some* Total_Unregistered
AGGREGATE (COUNT) for Student Registered	∀hasMeasureRegister ← Total_Registered hasMeasureRegister only Total_Registered

AGGREGATE (COUNT) for Student Unregistered	∀hasMeasureRegister ← Total_Unregistred hasMeasureRegister only Total_Unregistered

This process ends when the ontology structure is reconstructed and rechecked by using *Pallet* reasoner (as explained in Section7.3). The new structure of merging DWRO and DSO with new classes is known as the merged ontology (MRO). In Protégé-OWL, each class and property is shownwith a label, which explains the relationship between class to class, and class to properties. The MRO diagram is shown in Figure 7.16.

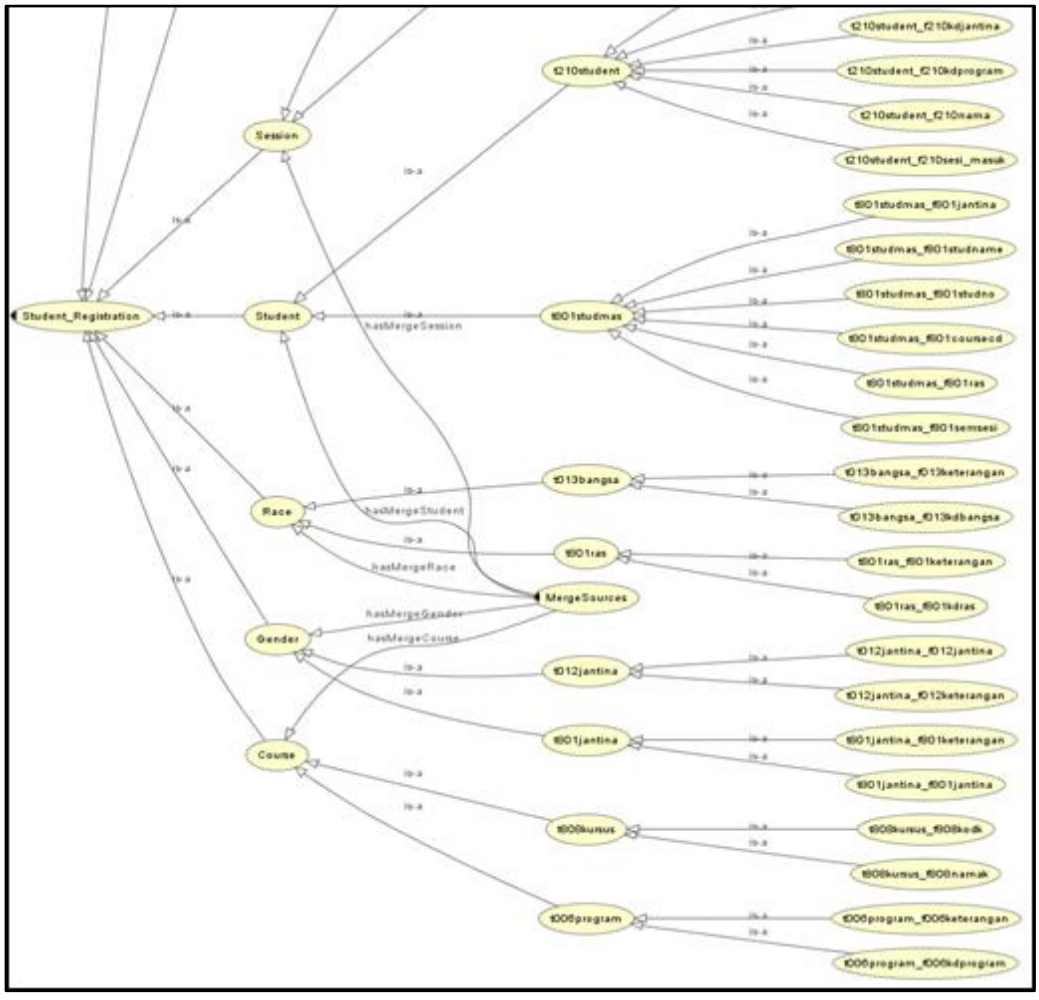


Figure 7.16: The MRO for Student Affairs

7.5.1.5 Results for Generating the ETL Processes Specifications

Producing the ETL processes specifications is part of the objective for this research. The ETL processes activities comprise the process of extract, retrieve, merge, filter, count, and load as shown in Table 7.5.

Table 7.5: The ETL Processes for Student Affairs

ETL Processes	Actions
EXTRACT()	Extract the data from the data sources ASIS and GAIS
RETRIEVE()	Retrieve the particular table from the ASIS and GAIS
MERGE()	Merge data set of ASIS and GAIS
FILTER()	Filters the merged data set according to specific conditions
COUNT()	Count for Total student registered, Total student Unregistered
LOADER()	Loads data into the DW

The ETL processes specifications were generated based on MRO that was represented by RDF/OWL. A snippet of the RDF/OWL is shown in Figure 7.17.

```
<!http://www.semanticweb.org/ontologies/2009/1/GoalRequirementOntology.owl#hasDimension
Register
<owl:ObjectProperty rdf:about="&GoalRequirementOntology;hasDimensionRegister">
<rdfs:range rdf:resource="&ApplicationOntology;Course"/>
<rdfs:range rdf:resource="&ApplicationOntology;Gender"/>
<rdfs:range rdf:resource="&ApplicationOntology;Race"/>
<rdfs:range rdf:resource="&ApplicationOntology;Session"/>
<rdfs:range rdf:resource="&ApplicationOntology;Student"/>
<rdfs:domain rdf:resource="&GoalRequirementOntology;Student_Registration"/>
</owl:ObjectProperty>

<!--
http://www.semanticweb.org/ontologies/2009/1/GoalRequirementOntology.owl#hasMeasureRegi
ster
<owl:ObjectProperty rdf:about="&GoalRequirementOntology;hasMeasureRegister">
<rdfs:domain rdf:resource="&GoalRequirementOntology;Student_Registration"/>
<rdfs:range rdf:resource="&GoalRequirementOntology;Total_Registered"/>
<rdfs:range rdf:resource="&GoalRequirementOntology;Total_Unregistered"/>
</owl:ObjectProperty>

<!-- http://www.semanticweb.org/ontologies/2009/1/MergeOntology3.owl#hasMalaysian
```

Figure 7.17: A snippet of MRO for Student Affairs

To generate the ETL processes specifications, a prototype of application for reading, and understanding the MRO was developed by using Java programming. This application manipulates the MRO through Jena 2 Framework that runs on Eclipse platform. The manipulation process is guided by the algorithm as proposed in Figure 6.9, Section 6.5.2, which is the ETL processes specifications generated. A part of the ETL processes specifications, results from the prototype application are shown in Figure 7.18.

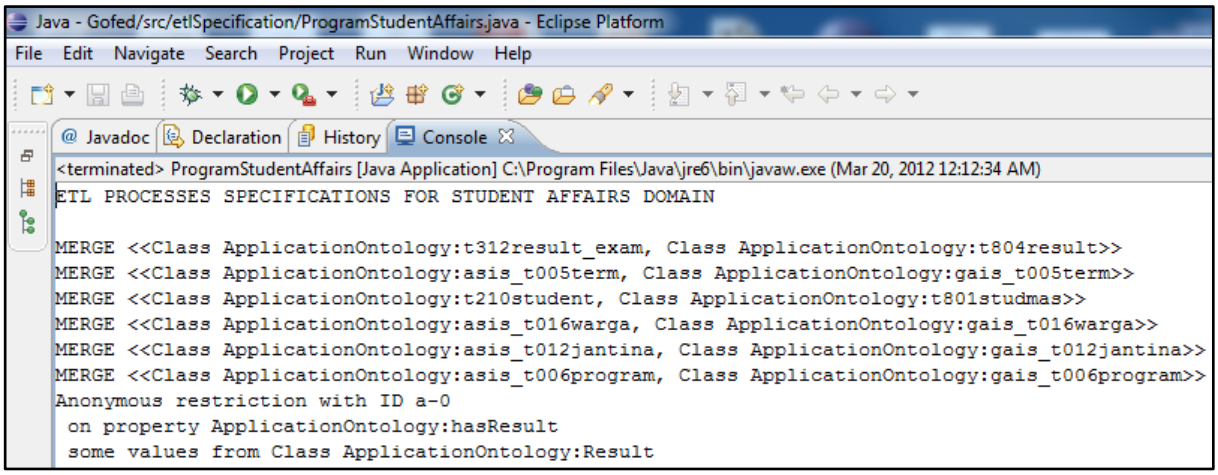


Figure 7.18: List of ETL Processes Specifications for Student Affairs

The results show that the ETL processes can be designed by RAMEPs and the ETL processes specifications from the ontology model of the DW requirements produced. In the future, the ETL processes specifications can be translated into SQL statements or applied directly into any ETL tools for implementing the ETL processes in the DW systems.

7.5.2 Case Study 2 – Billing Utility Area in GAS MALAYSIA

Gas Malaysia (M) Sdn. Bhd is a company to promote, construct, and operate the Natural Gas Distribution System (NGDS) within Peninsular Malaysia. The main office is located in Shah Alam, supported by three regional offices at Prai, Gebeng and Pasir Gudang and seven branch offices throughout Peninsular Malaysia. The company's mission of providing the cleanest, safest, cost effective, and reliable energy solutions has motivated them to provide innovative energy solutions to the nation. Billing utility is one of the important functions for ensuring all the company businesses operate efficiently to support the mission.

7.5.2.1 DW System Environment

The business activities or business process for billing domain in Gas Malaysia can be categorized into four main activities: i) Utility Billing Information System for Residential Consumers, ii) Industrial Billing Information System for Industrial and Large Commercial Consumers, iii) Call Center for Customer Complaining System, and iv) Enterprise Resource Planning System (ERP). In detail, the main business processes on each category are identified as: i) New Services, ii) Route and Billing Management, iii) Payment, iv) Deposit Management, v) Enforcement, vi) Meter Reading, vii) Work Order Management, viii) Call Center, ix) Communication Billing, x) Gas Production, xi) Human Resource, xii) Maintenance, xiii) Expenditure, and xiv) Project Development. This can be illustrated as shown in Figure 7.19.

The information required by the company in order to support decision making process is provided from the monitoring and analysis of billing transactions. The stakeholder views and requirements are reconciled and reorganized toward the information structure (i.e., DW modeling) as needed by the stakeholders. To provide the analytical and strategic information for management, the data needs to be integrated from the various sources as defined from the main activities. This creates the data heterogeneity problems since the data sources contain synonym and homonym data semantics.

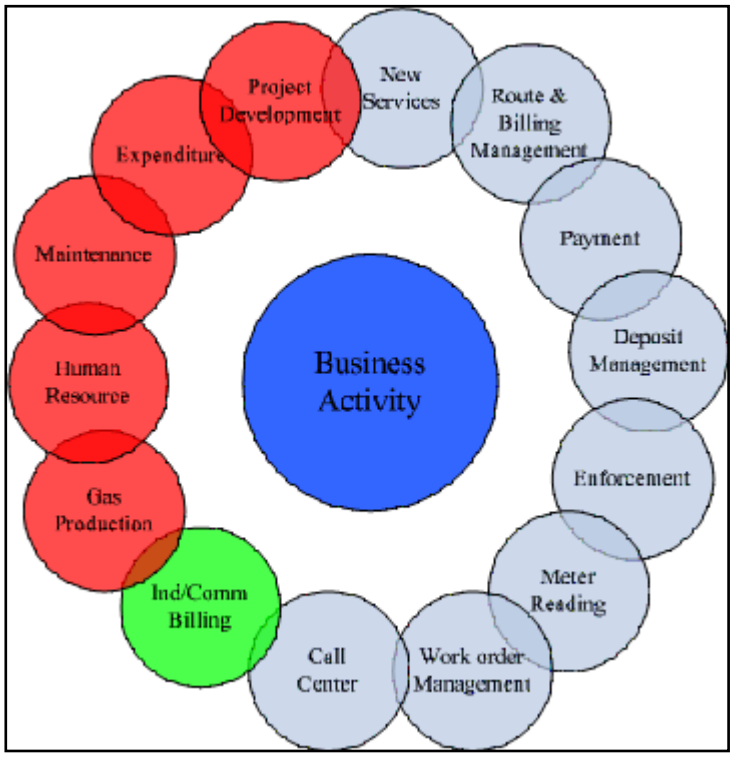


Figure 7.19: Business Activity for Gas Malaysia

The DW system environment in this case study uses the data sources from three different systems: i) Utility Billing Information System (UBIS), ii) JDE System

(Industrial and Large Commercial Data), and iii) Call Center System (Customer Complaining System).

7.5.2.2 Goal-Oriented Requirement Analysis

The requirements of collection and gathering are carried out with the company stakeholders and focus on the billing information needed by the Gas Malaysia (M) Sdn. Bhd. These requirements focus on the billing area, which comprises billing transaction and call center activities. The billing system is implemented by the Utility Billing Information System (UBIS) that handles the residential consumers and supported by the external application JDE System and Call Center System (CCS). These external systems are provided by the various vendors.

The main goal of the company is to be an *Innovative Value for Energy Solutions Provider.* This main goal is supported by four sub-goals that need to be fulfilled for achieving the main goal. To simplify the process in evaluating the RAMEPs, this case study focuses on the *Cost Effective Energy Solution* that is related to the billing domain.Briefly, the scenario of Billing Department that needs the information from the DW system from the stated goal can be described as follows:

> *"The Billing Department depends on the Billing Operator for achieving the goal billing without Error and Billing Operator depends on the Customer for the goal Cost Effective Energy Solutions. Moreover, the Call Center Department depends on the Customer for achieving the goal Controllable Customer Complaints"*.

The elicitation and analysis process applies the goal-oriented approach and the main goals of the Gas Malaysia identified are shown in Figure 7.20.

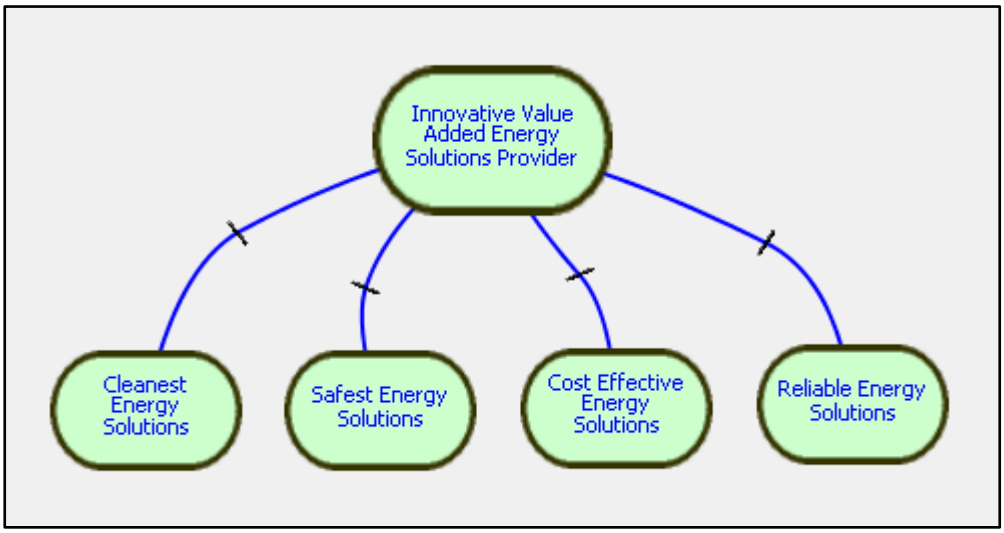


Figure 7.20: Gas Malaysia Main Goals

Based on this scenario, the analysis starts by modeling the requirements in the perspective of organization (i.e., the Billing Department). In the first step, the stakeholders involved in the billing domain were identified and represented by using actor diagram. An Actor diagram explains about dependencies among actors such as billing department, customer, billing operator, and call center department in Gas Malaysia. The actor diagram is shown in Figure 7.21.

The analysis on the billing domain actors producesthe actor diagram that presents the information about actor, sub-actor, type, goals, and dependencies. To fulfill the Billing Department and Gas Malaysia goal, the DW system needs to be supported by UBIS, JDE, and CCS. Although all these systems are not similar, it must support the

information required by the Gas Malaysia stakeholders. In other words, these systems need to be integrated for producing the information as required.

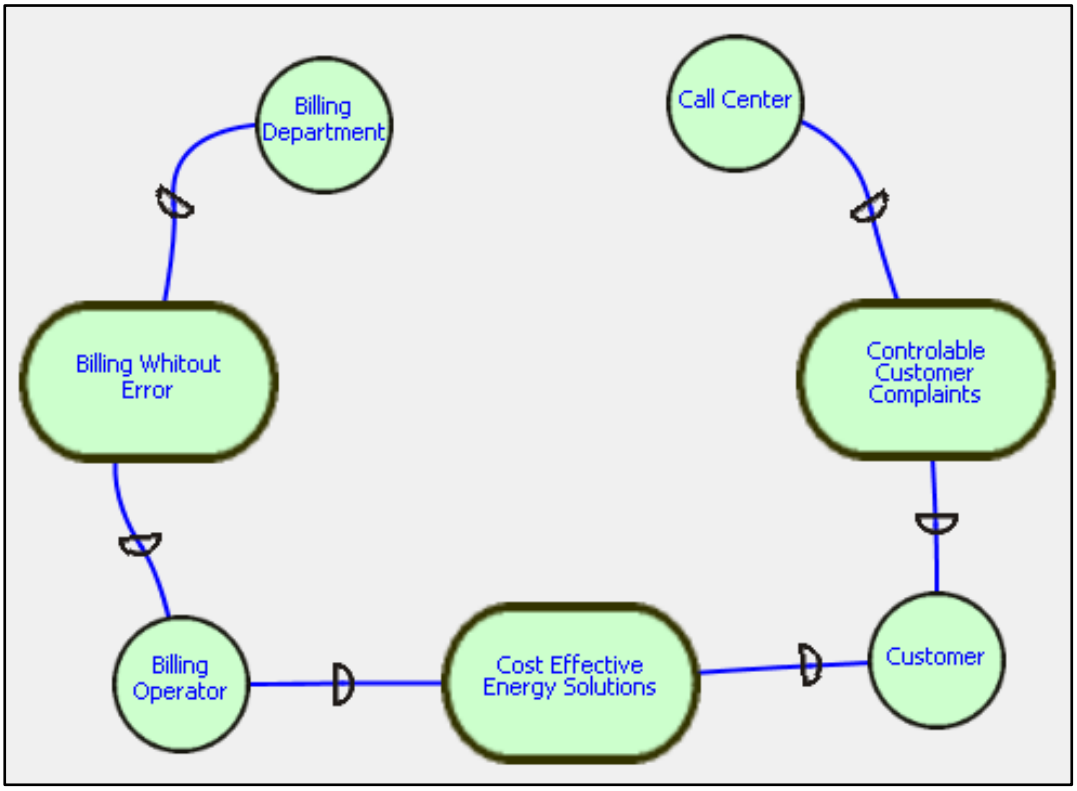


Figure 7.21: The Actor Diagram for Billing domain, Gas Malaysia

The next tasks are to analyze the DW requirements within the perspectives of organization, decision-maker, and developer of Billing Department. All these works are presented in Appendix B, and the final results are discussed in the next Section 7.5.2.3.

7.5.2.3 Results for Goal-Oriented Requirement Analysis

Goal-oriented analysis for billing utility ends after the transformation analysis task is completed. The diagram is focusing on the goal *Sale Volume and Revenue* and

Customer and Billing Status. Both diagrams are representing the final DW requirements' model, which contains facts, dimensions, measures, actions, and business rules. However, as the transformation analysis is carried out, all the DW components are used to design the ETL processes as required by decision goal to be fulfilled. The transformation analysis tasks require a clear understanding of decision makers in order to define suitable transformation activities for the ETL processes. Moreover, these DW components are used to construct an ontology model for supporting the design of the ETL processes. These diagrams are depicted in Figure 7.22 and Figure 7.23 respectively.

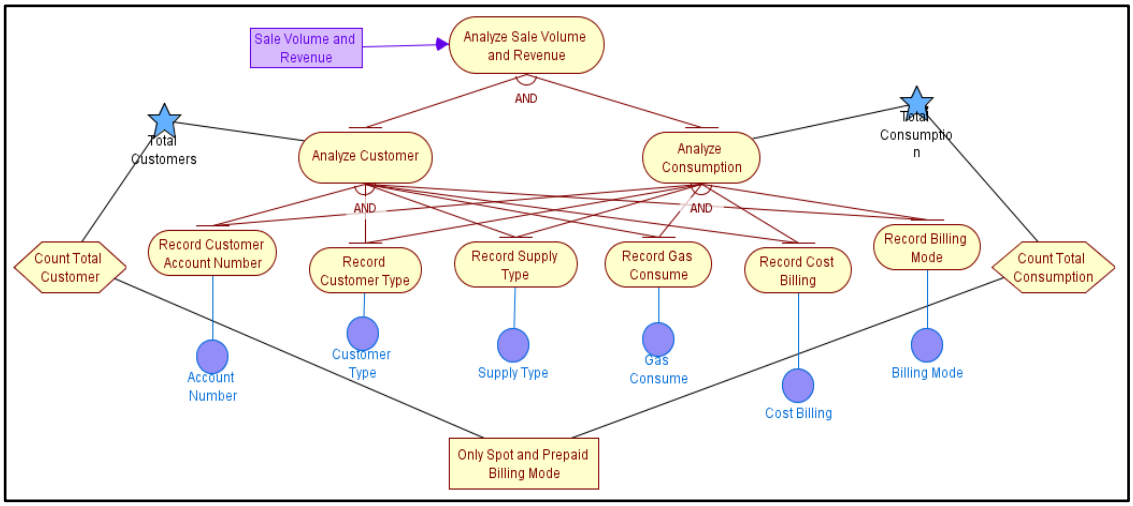


Figure 7.22: Sale Volume and Revenue Goal Diagram

In Figure 7.22, the goals of *Analyze Customer* and *Analyze Consumption* are based on the facts of *Sale Volume and Revenue*. In order to provide information for these goals, appropriate plans are decomposed into two: *Count Total Customer* and *Count Total Consumption* with support for the business rule *only for the spot and prepaid*

billing.The proposed plans are to achieve the goals of *Analyze Customer* and *Analyze Consumption.* Therefore, the proposed DW schemas are as follows:

Fact(Sale Volume and Revenue)

Dimension(Account Number, Customer Type, Supply Type, Gas Consume, Cost Billing, Billing Mode)

Measure(Total Customer, Total Consumption)

Action(COUNT Total Customer, COUNT Total Consumption)

Business Rule(Only Spot and Prepaid Billing Mode)

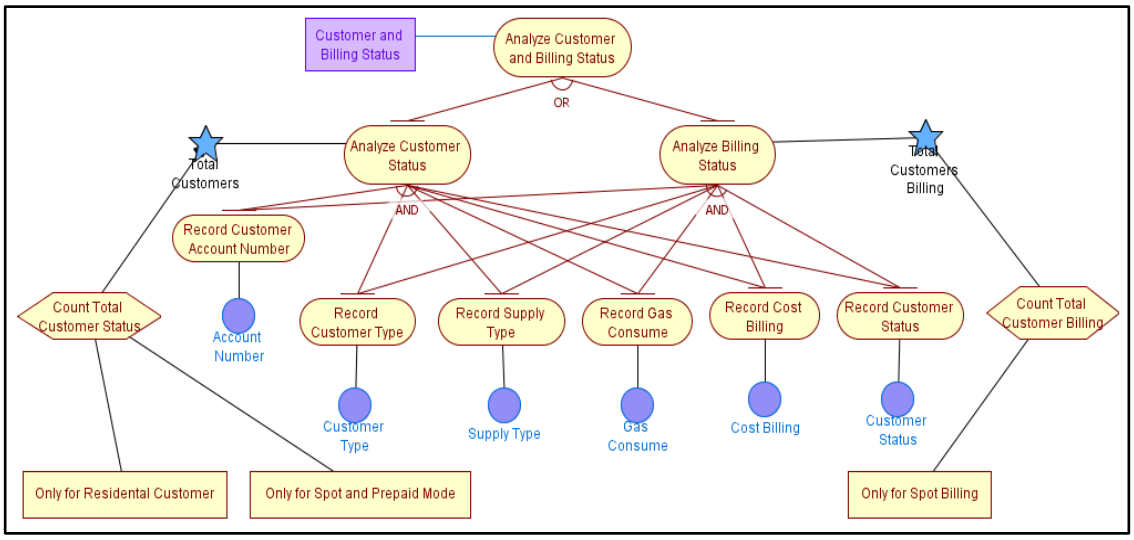


Figure 7.23: Customer and Billing Status Goal Diagram

Figure 7.23 explains the transformation analysis for *Customer and Billing Status*, which proposes the plans for achieving the *Analyzed Billing Status* goal. The plan consists of action *Count Total Customer Billing* with support for the business rule *only for spot billing* and action *Count Total Customer Status* supported by the

business rules *only for residential customer* and *only for the spot and prepaid billing*. Finally, the extended rationale diagram for BM is completed when each of the decision-goal contained plans support the information required by decision makers. Therefore, the proposed DW schemas are as follows:

Fact(Customer and Billing Status)

Dimension(Account Number, Customer Type, Supply Type, Gas Consume, Cost Billing, Customer Status)
Measure(Total Customer, Total Customer Billing)

Action(COUNT Total Customer Status, COUNT Total Customer Billing)

Business Rule(Only for Residential Customer, Only Spot and Prepaid Billing Mode, Only for Spot Billing)

The results for these analyses are used in constructing the ontology model for DW requirements. This is presented in the next Section 7.5.2.4.

7.5.2.4 Results for Ontology Modeling

The design of the ETL processes has been conceptualized by the DW components produced from the goal-oriented requirement analysis. The DW components used to construct an ontology structure are based on the ontology model O = (F, D, M, Br, Ac). Details for this task are presented in the Appendix B. In DWRO, four classes of measure have been identified as *Total Customer*, *Total Consumption*, *Total Customer Status*, and *Total Customer Billing.* Each of the classes contains properties such as *account number*, *customer type*, *supply type*, *gas consumed*, *cost billing*,

billing mode, and *customer status*. The relationship between classes and properties are defined as *hasMeasureTotalCustomer,hasMeasureSumConsumption, hasActionCountCustomer, hasActionSumConsumption,* and so on. Additionally, the axioms are described based on business rules such as *Only Spot and Prepaid Billing*, and actions (e.g., aggregation – SUM for usage of gas in volume).

The ontology model for data sources are constructed based on two different data sources UBIS and JDE. Both databases handle the billing transaction for gas consumption of residential and industrial consumers respectively. These data sources are implemented in different system that is dissimilar in their data structures and semantics. This scenario creates the heterogeneity problems during the integration and transformation of the data sources in the ETL processes. Therefore, the integration of both data sources based on ontology structure clarifies the semantic heterogeneity on the concepts or classes of data sources. The data integration and transformation are done through a proper mapping process.

The mapping and matching process involve the identification of similarity and dissimilarity of concepts and associate attributes of DWRO and DSO. These elements are represented in the ontology as follows:

- Concept is represented byclasses such as Sale Volume and Revenue, Customer and Billing Status.
- Relationship is represented byproperties such ashasMeasureTotalCustomer, hasDimensionCustomerType, hasActionCountCustomer.

- Specific element in DW setting is represented by new Classes such as SUM, COUNT.
- Restriction is represented byaxioms such as “Only for Spot and Prepaid Billing”.

Based on the mapping definition as described in Table 6.6, Chapter 6.5.2, the ontology mapping between DWRO and DSO is shown in Table 7.6.

Table 7.6: DWRO and DSO mapping for Sale Volume and Revenue

DWRO	**DSO**	**The Mapping**
Fact (Sale Volume and Revenue)	UBIS, JDE	Concept: Sale Volume, Sale Revenue
Dimension (account number, customer type, supply type, gas consume, cost billing, billing mode)	Concept: Mode Billing (tbillmode, -) Concept: Customer Type (tbConsType, CommType) Concept: Customer Profile (tbConsumer, Customer) Concept: Supply Type (tbSuppType, SupplyType) Concept: Billing Transaction (tbOpItems, Billing)	Billing Mode ↔ Concept: Mode Billing Customer Type ↔ Concept: Customer Type Customer, Account number * ↔ Concept: Customer Profile Supply Type ↔ Concept: Supply Type Cost Billing ↔ Concept: Billing Transaction *- Two dimensions were mapped to one concept
Measure (Total Customer, Total Consumption)	- Concept: Customer Profile (tbConsumer, Customer) - Concept: Billing Transaction (tbOpItems, Billing)	[Total Customer] ↔ [Customer Profile (COUNT All Records)] [Sum Consumption] ↔ [Billing Transaction (SUM (tbOpItems.Cons, Billing.Cons))]
Business Rule (Categorized by Gas Supply and Customer Type, Only for Spot and Prepaid Billing Mode)	Concept: Supply Type (tbSuppType, SupplyType) Concept: Customer Type (tbConsType, CommType) Concept: Mode Billing (tbillmode, -)	[Categorized by Gas Supply] ↔ [Concept: Supply Type (tbSuppType, SupplyType)] [Categorized by Customer Type] ↔ [Concept: Customer Type (tbConsType, CommType)] [Only for Spot and Prepaid ↔ [Billing Concept: Mode

Action (MERGE UBIS and JDE, FILTER for Spot and Prepaid Billing, COUNT Total Customer, SUM Total Gas Consumption)	Concept: Supply Type (tbSuppType, SupplyType) Concept: Customer Type (tbConsType, CommType) Concept: Mode Billing (tbillmode, -)	Billing (tbillmode, -)] • [MERGE for UBIS and JDE] ↔ [Customer Type (tbConsType, CommType), Customer Profile (tbConsumer, Customer), Supply Type (tbSuppType, SupplyType), Billing Transaction (tbOpItems, Billing) • [FILTER Spot and Prepaid Billing Only] ↔ [Billing Mode (tbillmode = "PP" and "SB")] • [COUNT Total Customer ↔ [Recno (Customer)] • [SUM Total Gas Consumption ↔ SUM (Billing Transaction.Cons)]

Table 7.6 presents the mapping specifications of DWRO and DSO that are derived from the analysis process of user requirements supported by the related data sources. To complete the entire cycle of ETL processes, the actions for extract and loading are shown in Table 7.7.

Table 7.7: The Extract and Loading for Sale Volume and Revenue

DWRO	**DSO**	**The mapping**
Action (RETRIEVE for account number, customer type, supply type, gas consume, cost billing, billing mode)	Concept: Mode Billing (tbillmode, -) Concept: Customer Type (tbConsType, CommType) Concept: Customer Profile (tbConsumer, Customer) Concept: Supply Type (tbSuppType, SupplyType) Concept: Billing Transaction (tbOpItems, Billing)	[RETRIEVE Billing Mode] ↔ (tbillmode) [RETRIEVE for Customer Type] ↔ (tbConsType, CommType) [RETRIEVE for Customer Profile] ↔ (tbConsumer, Customer) [RETRIEVE for Supply Type] ↔ (tbSuppType, SupplyType) [RETRIEVE for Billing Transaction ↔ (tbOpItems, Billing)

Action (LOADING for Fact)	Concept: Mode Billing (tbillmode, -) Concept: Customer Type (tbConsType, CommType) Concept: Customer Profile (tbConsumer, Customer) Concept: Supply Type (tbSuppType, SupplyType) Concept: Billing Transaction (tbOpItems, Billing)	[LOADING Mode Billing] ↔ DW_ModeBilling [LOADING Customer Type] ↔ DW_CustomerType [LOADING Customer Profile] ↔ DW_Customer [LOADING Supply Type] ↔ DW_SupplyType [LOADING Billing Transaction] ↔ DW_TotalCustomer, DW_TotalCustomer

Based on the mapping results, new classes and properties pertaining to the merging ontology (i.e., DWRO and DSO) are produced. These new classes and properties are shown in Table 7.8.

Table 7.8:New classes and Properties for Sale Volume and Revenue

Classes	**Type of Classes**
Total Customer	Aggregated class type
Total Consumption	Aggregated class type
Categorized by Gas Supply and Customer Type	Ranged class type
RETRIEVE	Table class type
MERGE	Merging class type
FILTER	Range class type
COUNT	Aggregation class type
LOADING	Table class type

These new classes are reorganized properly into the DWRO after merging through Protégé-OWL. The ontology merging is done through the ontology setting as defined in Table 7.9.

Table 7.9: Setting for Ontology Merging for Sale Volume and Revenue

Mapping List	Ontology Setting
Merge UBIS, JDE	Classes Billing Mode : tbillmode Customer Type : tbConsType ∪ CommType Customer Profile : tbConsumer ∪ Customer Supply Type : tbSuppType ∪ SupplyType Billing Transaction : tbOpItems ∪ Billing MergeSources: hasMergeCustomer *some* CustomerProfile, hasMergeSupply *some* SupplyType … Properties hasMergeCustomer(Domain:CustomerProfile, Range:tbConsumer, Customer) hasMergeSupply(Domain:SupplyType, Range:tbSuppType, SupplyType) …
FILTER Customer for "Only for residential customer"	∃hasMeasureTotalCustomer ← Total Customer hasMeasureTotalCustomer *some* Total Customer ∃hasMeasureTotalConsumption ← Total Consumption hasMeasureTotalConsumption *some* Total Consumption
AGGREGATE (COUNT) for Total Customer	∀hasMeasureTotalCustomer ← Total Customer hasMeasureTotalCustomer *only* Total Customer
AGGREGATE (COUNT) for Total Consumption	∀hasMeasureTotalConsumption ← Total_Consumption hasMeasureTotalConsumption *only* Total_Consumption

This process ends when the ontology structure is reconstructed and rechecked by using **Pallet** reasoner. The new appearance of MRO with new classes that represent the ETL processes specifications is shown in Figure 7.24.

7.5.2.5 Results for Generating the ETL Processes Specifications

The ETL processes specifications are produced from the MRO, which is the knowledge representation of ETL processes operations of Utility Billing. This task is realized by manipulating the semantic annotation of DW requirements and data sources in MRO. The manipulation process proposesthe sets of ETL processes specifications that transform the data sources to the DW schemas as presented in the

previous results. The ETL processes specifications comprise the process of extract, transform, and loading that are shown in Table 7.10.

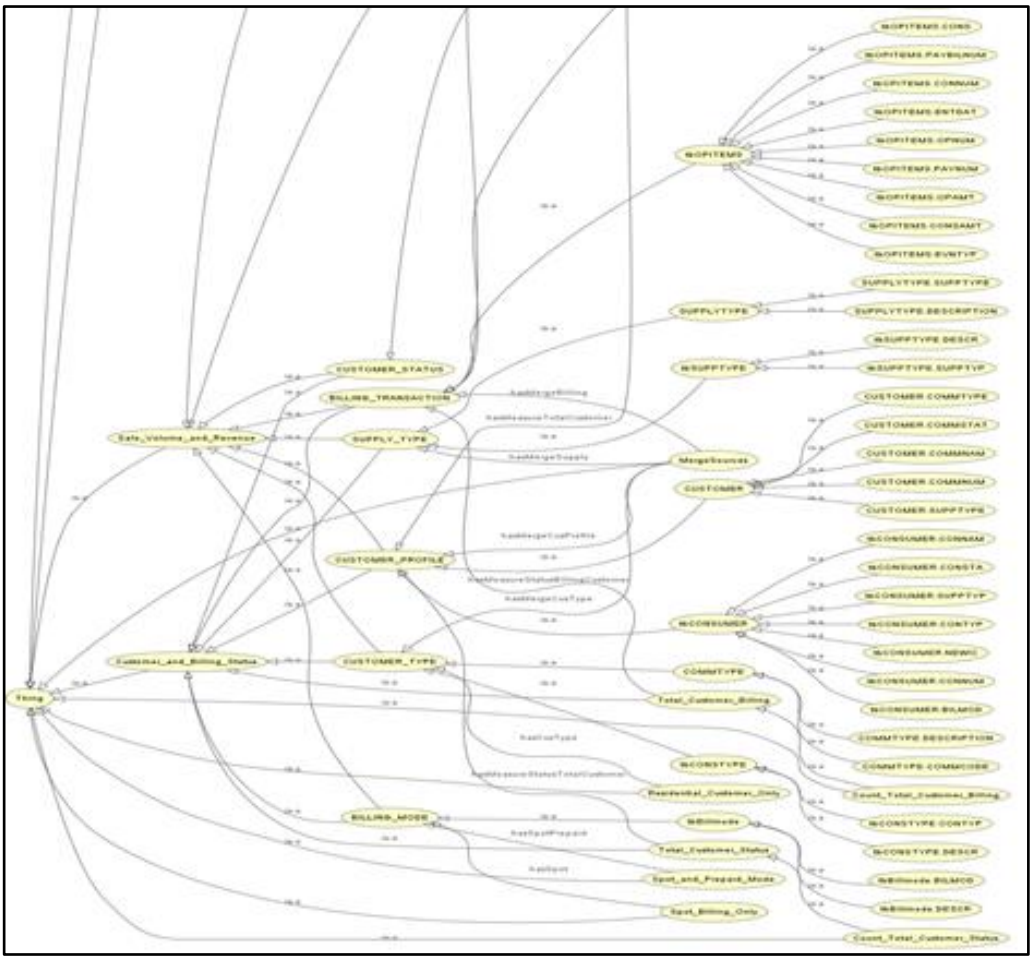


Figure 7.24: MRO for Gas Malaysia

Table 7.10: The ETL Processes for Gas Malaysia Utility Billing

ETL Processes	**Actions**
EXTRACT()	Extract the data from the data sources UBIS and JDE
MERGE()	Merge data set of UBIS and JDE
FILTER()	Filters the merged data set according to specific conditions.
CONVERT()	Convert set of data sources to another format or type
AGGREGATE()	Count for Total Customer, Sum for Total Consumption, Count Total Customer Billing, Count Total Customer Status
LOADER()	Loads data into the DW

As stated in MRO, the knowledge about information as required and their related data sources are defined according to RDF/OWL based language. Thus, the MRO is processed according to an appropriate reasoning in order to identify and propose a set of ETL processes specifications. The MRO contains set of RDF/OWL triples, which identify the nodes/classes and arcs/properties. Ontology reasoning is used on classes and their related properties to derive the ETL processes specifications according to the generic ETL processes tasks as shown in Table 7.10.

Based on the proposed algorithm as defined in Figure 6.9, Section 6.5.2, ETL processes specifications are derived automatically. A snippet of MRO is shown in Figure 7.25.

```
<!-- http://www.semanticweb.org/ontologies/2010/7/gasmalaysia_MRO.owl#hasSpot -->
<owl:ObjectProperty rdf:about="#hasSpot">
<rdfs:range rdf:resource="&gasmalaysia_datasources;BILLING_MODE"/>
<rdfs:domain rdf:resource="&gasmalaysia_requirements;Customer_and_Billing_Status"/>
</owl:ObjectProperty>
<!--http://www.semanticweb.org/ontologies/2010/7/gasmalaysia_MRO.owl#hasSpotPrepaid -->
<owl:ObjectProperty rdf:about="#hasSpotPrepaid">
<rdfs:range rdf:resource="&gasmalaysia_datasources;BILLING_MODE"/>
<rdfs:domain rdf:resource="&gasmalaysia_requirements;Customer_and_Billing_Status"/>
<rdfs:domain rdf:resource="&gasmalaysia_requirements;Sale_Volume_and_Revenue"/>
</owl:ObjectProperty>
<!-- http://www.semanticweb.org/ontologies/2010/6/gasmalaysia_datasources.owl#BILLING --
>
<owl:Class rdf:about="&gasmalaysia_datasources;BILLING">
<rdfs:subClassOf rdf:resource="&gasmalaysia_datasources;BILLING_TRANSACTION"/>
</owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/6/gasmalaysia_datasources.owl#BILLING.COM
MNUM -->
<owl:Class rdf:about="&gasmalaysia_datasources;BILLING.COMMNUM">
<rdfs:subClassOf rdf:resource="&gasmalaysia_datasources;BILLING"/>
</owl:Class>
<!--
```

Figure 7.25: A snippet of MRO of Gas Malaysia

To generate the ETL processes specifications, a prototype application for reading, and manipulating MRO was developed. The MRO is manipulated and generate the ETL processes specifications as shown in Figure 7.26.

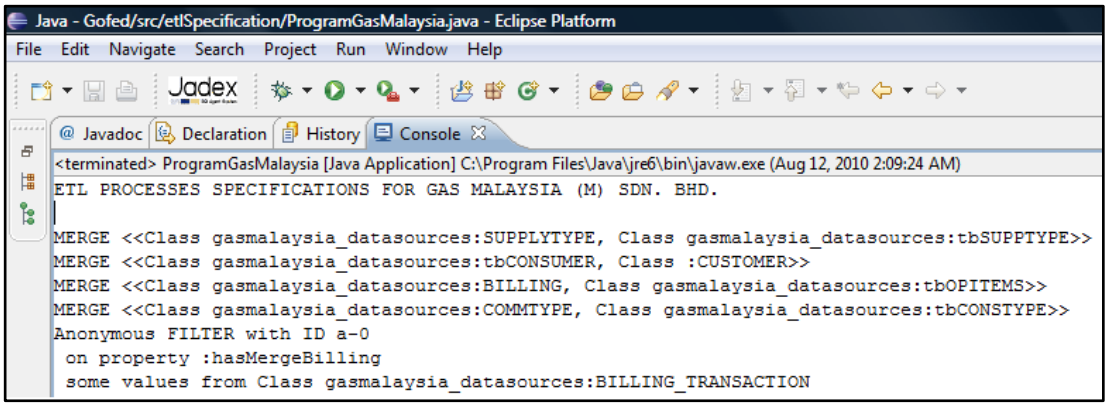


Figure 7.26: List of ETL Processes Specifications for Gas Malaysia

The results show that the ETL processes specifications for Utility Billing can be derived from the ontology model of the DW requirements. The ETL processes design was systematically carried out using RAMEPs. Although the DW environment is more complex than case study 1, nevertheless the RAMEPs were successfully applied. Furthermore, the complexity of ETL processes causes the entire process design to produce a large set of diagrams.

7.5.3 Case Study 3 – Student Entrepreneur in the MoHE Domain

Entrepreneurship is one of the main programs in the Ministry of Higher Education (MoHE) to support the transformation of the economy within the framework of a new economic model. This program is supported by the Entrepreneur Development Policy for Institutions of Higher Education (IHE) that was publicized in early 2010. The objective of this policy is to promote and strengthen the entrepreneur program in

an organized and holistic way among the IHEs. The entrepreneur program aims to produce high quality students who possess intellectual, attributes, and entrepreneur values. Currently, the entrepreneur program for student development is a core task in IHE, and has become a national agenda (MoHE, 2010).

To execute the entrepreneur development policy, MoHE has established a department for organizing and monitoring the entrepreneur program in IHE. The MoHE needs to organize the resources and information about an entrepreneur through IHE representative known as MAKMUM (Majlis Keusahawan Universiti-Universiti Malaysia). The MAKMUM and IHE collect the entrepreneur data and provide the information to the MoHE. However, the information about entrepreneur cannot be easily collected and analyzed due to the fact that the data is located in all IHEs. Moreover, most of the data is unavailable in the computer systems and if it does exist, the data is presented in various formats and structures.

The challenges to develop the entrepreneur DW system for MoHE are in integrating and consolidating the heterogeneous entrepreneur data sources from 20 IHEs. The data integration must produce complete and accurate data for analyzing the required information.

7.5.3.1 DW Systems Environment

The business activities for the Entrepreneur Unit (EU) are to plan and monitor the entrepreneur development program in IHEs. The planned activities involve the budget and roadmap of IHEs' entrepreneur program. Meanwhile, the monitoring

activities focus on the program implementation and assessment. The position of the EU in MoHE'sorganizational hierarchy is illustrated in Figure 7.27.

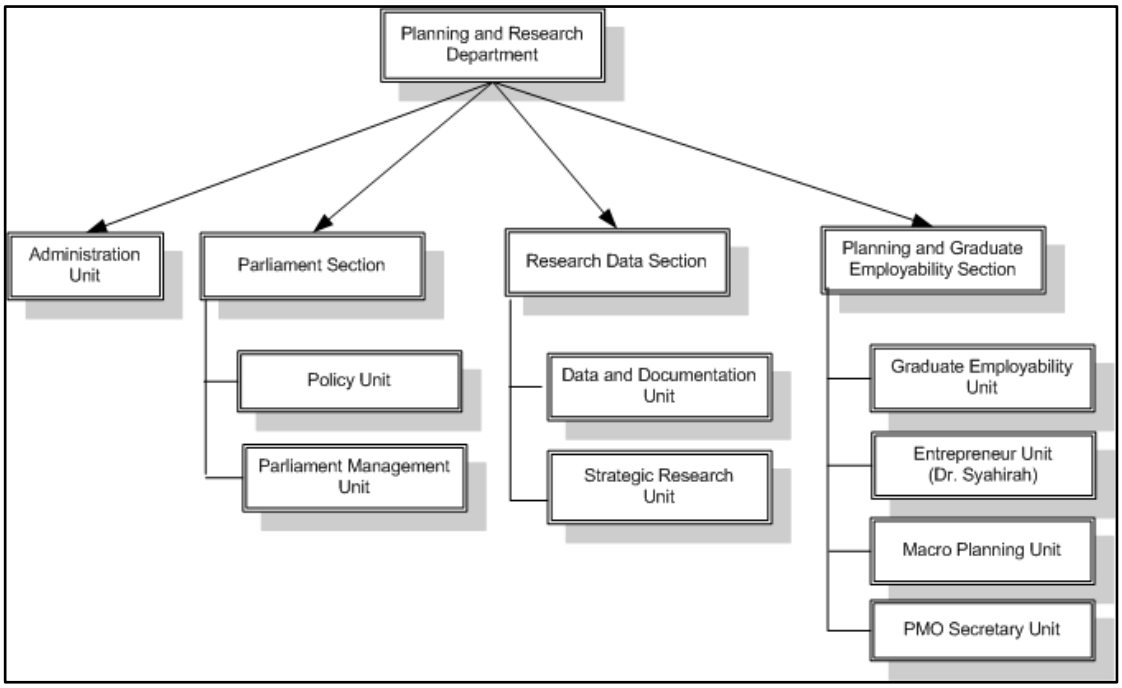


Figure 7.27: Entrepreneur Unit of MoHE

The EU is positioned within the Plan and Marketability of Graduates under the Planning and Research Department. Headed by one director, the EU manages the analyzed information about IHEs' entrepreneurs for the minister.

7.5.3.2 Scope of the Study

The roles of EU are to ensure the entrepreneur program in IHEs progress accordingly for the benefit of students, IHEs and the government. Several objectives need to be achieved in order to ensure the entrepreneur policy is implemented. Therefore, this case study focuses on the objectives that are relevant for DW. The interview with the EU and two IHEs personnel was conducted to gather and analyze the requirements

for an entrepreneur DW. These two IHEs were Universiti Utara Malaysia (UUM) represented by the Co-operative and Entrepreneurship Development Institute (CEDI) and Universiti Malaysia Perlis (UniMAP) represented by their EU.

The goals of the EU of MoHE were identified and explored based on the data required by the entrepreneur DW for supporting the EU's main goal. The interview with UUM and UniMAP was carried out for defining the entrepreneur data as required by the EU of MoHE. The information about an entrepreneur in IHEs and the goals of EU were analyzed, reorganized and presented in the actor diagram as defined by RAMEPs. The RAMEPs process for this case study is explained in the next Section 7.5.3.3.

7.5.3.3 Goal-Oriented Requirement Analysis

The elicitation and analysis process using the goal-oriented approach emphasizes achieving of goals, performing related tasks and furnishing the resources as required. Thus, the main goals of EU as shown in Figure 7.27 need to be analyzed systematically. In organization modeling, DW requirements of the EU are determined by exploring the EU goals, sub-goals, and related stakeholders involved in utilizing the entrepreneur information. The goals and sub-goals are derived from the EU mission and vision statements of entrepreneur development program policy that drives the synergized of EU, MAKMUM and IHEs to produce successful student entrepreneurs in future. Based on the results of the interview, the EU goals were analyzed and identified in order to ensure the entrepreneur DW requirements

are gathered according to goals as specified. The goals and sub-goals of the EU are presented in Figure 7.28.

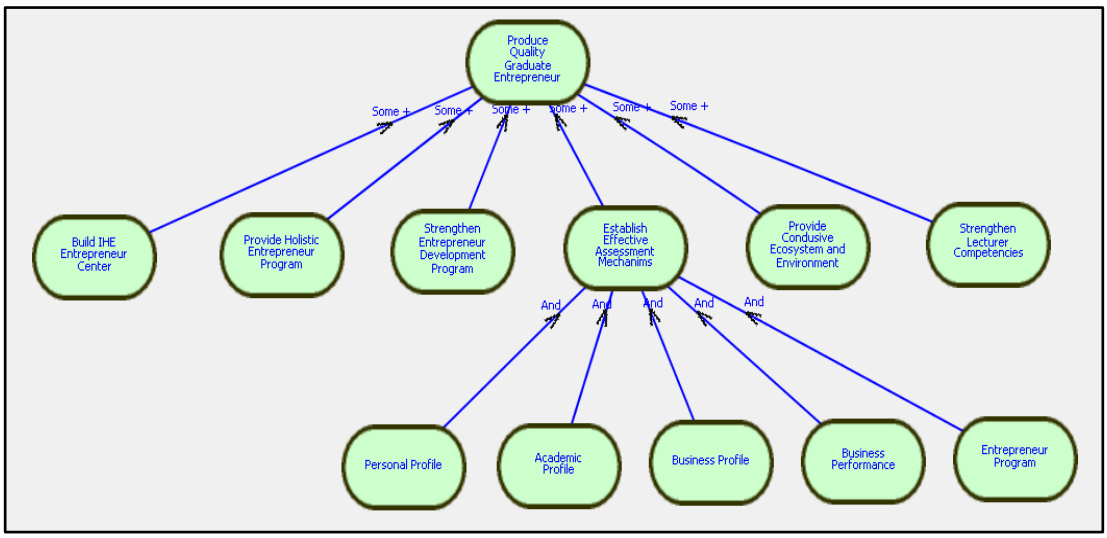


Figure 7.28: EU Goals and Sub-Goals

In Figure 7.27, the main goal is to *Produce Quality Graduate Entrepreneurs*, and six sub-goals. These sub-goals have some contribution to the fulfillment of the main goal, which are identified as *Build Entrepreneur Center*, *Provide Holistic Entrepreneur Program*, *Strengthen Entrepreneur Development Program*, *Establish Effective Assessment Mechanism*, *Provide Conclusive Ecosystem and Environment*, and *Strengthen Lecturer Competencies*. Based on the case study setting, the research is focused on the *Establish Effective Assessment Mechanism* sub-goal that is most related to the entrepreneur DW system. This sub-goal is supported by the entrepreneur information such as *Personal Profile*, *Academic Profile*, *Business Profile*, *Business Performance*, and *Entrepreneur Program.*

The scenario of EU that needs the information from the DW, which supports the goals that can be described as follows:

> *"The Head of EU depends on the IHEs for providing the entrepreneur reports to the minister and depends on MAKMUM for organizing the entrepreneur information. The IHEs depends on the entrepreneur for collecting the entrepreneur data and depends on the MAKMUM for organizing the entrepreneur information"*

Thus, the next task of requirement analysis is focused on *Establish Effective Assessment Mechanism* sub-goal. The stakeholders involved in EU were identified as Head of EU, IHEs, MAKMUM and Entrepreneur. The Actor diagram explains about dependencies among actors in EU DW system, which is produced from the requirements' documentation organized in three different templates namely: main actor (actor, objectives), sub-actor (sub-actor, type, goals), and dependencies (depender, dependee, goal). The actor diagram is shown in Figure 7.29.

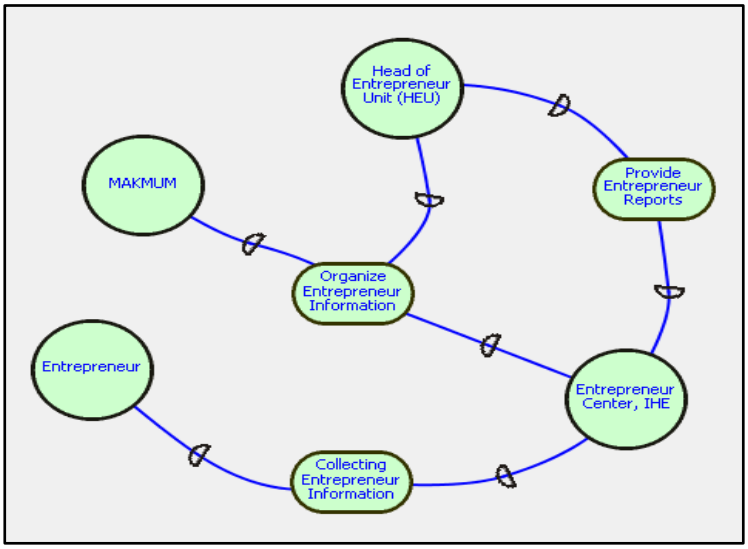


Figure 7.29: The Actor Diagram for EU of MoHE

The next tasks were to analyze the EU DW requirements in the perspectives of organization, decision-maker, and developer. All these analyses were conducted sequentially, and presented in Appendix C. Section 7.5.3.4 discusses the final results in goal-oriented and ontology modeling, and importantly the ETL processes specifications generation.

7.5.3.4 Results for Goal-Oriented Requirement Analysis

Based on the EU diagram, the final diagram for DW requirements, which are defining the information for *Total Entrepreneur* and *Sum of Funding*, *Total Company* and *Sum of Initial Capital*, and *Total Program* and *Total Budget* are presented in Figure 7.30, Figure 7.31, and Figure 7.32 respectively. In summary, the DW requirements analysis proposed the DW schemas as presented in Table 7.11.

Table 7.11: The glossaries of DW requirements

Fact	Dimension	Measure	Business Rule	Action
Entrepreneur Profile	Personal, Status, Gender, Race, University, State	Total of Entrepreneur, Sum of Funding	Only for UUM and UniMAP	Count Total Entrepreneur, Sum Total Funding
Business Profile	Business Info, Type, Category, Performance	Total Company, Sum of Initial Capital	Only for UUM and UniMAP	Count Total Company, Sum for Initial Capital
Entrepreneur Program	Program Training, Budget Program	Total Program, Total Budget	Only for UUM and UniMAP	Count Total Program, Count Total Budget

Based on the DW schemas, the next task is to develop the ontology model for the DW student entrepreneur system. The ontology model is used to generate the ETL processes specifications.

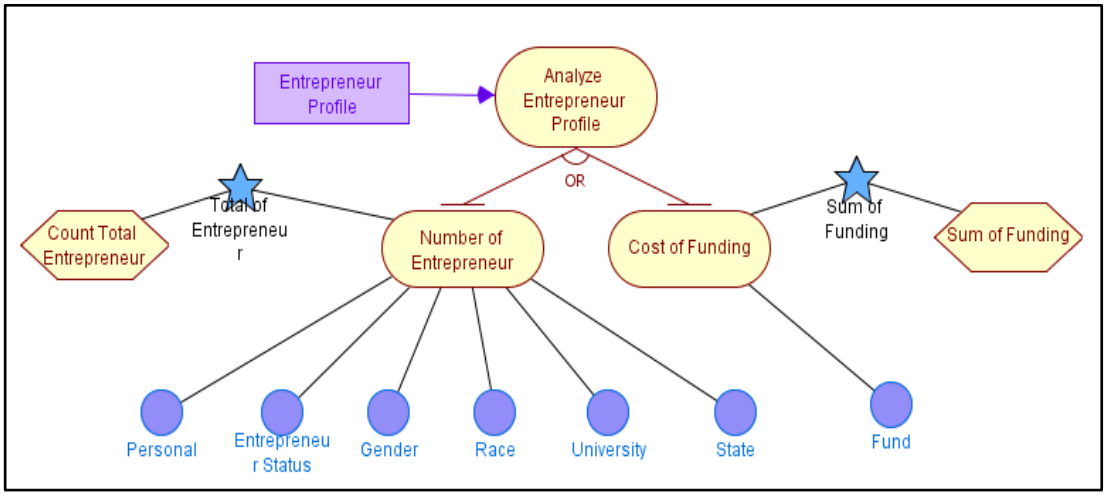


Figure 7.30: Goal Diagram for Entrepreneur Profile

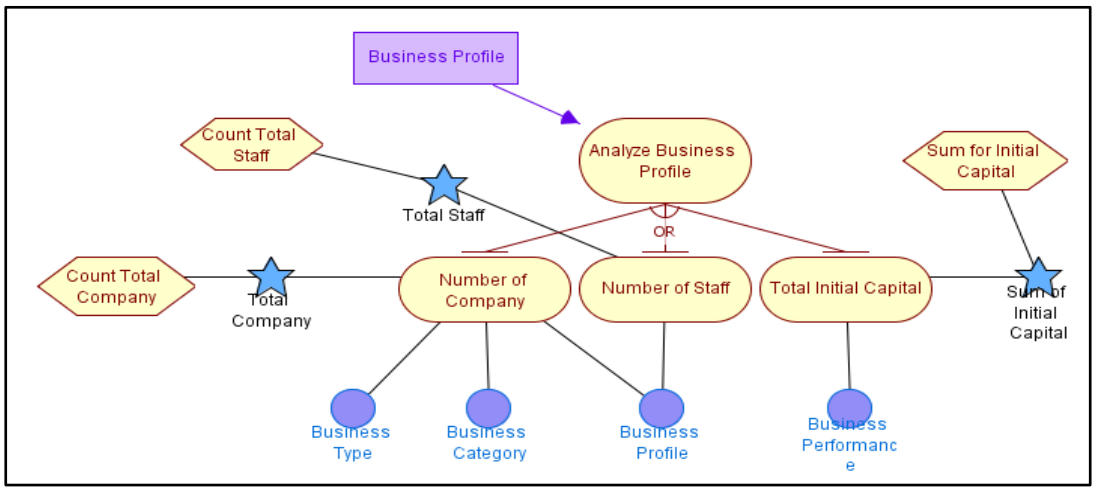


Figure 7.31: Goal Diagram for Business Profile

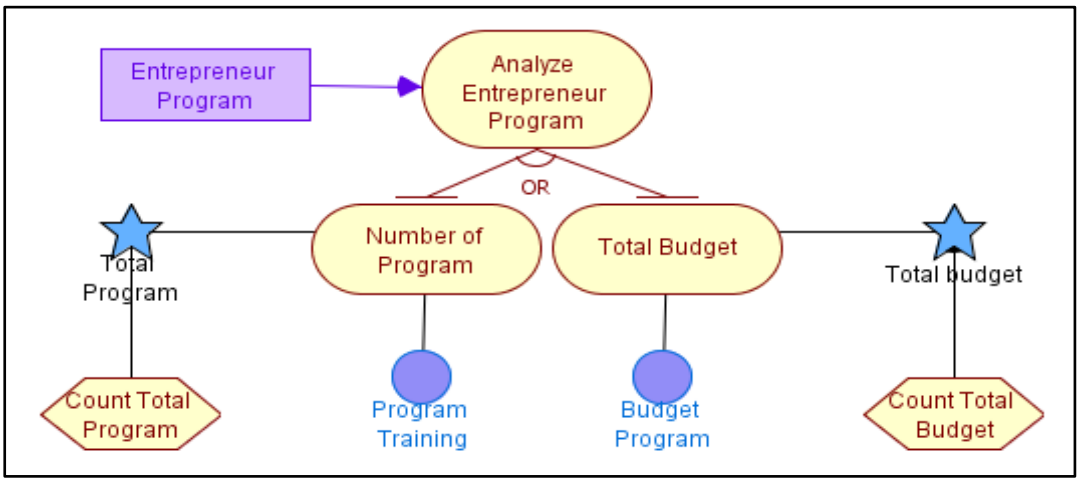


Figure 7.32: Goal Diagram for Entrepreneur Program

7.5.3.5 Results for Ontology Modeling

The DW requirements for Student Entrepreneur were constructed into the ontology based on the model O = (F, D, M, Br, Ac). Generally, six classes of measure were identified as *Total of Entrepreneur, Sum of Funding, Total Company, Sum of Initial Capital, Total Program,* and *Total Budget*. Then, each of the classes contained properties such as *Personal, Entrepreneur Status, Gender, Race, University, State, Business Info, Business Type, Business Category,* and *Business Performance.* These properties represented the dimension or measure components of the DW. The axiom thatdetermines the relationship between classes and properties is defined as *hasMeasureTotalEntrepreneur,hasActionFunding,hasMeasureTotalCompany, hasMeasureSumFunding, hasActionInitialCapital* and others. All these definitions are translated into the ontology model as presented in Appendix C.

The data sources are accessed from public IHLs that are implemented in different systems and platforms, which have different data structures and semantics. This case study utilizes entrepreneur data sources from UUM and UniMAP. The heterogeneity problems in this case study occurred during the analysis of both data structure for forming the entrepreneur data sources. After studying the data sources' schemas, the integration of both data sources should provide a single structure. Consequently, the data sources were modeled into ontology structure, which clarified the concepts or classes of the data sources. Moreover, by using ontology, the semantics difference or similarity is defined properly according to the DW requirements and ETL processes definitions. This is shown clearly in Appendix C.

The mapping process involves the identification of similarity and dissimilarity of concepts and associate attributes of DWRO toward the DSO. These elements are represented in the Table 7.12.

Table 7.12: The Ontology Elements

Elements	**Descriptions**
Concept or Classes	Entrepreneur Profile, Business Profile, Entrepreneur Program, Personal, Entrepreneur Status, Gender, Race, University, State, Total of Entrepreneur, Sum of Funding, Count Total Entrepreneur, Sum Total Funding, …. etc.
Relationship or Properties	hasMeasureTotalEntrepreneur, hasMeasureTotalCompany, hasMeasureSumFunding, hasActionFunding, hasActionInitialCapital
Restriction or Axioms	Only for UUM and UniMAP

Based on the mapping definition, the ontology mapping between DWRO and DSO is shown in Table 7.13. The mapping represents the semantics of user requirements as defined in the DWRO.

Table 7.13: DWRO and DSO mapping for Entrepreneur Profile

DWRO	**DSO**	**The Mapping**
Fact (Entrepreneur Profile)	Entrepreneur data sources	Concept: Entrepreneur profile
Dimension (Personal, Entrepreneur Status, Race, University, State, Fund)	Concept: Personal (tbPersonal) Concept: Entrepreneur Status (tbGStatus) Concept: Race (tbRace) Concept: University (tbUniversity) Concept: State (tbState) Concept: Fund (tbFund)	Personal ↔ Concept: Personal Entrepreneur Status ↔ Concept: Entrepreneur Status Race ↔ Concept: Race University ↔ Concept: University State ↔ Concept: State Fund ↔ Concept: Fund
Measure (Total of Entrepreneur, Sum of Funding)	Concept: Personal (tbPersonal) Concept: Fund (tbFund)	[Total of Entrepreneur] ↔ [Personal (COUNT All Records)]

		[Sum of Funding] ↔ [Fund (SUM (tbFund.FundAmt)]
Business Rule (Only for UUM and UniMAP)	Concept: University (tbUniversity)	[Only for UUM and UniMAP] ↔ [Concept: University (tbUniversity)]
Action (FILTER for UUM and UniMAP, COUNT Total Entrepreneur, SUM Total Funding)	Concept: University (tbUniversity) Concept: Personal (tbPersonal) Concept: Fund (tbFund)	• [FILTER for UUM and UniMAP] ↔ [University (idUniversity = "UUM" and "UniMAP")] • [COUNT Total Entrepreneur ↔ [Recno (Personal)] • [SUM Total Funding ↔ SUM (Fund.FundAmt)]

Table 7.13 presents the mapping specifications of DWRO and DSO derived from the analysis process of user requirements and supported by the related data sources. However, the actions for extract and load need to be included for completing the entire cycle of the ETL processes. This is shown in Table 7.14.

Table 7.14: The Actions for Extract and Loading for Entrepreneur Profile

DWRO	**DSO**	**The mapping**
Action (RETRIEVE for Personal, Entrepreneur Status, Race, University, State, Fund)	Concept: Personal (tbPersonal) Concept: Entrepreneur Status (tbGStatus) Concept: Race (tbRace) Concept: University (tbUniversity) Concept: State (tbState) Concept: Fund (tbFund)	[RETRIEVE Personal] ↔ (tbPersonal) [RETRIEVE Entrepreneur Status] ↔ (tbGStatus) [RETRIEVE Race] ↔ (tbRace) [RETRIEVE University] ↔ (tbUniversity) [RETRIEVE State ↔ (tbState) [RETRIEVE Fund ↔ (tbFund)
Action (LOADING for Entrepreneur Profile Fact)	Concept: Personal (tbPersonal) Concept: Entrepreneur Status (tbGStatus) Concept: Race (tbRace) Concept: University (tbUniversity) Concept: State (tbState) Concept: Fund (tbFund)	[LOADING Personal] ↔ DW_Personal [LOADING Entrepreneur Status] ↔ DW_EntrepreneurStatus [LOADING Race] ↔ DW_Race [LOADING University] ↔ DW_University [LOADING State]

	↔DW_State [LOADING Total Entrepreneur ↔ DW_TotalEntrepreneur [LOADING Sum Funding ↔ DW_SumFunding

Based on the mapping results, new classes and properties pertaining to the merging ontology (i.e., DWRO and DSO) were produced. These new classes and properties are shown in Table 7.15.

Table 7.15: New Classes and Properties Student Entrepreneur

Classes	**Type of Classes**
Total of Entrepreneur	Aggregated class type
Sum of Funding	Aggregated class type
UUM and UniMAP entrepreneur only	Ranged class type
RETRIEVE	Table class type
FILTER	Range class type
COUNT	Aggregation class type
LOADING	Table class type

These new classes are reorganized properly into the MRO after merging through Protégé-OWL. The merging process is done through the ontology setting as defined in Table 7.16.

Table 7.16: Ontology Setting for MRO of Entrepreneur Profile

Mapping List	**Ontology Setting**
FILTER University for "UUM" and "UniMAP"	∃hasUniversity ← Total_Entrepreneur, Sum_Funding hasUniversity *only* Total_Entrepreneur hasUniversity *only* Sum_Funding
AGGREGATE (COUNT) for Total Entrepreneur	∀hasMeasureTotalEntrepreneur ← Total_Entrepreneur

	hasMeasureTotalEntrepreneur *only* Total_Entrepreneur
AGGREGATE (COUNT) for Sum Funding	∀hasMeasureSumFunding ← Sum_Funding hasMeasureSumFunding *only* Sum_Funding

This process ends when the ontology structure is reconstructed and rechecked by using **Pallet** reasoner. The new appearance of Student Entrepreneur MRO is shown in Figure 7.33. Each node/class and arc/property is shown with labels, which explains the relationship between nodes/class. The ETL processes specifications are produced from the MRO, which is the knowledge representation of DW requirements and ETL operations of Student Entrepreneur DW.

7.5.3.6 Results for Generating the ETL Processes Specifications

The ETL processes specifications comprising the process of extract, transform, and loading are shown in Table 7.17.

Table 7.17: The ETL Processes for Student Entrepreneur

ETL Processes	**Actions**
EXTRACT()	Extract the data from the Entrepreneur data sources
FILTER()	Filters the data sources for entrepreneur from UUM and UniMAP
AGGREGATE()	Count for Total Entrepreneur, and Sum for Funding
LOADER()	Loads the selected data sources into the DW

As stated in MRO, the knowledge about information as required and their related data sources are defined according to RDF/OWL based language. Thus, the MRO was processed according to an appropriate reasoning in order to identify and propose

a set of ETL processes specifications. This method is based on the RDF/OWL data model that contains nodes (i.e., subject and object) and arcs (i.e., links between nodes) represented by OWL visual graph (OWLViz[23]).

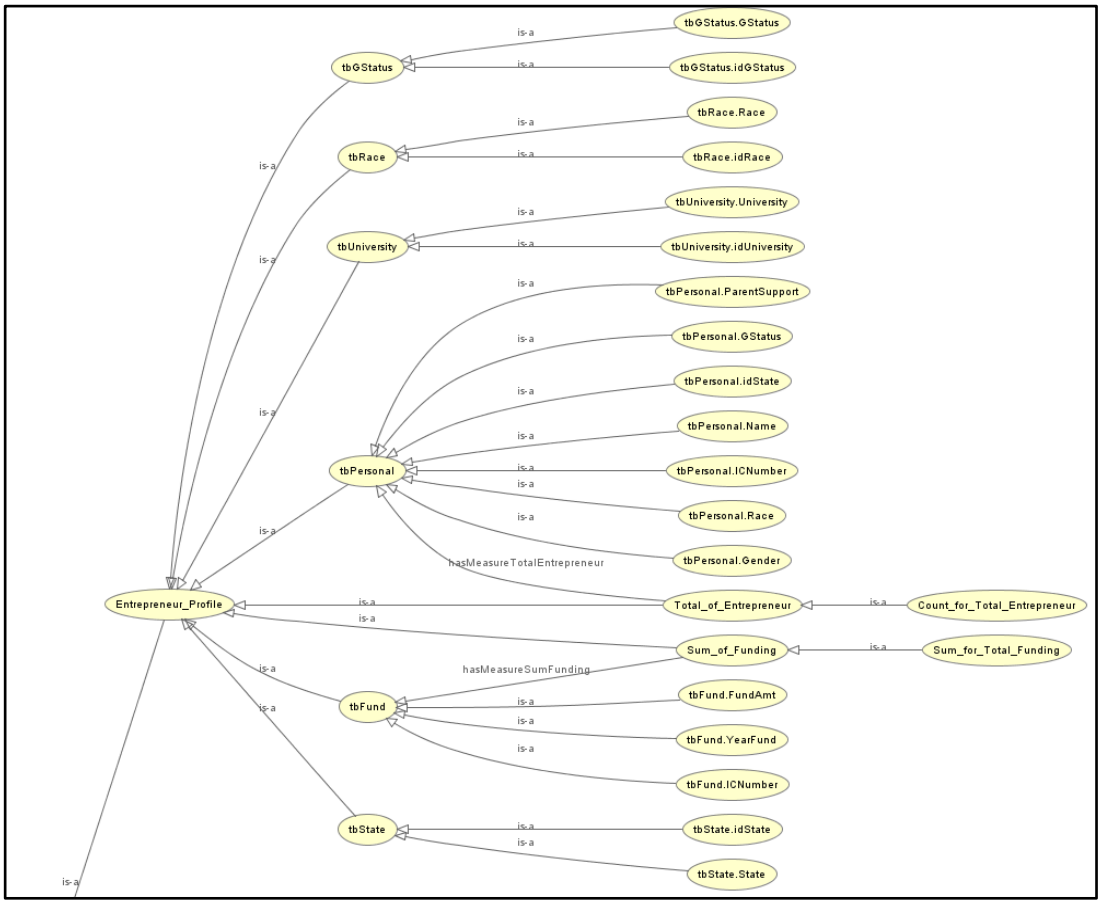


Figure 7.33: MRO for Student Entrepreneur

The MRO contains a set of RDF/OWL triples, which can be read and manipulated. The process identifies the nodes/classes and arcs/properties, and rechecks the mapping nodes that represent the DW requirements and data sources classes. Then, the ontology reasoning is used on classes and their related properties to derive the

[23]http://www.co-ode.org/downloads/owlviz/OWLVizGuide.pdf

ETL processes specifications according to the generic ETL processes tasks as shown in Table 7.17.

To generate the ETL processes specifications, the MRO is read and manipulated based on the same algorithm used in the previous case studies. The ETL processes specifications are derived from the MRO, where the MRO becomes the input, and the *ListOfETL* variable becomes the output. The ETL processes specifications are generated automatically and produce the results as a series of ETL processes for EU DW. A snippet of MRO is shown in Figure 7.34, and the ETL processes specifications are shown in Figure 7.35.

```
<!--
http://www.semanticweb.org/ontologies/2010/7/Entrepreneur_DWRO.owl#hasMeasureTotalProgr
am -->
<owl:ObjectProperty rdf:about="&Entrepreneur_DWRO;hasMeasureTotalProgram">
<rdfs:range rdf:resource="&Entrepreneur_DWRO;Count_for_Total_Program"/>
<rdfs:domain rdf:resource="&Entrepreneur_DWRO;Total_of_Program"/>
</owl:ObjectProperty>

<!-- http://www.semanticweb.org/ontologies/2010/7/Entrepreneur_MRO.owl#hasUniversity -->
<owl:ObjectProperty rdf:about="#hasUniversity">
<rdfs:range rdf:resource="&Entrepreneur_DSO;tbUniversity"/>
<rdfs:domain rdf:resource="&Entrepreneur_DWRO;Business_Profile"/>
<rdfs:domain rdf:resource="&Entrepreneur_DWRO;Entrepreneur_Profile"/>
<rdfs:domain rdf:resource="&Entrepreneur_DWRO;Entrepreneur_Program"/>
</owl:ObjectProperty>

<!-- http://www.semanticweb.org/ontologies/2010/7/Entrepreneur_DSO.owl#tbBusiness -->
<owl:Class rdf:about="&Entrepreneur_DSO;tbBusiness">
<rdfs:subClassOf rdf:resource="&Entrepreneur_DWRO;Business_Profile"/>
```

Figure 7.34: A snippet of MRO of the EU

The results have shown that the ETL processes specifications can be derived from the ontology model of the Student Entrepreneur DW requirements. The ETL processes specifications can be further translated into SQL statements or applied to any ETL tools for DW systems implementation. The sequence of the ETL processes executionsare based on the results produced from the ETL processes generation. However, the execution order does not necessarily follow the sequences since the best practices are still dependenton the developers' efforts and knowledge.

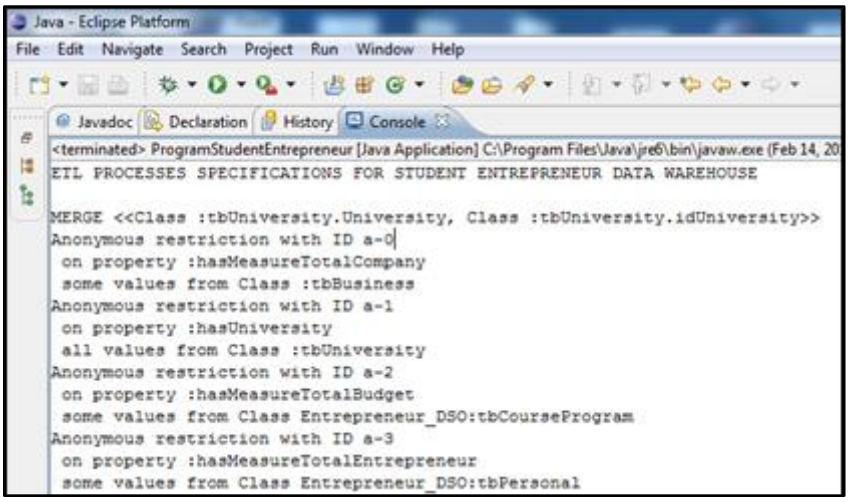


Figure 7.35: List of ETL Processes Specifications for Entrepreneur Profile

7.6 Expert Reviews

The expert reviews were conducted to help clarify the strengths and weaknesses of RAMEPs by using DW scenario of the case study. This method known as an *exemplar* is used for evaluating the methodology, especially for requirement engineering approach (Cysneiros, Werneck, & Yu, 2004). A set of questionnaires together with the case study was given to seven DW developers, where three of them are from government agencies, and the others are from DW companies. The seven

DW developers selected were qualified to assess the features of the RAMEPs, since the appropriate number forfocused participants ranges from six to nine users (Nielsen, 1997; Sobreperez, 2008). Moreover, their experiences are within the ranges of three to seventeen years in developing and implementing the DW systems in various organizations.

7.6.1 Setting of the Questionnaires

The set of questionnaires were adopted from Yu and Cysneiros(2002)and adapted within the scope of RAMEPs. The questions (given in Appendix D) aim to highlight the issues of abstraction level, participants in a domain, understanding terminology, requirement elicitation and analysis, DW and ETL design decision, DW evaluation and evolution, tools used and learning curve. The questionnaire was designed in order to capture feedback about RAMEPs processes within the knowledge scales of *yes*, *no*, and *neutral*. This scale is sufficient to probe the capabilities and limitations of the RAMEPs methodology by supporting the open-ended real-world exemplar.

The real-world exemplar was selected from the Student Affairs DW scenario in a university domain. The scenarios are explainedin detail in case study 1(as explained in Section 7.5.1), and the description about the RAMEPs was given to the experts for their review and evaluation. Briefing and explanation about the RAMEPs was given to the experts in separate occasion. Based on their knowledge and experiences, the experts responded to the questionnaires as shown in the next Section 7.6.2.

7.6.2 Results for Expert Reviewing

The summarized responses from the DW experts are presented in Table 7.18. The number represents the number of experts who said*yes*, *no* or *neutral*to a particular question. However, the responses do not include the comments from the experts.

Table 7.18: The summarized of Seven DW expert reviews

The Questions	Yes	No	Neutral	Total
Different Level of abstraction	7	0	0	7
Identifying participants in the domain	6	0	1	7
Capturing, understand, and registering terminology	6	1	0	7
Domain Analysis	6	0	1	7
Finding Requirements	7	0	0	7
DW Design	5	0	2	7
DW Evolution	5	1	1	7
ETL Processes Design	5	1	1	7
Eliciting Non-Functional Aspects	5	2	0	7
Formal Verification and Validation	6	0	1	7
Tool Supports	3	0	4	7
Learning Curve	1	4	2	7
Methodology for Simpler Problem	5	0	2	7

Based on the feedback, the experts generally agreed that RAMEPs can be implemented by using proper tools and going through proper learning exercises. This finding is clear from the higher number of *yes* for the question 1 to 9 that explain about the ETL processes design, and the lower number of *No* for the question 11 to 13 that explain about RAMEPs learning process. Specifically, the feedback for finding requirements, ETL processes design, tool supports, and learning curve are

illustrated in Figure 7.36, Figure 7.37, Figure 7.38, and Figure 7.39 respectively. Four types of feedbacks were selected because they were more significant to the ETL processes design and also they were well-commented by most DW experts.

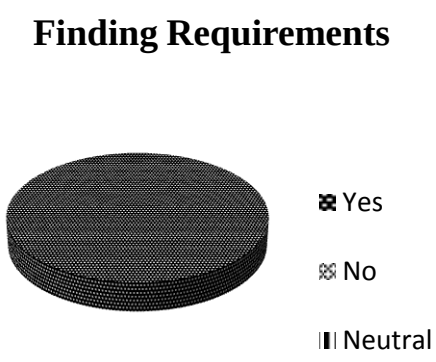


Figure 7.36: Finding Requirements

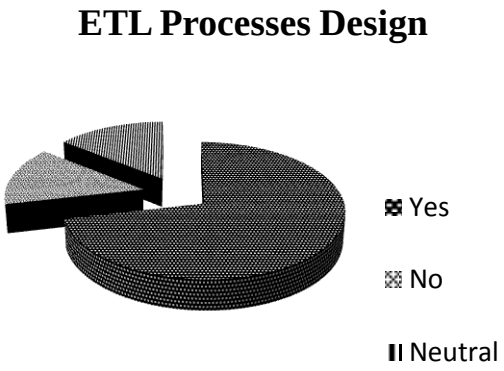


Figure 7.37: ETL Processes Design

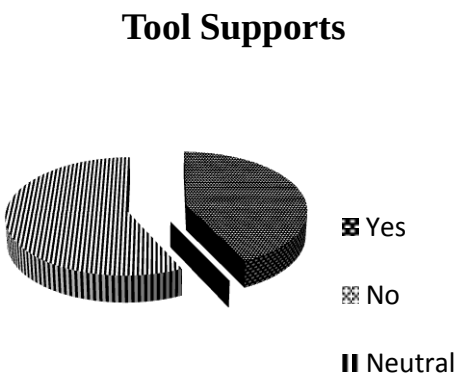


Figure 7.38: Tool Supports

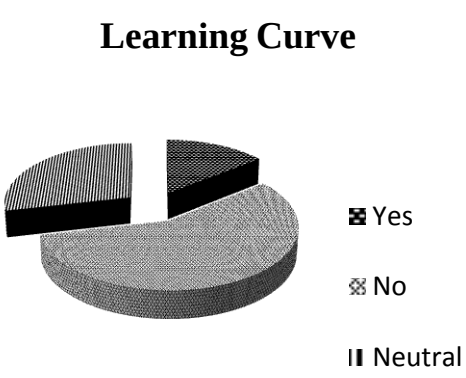


Figure 7.39: Learning Curve

The implementation of RAMEPs in real environments is challenging because of the complexity of DW model that requires longer time for learning the RAMEPs. Nevertheless, the RAMEPs approach enables DW developers to model the DW

system from the early phases to the generation of ETL processes, in whichcurrently, no specific tools are supported. However, the RAMEPs has facilitated most of the important activities in the DW systems development, especially in the ETL processes design.

Most of the experts agreed that the RAMEPs can be used to design the ETL processes by proper learning process of methods and tools. By using proper tools, the design process will be easy to implement and establish the formal verification and validation of the ETL processes specifications. However, some of the experts believed that, the design process will take time to be implemented in the real environments because of the complexity of current organization model and business rules. This is concluded from the *Learning Curve* feedback, which thinks the learning process will consume time to understand that the real DW environment is complex and tedious. In Table 7.18, all the expert review feedbacks are depicted in Figure 7.40.

Nevertheless, the experts believed that the RAMEPs approach enables DW designers to model the DW systems from the beginning to the end of DW systems development. Moreover, the ETL processes design can be automated to accelerate the execution of the DW systems. The complete responses from the DW experts are shown in Appendix E.

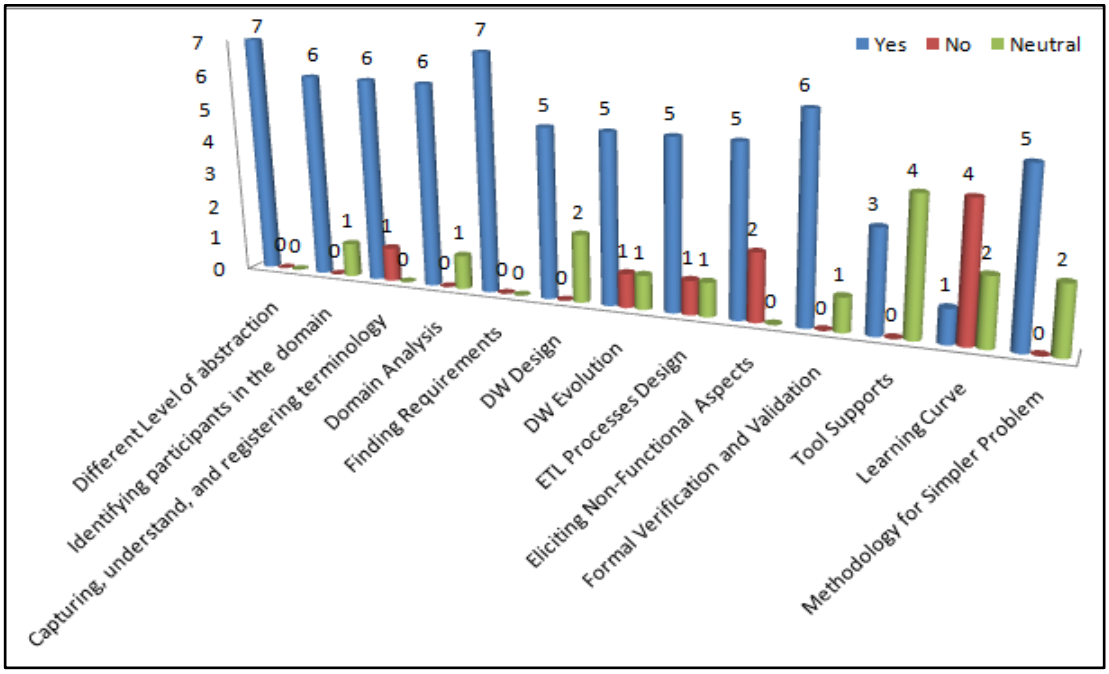


Figure 7.40: All Expert Reviews Feedbacks

7.7 Summary and General Finding

The RAMEPs approach has proven that the ETL processes specifications can be derived from the early stages of DW systems development. The methodology used in analyzing the user requirements was validated by compliant tools (i.e., DW-Tool and Protégé-OWL) successfully. Additionally, the evaluation was carried out by implementing the RAMEPs into various domains of case studies. This gives the multiple views of information in the heterogeneous environment of the DW systems.

Importantly, the RAMEPs show that, the ETL processes can be designed from the early phases of DW systems development, which obviously describe the requirements of DW in an abstract level of representation. By properly analyzing the requirements within the organization, decision-maker, and developer perspectives, the main components of DW systems (i.e., DW schemas and data integration and

transformation specifications) were successfully captured. This was shown in the case studies which has also answered the main research question.

Finally, the DW experts reviewed the RAMEPs and positively supported the method to be implemented in the real DW environment. Most of the experts believed that, the adoption of this method can help developers to define the ETL processes specifications prior to the detailed design of DW schemas, which can accelerate the implementation of DW systems. Furthermore, the use of ontology helps developers to resolve semantic heterogeneity problems during data integration and generation of the ETL processes specifications.

It is not an easy task to map high-level user requirements to DW design model, especially on the design of ETL processes. Most of the previous approaches such as using ERD-based (Kimball and Ross, 2002), UML-based (Lujan-Mora, 2005), and adhoc-based (Rizzi, 2007) do not provide adequate formalisms and techniques to derive the ETL processes specifications from the design model that are built from the users' goals. The previous approaches were only focused on the DW design, which was not well treated on ETL processes design. However, the RAMEPs was capable of deriving the DW schemas and ETL processes from the users' goals, and have resolved two main problems in designing the ETL processes. The RAMEPs was filled the gaps by provided approach to design ETL processes, and generate the ETL processes specification automatically. This new approach can help to reduce the DW project failure and support the advancement of DW tools.

The validation process of the RAMEPs was conducted in the proposed model checking as explained in Section7.1. The checking model emphasized the validity of the design diagram on each perspective (i.e., organization, decision-maker, and developer) by ensuring the correctness of captured DW components and linkages between diagrams to another diagram. The compliant tools (i.e., DW-Tool and Protégé-OWL) are used to ensure the DW components are properly captured and are able to support for the next diagram's inter and intra-perspectives.

Since the DW-Tool and Protégé-OWL are two different tools with diverse purposes, the only thing connecting them is data representation. Both tools use the XML-based data structure to represent the diagrams (for DW-Tool) and ontology structure (for Protégé-OWL). However, in this research, the ontology structure is represented by OWL language that is an enhancement from the XML structure thatis completely supported by the Protégé-OWL. Although OWL structure needs to be prepared according to the proper syntax, the XML-based source is easily transformed to the OWL language according to structure similarity. By utilizing XML-based data sources in both tools, the linkages between diagrams and ontology can be easily organized and maintained.

The facts, dimensions, measures, attributes and actions, which are produced from the organizational, decision-maker, and developer modeling were carefully translated into ontology modeling (see Section 7.4.4). These DW requirements were modeled as ontology structure and rechecked for correctness by using built-in ontology *Pallet* reasoner. The *Pallet* was applied to ensure the building of DWRO and MRO are

consistent with the goal-oriented requirement's model. The correctness of ontology is important in order to enable the ETL processes specifications to be generated accordingly and ready for implementing the DW systems.

The evaluation of three case studies (discussed in Sections 7.5.1, 7.5.2, and 7.5.3) has demonstrated that the implementation of RAMEPs works in real DW systems development. The RAMEPs has shown that, the three different heterogeneous data sources setting and underlying various goals were consistently supporting the design of the ETL processes. The use of tools (i.e., DW-Tool, and Protégé-OWL) and associated diagrams (i.e., actor, goal, dimension, measure, and action) have encouraged the reuse of models in the design tasks. Moreover, each tool is capable of checking the model correctness, supported by Java application for generating the ETL processes specifications.

7.8 Conclusion

The validation and evaluation process show that the RAMEPs can be used confidently in designing the ETL processes. It proves that the ETL processes can be designed from an early requirement phase of DW systems development. Consequently, the goal diagrams extension and ontology structure are technically correct for representing the DW schemas and ETL processes design. The linkages between goal-oriented and ontology model confirm the validity of the notations used in the design of the ETL processes. Chapter 8 concludes the research by highlighting the achievement of the research objectives, contributions, limitations, future works and concluding remarks.

CHAPTER EIGHT–CONCLUSIONS AND FUTURE WORKS

This chapter reviews all the findings and concludes the research work by giving a holistic view according to the research objectives. The main contributions are provided.However more work needs to be done to design efficient and effective ETL processes due to the complexity of the DW domain. Therefore, the limitations in designing the ETL processes are discussed and overviews of the solutions are highlighted. This chapter ends by proposing the future work.

8.1 Examining Research Objectives

The main objective of this research is to facilitate, manage, and enhance the design of the ETL processes from the early phases of DW systems development based on goal-oriented and ontology approach. The specific objectives are:

Research objective 1: To define the semantic framework of DW systems development for guiding the ETL processes design –discussed and presented in Chapter 2, Chapter 3, and Chapter 4.

Research objective 2: To develop the requirements analysis method by using a goaland ontology for designing the ETL Processes - discussed and presented in Chapter 5 and Chapter 6.

Research objective 3: To develop an algorithm and demonstrating the process to generate the ETL processes specifications- discussed and presented in Chapter 6.

Research objective 4: To validate and evaluate the approach by usingcompliant tools and applying to real case studies - discussed and presented in Chapter 7.

The summary of the research works related to research objectives are discussed in the following section.

8.1.1 An analysis of DW and ETL Processes Design Problems

A review and analysis about the DW and ETL processes development identified the common problems for designing the DW systems. These common problems are: i) the complexity and hugeness of the DW, ii) inefficiency of data loading, iii) data integration and transformation process, and iv) generating the data integration and transformation specifications. Two of these common problems (i.e., data integration and transformation process and generating the data integration and transformation) are directly related to the design of the ETL processes that provided the impetus for this research.

However, detailed exploration of the ETL processes design found that the issues related to these common problems are: i) defining and maintaining the ETL processes specifications, and ii) handling the semantic heterogeneity in data integration and transformations. These issues were traced back to the general failures of software systems developmentrelated to the design-process. In particular, a familiar design-related failure was considered due to the failure in requirements elicitation and analysis. This is more pressing on the DW project due to some reasons such as: i) DW is a long-period project and various information for decision-

makersare anticipated, ii) requirements difficult to specify because data is poorly shared across the organization, and iii) data needs to be validated and transformed, where some data does not exist.

The DWsystem's design involves several tasks such as defining the DW schemas and the ETL processes specifications, and these have been extensively studied and practiced for many years. However, the problems in heterogeneous data integration are still far from being resolved due to the complexity of ETL processes and the fundamental problems of data conflicts in information sharing environments. Current approaches that are based on existing software requirement methods have limitations on translating the business semantics for DW requirements toward the ETL processes specifications. This research explored the enhancement for designing the ETL processes and resolving the design-related problems during the design-process within the suitable framework of DW systems development. The semantic framework of DW system supports the semantic component of the ETL processes that produced from the extensive review on this area. The framework is important in order to reduce the business semantic inconsistencies in the ETL processes. This was aligned and achieved with the research objective one.

8.1.2 The Use of Goal-Oriented and Ontology Approach for Resolving the ETL Processes Design Problems

This research addresses two main issues in designing the ETL processes. Firstly, to define and maintain the ETL processes specifications, which is required by DW systems prior to implementation. Normally, the ETL processes specification is

derived from the representation and modeling of the ETL processes that comes from the traditional ways of requirement elicitation. This requirement is usually incomplete and prone to errors because it is difficult to maintain in the application-driven and platform-dependent methods. The required information according to the relevant semantics of the data sources are likely to be ignored. This creates an unmanageable situation for generating the ETL processes specifications from its designs and contributes to the failure of the DW projects.

Therefore, this research uses the goal-oriented approach for gathering and analyzing the DW requirements that are based on the Tropos methodology. The goal-oriented approach focuses on the analysis of the high-level objectives of the stakeholders and setting of the organizations rather than a specific function that the DW systems should have. This encourages the DW developer to explore and understand the reasons of user requirements prior to sourcing out the possible solutions to be implemented. In software engineering discipline, this is known as analyzing early requirements that significantly decrease the possibility of doubts over the user requirements and, consequently, reduce the failure of the DW project.Moreover, the DW developer uses these requirements within the implicit knowledge of the DW domain to create a shared repository that guarantees the right semantics and maintains the artifacts' design.

The second task is to handle the semantic heterogeneity problems in data integration and transformation. The significant challenge in the early phases of the DW design is to map the user requirements to the appropriate data sources and handling by the

sequences of data operations and transformations. The aim of this process is to provide the correct semantics of information regardless of the DW structure and semantic heterogeneity problems. Therefore, the ontology is used to model and represent the DW requirements for describing the semantics of the user requirements toward the corresponding data sources. Realizing the fact that the design of the ETL processes need to be efficient, robust, and evolvable, thusdeveloping a formal and structure-driven model to allow a high possibility of automation of the ETL processes design is quite relevant in the current environment. Given the ontology of DW requirements, the application is constructed based on the reasoning tasks that are performed to facilitate the generation of the ETL processes specifications in an automated way. This was aligned and achieved with the research objective two.

8.1.3 Development of RAMEPs for Designing the ETL Processes

This research proposed a method for designing ETL processes called RAMEPs. The development of RAMEPs is based on the Tropos methodology that was developed from the well-accepted i* conceptual framework of an agent-based software development process. The aim of RAMEPs is to provide the decisional information from the perspectives of organization, decision-maker, and developer. The requirement analysis approach determined the components of a DW and ETL processes model through the goal-oriented diagrams. These diagrams represented in specific symbols explained their roles and activities (e.g., facts, dimensions, measures, business rules, actions). The data needed by the decision maker is provided by the organizational, decisional, and developer models related to the

components of ETL processes. Particularly, these models help developers to generate appropriate actions for populating the data sources to the DW.

All activities in RAMEPs were carried out in sequences, and cannot be implemented in parallel. The activities are: i) gather user requirements from stakeholders, ii) analyze requirements based on the organization, decision-maker, and developer perspectives, iii) construct ontology for user requirements, data sources, and data transformation, iv) map the requirements' ontology with the data sources, v) refine the merging ontology to fully satisfy the user requirements, and vi) construct the ETL processes specifications from the merging ontology. The focused of RAMEPsis to perform the analysis on data transformation that belongs to the intention of ETL developers. The ontology is used to conceptualize the facts that are produced from the analysis of data integration and transformation. This was aligned and achieved with the research objective three.

8.1.4 RAMEPs Validation, Evaluation, and Implementation

The RAMEPs approach has shown that the ETL processes specifications can be derived from the early phases of DW systems development. This was successfully proven by the validation of the goal-oriented diagrams produced by the DW-Tool. Each diagram represented for every modeling perspective was systematically checked on their correctness by an appropriate reasoning of the DW-Tool. Moreover, the diagram transitions between modeling perspectives were properly treated through the intermediateinterfaces. Therefore, the TA-Tool was developed for inserting new actions and business rules for analyzing the data transformations. This tool used the

XML-based data that represented the goal diagrams to be manipulated and reorganized into the DW-Tool. Finally, the goal diagrams in DW-Tool were stored in XML-based structure and presented the new transformation diagrams as required.

The ontology helped to resolve semantic heterogeneity problems during data integration and transformation and finally facilitated the generation of the ETL processes specifications automatically. Prior to implementation of ontology, the structure of the ontology that represents the ETL processes design was validated by using ontology reasoner. This research used*Pallet* among the good reasoners provided by Protégé-OWL to check the correctness of the ontology structure. Incorrectness of the ontology is identified and immediately corrected by using the Protégé-OWL. In-built *Pallet* reasoner accelerated the correction of ontology and built-up the ontology faster.

The evaluation approach was carried out by implementing the RAMEPs into various domains of case studies. This gave multi views of information in the DW systems. The case studies were selected from three different domain and data sources setting, which demonstrated different scenario of semantics heterogeneity problems. The first case study was conducted in UUM Student Affairs Department, which utilized two different student databases (i.e., Undergraduate and Postgraduate students) under the supervision of the UUM Student Affairs Department. Historically, these databases are not combined because of the roles played by the department, and it remains implemented until now. Basically, under this scenario, it is not difficult to get consensus on terms used in ETL processes modeling. However, the challenges

are to represent the different semantics of particular data sources schemas (e.g., semester) that can be cross-referenced to the user requirements.

The second case study was conducted at Gas Malaysia specifically focused on the Billing utility area. Gas Malaysia is a large utility company in Malaysia that provides gas for residential and industrial consumers. These two types of consumers are organized in separate databases, i.e., residential database (UBIS) which is handled by Gas Malaysia, while the industry's database (JDE) is handled by an external party. The tasks to integrate the UBIS and JDE databases are the challenges because of lack of knowledge on the JDE database developed by the external party (JD Edward Malaysia). Moreover, it is difficult to streamline these databases schemas due to non-authorization by JDE. This research handles the semantics heterogeneity problems based on the limited knowledge of the JDE database.

The third case study was conducted in MoHE that particularly focused on the entrepreneur area. The case study focused on the development of a student entrepreneur DW, which currently is not available. The main user of this DW is the EU of MoHE, which currently obtains the student entrepreneur information by requesting the IHLson a periodic basis. The main challenges are to collect and integrate the entrepreneur data (e.g., personal profile, academic profile, and business profile) from the various IHLs. This research explored the semantics heterogeneity problems dealing with two IHLs (i.e., UUM and UniMAP) in data sources integration. The different structure of entrepreneur data clearly showed the inconsistency of entrepreneur information provided by UUM and UniMAP. The

analysis of user requirements was properly synchronized on the terms used by UUM and UniMAP, and finally utilized in the entrepreneur DW systems.

The DW experts reviewed the RAMEPs and positively supported the method to be implemented in the real environment. They believed that the adoption of this method can help developers to systematically design the ETL processes and accelerate the implementation of DW systems. Initially, the reviews were conducted to help clarify the strengths and weaknesses of RAMEPs by using DW scenario of the case study. The expert reviewed a method known as an *exemplar* used for evaluating the methodology, especially for requirement engineering approach. A set of questionnaires together with the case study was given to seven DW developers, three of them from government agencies, and the others from DW companies. Their experiences are within the ranges of three to seventeen years in developing and implementing the DW systems in various domains and organizations. The set of questionnaires were accommodated within the scope of RAMEPs.

In summary, the experts agreed that the RAMEPs can facilitate the ETL processes design by using proper tools, but it will take time to implement in the real environments because of the complexity of model and business rules. Nevertheless, the RAMEPs approach enables a developer to design the DW systems from the beginning to the end. However, the changing goals and policies, which reflect the existing DW schemas and ETL processes specifications, need to be tackled properly by RAMEPs. This was aligned and achieved with the research objective four.

8.2 Research Contributions

This section highlights the contribution to the DW area, primarily in the modeling and designing of the ETL processes. For the ETL processes design, a goal-oriented and ontology-based approach was coherently applied in the process. The requirement analysis approach helpsthe DW developer to design and generate the ETL processes specifications for implementation in DW systems. Sections8.2.1 to 8.2.7summarize this research's key contributions.

8.2.1 Comparative Analysis of ETL Processes Requirements Approaches

Currently, DW and ETL processes modeling are well established, widely adopted, successful, and fully supported by the industries. The dimensional modeling that founded the DW is a mature methodology that organizes data into a simple and intuitive representation for the decision-makers' view and analyzes data. This research analyzes the DW requirements from the high level abstract of user requirements (e.g., goal, sub-goal, stakeholder, resources) toward the detailed specifications (e.g., extracting, filtering, conversion, loading) which are important for handling the complexity of the ETL processes design and ensure that the DW systems implementation is successful.

Therefore, this research gives attention to the DW and ETL processes requirement analysis approaches as discussed in Chapter 3 and appropriate framework for DW system development as discussed in Section 4.3.1, Chapter 4. This framework includes the goal and ontology approach for analyzing and representing the DW requirements. This research contributes a comparative analysis of ETL processes

requirements approaches as presented in Table 3.3, Chapter 3. The survey of literature on DW and ETL processes design by highlighting the works of requirement analysis approaches in this research area are also provided. This supports the **research objective 1:** *to define the semantic framework of DW systems development for guiding the ETL processes design*.

8.2.2 A Systematic Approach for Designing the ETL Processes

To model and design the ETL processes required knowledge about current DW and data sources schemas. The DW schema is produced from the user requirements that were analyzed to define the DW components (e.g., fact, dimension, measure). Underlying the current practices and focused problems, this research contributes to a systematic approach to design the ETL processes by utilizingthe semantic framework of DW systems development. An explanationabout the use of goal-oriented and ontology approaches in the context of the early phases of DW systems development were discussed in detail in Chapter 3 and Chapter 4.

In summary, a systematic approach RAMEPs considered the whole designing tasks into unified perspectives of user requirements. This approach was established from the theory of an organization, decision-making, and socio-technical system. These theories have common suggestionson developing information system, which requires strong interaction among organization, decision-makers, and resources to provide complete and satisfactory information for the users. The motivation from these theories derived the requirements that need to be elicited and analyzed from the perspectives of organization, decision maker, and developer. The proposed RAMEPs

is in line with the **research objective 2**: *to develop the requirements analysis approach by using a goal and ontology for designing the ETL Processes*, which was discussed and presented in Chapters 5 and 6.

8.2.3 Automate the Generation of ETL Processes Specifications

Generation of codes from software design is provoked by the current software tools. However, the success of tools is still questionable and seems to require a long time to support the ETL processes implementation. The possibility to generate the coding automatically from the design artifacts was practically promoted by domain specific-modeling (DSM) approach(Kelly & Pohjonen, 2009), supported by the mainstream modeling approach such as UML and ORM. However, in DW systems design, this research contributes to automatically generate the ETL processes specifications by using the newly developed algorithm. Then, the ETL processes specifications can be used directly or indirectly by invoking the existing ETL tools for implementing the DW systems. This is in line with the **research objective 3**: *to develop an algorithm and demonstrating the process to generate the ETL processes specifications*, which was discussed and results presented in Section 6.5, Chapter 6.

8.2.4 Model Checking with Modified and Newly Developed Compliant Tools

Basically, model checker methods are used to verify the correctness of software systems at a design stage. The method proposed by Ogawa et al. (2008) was adopted to validate the DW components by using compliant tools (i.e., DW-Tool and Protégé-OWL). This research did not followthe whole process proposed by the

method as the field of model checker process is complex and diverse for ensuringthe correctness of software systems. Moreover, almost none of any model checker methods are based on DWRO or DW ontology. Therefore, this research contributes to the variance of model checking methods and gives several benefits as presented in Chapter 7. This in line with the **research objective 4**: *to validate and evaluate the approach by using compliant tools and applying to real case studies.*

8.2.5 Bridge the Gap Between Conceptual to Detail Design of the ETL Processes

Traditionally, a software development is a process of mapping from the domain idea, to design models, and finally to the source codes' generation. It is generally knownthat these mappings create the gap between conceptual and detailed design because the process tends to be slow and leads to errors for the software system. Therefore, the notion of DSM has emerged to address these problems by avoiding unnecessary mappings and focusing on the solution at the same level of abstraction with the domain. This research contributes to the DSM by applying an enhanced concept in the DW area by bridging the gap between conceptual and detailed design by giving better understanding on the early requirements toward the specification of the ETL processes.

Applying the goal-oriented approach, supported by ontology for modeling the ETL processes has raised the level of abstraction and focuses on information that needs to be modeled. Using ontology (i.e., OWL) as modeling language is closer to the model to the perceived domain, and highly automates the generation of ETL processes specifications.

8.2.6 Development of DW Requirements Ontology

Building ontology in various domains has attracted tremendous attention from the research community. The development of DW requirements ontology (DWRO) is specialized for organizing and representing the DW components that emphasize the ETL processes activities. DWRO is an application ontology that allows the DW components to be modeled in the ontology structure. This research contributes to the enhancement of the ontology coverage by building the DWRO in the DW or BI domain. Particularly, this research also has contributed to the area of domain engineering by introducing the DWRO based on the organizational, decisional, and socio-technical theory. This is part of the solution for supporting the **research objective 2**: *to develop the requirements analysis method by using goal and ontology for designing the ETL processes*, which was discussed and presented in Chapter 6.

8.2.7 Extending the Use of i* Modeling Concepts and Notations

This research utilized the modeling concepts and notations from the Tropos methodology, which fundamentally adopted the concepts and notations froman i* framework. Currently, the i* framework is an international standard for complex reactive, distributed and dynamic systems applications by proposing the User Requirements Notation (URN). The URN consists of Goal-oriented Requirements Language (GRL), and Use Case Maps (UCM) for supporting the modeling of functional and non-functional requirements. Therefore, the Tropos methodology is widely adopted in modeling and designing the goal-oriented based software system. However, the adoption in DW systems development is immature.

This research contributes to expedite and widen the application of URN by adapting the concepts and notations RAMEPs. The *plan* modelingconcepts and notations in Tropos are one of the important analysis tasks in RAMEPs which were successfully adapted for modeling and designing the ETL processes. Moreover, the completeness of the transformation analysis model was supported by the *resource* concept and notation, which represents a physical or informational entity. Nevertheless, the use of these notations required support from the DW-Tool, and this has been tackled by developing an intermediate application TA-Tool.

8.3 Research Limitations

This section highlights the limitations on developing the method for modeling and designing the ETL processes in the DW systems. These limitations narrow the option for proving the ETL processes design method. In summary, these limitations are as follows:

8.3.1 Limited Compliant Tools

Very limited tools can support the modeling of goal-oriented diagrams, especially in modeling and designing the DW systems and ETL processes. Current tools emphasize executions of the ETL processes specifications and implementation of the DW systems. The functions played by the tools,do not include the in-depth modeling of particular tasks. Moreover, the goal-oriented approach requires some *social* features that represent roles such as actor, goal, resource, dependency, and others. These features were not supported by the well-known modeling tools such as UML

or ORM. Therefore, this research was subjected to these constraints and had to utilizeappropriate research tools (e.g., DW-Tool and Protégé-OWL) for supporting the proposed solutions. Indeed, the development of this tool is not included in this research work.

8.3.2 Mapping between DWRO and DSO

The mapping between DWRO and DSO was carried out manually because of unavailable appropriate tools. Current tools (e.g., Protégé-OWL, OntoClean) were impossible to apply because of insufficient reasoning in supporting the mapping processes, which involved matching between class to class, property to property, and relationship to a relationship. Moreover, the mapping functions are built for a specific domain (e.g., Protégé-2000 for frame-based ontology structure), which was unsuitable for this research. The manual process slows the mapping process and increases the possibility of errors on the concepts of matching. Although this research does not focus on the development of mapping tools, the guidelines on the mapping process are properly presented and applied. This at least has reduced the uncertainty on mapping between concepts from DWRO and DSO. However, if the mapping process can be facilitated by the tools, the automation mapping can be carried out, and the building of MRO can be accelerated.

8.4 Future Work

In respect to the modeling and designing the ETL processes, this research suggests the following issues to tackle the current limitations towards future research work.

The suggestion can be classified into either short (i.e., immediately implement after this research) and long term (i.e., possibility involving new trends in DW and semantic web technology). Nevertheless, future work can complete the application prototype for generating the ETL processes specifications and implemented in the case study. However, several research issues are still left open on the grounds of this research work and discussed in the next Section8.4.1 to 8.4.4.

8.4.1 Softgoals for ETL Processes Quality Measures

In goal-oriented modeling, a softgoal is used to model quality attributes for which the criteria for satisfaction are sufficiently judged by actors (Yu et al., 2011). Quality attributes are derived from non-functional requirements (NFR) of DW, which are not well treated in this research. Generally, quality attributes are defined by stakeholders and can be classified such as security, performance, availability, cost, accuracy, usability, reliability, and others. Some attributes have been supported by Tropos methodology, and current research introduced a security concept in Tropos methodology for modeling the software system (Mouratidis et al., 2009). This provides the foundation for modeling the security concepts in the ETL processes model. Therefore, the short term future work for modeling and designing the ETL processes should include the NFR for coverage of the entire element of requirements in DW systems.

8.4.2 Impact Analysis for DW Requirements

New computing challenges that reflect changing stakeholder needs have arisen. The investigation on these challenges needs to be done to tease out the essential goals and to assess their impact on the DW requirement engineering tasks. Generally, in a software system, the design decision adheres to requirements that reflect on the modularity or performances of the system (Cheng & Atlee, 2007). In RAMEPs, the changes of stakeholder needs will reflect the goal-modeling and ontology structure of DW requirements. Therefore, some impact analysis method needs to be proposed and implemented in the DW requirement produced by the RAMEPs. The impact in ETL processesneeds to be considered prior to design tasks. The short term future work will focus on the impact analysis method and predict the successful ETL processes implementation according to the changes of stakeholder requirements. Some adjustments on the goal and ontology modeling need to be made for complying with the requirement changes without trade-off of the modularity and performances of the DW systems.

8.4.3 Applying RAMEPs in a Complex Organization and DecisionProcess

One of the comments forwarded by the DW experts is about the possibility of RAMEPs to be implemented in a complex organization and decision process environments. This phenomenon is with regards to the issues of algorithm complexity for identifying the data transformations' activity in ETL processes specifications. Basically, these algorithms explain the classes, properties, and relationships in MRO to be inferred for defining the appropriate transformation

activities between the data sources and DW. The complexity of the organization and decision process will cause the goal and ontology modeling of the ETL processes to become large. The long term future work should enhance the RAMEPs method for better handling of the complexity of DW requirements that come from the complex organization and decision process. Furthermore, the ETL processes specifications should be easy to produce by a better design decision of complex organization and design processes.

8.4.4 Developing Tools for RAMEPs

As mentioned, due to limitations of this research, an appropriate tool, which is a complete tool that can support from requirement analysis using goal-oriented until the generation of the ETL processes to support the implementation of RAMEPs was not developed. Current works are using a number of tools that comes from previous research work, supported by intermediate tools for connecting between the tools. Clearly, the problem arose on how to consolidate all tasks in RAMEPs into a single tool. Therefore, a new tool needs to be developed in order to tackle this problem and accelerate the process to design the ETL processes. The long term future research needs to develop a tool that enables the defining, adjusting, generating, and executing of the early phases of requirement analysis to the implementation of the ETL processes specifications. This tool should enable the capturing and managing the ontology, which has been recently introduced in one of the DW solution providersnamedExpressor®[24]. However, Expressor® is mainly for a data integration

[24] http:// http://www.expressor-software.com/

solution and does not support the ETL processes design, especially from the early phases of DW requirements.

8.5 Conclusion and Final Remarks

The adoption of RAMEPs method can help developers to define clearly the user requirements prior to the detailed design of ETL processes in a DW systems environment. The ontology model helps developers to resolve semantic heterogeneity problems during data integration and transformation activities. The RDF/OWL language is easy to maintain and the ETL processes specifications are easily produced, although the changes in user requirements frequently occur.The RAMEPs has proven the ETL processes specifications can be derived from the early phases of DW systems development. The methodology used in analyzing the DW requirements was validated by DW-Tool and Protégé-OWL successfully. The RAMEPs was evaluated by implementing it into various domains of case studies and reviewed by the DW experts for identifying strengths and weaknesses of the approach. The case studies were given multi-views of information and diversification of ETL processes workflows. Moreover, the RAMEPs can beused in EAI and EII environments to achieve a consolidated data for analyzing and reporting. The semantic web-based applications for DWseem to have promising prospects to adopt RAMEPs approach for data integration and transformation.

REFERENCES

Abdullah, M. S. (2006). *A UML Profile for Conceptual Modelling of Knowledge-Based Systems.*Unpublished PhD, University of York.

Abello, A., Samos, J., & Saltor, F. (2002).*YAM2 (Yet Another Multidimensional Model): An extension of UML*. Paper presented at the IDEAS'02.

Abiteboul, S., Cluet, S., Milo, T., Mogilevsky, P., Simeon, J., & Zohar, S. (1999). Tools for Data Translation and Integration.*Bulletin IEEE Computer Society Technical Committee on Data Engineering*.

Agosta, L. (2002). Market Overview Update: ETL. Retrieved September 20, 2007, from http://www.gigagroup.com/

Ahmad, M. N., & Colomb, R. M. (2007).Overview of Ontology Servers Research, Webology 4(2), Article 43.*Webology* Retrieved January 15, 2008, from http://www.webology.ir/2007/v4n2/a43.html

Akkaoui, Z. E., Mazón, J.-N., Vaisman, A., & Zimányi, E. (2012).*BPMN-Based Conceptual Modeling of ETL Processes.* Paper presented at the 14th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2012), Vienna University of Technology.

Aleksovski, Z. (2008). *Using Background Knowledge in Ontology Matching.*Vrije University.

Alexiev, V., Breu, M., Bruijn, J. d., Fensel, D., Lara, R., & Lausen, H. (2005).*Information Integration with Ontologies: Experiences from an Industrial Showcase*: John Wiley & Son Ltd.

Ali, R., Dalpiaz, F., & Giorgini, P. (2010).A Goal-based Framework for Contextual Requirements Modeling and Analysis.*Springer - International Journal of Requirements Engineering, 15*(4), 439–458.

Allemang, D., & Hendler, J. (2008).*Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*: Morgan Kaufmann.

An, Y. (2007).*Discovering and Using Semantics for Database Schemas.*Unpublished PhD, University of Toronto.

Antonio, A. d., Ramırez, J., Imbert, R., & Mendez, G. (2005). Intelligent Virtual Environments for Training: An Agent-Based Approach. *LNAI*(3690), 82-91.

Antoniou, G., & Harmelen, F. V. (2003). *Web Ontology Language: OWL*: Springer-Verlag.

Aparício, A. S., Farias, O. L. M., & Santos, N. d. (2005).*Applying Ontologies in the Integration of Heterogeneous Relational Databases.* Paper presented at the Conferences in Research and Practice in Information Technology (CRPIT), Sydney.

Archer, M. S., & Tritter, J. Q. (2000).*Rational choice theory: resisting colonization*: Routledge.

Arpírez, J. C., Corcho, O., Fernández-López, M., & Gómez-Pérez, A. (2003).WebODE in a Nutshell.*AI Magazine, 24*(3), 37-48.

Baader, F., Horrocks, I., & Sattler, U. (2005).Description Logics as Ontology Languages for the Semantic Web.*Lecture Notes in Artificial Intelligence, 26*(5), 228-248.

Barzdins, G., Barzdins, J., & Cerans, K. (2009).From Databases to Ontologies.*IGI Global*, 242-266.

Bekke, J. H. t. (1992).*Semantic Data Modeling*: Prentice Hall.

Beneventano, D., Bergamaschi, S., Guerra, F., & Vincini, M. (2003).*Building an integrated Ontology within SEWASIE system.* Paper presented at the 1st InternationalWorkshop on Semantic Web and Databases (SWDB), Berlin, Germany.

Berenbach, B., Paulish, D. J., Kazmeier, J., & Rudorfer, A. (2009).*Software & Systems Requirements Engineering: In Practice*: McGraw-Hill.

Berners-Lee, T., Hall, W., Hendler, J., Shadbolt, N., & Weitzner, D. J. (2006).Creating a Science of the Web.*Science, 313*(5788), 769-771.

Booch, G., Rumbaugh, J., & Jacobson, I. (1999).*The Unified Modeling Language User Guide*: Addison Wesley.

Bostrom, R. P., & Heinen, J. S. (1977). MIS Problems and Failures: A Socio-Technical Perspective. *MIS Quarterly, 1*(3), 17-32.

Bouzeghoub, M., Fabret, F., & Matulovic-Broqué, M. (1999).*Modeling Data Warehouse Refreshment Process as a Workflow Application.* Paper presented at the International Workshop on Design and Management of Data Warehouse (DMDW'99), Heidelberg, Germany.

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2004). Tropos: An Agent-Oriented Software Development Methodology. *Autonomous Agents and Multi-Agent Systems, 8*(3), 203-236.

Brisaboa, N. R., Penabad, M. R., Places, A. S., & Rodriguez, F. J. (2002).Ontologies for Database Federation.*UPGRADE, Vol. III*(3), 52-61.

Bruckner, R. M., List, B., & Schiefer, J. (2001).*Developing Requirements For Data Warehouse Systems With Use Cases.* Paper presented at the 7th Americas Conference on Information Systems.

Bruckner, R. M., List, B., & Schiefer, J. (2002).*A Holistic Approach for Managing Requirements of Data Warehouse Systems.* Paper presented at the 8th Americas on Information Systems.

Buccella, A., Cechich, A., & Brisaboa, N. R. (2003).*An Ontology Approach to Data Integration.* Paper presented at the JCS&T.

Calvanese, D., Giacomo, G. D., Lenzerini, M., Nardi, D., & Rosati, R. (1998, 20-22 August).*Information Integration: Conceptual Modeling and Reasoning Support.* Paper presented at the 3rd IFCIS International Conference on Cooperative Information Systems, New York, NY, USA.

Calvanese, D., Giacomo, G. D., Lenzerini, M., Nardi, D., & Rosati, R. (2001). Data Integration In Data Warehousing. *International Journal of Cooperative Information Systems, 10*(3), 237-271.

Cao, L., Zhang, C., & Liu, J. (2005). Ontology-based Integration of Business Intelligence. *International Journal of Web Intelligence and Agent System*.

Chaudhri, V. K., Farquhar, A., Fikes, R., Karp, P. D., & Rice, J. P. (1998, July 26-30). *OKBC: A Programmatic Foundation for Knowledge Base Interoperability.* Paper presented at the AAAI-98, Madison, WI.

Chaudhuri, S., & Dayal, U. (1997). An Overview of Data Warehousing and OLAP Technology.*ACM SIGMOD Record, 26*(1).

Chen, P. P.-S. (1976). The Entity-Relationship Model-Toward a Unified View of Data. *ACM Transactions on Database Systems, 1*(1), 9-36.

Cheng, B. H. C., & Atlee, J. M. (2007).*Research Directions in Requirements Engineering*. Paper presented at the 2007 Future of Software Engineering.

Codd, E. F. (1979). Extending the Database Relational Model to Capture More Meaning.*Communications of the ACM, 4*(4), 397-434.

Connolly, T., & Begg, C. (2005).*Database System A Practical Approach to Design, Implementation, and Management* (4th ed.): Pearson Education Limited.

Corcho, O., Fernandez-Lopez, M., Gomez-Perez, A., & Lopez-Cima, A. (2005).*Building Legal Ontologies with METHONTOLOGY and WebODE.* Paper presented at the Law and Semantic Web Conference.

Cui, Z., & O'Brien, P. (2000).*Domain Ontology Management Environment.* Paper presented at the 33rd International Conference on System Sciences, Hawaii.

Cure, O., & Jablonski, S. (2007). *Ontology-Based Data Integration in Data Logistics Workflows.* Paper presented at the ER Workshops CMLSA, Auckland, New Zealand.

Cysneiros, L. M., Werneck, V., & Yu, E. (2004).*Evaluating Methodologies: A Requirements Engineering Approach Through the Use of an Exemplar.* Paper presented at the 7th Workshop on Autonomous Agents.

Daconta, M. C., Obrst, L. J., & Smith, K. T. (2003).*The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*: Wiley Publishing Inc., Indianapolis, Indiana.

Daft, R. L. (2008). *Organization Theory and Design* (10 ed.). Mason, USA: South-Western Cengage Learning.

Denny, M. (2004).Ontology Tools Survey, Revisited. Retrieved September 20, 2007, from http://www.xml.com/pub/a/2004/07/14/onto.html

Doan, A., & Halevy, A. Y. (2005). Semantic-integration research in the database community: A Brief Survey. *AI Magazine, 26*(1).

Dou, D., & LePendu, P. (2006).*Ontology-based Integration for Relational Databases.* Paper presented at the SAC'06, Dijon, France.

Drucker, P. F. (1974). *Management: tasks, responsibilities, practices*: Butterworth-Heinemann.

Farhan, M. S., Marie, M. E., El-Fangary, L. M., & Helmy, Y. K. (2012).Transforming Conceptual Model into Logical Model for Temporal Data Warehouse Security: A Case Study.*International Journal of Advanced Computer Science and Applications (IJACSA), 3*(3), 115-122.

Fellbaum, C. (1998). A Semantic Network of English: The Mother of All WordNets. *Computers and the Humanities, 32*(2), 209-220.

Fensel, D. (2004). *Ontologies: A Silver Bullets for Knowledge Management and Electronic Commerce* (2nd ed.): Springer-Verlag.

Firat, A., Madnick, S., & Grosof, B. N. (2002). Knowledge Integration to Overcome Ontological Heterogeneity: Challenges from Financial Information Systems. *Twenty-Third International Conference on Information Systems*.

Fonseca, F. T., & Martin, J. (2007). Learning The Differences Between Ontologies and Conceptual Schemas Through Ontology-Driven Information Systems. *Association for Information Systems, 8*(2), 129-142.

Franconi, E., & Kamble, A. (2004).*A DataWarehouse Conceptual Data Model.* Paper presented at the SSDBM.

Friedman, T., & Gassman, B. (2005).Magic Quadrant for Extract, Transformation and Loading. Retrieved September 20, 2007, from http://www.gartner.com/

Galhardas, H., Florescu, D., Shasha, D., & Simon, E. (2000).*Declaratively Cleaning Your Data Using AJAX*. Paper presented at the Journ?es Bases de Donn?es Avanc?es (BDA), Portugal.

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., & Schneider, L. (2002). Sweetening Ontologies with DOLCE. *LECTURE NOTES IN COMPUTER SCIENCE*(2473), 166-181.

Garcia-Molina, H., Labio, W., & Yang, J. (1998).*Expiring Data in a Warehouse.* Paper presented at the 24th VLDB'98, San Mateo, CA.

Gardner, S. P. (2005). Ontologies and Semantic Data Integration.*Drug Discovery Today, 10*(14), 1001-1007.

Gennari, J. H., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubézy, M., Eriksson, H., et al. (2003). The Evolution of Protégé: An Environment for Knowledge-Based Systems Development. *International Journal of Human-Computer Studies, 58*(1), 89-123.

Geroimenko, V. (2004).*Dictionary of XML technologies and the semantic Web*: Springer.

Giorgini, P., Rizzi, S., & Garzetti, M. (2008). GRAnD: A Goal-Oriented Approach to Requirement Analysis in Data Warehouses. *Decision Support Systems, 45*, 4-21.

Giunchiglia, F., & Shvaiko, P. (2004).Semantic Matching.*The Knowledge Engineering Review, Cambridge Univ Press*.

Gogolla, M., Bohling, J., & Richters, M. (2005).Validating UML and OCL models in USE by automatic snapshot generation.*Software and Systems Modeling, 4*(4), 386-398.

Goh, C. H. (1997). *Representing and Reasoning About Semantic Conflicts in Heterogenous Information Systems.* Unpublished PhD, MIT.

Golfarelli, M. (2010).From User Requirements to Conceptual Design in Data Warehouse Design - a Survey.*Data Warehouse Design and Advanced Engineering Applications: Methods for Complex Construction*, 1-16.

Golfarelli, M., Maio, D., & Rizzi, S. (1998). THE DIMENSIONAL FACT MODEL: A CONCEPTUAL MODEL FOR DATA WAREHOUSES. *International Journal of Cooperative Information Systems, 7*(2-3), 215-247.

Graciela, B., Ma. Laura, C., & Omar, C. (2006).*A process for building a domain ontology: an experience in developing a government budgetary ontology*. Paper presented at the Proceedings of the second Australasian workshop on Advances in ontologies - Volume 72.

Gruber, T. R. (1993). A translation approach to portable ontology specifications.*Knowledge Acquisition, 5*(2), 199–220.

Gruber, T. R. (1994). Toward Principles for the Design of Ontologies Used for Knowledge Sharing.*IJHCS, 43*(5/6), 907-928.

Guarino, N. (1998). *Formal Ontology and Information Systems.* Paper presented at the FOIS'98, Trento, Italy.

Guo, M., Li, S., Dong, J., Fu, X., Hu, Y., & Yin, Q. (2003). *Ontology-based Product Data Integration.* Paper presented at the 17th International Conference on Advanced Information Networking and Applications (AINA'03), Xidian University, Xi'an, China.

Haas, L. M., Miller, R. J., Niswonger, B., Roth, M. T., Schwarz, P. M., & Wimmers, E. L. (1999).Transforming Heterogeneous Data with DatabaseMiddleware: Beyond Integration.*Bulletin Data Engineering, 22*(1), 31-36.

Halevy, A. Y. (2005). Why Your Data Won't Mix: Semantic Heterogeneity. *ACM Queue, 3*(8).

Halpin, T. (2001).*Information Modeling and Relational Databases - From Conceptual Analysis to Logical Design*: Morgan Kaufman.

Hammond, M. (2004).*The Fact Gap: The Disconnect Between Data and Decisions*: Business Onjects.

Hansson, S. O. (1994). *Decision Theory - A Brief Introduction*. Royal Institute of Technology (KTH), Stockholm: Uppsala University.

Hatch, M. J., & Cunliffe, A. L. (2006).*Organization theory: modern, symbolic, and postmodern perspectives*: Oxford University Press.

Hellerstein, J. M., Stonebraker, M., & Caccia, R. (1999).Independent, Open Enterprise Data Integration.*Bulletin IEEE Computer Society Technical Committee on Data Engineering*.

Horkoff, J. M. (2012). *Iterative, Interactive Analysis of Agent-Goal Models for Early Requirements Engineering.* Toronto.

Husemann, B., Lechtenborger, J., & Vossen, G. (2000).*Conceptual Data Warehouse Design.* Paper presented at the DMDW, Stockholm, Sweden.

Hutter, D., Stephan, W., Baader, F., Horrocks, I., & Sattler, U. (2005).Description Logics as Ontology Languages for the Semantic Web.In *Mechanizing Mathematical Reasoning* (Vol. 2605, pp. 228-248): Springer Berlin / Heidelberg.

Hwang, M. I., & Xu, H. (2007). The Effect of Implementation Factors on Data Warehouseing Success: An Exploratory Study. *Journal of Information, Information Technology and Organizations, 2*(1).

IEEE.(2004). *SWEBOK - Guide to the Software Engineering Body of Knowledge*. Los Alamitos, CA.

Inmon, W. H. (2002). *Building the Data Warehouse - Third Edition*: John Wiley & Sons, Inc.

Jacky, A., Isabelle, C.-W., & Nicolas, P. (2001).*Dimension hierarchies design from UML generalizations and aggregations.* Paper presented at the Conceptual modeling - ER 2001, Yokohama.

Jarrar, M. (2005).*Towards Methodological Principles for Ontology Engineering.* Unpublished Computer Science, Vrije Universiteit Brussel.

Jasper, R., & Uschold, M. (1999).*A Framework for Understanding and Classifying Ontology Applications.* Paper presented at the IJCAI-99 ontology workshop.

John, T., Lin, P., & James, H. (2002). Representation and reasoning for goals in BDI agents. *Aust. Comput. Sci. Commun., 24*(1), 259-265.

Jureta, I. J., Faulkner, S., & Schobbens, P.-Y. (2007). Achieving, Satisficing, and Excelling. *LECTURE NOTES IN COMPUTER SCIENCE, 4802*, 286-295.

Kaiya, H., & Saeki, M. (2006).*Using Domain Ontology as Domain Knowledge for Requirements Elicitation.* Paper presented at the 14th IEEE Requirements Engineering, Minneapolis/St. Paul, MN.

Karp, P. D., Chaudhri, V. K., & Thomere, J. (2000).*XOL: An XML-based ontology exchange language, Version 0.5, February 17, 2000*.

Kelly, S., & Pohjonen, R. (2009).Worst Practices for Domain-Specific Modeling.*Software, IEEE, 26*(4), 22-29.

Kerremans, K., Temmerman, R., & Tummers, J. (2003).*Representing Multilingual and Culture-Specific Knowledge in a VAT Regulatory Ontology: Support from the Termontography Method.* Paper presented at the OTM 2003 Workshops.

Kim, W., Hong, B.-J., Hong, E.-K., Kim, S.-K., & Lee, D. (2003).*A Taxonomy of Dirty Data.* Paper presented at the Data Mining and Knowledge Discovery.

Kimball, R. (1996). *The Data Warehouse Toolkit - Practical Techniques for Building Dimensional Data Warehouses*: John Wiley & Son.

Kimball, R. (2006). Kimball University: Integration for Real People. Retrieved June 15, 2007, from http://www.intelligententerprise.com/showArticle.jhtml?articleID=190500064

Kimball, R., & Caserta, J. (2004).*The Data Warehouse ETL Toolkit. Practical Technique for Extracting, Cleaning, Conforming and Delivering Data*: Wiley Publishing, Inc., Indianapolis.

Kimball, R., & Ross, M. (2002).*The Data Warehouse Toolkit - The Complete Guide to Dimensional Modeling* (Second ed.): John Wiley and Sons.

Lamsweerde, A. v. (2009). *Requirements Engineering - From System Goals to UML Models to Software Specifications*: John Wiley & Sons Ltd.

Leffingwell, D., & Widrig, D. (2003).*Managing software requirements: a use case approach*: Pearson Education, Inc.

Lenat, D. B. (1995). Cyc: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM, 38*(11).

Lenzerini, M. (2002).*Data Integration: A Theoretical Perspective.* Paper presented at the 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, Madison, Wisconsin.

Leuf, B. (2006). *The Semantic Web - Crafting Infrastructure for Agency*: John Wiley & Son, Ltd.

Levy, A. Y. (1999).*Logic-Based Techniques In Data Integration.* Paper presented at the Workshop on Logic-Based Artificial Intelligence, Washington, DC.

Lujan-Mora, S. (2005).*Data Warehouse Design With UML.*Unpublished PhD, University of Alicante.

Lujan-Mora, S., Trujillo, J., & Song, I.-Y.(2006). A UML Profile for Multidimensional Modeling in Data Warehouse.*Data & Knowledge Engineering, 59*(3), 725 - 769.

Maedche, A., Staab, S., Studer, R., Sure, Y., & Volz, R. (2002).SEAL - Tying Up Information Integration and Web Site Management by Ontologies.*Buletin of The IEEE*.

Mazon, J.-N., Pardillo, J., & Trujillo, J. (2007).A Model-Driven Goal-Oriented Requirement Engineering Approach for Data Warehouses.*LECTURE NOTES IN COMPUTER SCIENCE*(4802), 255–264.

Mazon, J.-N., Trujillo, J., Serrano, M., & Piattini, M. (2005).*Designing Data Warehouse: From Business Requirement Analysis to Multidimensional Modeling.* Paper presented at the REBNITA.

Meersman, R. (2001). *Ontologies and Databases: More than a Fleeting Resemblance.* Paper presented at the OES/SEO Workshop, Rome.

MoHE.(2010). *Dasar Pembangunan Keusahawanan Institusi Pengajian Tinggi*.Retrieved.from http://www.mohe.gov.my/portal/pelajar/program-keusahawanan.html.

Moss, L. (2005).Ten Mistakes to Avoid for Data Warehouse Project Managers.*TDWI's Best of Business Intelligence, 3,* 16-23.

Mouratidis, H., Giorgini, P., Barley, M., Mouratidis, H., Unruh, A., Spears, D., et al. (2009). Enhancing Secure Tropos to Effectively Deal with Security Requirements in the Development of Multiagent Systems

Safety and Security in Multiagent Systems. In (Vol. 4324, pp. 8-26): Springer Berlin / Heidelberg.

Mumford, E. (2000). A Socio-Technical Approach to Systems Design.*Requirement Engineering, 2000*(5), 125-133.

Mumford, E. (2003). *Redesigning Human Systems*: IRM Press.

Niedrite, L., Solodovnikova, D., Treimanis, M., & Niedritis, A. (2007, February 16-19). *Goal-Driven Design of a Data Warehouse-Based Business Process Analysis System* Paper presented at the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Corfu Island, Greece.

Nielsen, J. (1997). The use and misuse of focus groups.*Software, IEEE, 14*(1), 94-95.

Nimmagadda, S. L., Dreher, H., & Rudra, A. (2005).*Ontology of Western Australian Petroleum Data for Effective Data Warehouse Design and Data Mining.* Paper presented at the 3rd IEEE International Conference on Industrial Informatics (INDIN).

Noy, N. F., & McGuinness, D. L. (2001).*Ontology development 101: A guide to creating your first ontology* (No. SMI-2001-0880): Stanford Medical Informatics.

Noy, N. F., & Musen, M. A. (2000).*PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment.* Paper presented at the AAAI'00.

Noy, N. F., Sintek, M., Decker, S., Crubézy, M., Fergerson, R. W., & Musen, M. A. (2001).Creating Semantic Web Contents with Protege-2000.*IEEE INTELLIGENT SYSTEMS AND THEIR APPLICATIONS, 16*(2), 60-71.

Nuseibeh, B., & Easterbrook, S. (2000). *Requirements Engineering: A Roadmap.* Paper presented at the The Future of Software Engineering, Limerick, Ireland.

Nwana, H. S., & Ndumu, D. T. (1999).A Perspective on Software Agents Research.*KNOWLEDGE ENGINEERING REVIEW, 14*(2), 125-142.

Ogawa, H., Kumeno, F., & Honiden, S. (2008).*Model Checking Process with Goal Oriented Requirements Analysis.* Paper presented at the 15th Asia-Pacific Software Engineering Conference.

Olivé, A. (2007). *Conceptual Modeling of Information System*: Springer-Verlag Berlin Heidelberg.

OMG.(2003). *Common Warehouse Metamodel (CWM) Specification*.

OMG.(2007). *Ontology Definition Metamodel* (No. ptc/2007-09-09).

Papastefanatos, G., Vassiliadis, P., Simitsis, A., & Vassiliou, Y. (2009). Policy-regulated Management of ETL Evolution. *Springer Journal on Data Semantics (JoDS XIII)*(5530), 146-176.

Parviainen, P., Tihinen, M., Lormans, M., & Solingen, R. V. (2005). Requirement Engineering: Dealing with the Complexity of Sociotechnical Systems Development *Requirement Engineering for Sociotechnical Systems*, 1-20.

Patel-Schneider, P. F., & Fensel, D. (2002).*Layering the Semantic Web: Problems and Directions.* Paper presented at the First International Semantic Web Conference (ISWC2002), Sardinia, Italy.

Patil, P. S., Rao, S., & Patil, S. B. (2011). Data Extraction, Transformation and Loading.*International Journal of Computer Science and Application*, 5.

Ponniah, P. (2007). *Data Modeling Fundamentals - A Practical Guide for IT Professionals*: John Wiley & Sons.

Prakash, N., & Gosain, A. (2008).An approach to engineering the requirements of data warehouses.*Requirements Engineering, 13*(1), 49-72.

Priebe, T., & Pernul, G. (2003).*Ontology-based Integration of OLAP and Information Retrieval.* Paper presented at the 14th International Workshop on Database and Expert System Applications (DEXA'03).

Rahm, E., & Bernstein, P. A. (2001).A Survey of Approaches to Automatic Schema Matching.*VLDB Journal, 10*, 334-350.

Rahm, E., & Do, H. H. (2000).*Data Cleaning: Problems and Current Approaches*.

Raman, V., & Hellerstein, J. M. (2001).*Potter's Wheel: An Interactive Data Cleaning System.* Paper presented at the 27th VLDB, Roma, Italy.

Rizzi, S. (2007).Conceptual Modeling Solutions for the Data Warehouse.*Idea Group Inc.*, 1-26.

Rizzi, S., Abello, A., Lechtenborger, J., & Trujillo, J. (2006).*Research in Data Warehouse Modeling and Design: Dead or Alive?* Paper presented at the DOLAP'06, Arlington, Virginia, USA.

Romero, O., & Abelló, A. (2007).*Automating Multidimensional Design from Ontologies.* Paper presented at the DOLAP'07, Lisboa, Portugal.

Ropohl, G. (1999). Philosophy Of Socio-Technical Systems. *Society for Philosophy and Technology, 4*(3).

Rudin, K., & Cressy, D. (2003). Will the Real Analytic Application Please Stand Up? Retrieved January 20, 2008, from http://www.dmreview.com/issues/20030301/6427-1.html

Rundensteiner, E. A., Koeller, A., & Zhang, X. (2000). Maintaining Data Warehouses Over Changing Information Sources. *Communication of ACM, 43*(6), 57-62.

Sane, S. S., & Shirke, A. (2009).*Generating OWL Ontologies from a Relational Databases for the Semantic Web.* Paper presented at the ICAC3 '09, Mumbai, India.

Sapia, C., Blaschka, M., Hofling, G., & Dinter, B. (1998).*Extending the E/R model for the multidimensional paradigm.* Paper presented at the ER Workshop on Data Warehouse and Data Mining.

Schreiber, Z. (2003). Semantic Information Architecture: Creating Value by Understanding Data. *DM Review*.

Schreiber, Z. (2004). *Semantics: Delivering One Language to The Enterprise.* Paper presented at the 2nd Semantic Technologies for eGov.

Schreiber, Z., & Gonchar, I. (2004).Industry Models for Semantic Information Management. Retrieved September 20, 2007, from http://www.dmreview.com/

Sell, D., Cabral, L., Motta, E., Domingue, J., & Pacheco, R. (2005, August 22). *Adding Semantics to Business Intelligence.* Paper presented at the Database & Expert Systems Application 2005 - 16th International Workshop.

Sen, A., & Sinha, A. P. (2007).Toward Developing Data Warehousing Process Standards: An Ontology-Based Review of Existing Methodology.*Transactions on Systems, Man, and Cybernetics, 37*(1), 17-31.

Shen, G., Huang, Z., Zhu, X., & Zhao, X. (2006).*Research on the Rules of Mapping from Relational Model to OWL.* Paper presented at the OWLED'06, Athens, Georgia (USA).

Shibaoka, M., Kaiya, H., & Saeki, M. (2007). GOORE: Goal-Oriented and Ontology Driven Requirements Elicitation Method. *ER Workshops (LNCS), 4802,* 225-234.

Simitsis, A. (2004). *Modeling and Optimization of Extraction-Transformation-Loading (ETL) Processes in Data Warehouse Environments.*Unpublished PhD, National Technical University of Athens, Athens.

Simon, H. A. (1996). *The sciences of the artificial*: MIT Press.

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). Pellet: A Practical OWL-DL Reasoner *Web Semantic, 5*(2), 51-53.

Skoutas, D., & Simitsis, A. (2006).*Designing ETL Processes Using Semantic Web Technologies.* Paper presented at the DOLAP'06, Arlington, Virginia, USA.

Skoutas, D., & Simitsis, A. (2007). Ontology-Based Conceptual Design of ETL Processes for Both Structured and Semi-Structured Data. *Semantic Web & Information Systems, 3*(4), 1-24.

Smith, B. (2003). Ontology.*Blackwell Guide to the Philosophy of Computing and Information*, 155-166.

Sobreperez, P. (2008). Using plenary focus groups in information systems research : more than a collection of interviews. *Electronic Journal of Business Research Methods, 6*(2), 181-188.

Sommerville, I. (2007). *Software Engineering* (Eighth ed.): Addison-Wesley.

Spyns, P., Meersman, R., & Jarrar, M. (2002).Data Modeling Versus Ontology Engineering.*ACM SIGMOD.*

Stefanov, V., & List, B. (2005).*Bridging the Gap between Data Warehouses and Business Processes: A Business Intelligence Perspective for Event-Driven Process Chains.* Paper presented at the 9th IEEE International EDOC Enterprise Computing (EDOC'05), Enschede, The Netherlands.

Stefanov, V., & List, B. (2007).*A UML Profile for Modeling Data Warehouse Usage.* Paper presented at the ER 2007 Workshops CMLSA, Auckland, New Zealand.

Storey, V. C. (1993).Understanding Semantic Relationships.*VLDB Journal, 2*, 455-488.

Stylianou, A. C., & Kuman, R. L. (2000). An Integrative Framework for IS Quality Management. *Communications of the ACM, 43*(9), 99-104.

Sumathi, S., & Esakkirajan, S. (2007).*Fundamentals of Relational Database Management Systems*: Springer-Verlag Berlin Heidelberg.

Sung, S., & McLeod, D. (2006). *Ontology-Driven Semantic Matches Between Database Schemas.* Paper presented at the 22nd Int'l Conference on Data Engineering.

Sure, Y., Angele, J., & Staab, S. (2002). Guiding Ontology Development by Methodology.*Inferencing, 31*(4), 18-23.

Ta'a, A., Abdullah, M. S., & Norwawi, N. M. (2008).*Ontology-Based Extraction-Transformation-Loading (ETL) Processes Model in Data Warehouse Environments.* Paper presented at the CAMP'08, UPNM, Kuala Lumpur.

Ta'a, A., Abdullah, M. S., & Norwawi, N. M. (2010). RAMEPs: A Goal-Ontology Approach To Analyse The Requirements For Data Warehouse Systems. *WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS, 7*(2), 295-309.

Ta'a, A., Bakar, M. S. A., & Saleh, A. R. (2008).*ACADEMIC BUSINESS INTELLIGENCE SYSTEM DEVELOPMENT USING SAS TOOLS.* Paper presented at the SAS Global Forum, San Antonio, Texas, USA.

Tang, Y., & Meersman, R. (2005).*Judicial Support Systems: Ideas for a privacy Ontology-Based Case Analyzer.* Paper presented at the OTM Workshops 2005, LNCS 3762.

Thayer, R. H., & Dorfman, M. (1990).*System and Software Requirements Engineering*. Los Alamitos, CA: IEEE Computer Society Press.

Tieniu, W., Jianhua, H., Haihe, Z., Yinglin, W., & Tianrui, L. (2011).Design and Implementation of an ETL Approach in Business Intelligence Project. In *Practical Applications of Intelligent Systems* (Vol. 124, pp. 281-286): Springer Berlin / Heidelberg.

Toivonen, S., & Niemi, T. (2004).*Describing Data Sources Semantically for Facilitating Efficient Creation of OLAP Cubes.* Paper presented at the 3rd International Semantic Web, Hiroshima, Japan.

Ullman, J. D. (2000). Information Integration Using Logical Views.*Theoretical Computer Science, 239*(2), 189-210.

Uschold, M., King, M., Moralee, S., & Zorgios, Y. (1998). The Enterprise Ontology.*Knowledge Engineering Review, 13*, 71–88.

Vassiliadis, P. (2000). *Data Warehouse Modeling and Quality Issues.*Unpublished PhD, National Technical University of Athens.

Vassiliadis, P., Simitsis, A., Georgantas, P., & Terrovitis, M. (2003).A Framework for the Design of ETL Scenarios.*LECTURE NOTES IN COMPUTER SCIENCE*(2681), 520-535.

Wache, H., Vogele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., et al. (2001).*Ontology-Based Integration of Information - A Survey of Existing Approaches.* Paper presented at the IJACAI-01.

Walker, G. H., Stanton, N. A., Salmon, P. M., & Jenkins, D. P. (2008). A review of sociotechnical systems theory: a classic concept for new command and control paradigms. *Theoretical Issues in Ergonomics Science, 9*(6), 479-499.

Wand, Y., & Wang, R. Y. (1996).Anchoring Data Quality Dimensions in Ontological Foundations.*Communications of the ACM, 39*(11), 86-95.

Wang, T.-W., & Murphy, K. E. (2006). Semantic Integration in Multidatabase Systems: How Much Can We Integrate? *Advanced Topics in Database Research, 3*(xxi), 420-439.

White, C. (2006). *The Next Generation of Business Intelligence: Operational BI* (Sponsored by Sybase): BI Research.

Winter, R., & Strauch, B. (2004).*Information Requirements Engineering for Data Warehouse Systems.* Paper presented at the ACM Symposium on Applied Computing.

Yu, E. (1995). *Modeling Strategic Relationships for Process Reengineering.*Unpublished Ph. D thesis, University of Toronto.

Yu, E., & Cysneiros, L. M. (2002).Agent-Oriented Methodologies - Towards a Challenge Exemplar.

Yu, E., Giorgini, P., Maiden, N., & Mylopoulos, J. (2011).*Social Modeling for Requirements Engineering*: The MIT Press.

Zeleznikow, J., & Stranieri, A. (2001).*An Ontology for the Construction of Legal Decision Support Systems.* Paper presented at the 2nd International Workshop on Legal Ontologies.

Zhao, G., Gao, Y., & Meersman, R. (2004). *An Ontology Based Approach To Business Modelling.* Paper presented at the International Conference of Knowledge Engineering & Decision Support (ICKEDS'04).

Zhuolun, Z., & Sufen, W. (2008, 12-14 Oct. 2008). *A Framework Model Study for Ontology-Driven ETL Processes.* Paper presented at the Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08.4th International Conference on.

Ziegler, P., & Dittrich, K. R. (2004).*Three Decades of Data Integration - All Problems Solved.* Paper presented at the 1st Int'l IFIP on Semantics of a Network World.

Appendix A– Case Study forStudent Affairin University

(i) Organizational Modeling

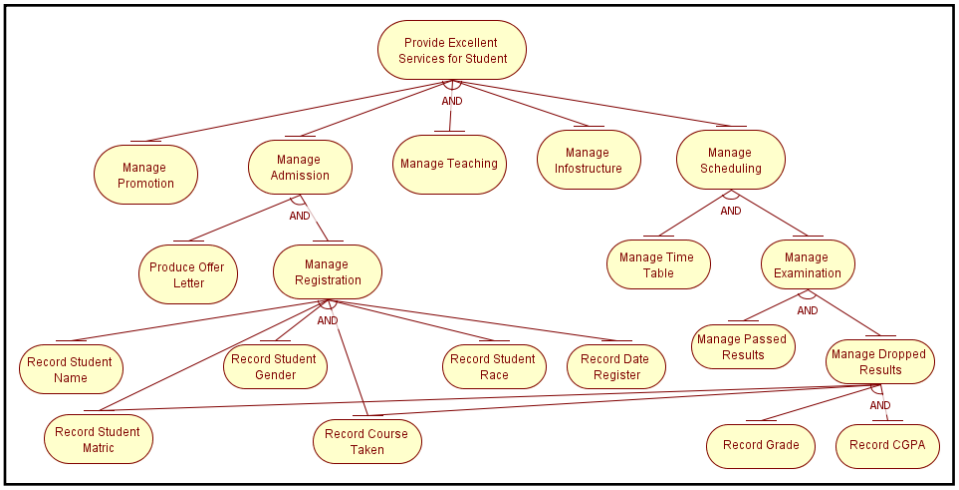


Figure A1.1: Goal Diagram for Student Affair Department

(ii) Fact and Attribute Analysis in Organizational Modeling

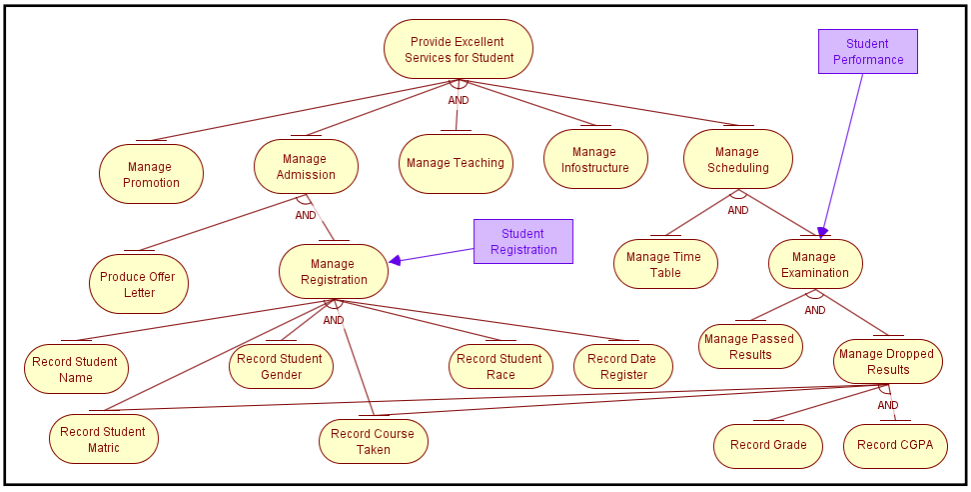


Figure A1.2: Extended goal diagram from the organizational perspectives

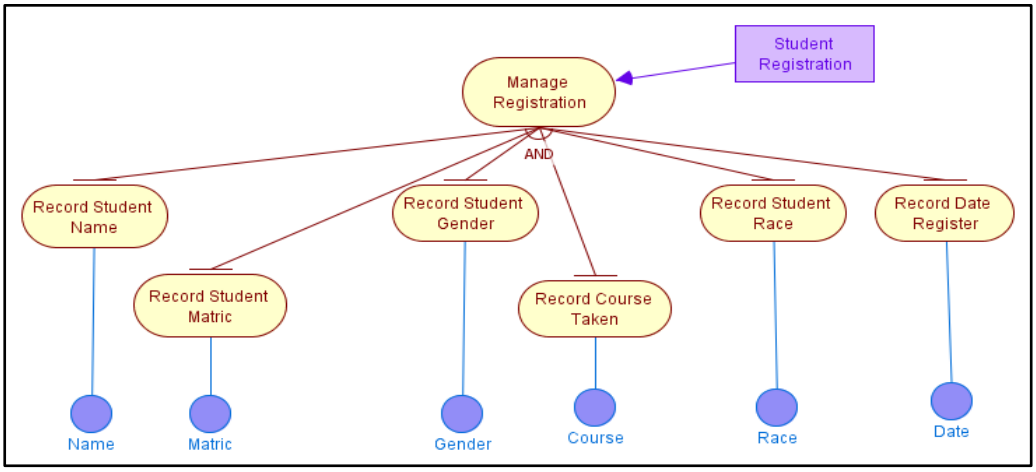


Figure A1.3: Extended goal diagram with attributes (Student Registration)

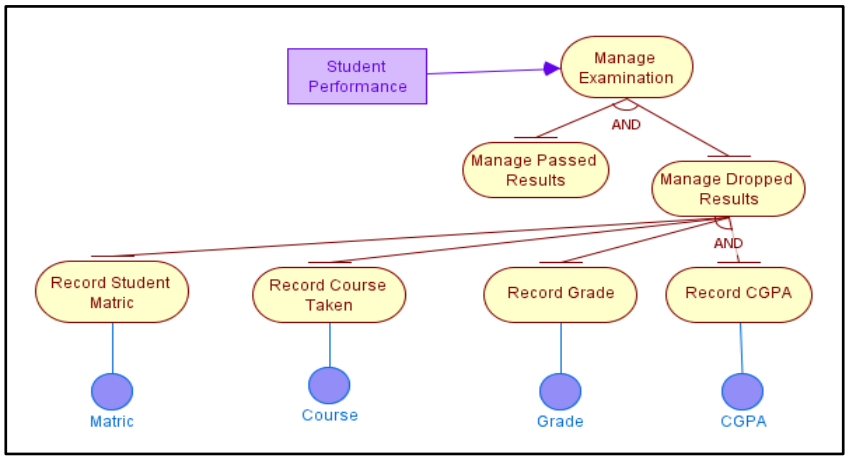


Figure A1.4: Extended goal diagram with attributes (Student Performance)

(iii) Goal Analysis in Decisional Modeling

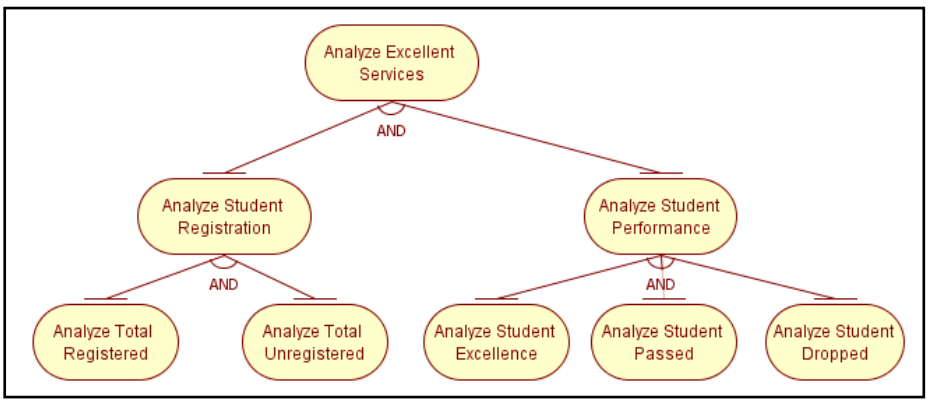


Figure A1.5: Goal diagram from the decision maker perspectives

(iv) Fact, Dimension, and Measure Analysis in Decisional Modeling

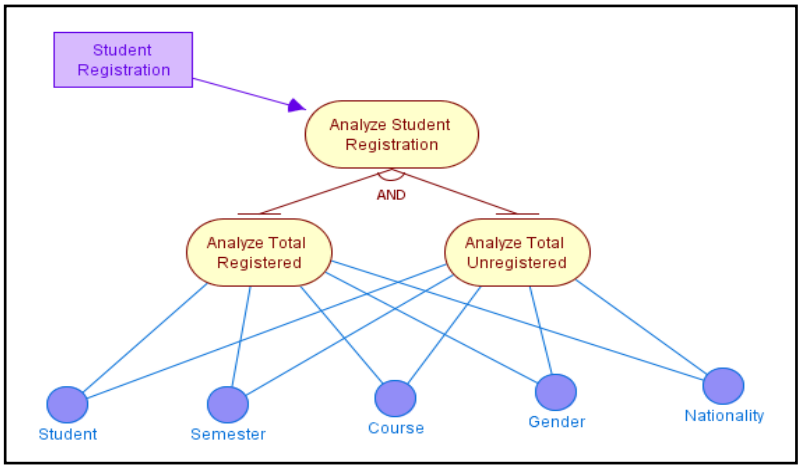


Figure A1.6: Extended goal diagram for Student Registration

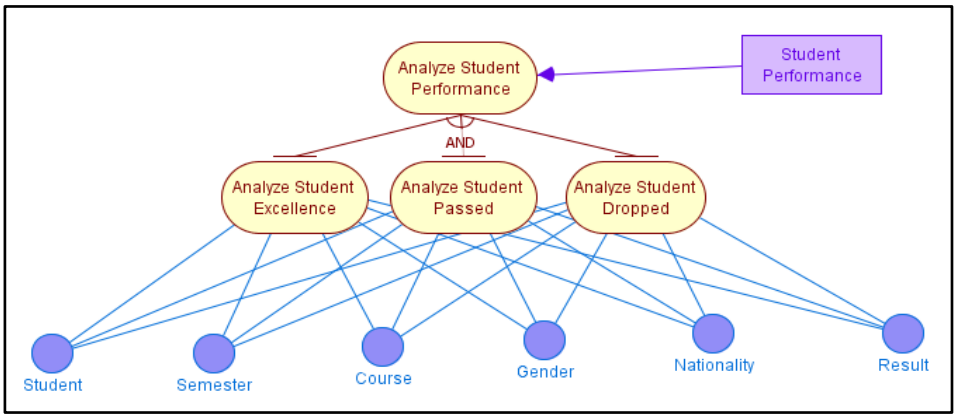


Figure A1.7: Extended goal diagram for Student Performance

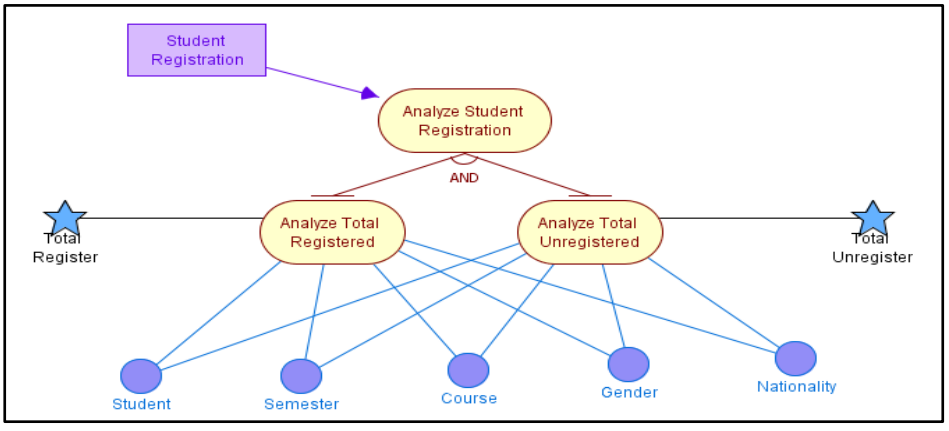


Figure A1.8: Extended goal diagram with measure for Student Registration

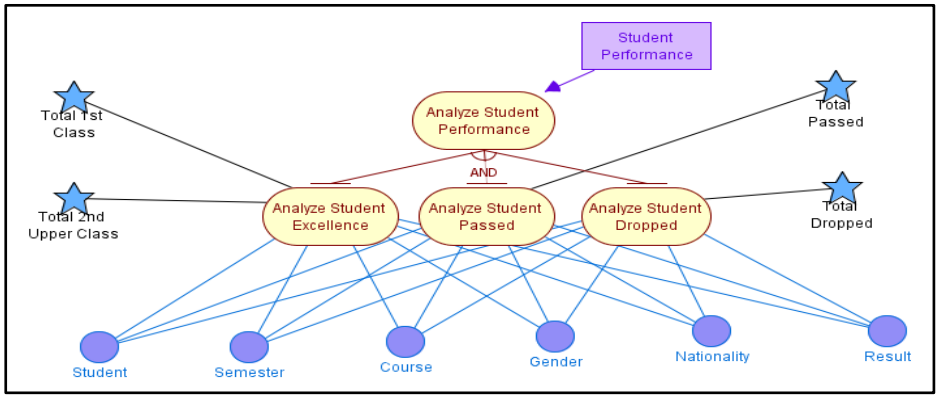


Figure A1.9: Extended goal diagram with measure for Student Performance

(v) Ontology Model – The DWROand DSO

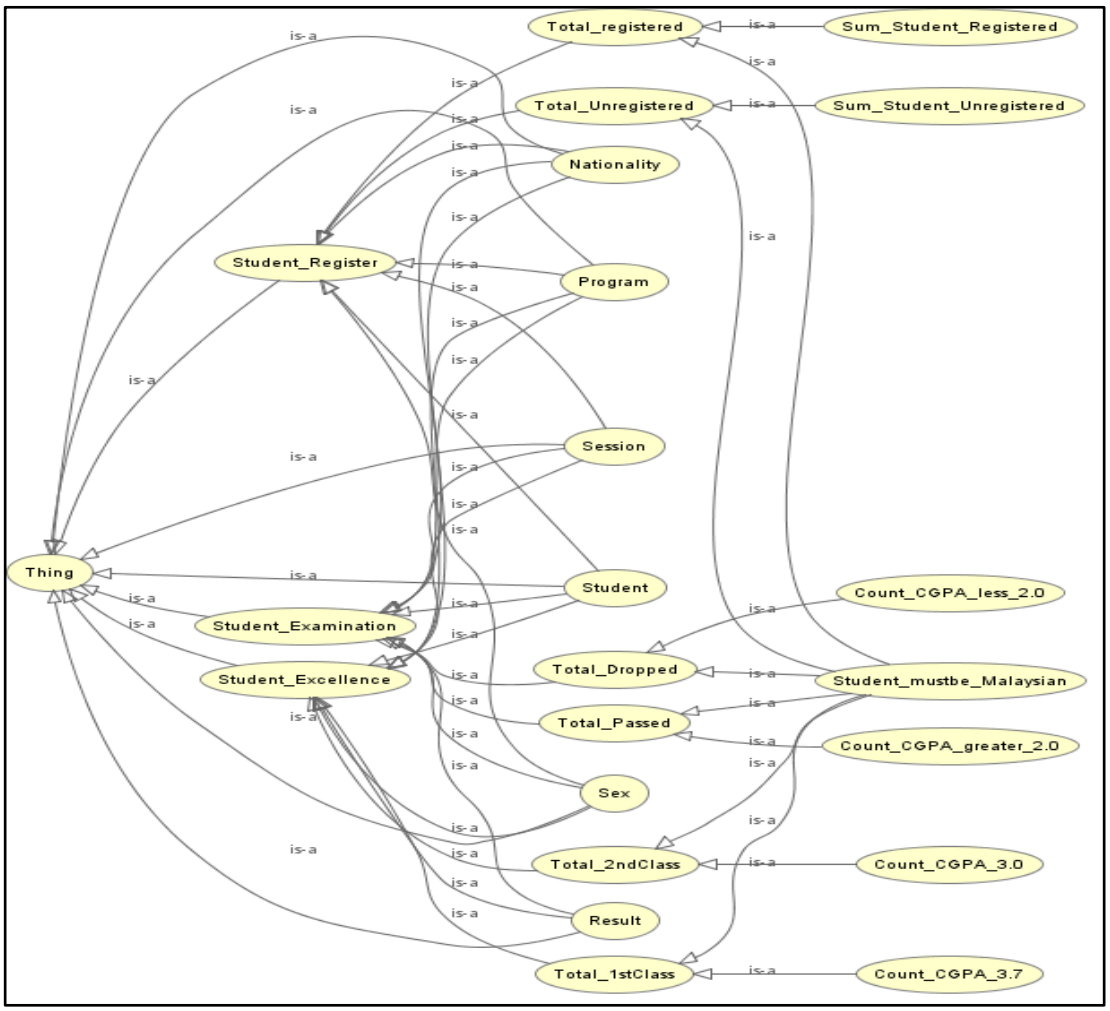


Figure A1.10:The DWRO for Student Affair

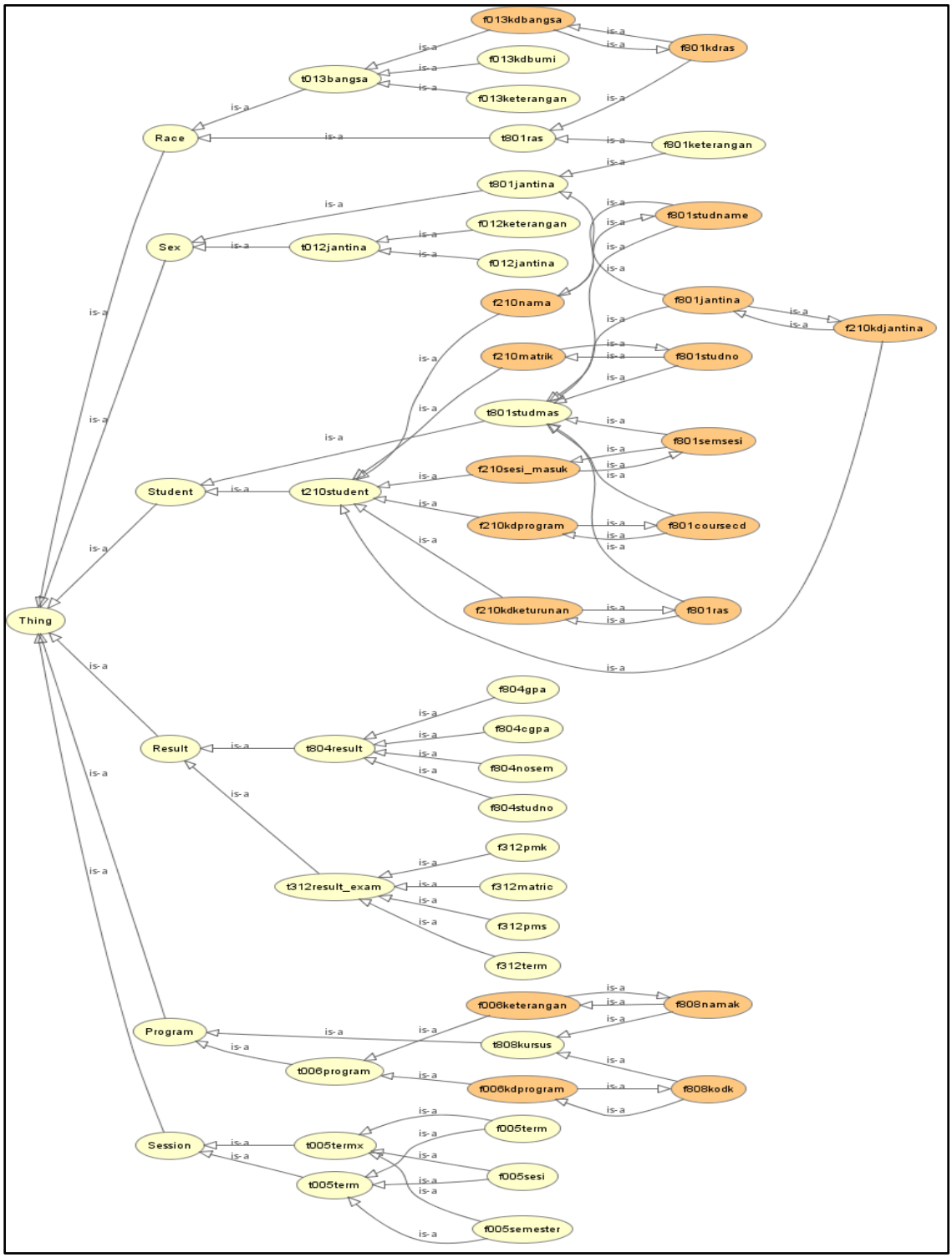


Figure A1.11:The DSO for Student Affair

Appendix B–Case Study forBilling Utilityin Gas Malaysia

(i) Organizational Modeling

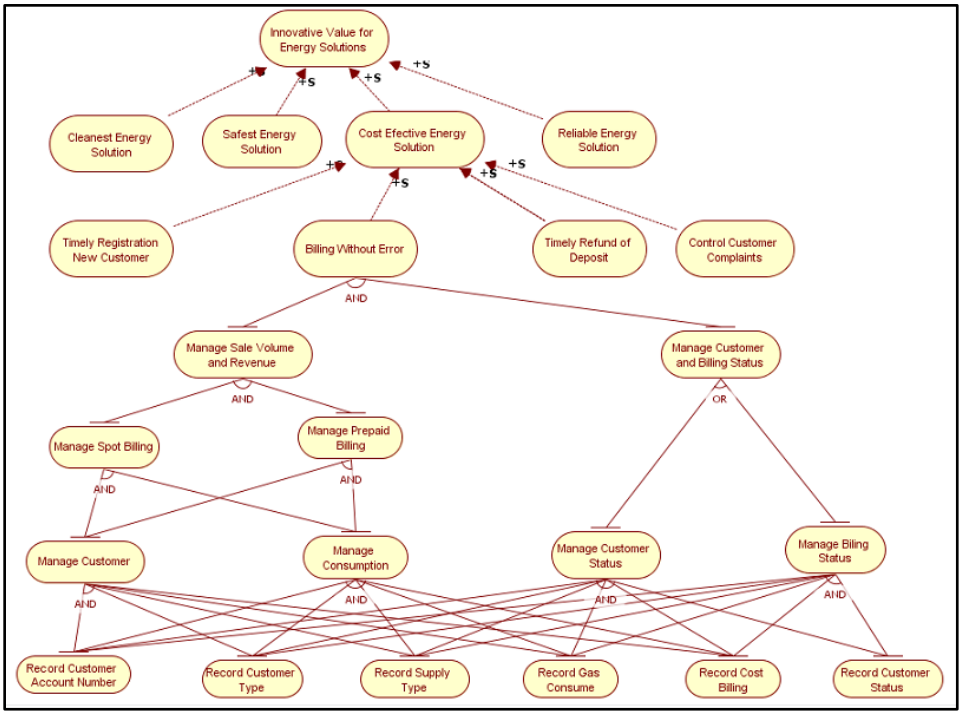


Figure B2.1: Rationale Goal Diagram for Gas Malaysia

(ii) Fact and Attribute Analysis

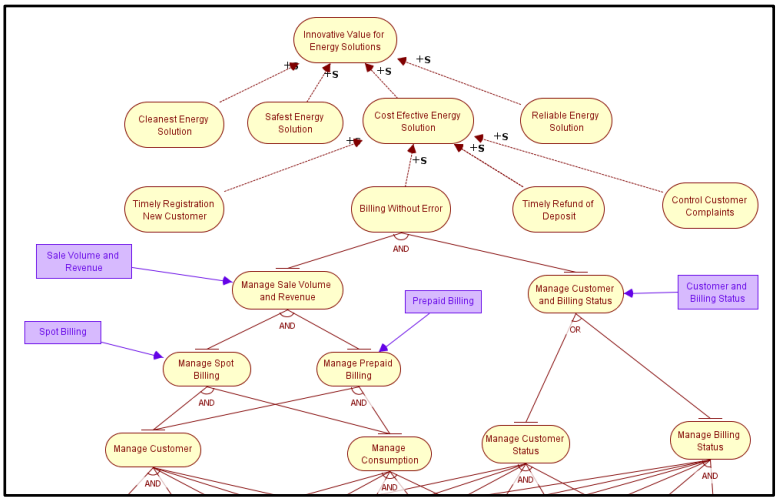


Figure B2.2: Goal Diagram with Facts for Gas Malaysia

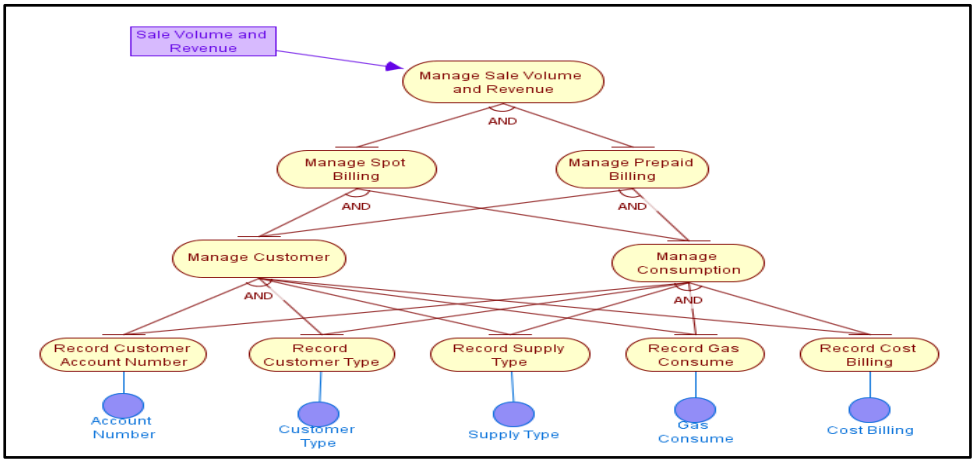


Figure B2.3: Attributes Analysis Diagram for Sale Volume and Revenue Fact

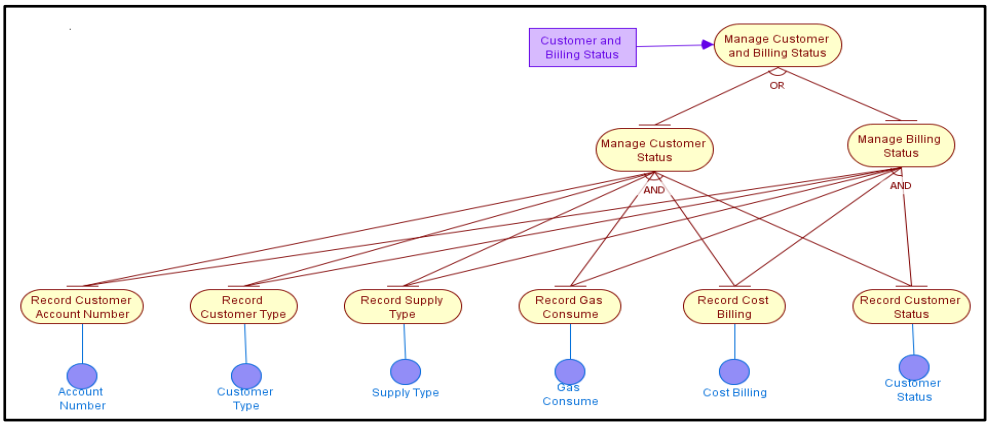


Figure B2.4: Attributes Analysis Diagram for Customer and Billing Status Fact

(iii) Decisional Modeling - Goal Analysis

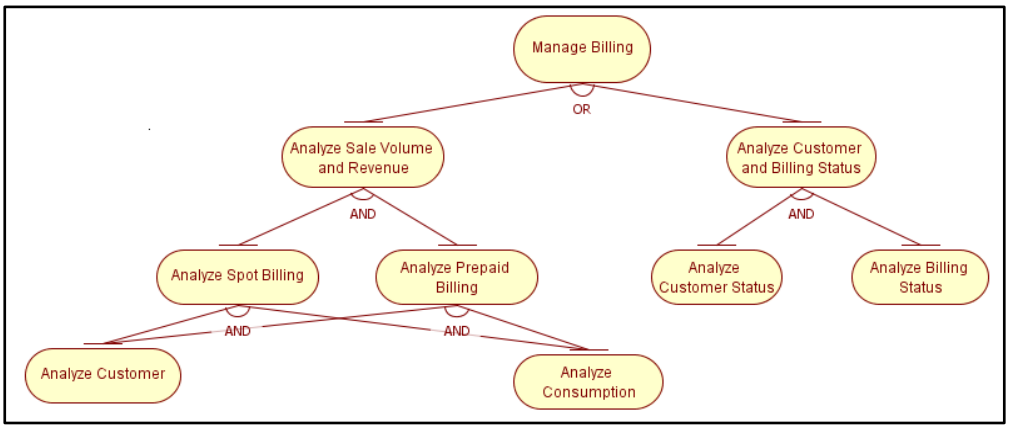


Figure B2.5: Goal Diagram for Billing Manager

(iv) Fact, Dimension and Measure Analysis

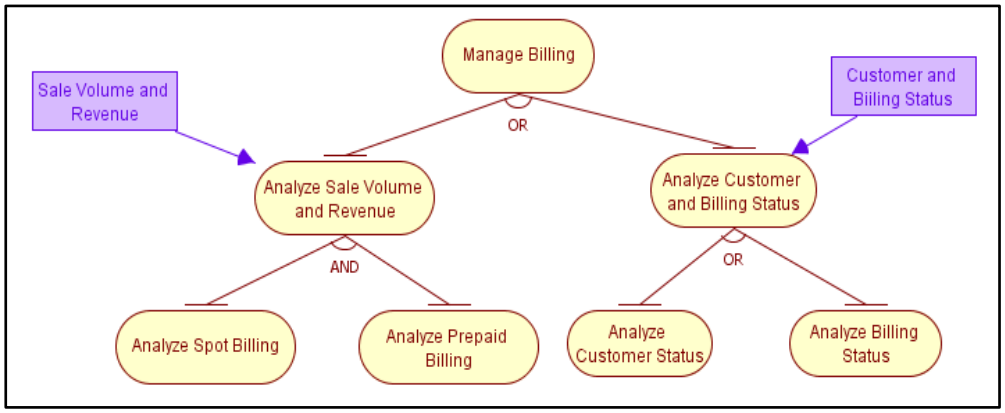


Figure B2.6: Extended Goal Diagram for BM with Facts

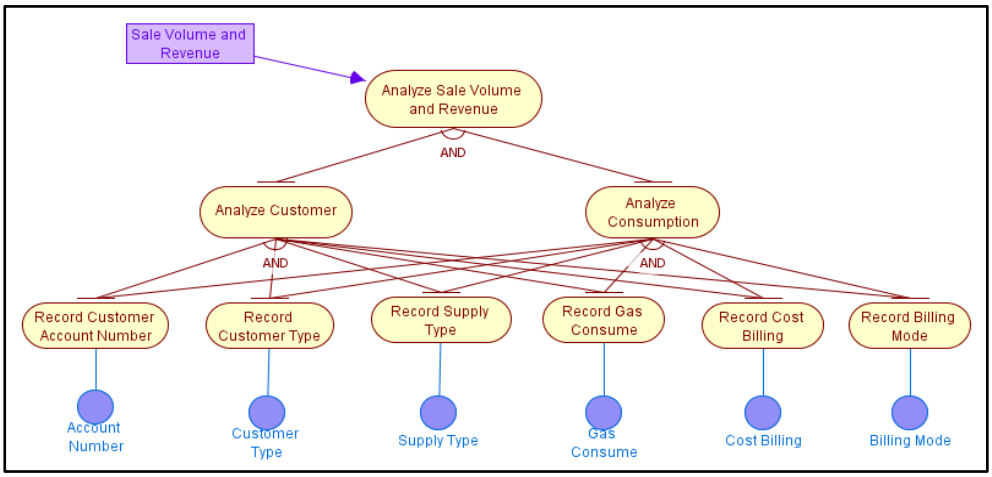


Figure B2.7: Goal Diagram for Sale Volume and Revenue with Dimension

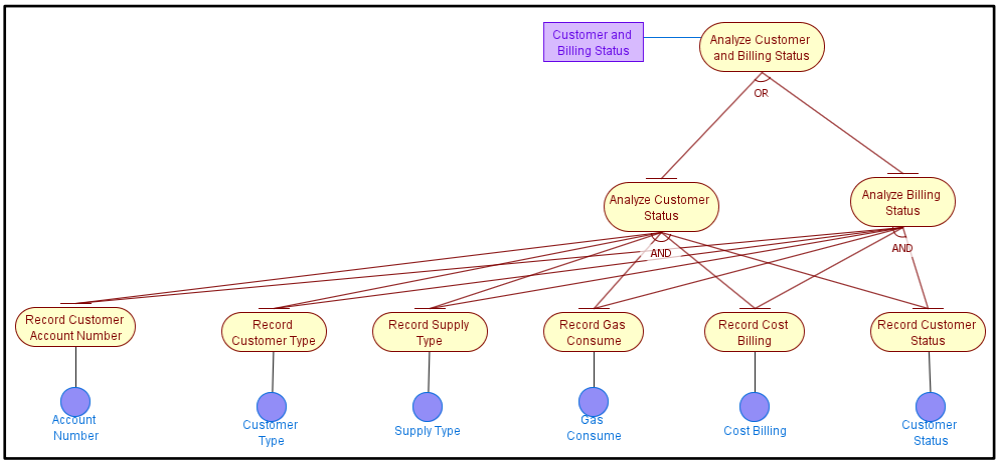


Figure B2.8: Goal Diagram for Customer and Billing Status with Dimension

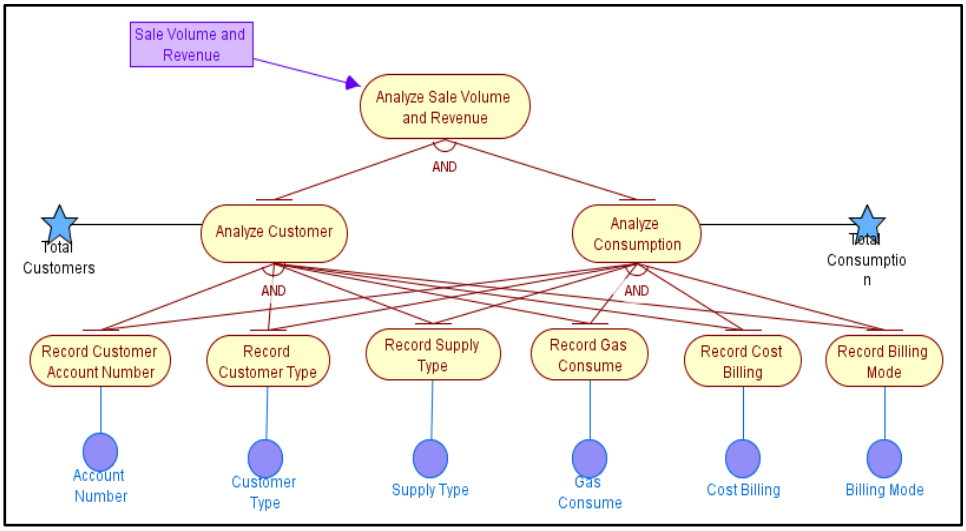


Figure B2.9: Goal Diagram for Sale Volume and Revenue with Measures

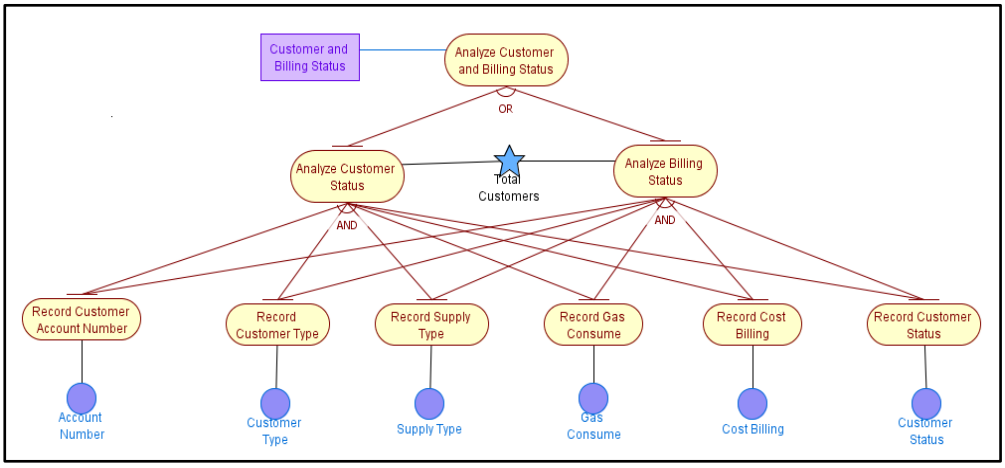


Figure B2.10: Goal Diagram for Customer and Billing Status with Measures

(v) Ontology Modeling–The DWRO and DSO

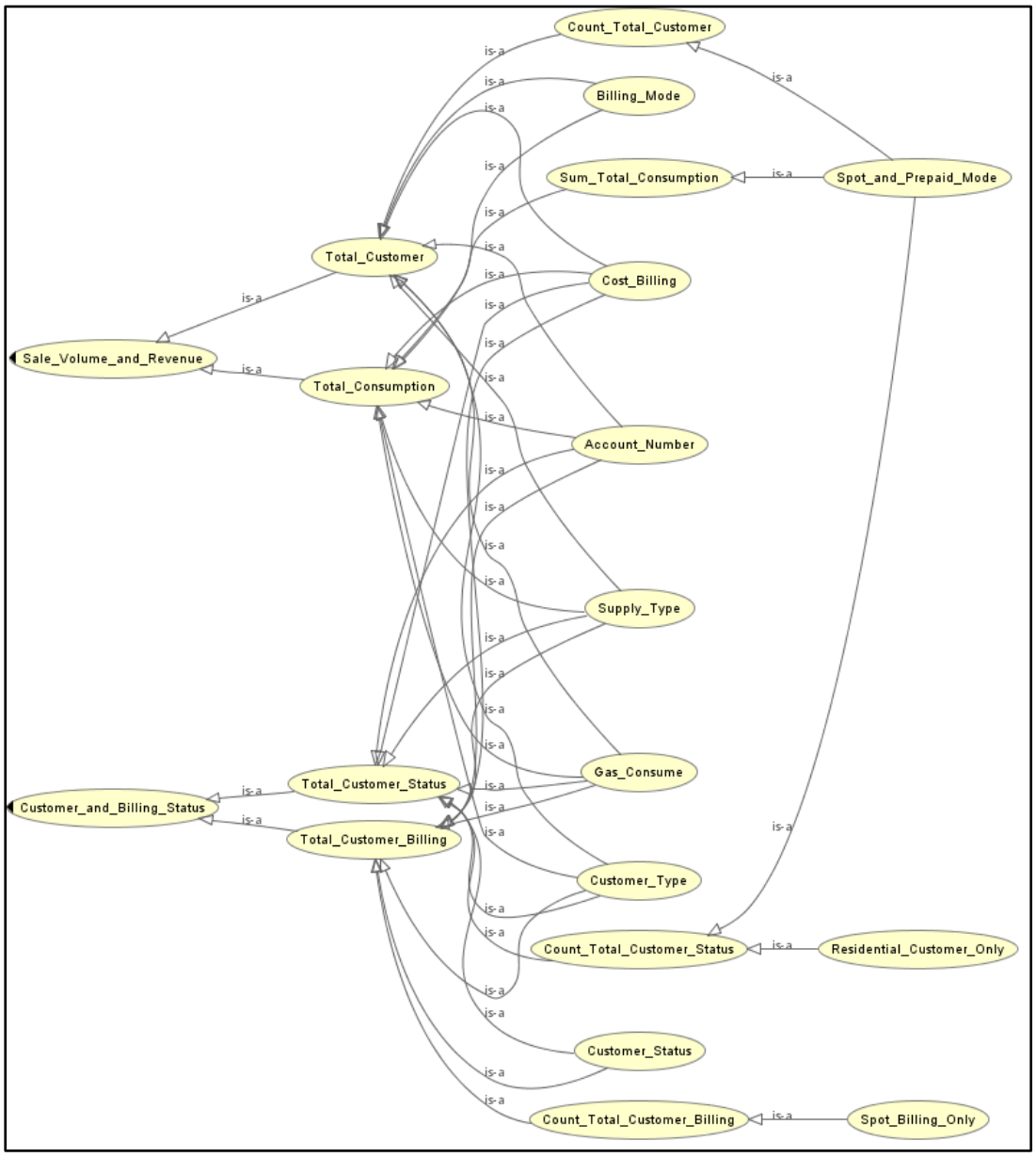


Figure B2.11: The DWRO for Billing Utility

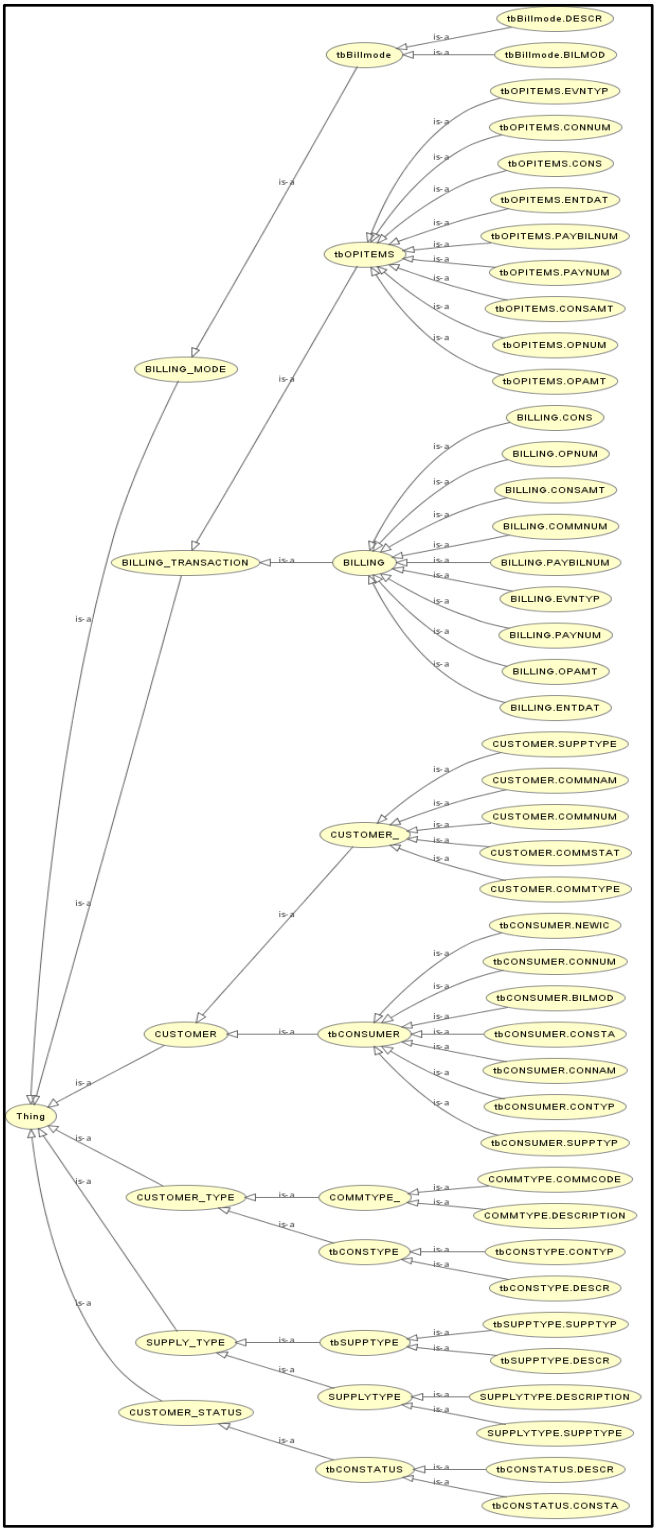


Figure B2.12: The DSO for Billing Utility

Appendix C–Case Study for Student Entrepreneur in MoHE

(i) Organizational Modeling

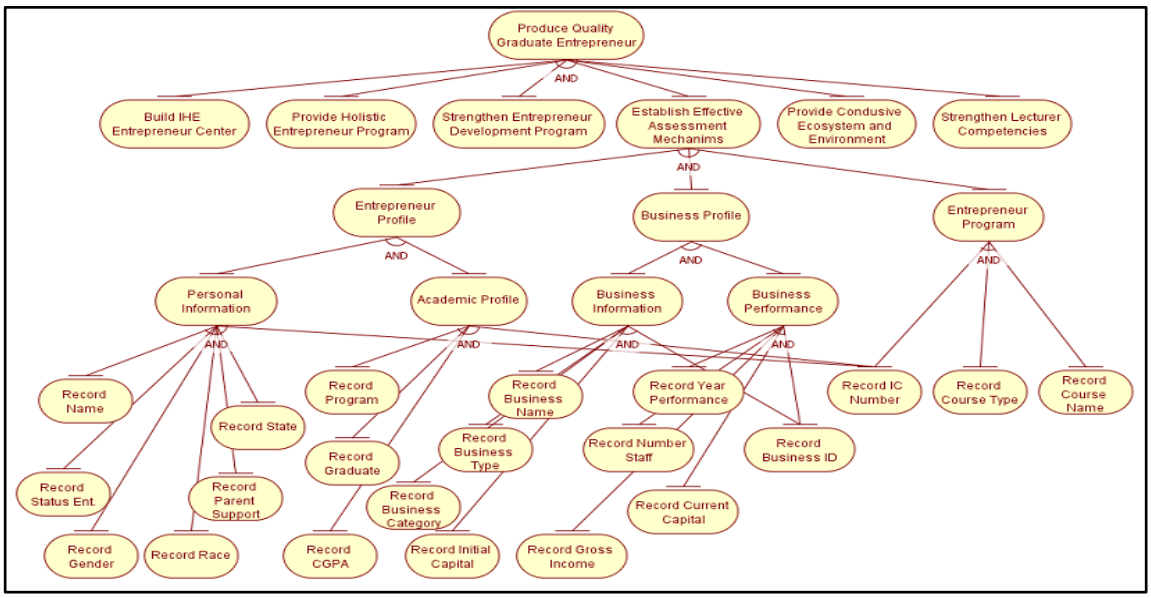


Figure C3.1: Goal Diagram for EU in Organizational Perspective

(ii) Fact and Attribute Analysis

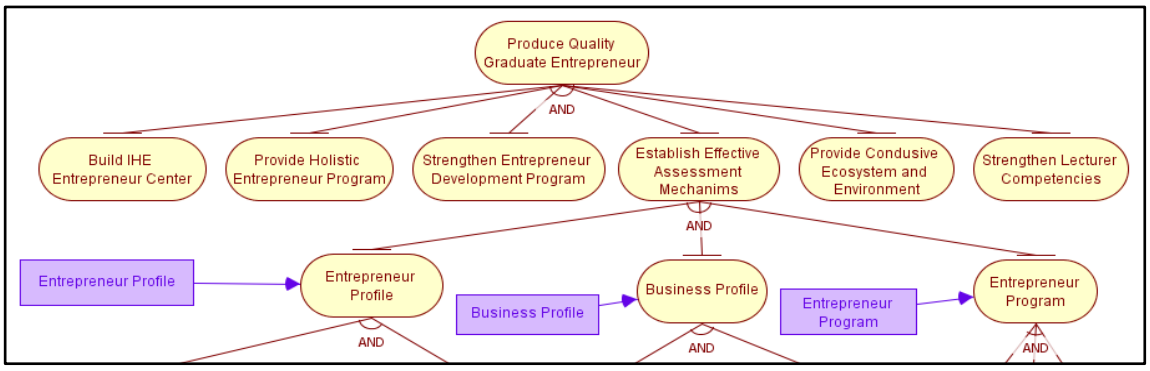


Figure C3.2: Extended Goal Diagram with Facts

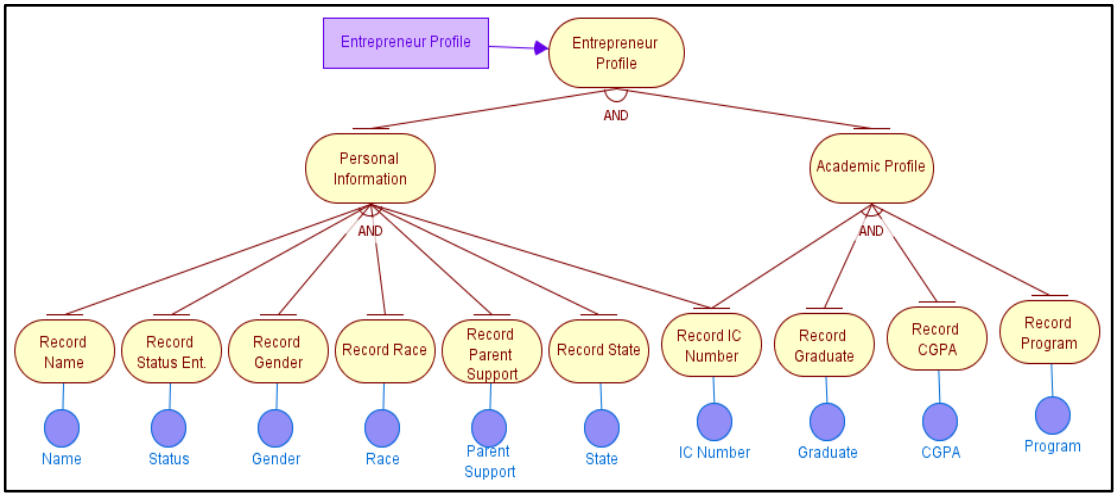


Figure C3.3: Goal Diagram with Attributes for Entrepreneur Profile

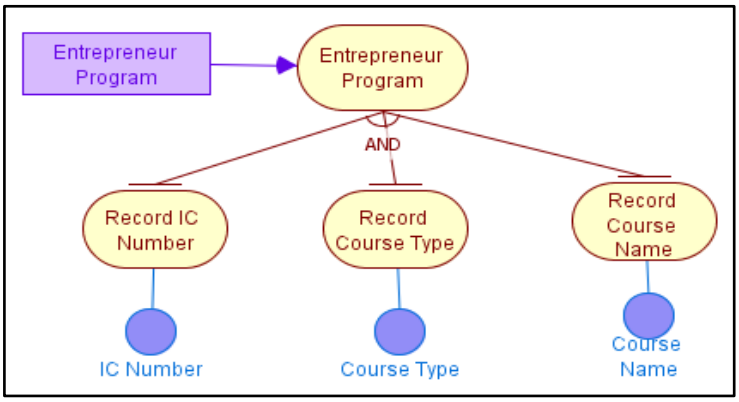


Figure C3.4: Goal Diagram with Attributes for Entrepreneur Program

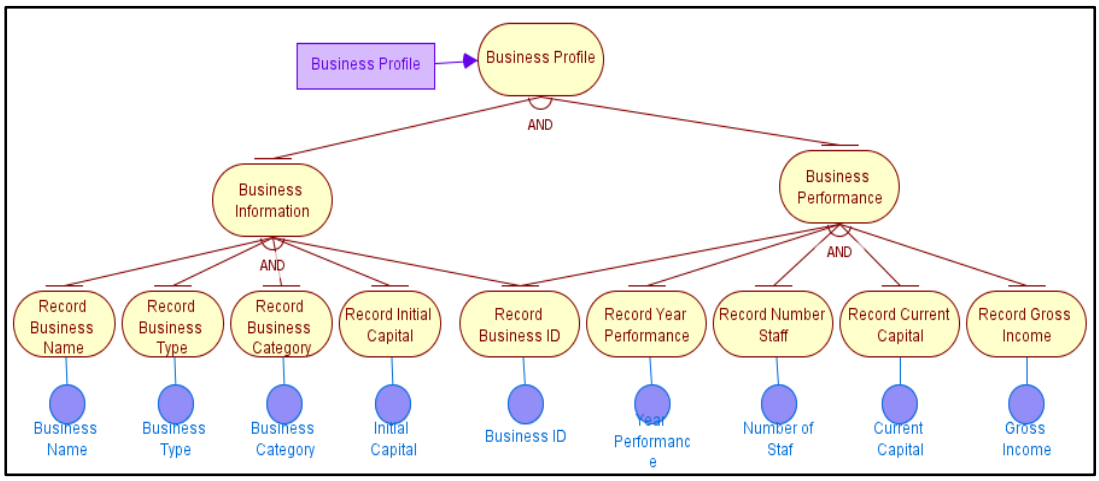


Figure C3.5: Goal Diagram with Attributes for Business Profile

(iii) Decisional Modeling

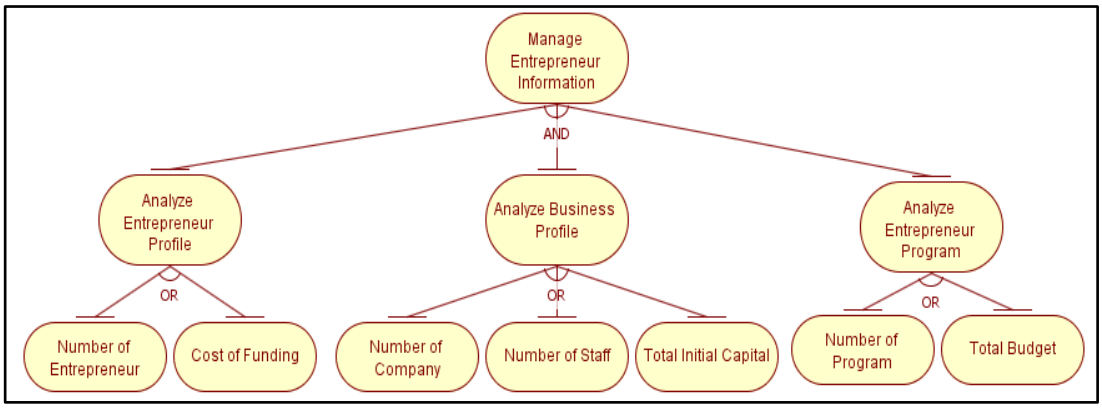


Figure C3.6: Goal diagram for EU

(iv) Fact, Dimension, Measure Analysis

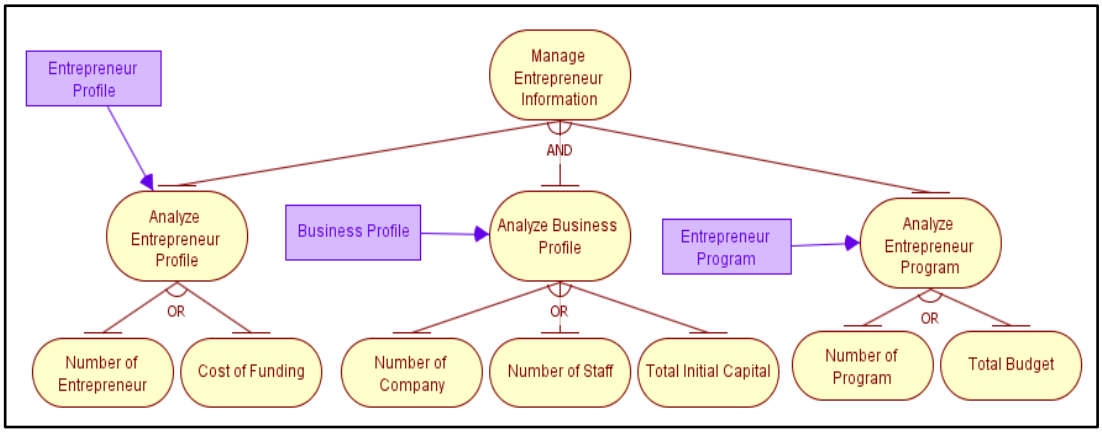


Figure C3.7: Extended Goal Diagram for EU with Facts

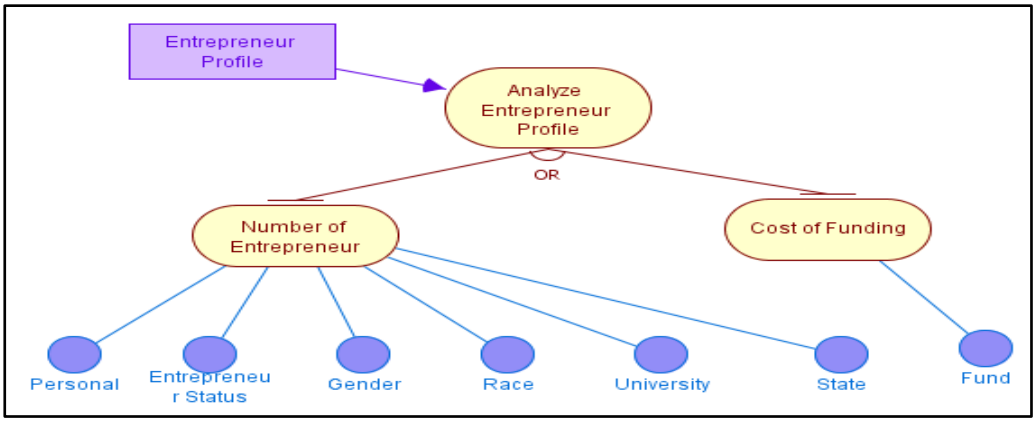


Figure C3.8: Extended EU Diagram with Entrepreneur Profile Dimensions

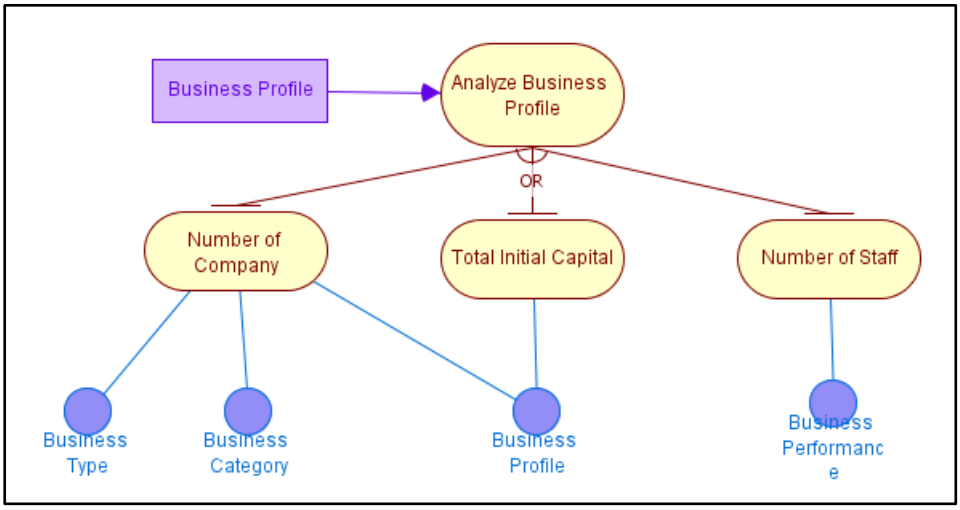


Figure C3.9: Extended EU Diagram with Business Profile Dimensions

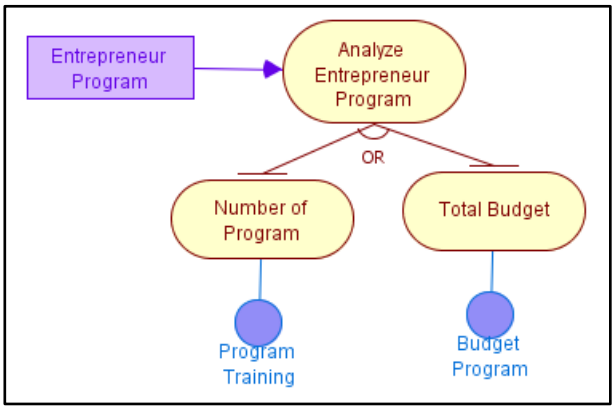


Figure C3.10: Extended EU Diagram with Entrepreneur Program Dimensions

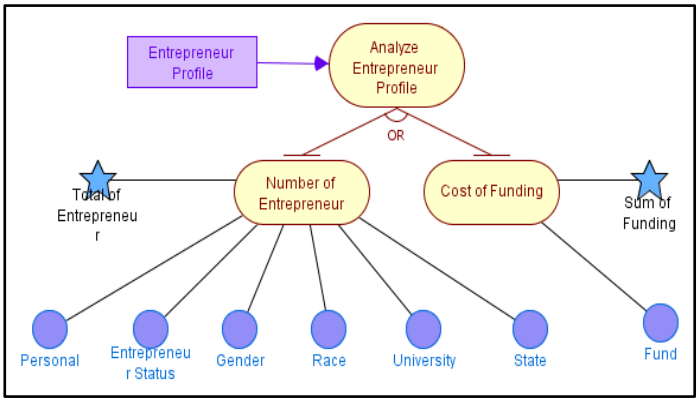


Figure C3.11: Extended EU Diagram with Entrepreneur Profile Measures

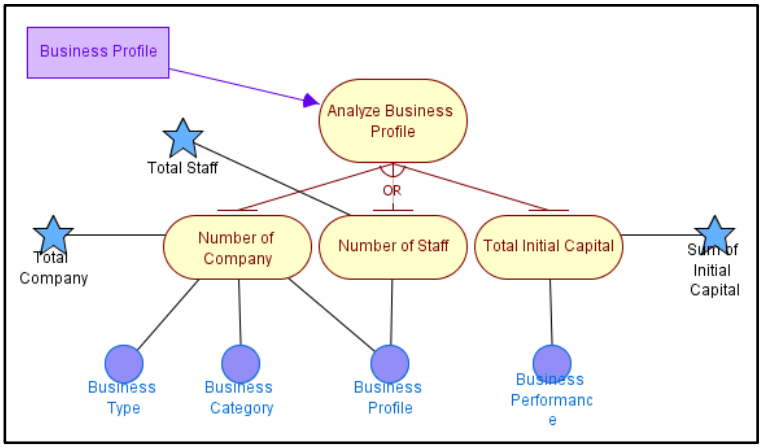


Figure C3.12: Extended EU Diagram with Business Profile Measures

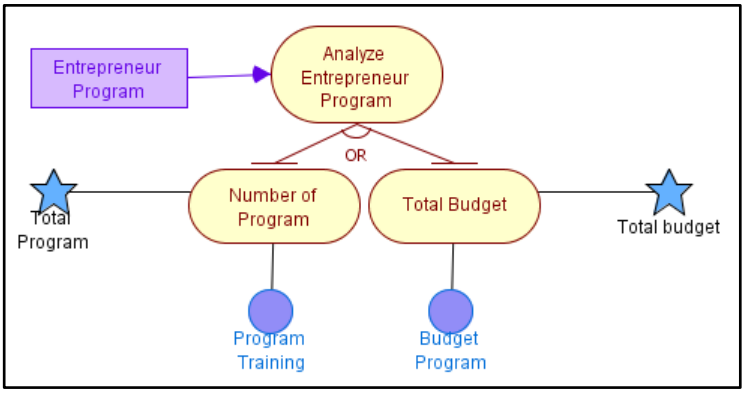


Figure C3.13: Extended EU Diagram with Entrepreneur Program Measures

(v) Ontology Model - The DWROand DSO

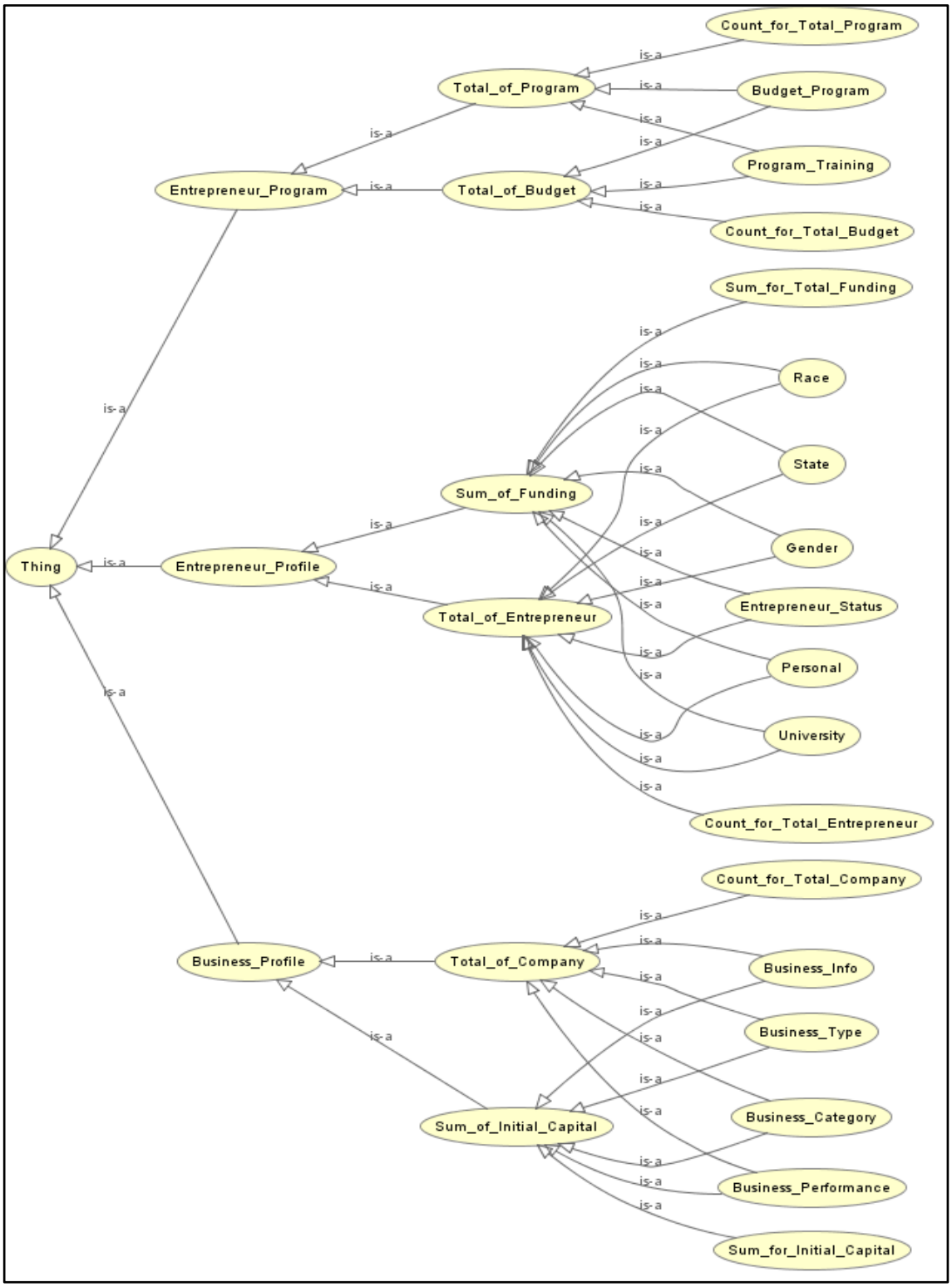


Figure C3.14: The DWRO for Student Entrepreneur

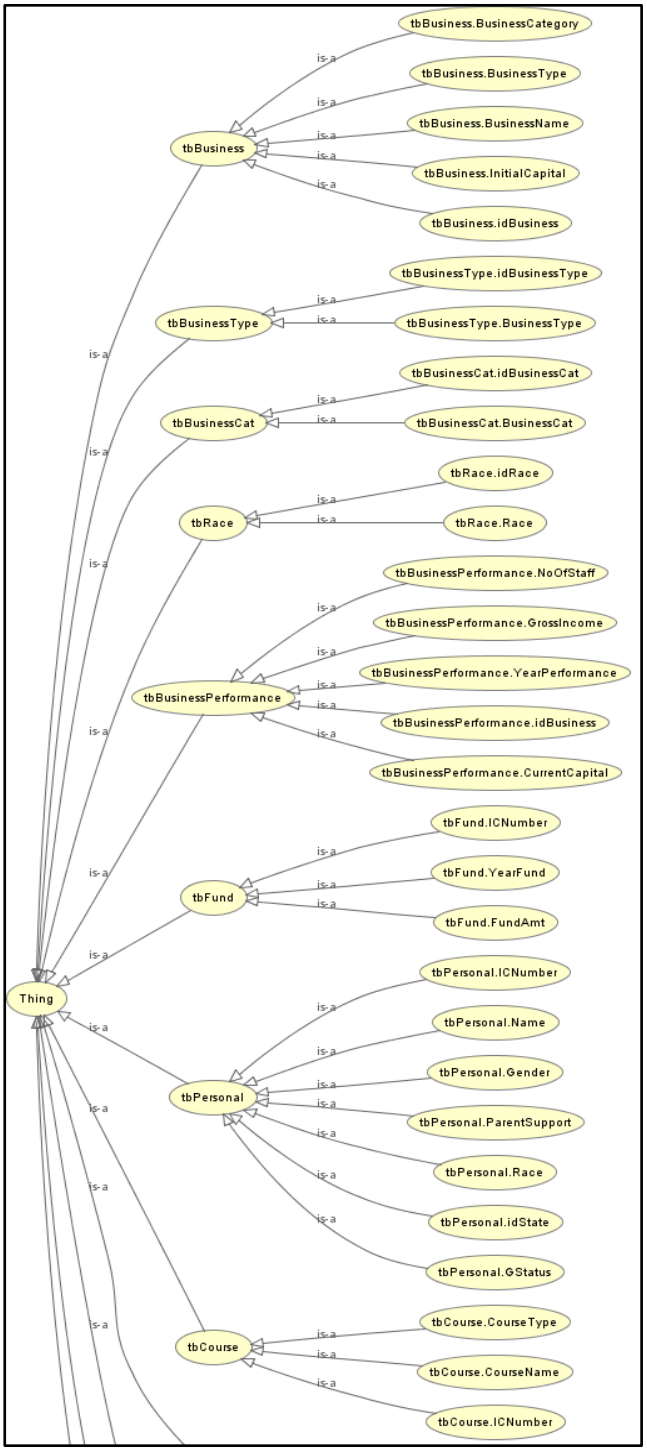


Figure C3.15: The DSO for Student Entrepreneur

Appendix D– Questionnaires for Expert Review

1. Different levels of abstraction – Does the RAMEPs supports navigating from the abstract levels of reasoning to the concrete one and vice versa?
   - Yes
   - No
   - Neutral

2. Identifying participants in the domain – In DW scenarios with many participants, do the RAMEPs help identify participants?
   - Yes
   - No
   - Neutral

3. Capturing, understand, and registering terminology – Would the RAMEPs help understand the different terms?
   - Yes
   - No
   - Neutral

4. Domain Analysis – Does the RAMEPs support the modeling and reasoning about the social relationship involved in the DW scenarios?
   - Yes
   - No
   - Neutral

5. Finding Requirements – Does the RAMEPs help in discovering and refining requirements?
   - Yes
   - No
   - Neutral

6. DW Design – DW system implied the use of heterogeneous databases for accessing data sources. Do the RAMEPs determine the modes of interaction with these data sources?
   - Yes
   - No
   - Neutral

7. DW Evolution – Would the RAMEPs support the fact that these data sources will be continuously evolving?
   - □ Yes
   - □ No
   - □ Neutral

8. ETL Processes Design – Would the RAMEPs supports the design of ETL processes and produced the ETL processes specifications automatically?
   - □ Yes
   - □ No
   - □ Neutral

9. Eliciting and reasoning about non-functional aspects – Does the RAMEPs facilitates the elicitation and reason of such non-functional requirements?
   - □ Yes
   - □ No
   - □ Neutral

10. Formal verification and validation – Does the RAMEPs provide any mean for formal verification and validation?
    - □ Yes
    - □ No
    - □ Neutral

11. Tool support – Does the RAMEPs supported by the current commercial tools?
    - □ Yes
    - □ No
    - □ Neutral

12. Learning curve – Can the DW developer learn the RAMEPs and its tools easily?
    - □ Yes
    - □ No
    - □ Neutral

13. Methodology for simpler problem – Does the RAMEPs scales from the complex to simpler problems?
    - □ Yes
    - □ No
    - □ Neutral

14. What do you think about RAMEPs and this expert review questionnaire?

_____

_____

_____

_____

_____

_____

_____

15. About yourself?

Name : _____

Position : _____

Company : _____

Year of experiences in DW system development : _____

DW tools used : _____

Thanks for sharing your opinions in this questionnaires.

Mr. Azman Ta'a
College of Arts and Sciences
Universiti Utara Malaysia
06000 UUM Sintok
Kedah Darulaman.
OP: 04-9284600, HP: 0184742680, emel: azman@uum.edu.my

Appendix E– Feedbacks from the DW Experts

		Industry			Government				Results		
No.	About the Questions	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	Expert 6	Expert 7	Yes	No	Neutral
1	Different Level of abstraction	0	0	0	0	0	0	0	7	0	0
2	Identifying participants in the domain	0	0	0	0	2	0	0	6	0	1
3	Capturing, understand, and registering terminology	0	0	0	1	0	0	0	6	1	0
4	Domain Analysis	0	0	0	2	0	0	0	6	0	1
5	Finding Requirements	0	0	0	0	0	0	0	7	0	0
6	DW Design	2	0	0	2	0	0	0	5	0	2
7	DW Evolution	1	0	2	0	0	0	0	5	1	1
8	ETL Processes Design	0	0	1	2	0	0	0	5	1	1
9	Eliciting Non-Functional Aspects	1	0	0	1	0	0	0	5	2	0
10	Formal Verification and Validation	2	0	0	0	0	0	0	6	0	1
11	Tool Supports	0	2	2	2	2	0	0	3	0	4
12	Learning Curve	1	2	1	1	0	1	2	1	4	2
13	Methodology for Simpler Problem	2	0	2	0	0	0	0	5	0	2

14	Opinion about RAMEPs	1. The explanation of the document is too high level 2. Most methodologies will be modified or customized when come to implementation 3. The methodology is a good start for more detail and complete	No opinions or comments given	1. In real environment the model and business rules are more complex 2. It take time to prepare all the diagram documentation 3. RAMEPs might have change it model into ETL jobs since not all tools are equal	1. The RAMEPs can be used to implement ETL processes 2. To automate the ETL processes is not easy in the real world 3. Normally developer pleasant to use SQL rather than auto generated codes 4. With current technology, defining the ETL processes is not too hard compare to design the fact-dimension	1. It good to have tool that support the RAMEPs approach 2. RAMEPs approach look can support consultant and business analyst in developing DW	1. RAMEPs can help to generate the ETL processes in large domain of DW 2. It also help documents each of the requirement analysis process 3. Clearly, the ETL processes support the organization goals	1. Many developer in DW development team can be easily managed and control 2. Maybe an issue of changing goals or policies will reflect the existing DW schemas			
15	Experiences in DW system Development	17 Years	3 Years	4 Years	5 Years	3 Years	6 Years	6 Years	3 - 17 Years		
16	DW Tools Used	Speedminer for DW and BI	MS SQL Server - Analysis Services, Integration Services, Reporting Services	SAS Data Integration Studio	SAS BI Studio, Microsoft SQL Server, Open Kettle, Sybase IQ	Business Object, Pentaho	Microsoft SQL Server - Business Intelligent	Microsoft SQL Server - Business Intelligent	MS SQL Server, SAS BI, Pentaho, Business Object		

Legend: 0 – Yes, 1 – No, 2 – Neutral

Appendix F–DW Experts Profile

No.	DW Expert Profile	
1.		Name : Thomas How Kok Sheng Position : Managing Director/Chief technology Officer Company : Speedminer Sdn. Bhd. Year of experiences in DW system development : 17 years DW tools used : Speedminer DW and BI
2.		Name : AlaaEddin H A AlMabhouh Position : DW specialist Company : AE ITQAN Year of experiences in DW system development : 3 Years DW tools used : MS SQL Server(Analysis Services, Integration Services, and Reporting Services)
3.		Name : Kong Khai Yun Position : BI Consultant Company : SAS Malaysia Institute Year of experiences in DW system development : 4 years DW tools used : SAS BI Studio.
4.		Name : Noorfaizalfarid Mohd Noor Position : IT Officer Company : Universiti Teknologi Mara (UiTM) Year of experiences in DW system development : 5 years DW tools used : SAS, Microsoft, Open Kettle, Sybase IQ.
5.		Name : Yahaya Hj Ismail Position : Senior Programmer Company : Universiti Utara Malaysia (UUM) Year of experiences in DW system development : 6 years DW tools used : Microsoft SQL Server, Microsoft Excel, Sybase IQ.
6.		Name : Nur Hani Zulkifli Abai Position : Senior Programmer Company : Universiti Utara Malaysia (UUM) Year of experiences in DW system development : 6 years DW tools used : Microsoft SQL Server, Microsoft Excel, Sybase IQ.
7.		Name : Idris Takyan Position : Senior IT Officer Company : Universiti Sains Islam Malaysia (USIM) Year of experiences in DW system development : 3 years DW tools used : Business Object, Pentaho