Journal of ICT, 10, pp: 15–43

GOAL-ONTOLOGY ETL PROCESSES SPECIFICATION

Azman Ta'a[1], Mohd. Syazwan Abdullah[2] and Norita Md. Norwawi[3]

College of Arts and Sciences
Universiti Utara Malaysia

azman@uum.edu.my[1]
syazwan@uum.edu.my[2]
norita@usim.edu.my[3]

ABSTRACT

The design-related problems for extract, transform, load (ETL) processes are far away from being resolved due to the ambiguity of user requirements and the complexity of operations. Current approaches are based on existing software requirement methods that have limitations on reconciliation of the requirement semantics toward generating the ETL processes specification. The solution is to apply the RAMEPs (Requirement Analysis Method for ETL Processes) that was developed to facilitate the design of the ETL processes in the perspectives of organization, decision-maker, and developer. The results are the ETL processes specification, which was validated on the correctness of the goal-ontology model and evaluated in the case study of Gas Malaysia Data Warehouse (DW) system. The case study illustrated how the goal-ontology approach was successfully implemented in designing and generating the ETL processes specification.

Keywords: ETL Processes, Data warehouse, Ontology, Business intelligence, Requirement analysis

INTRODUCTION

Data Warehouse (DW) is a system for gathering, storing, processing, and providing huge amounts of data with analytical tools to present complex and meaningful information for decision makers (Ta'a, Abdullah & Norwawi, 2010). These data are collected, stored, and accessed in centralized databases in order to sustain competitiveness in the business environment. However, the DW system requires the ETL processes for providing the required data (Kimball & Caserta, 2004). Specifically, the success of the DW system is

highly dependent on the ETL processes specification. Many issues in modelling and designing the ETL processes are due to the problems of capturing and analysing the appropriate requirements of DW (Simitsis, 2004; Kimball & Caserta, 2004). Moreover, the design tasks need to tackle the complexity of the ETL processes from the early phases of the DW system development to ensure the appropriateness of the information produced from the DW systems (Giorgini, Rizzi & Garzetti, 2008).

The complexity of the ETL processes always refers to the problem of defining the integration and transformations of data sources. These transformations involve the reconciliation semantic of user requirements and data sources heterogeneity (Alexiev, Breu, Bruijn, Fensel, Lara & Lausen, 2005; Allemang & Hendler, 2008). Generally, an ambiguous definition of user requirements occurs because the users are unable to define their requirements precisely and clearly (Inmon, 2002). Moreover, various meanings of data (e.g. attributes, tables, constraints) make it difficult for integrating the user requirements to the appropriate data sources. Thus, reconciliations of user requirements and data sources are important for generating the ETL processes accordingly. The designing of the ETL processes should commence from the early phases of the DW system development and guided by a suitable methodology. Therefore, this research has applied the RAMEPs, a requirement analysis method for the ETL processes (Ta'a et al., (2010), and generated the ETL processes specification from the Gas Malaysia DW system case study.

REQUIREMENT ANALYSIS MODEL FOR ETL PROCESSES

Requirement analysis of the ETL processes focuses on the informal statements of user requirements into a formal expression of the ETL processes specifications. The informal statements are derived from the requirement of users and analysed from the organization and decision-maker perspectives (Giorgini et al., 2008). However, we argue that analysing the user requirements toward the ETL processes specification is better defined by supporting of the developer perspectives. This is clearly important for tackling the complexity issues, which analyse abstract knowledge to the detail execution of the ETL processes. It is widely accepted that the early requirement analysis significantly reduces the possibility of misunderstanding user requirements (Yu, 1995). The better the understanding among users, the higher are the chances of agreeing on the terms and definitions used in the ETL processes specification. Therefore, the RAMEPs is centered on the organizational and decisional modeling and focuses on the transformation analysis from the perspective of a developer model.

The focus on the works in RAMEPs are highlighted in the thick box area of Figure 1. The organizational modeling is used to identify the goals that are related to facts, and attributes. The decisional modeling is focused on the information needs decision makers and related to facts, dimension and measures. The developer modeling is used to define the related actions and business rules for the data sources. Detailed explanation on the RAMEPs can be referred to in Ta'a et al., (2010).

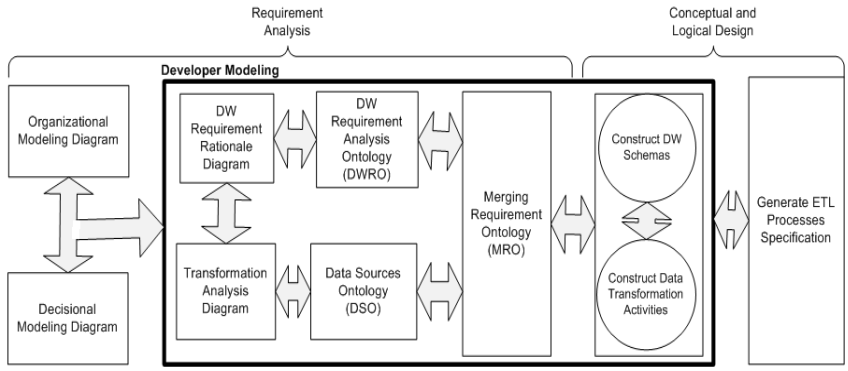

Figure 1. The RAMEPs

GOAL-ONTOLOGY FOR REQUIREMENT ANALYSIS MODEL

The aim of RAMEPs is to facilitate the design of the ETL processes by analysing and producing the DW requirements for decision-makers and organizations. Through RAMEPs, the ETL processes are modeled and designed by capturing important knowledge such as the DW schemas and transformation activities. Indeed, the goal-oriented approach is utilized to capture and represent the knowledge about the ETL processes as defined from the perspectives of the organization, decision-maker, and developer. Developer perspectives are pressing beside the organization and the decision-maker perspectives in order to complete the requirements analysis model of the ETL processes.

Goal-orientedApproach

Goal-oriented is utilizing goals for requirements elicitation, evaluation, negotiation, elaboration, structuring, documentation, analysis and evolution of the software system through the cooperation of its agents (Bresciani, Perini,

Giorgini, Giunchiglia & Mylopoulos, 2003; Lamsweerde, 2010). Specifically, this research uses goals to elicit and analyse the DW requirements with the cooperation of three modeling: organizational, decisional, and developer.

Organizational Modelling - consists of three different analyses, which are produced in the iterative process. The types of analyses is goal analysis, fact analysis, and attributes analysis. All goals, facts, and attributes are defined in the context of organization views.

Decisional Modelling - consists of four different analyses, which are produced in the iterative process. However, these analyses are focused on the goal of a decision-maker that is represented by the actors as defined in the organizational model. The types of analyses are goal analysis, fact analysis, dimension is required for supporting the decision making.

Developer Modelling - consists of three different analyses, which are produced in the iterative process. The analysis is focused on the goal of a decision-maker which is represented by the actors as defined in the decisional model. The analyses are data sources analysis, business rules analysis, and transformation analysis. The transformation analysis is based on plan modelling in Tropos methodology (Bresciani, Perini, Giorgini, Giunchiglia & Mylopoulos, 2004). The developer modelling explains the facts about actions and rules applied to the data sources in the perspectives of ETL developers. The developer modelling will complete the goal-driven analysis of user requirements in order to produce the ETL processes model for the DW system.

As comparism, the outcome of each of the modelling is presented in Table 1.

Table 1

Outcomes of the Modelling

Modelling	Outcomes	Notes
Organizational	- List of Facts - List of Attributes	Represent the main data in the organization and comprise most relevant attributes as exist in data sources.
Decisional	- List of Facts - List of Dimensions - List of Measures	Represent decision-maker needs, summarizing the role played in the glossary based requirements.
Developer	- List of Actions - List of Business Rules - List of Tables	Represent the information within the developer needs to define the transformations.

Ontology-based Approach

The organizational, decisional, and developer models have determined the ETL processes glossaries (i.e.facts, dimensions, measures, attributes, business rules, actions) through goal-driven diagrams. The glossaries for facts, dimensions, attributes, measures, and actions must be agreed up on by the users. This will be used for building the conceptual design of ETL processes according to RAMEPs approach. Since these agreeable glossaries will be mapped to the heterogeneous data sources, the semantic heterogeneity problems will remain 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 (Alexiev, Breu, Bruijn, Fensel, Lara & Lausen, 2005; Allemang & Hendler, 2008).

In ontology modelling, the process for constructing the DW requirements ontology (DWRO) is semantically described in the requirement glossaries. The semantics of the DW requirements are described in high-level meaning, so that the DW requirements can be possibly mapped to the data sources ontology (DSO) for accomplishing the transformation and integration process. The strong linkages between requirement glossaries and appropriate data sources through ontology structure will produce the ETL processes specification automatically. This can be done through invoking an appropriate algorithm and reasoning. In particular, the use of ontology is based on description logic (DL), which constitutes the most commonly use of knowledge representation formalism (Hutter, Stephan, Baader, Horrocks & Sattler, 2005). This research uses the OWL language for knowledge representation that adopted the DL formalism. The Resource Description Framework (RDF) is used together with OWL in presenting the DWRO and DSO for modelling the concepts of the domain, relationships between concepts to attributes, and the attributes and relationship that belong to each attribute. The concepts refer to the facts, whereas the dimensions, measures,business rules, and actions refer to the attributes. The concepts of the domain are represented by classes, and relationships or attributes are represented by the properties. Moreover, theconcept, relation, and attribute components were also applied in representing the semantic sentences through the concept relational model (CRM), which was concerned with the ambiguity of sentences (Abdullah, Selamat, Ibrahim, Chulan & Nasharuddin, 2009).

Due to the specialty of aggregation and population operation in the DW systems, specific representation classes are necessary to specify. However, the RDF/OWL features need to be suited for high-level representation, since

all the glossaries are defined in the abstract form. For this purpose, the RDF/ OWL features and ontology notations were adopted as shown in Table 2.

Table 2

OWL Features and Ontology Notations

Notation	Name	Description
C	Class	Classes represent the concepts of the domain being modelled.
C_1 C_2	Equivalent	States two classes are equivalent.
C_1 C_2	Sub Class of	Creates class hierarchies.
C_1 C_2	Disjoint with	States that two classes two $C_1$and C_2 are disjoints.
C_1 C_2	Union of	The union of two classes C_1 and $C_2$are joined.
P	Property	Represents attributes of concepts and relationships between concepts.
dom(P)	Domain	Specifies the class (-es) to which the property belongs to.
rang(P)	Range	Specifies the class (-es) to which the value of the property belongs to.
P.C	Some values from	Restrict the range of property to participate in at least one relationship.
P.C	All values from	Restrict the range of property to only have relationships with this property.
nP, nP	Mix/max cardinality	Specifies the min/max cardinality of a property.

The mapping results between DWRO and DSO created new classes and properties pertaining to the ETL processes activities, and produced merging requirement ontology (MRO). These new classes and properties will capture the knowledge of the ETL processes such as aggregated, aggregation, range, table, formation, and others. The type of knowledge applied for this case study is shown in Table 3.

Table 3

Description of New Classes for ETL Processes

Type of Knowledge	Classes: Examples	Description
Concept	Sale Volume and Revenue	Represents the concept of Sale Volume and Revenue.
Aggregated	Count Total Customers	Represents the measure of Total Customer.
Range	Only for Residential Customers	Represents the business rule for the measure.
Aggregation	COUNT, SUM, AVERAGE	Represents the calculation operation for the measure.
Table	RETRIEVE, LOADING	Represents the accessing and pushing of the data.
Formation	CONVERSION	Represents the transformation of data format.

These new classes need to be organized accordingly into the MRO. Again, this task is done through Protégé-OWL. This process is finished until the ontology structure is reconstructed and rechecked by using the Pellet reasoner. New RDF/OWL document is produced to represent the entire specification of the ETL processes. Then, the RDF/OWL document is used to determine the appropriate ETL processes specification. However, few refinements on the MRO need to be carried out in order to ensure the ontology structure is fully satisfying the ETL processes operation. Through a reasoning process, the inferred MRO is semantically organized in presenting the knowledge of the ETL processes. By using the semantic Jena 2 framework as a web-programming language, the ETL processes specification is produced. Furthermore, the generated ETL processes specification can be used for implementing the DW system.

THE CASE STUDY

The case study discussed in this paper is focused on the Gas Malaysia utility company. This company promotes, constructs, and operates the Natural Gas Distribution System within Peninsular Malaysia. The company's mission in providing the cleanest, safest, cost effective, and reliable energy solutions

were motivated them to provide innovative energy solutions to the nation. The requirements gathering was carried out with the company stakeholders and focused on the information needed. These requirements were focused on the billing area, which was comprised of billing transaction activities. The billing system was implemented in the Utility Billing Information System (UBIS). It was focused on the residential customers and supported by the external application systems namely the JDE System and the Call Center System (CCS). The main goals of Gas Malaysia are shown in Figure 2.

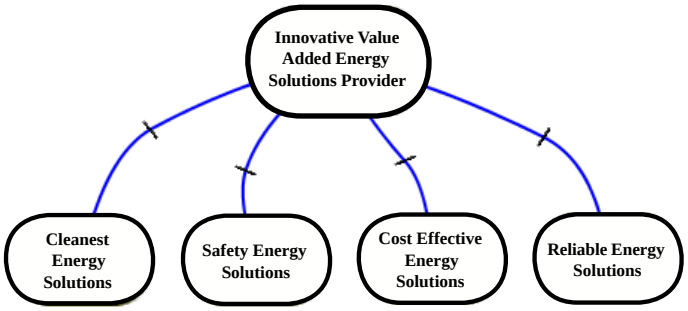

Figure 2. Gas Malaysia Main Goals

Organization Modelling

The main goal of the company is *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 evaluation process, the case study is focused on the *Cost Effective Energy Solution* that is related to the billing area. The analyses task was commenced by modelling the DW requirements in the perspective of the Billing Department. The stakeholders involved in the billing area were identified and represented by using an actor model. An actor model explains the dependencies among the actors (i.e. billing department, customer, billing operator, call center department). The next step was to analyse the facts that aim to identify all the relevant facts for the billing area. The facts explain the information required within the goal structure in the billing area. Thus, the analysis was carried out by identifying the facts for each goal from top to down of the goal hierarchy. The final diagram for organization modelling that defines two facts (i.e. *Sale Volume and Revenue* and *Customer and Billing Status*) is illustrated in Figure 3 and Figure 4 respectively.

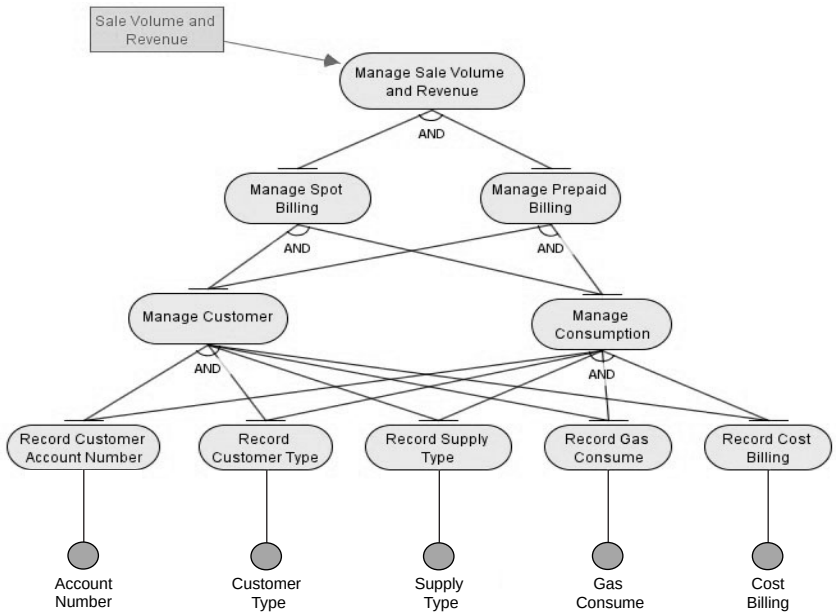

Figure 3. Sale Volume and Revenue Fact

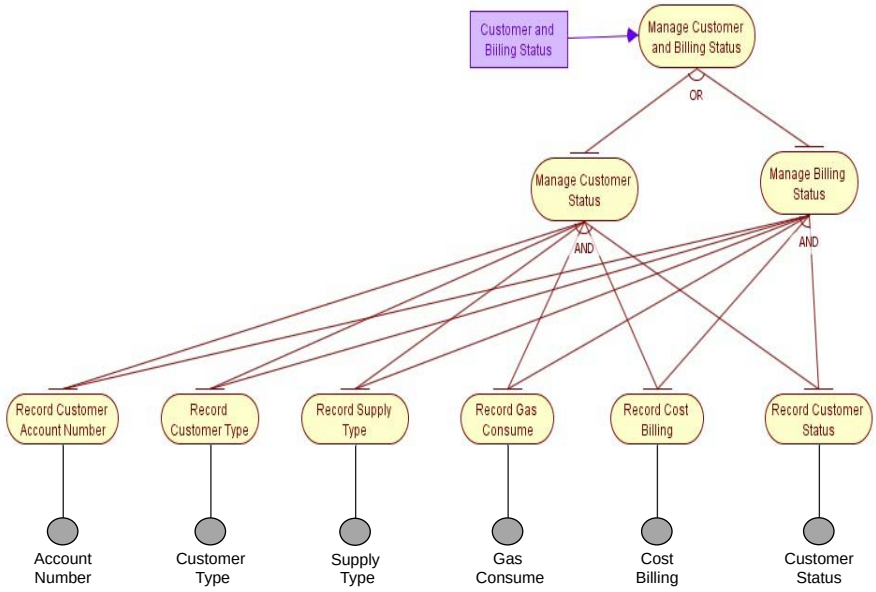

Figure 4. Customer and Billing Status Fact

Decisional Modelling

There are four phases in decisional modelling: goal analysis, fact analysis, dimension analysis, and measure analysis. All four analyses are connected to each other and aimed to identify the DW components. The analysis is focused on the decision-maker goals in order to define the requirements. The analysis process starts with identifying the actors in the goal analysis, and extends to the fact, dimension, and measure. In this case study, a Billing Manager (BM) was selected as an actor for the decision maker. In previous approaches, the requirement analysis process ended at this stage. The knowledge of facts, dimensions, attributes, and measures will be used in further designs of the is the DW and ETL processes. However, the extended analysis of data transformation thats related to defining facts, dimensions, and measures needs to be carried out to ensure the successful implementation of the DW system. The final diagram for decisional modelling that define the few measures is illustrated in Figure 5 and Figure 6 respectively.

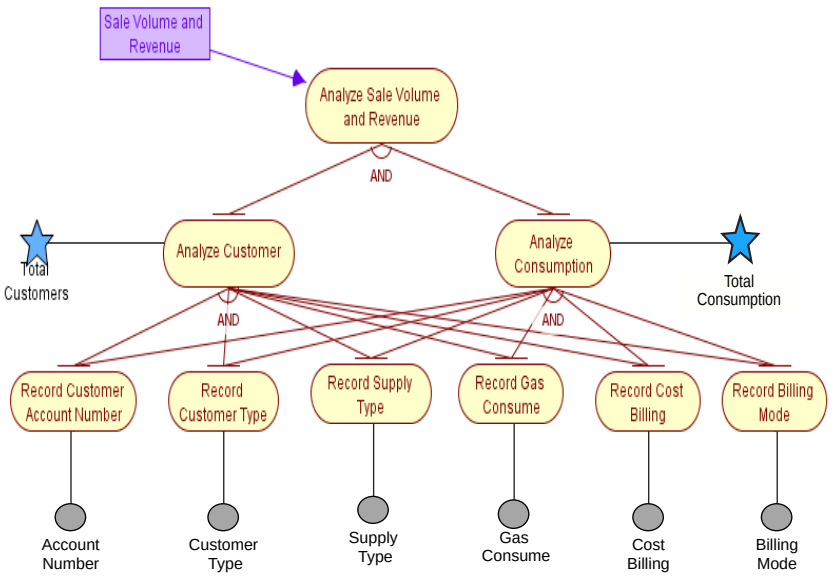

Figure 5. Sale Volume and Revenue Measures

Developer Modelling

In business rule analysis, the developer needs to identify the restrictions applicable to the ETL processes according to the user requirements. The ETL processes will populate the data sources according to the restrictions given. In

this case study, the business rules were identified for facts of *Sale Volume and Revenue* and *Billing and Customer Status*. According to the analysis, list of business rules is presented in Table 3.

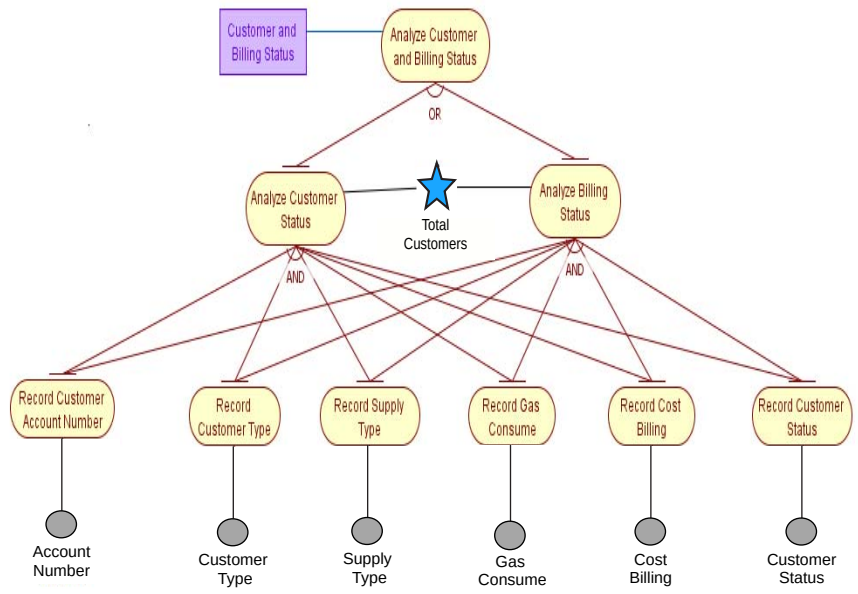

Figure 6. Customer and Billing Status Measure

Table 3

List of Business Rules

Facts	Measures	Business Rules
Sale Volume and Revenue	Count Total Customers	Only for spot billing and prepaid billing mode.
Billing and Customer Status	Count Total Consumption	Only for spot billing and prepaid billing mode.
	Total Customers	• Only for residential customer • Only for spot billing and prepaid billing mode.
	Total Billing	Only for spot billing.

Based on the business rules given, the transformation analysis can be carried out for conceptualizing the actions to be taken for transforming data sources to the DW. The transformation analysis emphasized on the achievement of the ETL processes model for user requirements and required business rules to absorb the complexity of the data sources. Based on the extended goal diagram of the BM, the actions for *Total Customers* and *Total Consumption* for *Sale Volume and Revenue* goal are presented in Figure7. The actions for *Count Total Customer Billing* and *Count Total Customer Status* for Customer and Billing Status are presented in Figure 8.

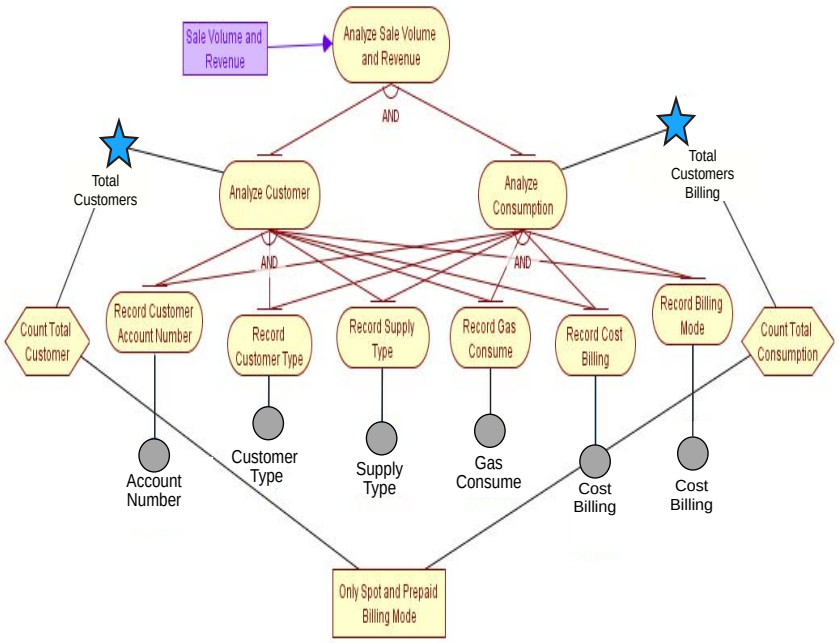

Figure 7. Sale Volume and Revenue Actions

The integration of UBIS and JDE data sources are based on the DSO, which clarifies the semantic of user requirements toward the data sources by defining the concepts, classes, properties, and relationships. The ontology mapping between DWRO and DSO is definedaccording to the mapping method used in RAMEPs. The example of mapping structure for *Sale Volume and Revenue* is shown in Table 4.

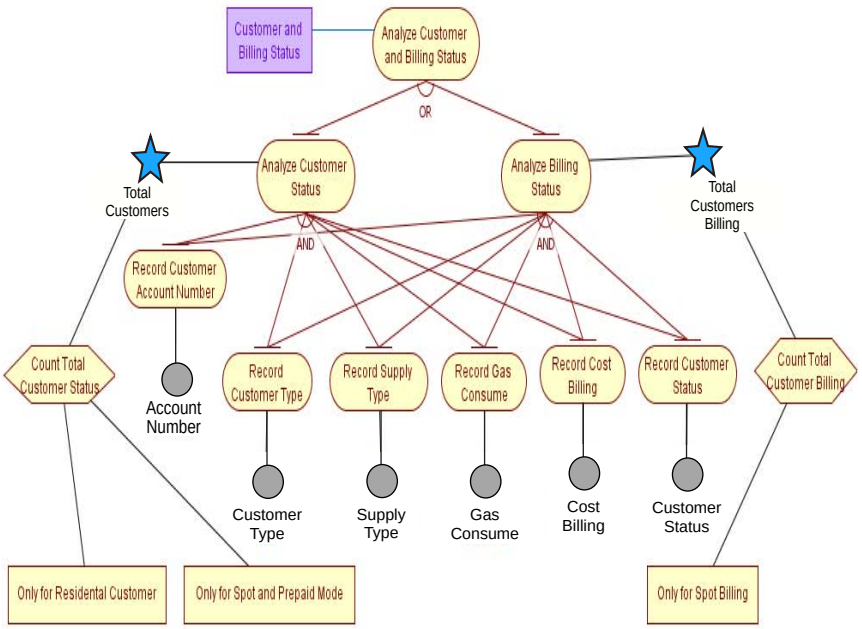

Figure 8. Customer and Billing Status Actions

Table 4

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) …	Billing Mode ↔Concept: Mode Billing Customer Type ↔ Concept: Customer Type Customer, Account number * ↔Concept: Customer Profile Supply Type ↔Concept: Supply Type …

Ontology Modelling

The design of the ETL processes is conceptualized by the DW components that are produced from the goal-oriented requirement analysis process. The DW components used to construct the DWRO are based on the ontology model (Ta'a et al., 2010):

DWRO = (F,D,M,Br,Ac)

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)

Based on DWRO, the four classes of measures 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 consume, cost billing, billing mode, and customer status*. The relationship between the classes and the properties are defined as *has Measure Total Customer, has Measure Sum Consumption, has Action Count Customer, has Action Sum Consumption*, and so on. Additionally, the axioms are defined based on business rules (e.g. *Only Spot and Prepaid Billing*) and actions (e.g. aggregation – SUM for usage of gas in volume).

The ontology model for data sources UBIS and JDE is constructed based on the ontology model (Shen, Huang, Zhu & Zhao, 2006):

DSO = (C, R, A, I)

Where: C = Finite set of concepts in the domain

R = Set of relations between concepts.

A = Set of axioms imply in property of concepts.

I = Instance of the concept that presents the values of the ontology tuple.

Both databases handle the billing transaction for gas consumption of the residential and the industrial respectively. These data sources are implemented in the different systems that is dissimilar in their data structure and semantics. This scenario creates the heterogeneity problems during the integration and transformation of the data sources. Therefore, the integration of both

data sources is resolves the semantic heterogeneity problems by defining the ontology concepts and classes between the data sources and the DW requirements.

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

- Concept = Classes (i.e. Sale Volume and Revenue, Customer and Billing Status)
- Relationship = Properties (e.g. has Measure Total Customer, has Dimension Customer Type, has Action Count Customer)
- Specific element in DW setting = new Classes (e.g. SUM, COUNT)
- The restriction = Axioms (e.g. "Only for Spot and Prepaid Billing")

Based on the mapping definition as described in Ta'a, Abdullah and Norwawi (2010), the ontology mapping between the DWRO and the DSO for *Sale Volume and Revenue* is shown in Table 5. This ontology mapping should maintain the semantics of user requirements as defined in the DWRO.

Table 5

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

(continued)

DWRO	DSO	The Mapping
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 Billing (tbillmode, -)]
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, -)	– [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 5 presents the mapping specifications of the DWRO and the DSO that are derived from the analysis process of user requirements and supported by the related data sources. To complete the entire cycle of the ETL processes activities, the actions for *extract* and *loading* are added in the actions plan. Based on the mapping results, new classes and properties pertaining to the merging ontology (MRO) were produced. These new classes and properties are shown in Table 6.

Table 6

New Classes and Properties

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 and properties are reorganized properly into the MRO after the merging process is successful through Protégé-OWL. Moreover, the ontology merging is done through the ontology setting as defined in Table 7.

Table 7

Ontology Merging Setting 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

(continued)

Mapping List	Ontology Setting
	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 is finished when the ontology structure is reconstructed and rechecked by using the Pallet reasoner. The new appearance of the MRO with the new classes and properties that represent the ETL processes specification is shown in Figure 9.

Constructing the ETL Processes Specifications

Using the MRO as knowledge representation of the DW requirements and the ETL operations of thebilling area, the generation of the ETL processes specification is done automatically. This task can be realized by manipulating the semantic annotation of the user requirements and the data sources in the MRO. The manipulation process proposes a set of ETL processes specifications that transform the data sources to the DW schemas as determined in the goal-oriented analysis approach. The generic data transformations used in this case study are EXTRACT, MERGE, FILTER, CONVERT, AGGREGATE, and LOAD. As presented in the MRO, the knowledge about the information

as required and their related data sources have been defined according to the RDF/OWL- based language. Thus, the MRO is processed to determine and propose a set of ETL processes specifications. Ontology reasoning is applied on classes and their related properties to derive the ETL processes specifications according to the generic data transformation tasks.

The MRO structure that is represented by the RDF/OWL language (as shown in Figure 10) is manipulated through the Jena 2 framework. To generate the ETL processes specifications, a prototype application for reading, and manipulating the MRO was developed by using Java running on the Eclipse platform. The derivative of the ETL processes specification is based on an algorithm (as shown in Figure 11) that is adapted from Skoutas and Simitsis (2007). The result of this application is the ETL processes specifications as shown in Figure 12.

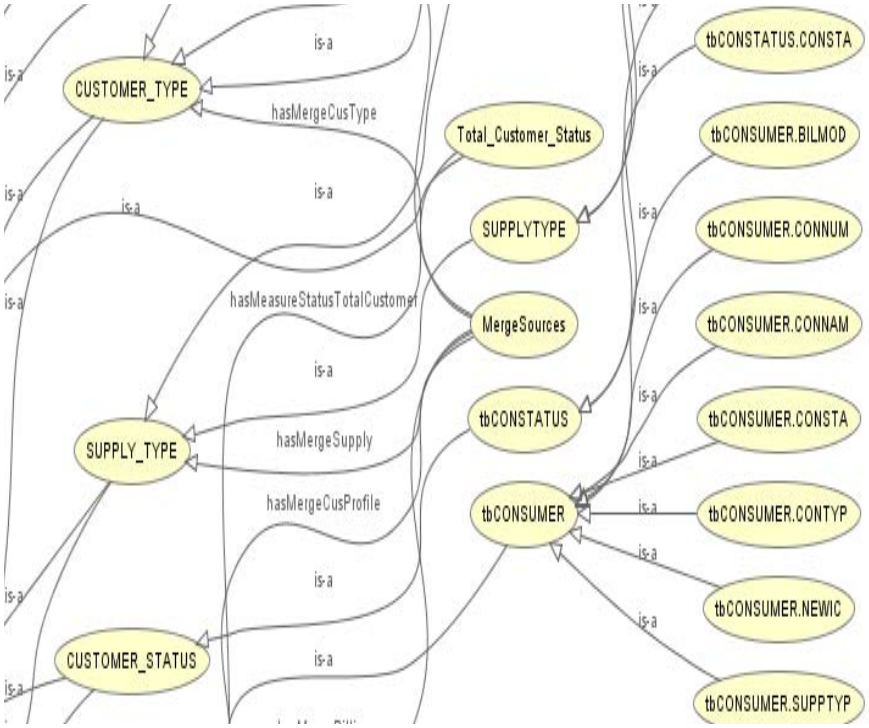

Figure 9. The MRO for Gas Malaysia

```
<!--http://www.semanticweb.org/ontologies/2010/7/gasmalaysia_MRO.
owl#hasSpot -->
<owl:ObjectPropertyrdf:about="#hasSpot">
<rdfs:rangerdf: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:ObjectPropertyrdf:about="#hasSpotPrepaid">
<rdfs:rangerdf:resource="&gasmalaysia_datasources;BILLING_MODE"/>
<rdfs:domainrdf:resource="&gasmalaysia_requirements;Customer_and_
Billing_Status"/>
<rdfs:domainrdf:resource="&gasmalaysia_requirements;Sale_Volume_and_
Revenue"/>
</owl:ObjectProperty>
<!--http://www.semanticweb.org/ontologies/2010/6/gasmalaysia_datasources.
owl#BILLING -->
```

Figure 10. The Snippet of MRO

```
Input: MRO
Output: A List of ETL Processes Specifications (ListOfETL)
Begin
        Cs ← Class corresponding to MRO nodes sources
        Cdw← Class corresponding to MRO nodes DW
        IF (Cs  Cdw)
        { ListOfETL ←   }
      ELSE {IF (Cdw Cs) {For each class Ci in the path from Cs to Cdw
{ IF ( Cg: Aggregate (Cg, Ci))
{C'← one or more classes Cior groups (Ci, C)
                ListOfETL ← add AGGREGATE FUNCTIONS (Cg, C') }
                ELSE {IF ( Cm: MergeSource (Cm, Ci))
{C' ← one or more classes Cior groups (Ci, C)
ListOfETL ← add MERGE (Cm, C') }
                ELSE { ListOfETL ← add FILTER (Ci) } } }
ELSE IF (  (C1, C2): Cs  C1 AND Cdw C2 AND ConvertTo (C1, C2)
{ ListOfETL ← add CONVERT (C1, C2); Cs ← C2
        Repeat for each class in the path from Cs to Cdw}
        ELSE { Cs← classes C0, Repeat for each class in the path from Cs to
Cdw}}}
End.
```

Figure 11. The Algorithm for Deriving the ETL Processes Specification

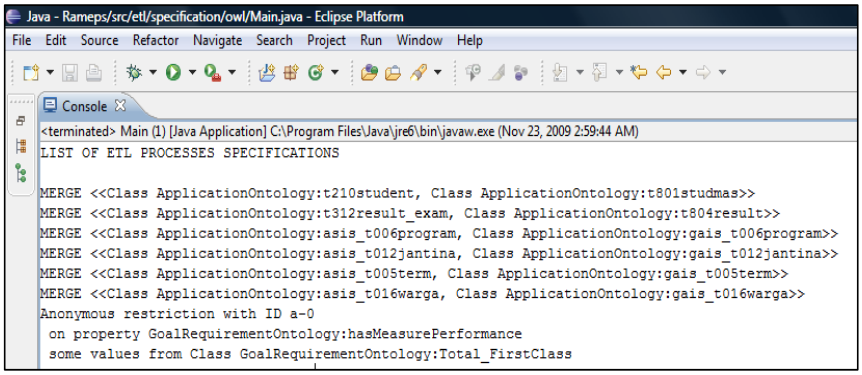

Figure 12. The ETL Processes Specification

VALIDATION AND EVALUATION

Since the RAMEPs is based on the goal-oriented and ontology approach, the validation process is emphasized on the correctness of both approaches. Consequently, the correctness of the RAMEPs is not enough until it can be evaluated in the real design of the ETL processes. To validate the correctness and ensuring the satisfaction of the RAMEPs, appropriate goal-oriented and ontology compliant tools are required for capturing and analysing the DW requirements. The compliant goal-oriented tools must be able to accommodate the elements of organizational, decisional, and developer into the modelling functionalities. Moreover, the compliant ontology tools must be able to capture and present the DW requirements and data sources in an ontology model. The evaluation was conducted for ensuring the usefulness of the RAMEPs for designing the ETL processes and was implemented in the real DW project case studies.

Generally, model checkers are used to verify the correctness of the software systems at the design stage. The correctness of a software system is verified according to the system's properties that must be a model-checked. System properties in the RAMEPs are the DW components (i.e. facts, dimensions, measures, business rules, measures) as defined from the goal-oriented analysis. The method proposed by Ogawa, Kumeno and Honiden (2008) was adopted to validate the DW components by using compliant tools (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 is focused on the sufficiency of design against requirements, which is similar to our objectives. However, our

approach is based on the Tropos model that emphasises the goal and resources that describe the DW characteristics. The model checking process and tools are illustrated in Figure 13.

In the checking method, the compliant tools are used to ensure the DW components are properly captured from one model to the next model. For examples, the goals, facts, and attributes in the organizational modelling correctly support the goals, facts, dimensions, and measures in the decisional modelling. These DW components in the decisional modelling correctly support the developer modelling. The complete DW requirements are modelled as ontology and rechecked for their correctness as ontology structure by using the Pallet ontology reasoner. Since the DW-Tool (Giorgini, Rizzi & Garzetti, 2008) does not support data transformation analysis as required for the ETL processes, a transformation analysis (TA-Tool) was developed to provide the data transformation diagram used in developer modelling.

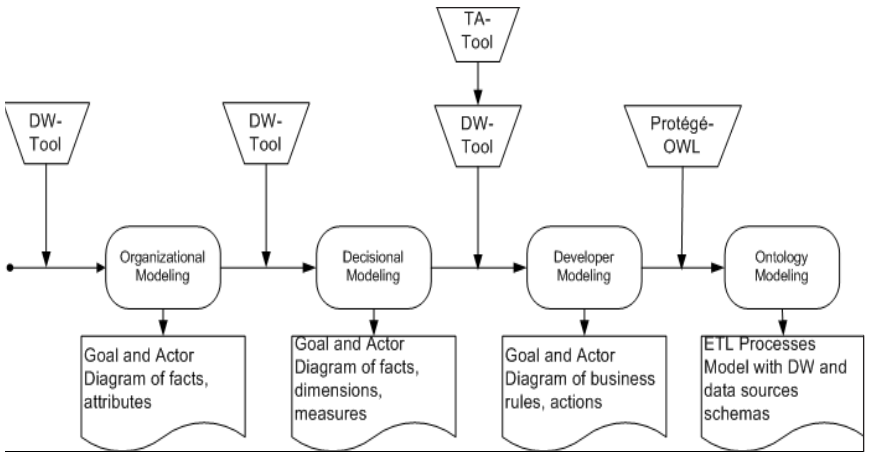

Figure 13. Model Checking Process and Compliant Tools

Model Checking Example for Goal Modelling

In the organizational modelling phase, the goal cannot be updated in the fact definition. Furthermore, the facts also cannot be updated in the dimension definition. The DW-Tool will ensure the goals only can be inserted or updated within the goal definition area as shown in Figure 14. The grey area of goal description explains the checking mechanism of the model correctness. This principle applies for all phases of modelling to avoid inconsistency among

the diagrams produced by the DW-Tool. The goal diagram in the DW-Tool is stored in XML and helps the developer to add new diagrams as proposed for the data transformation analysis.

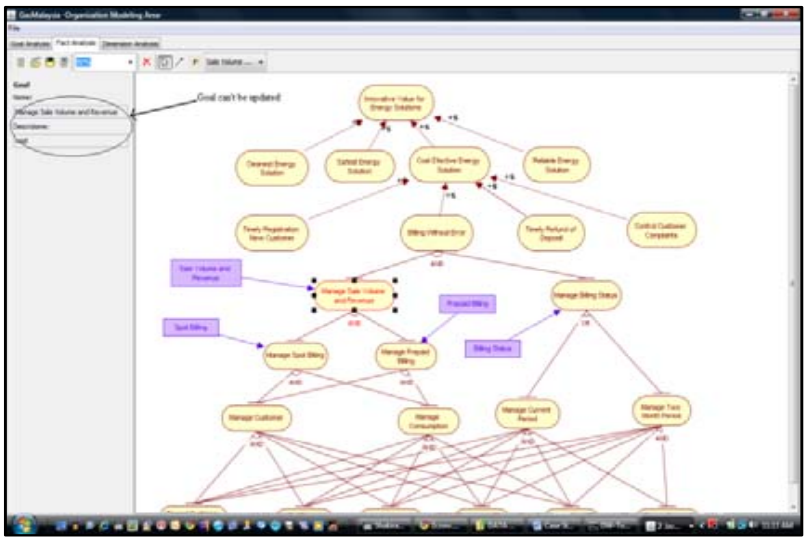

Figure 14. Goal Model Checking for Consistency

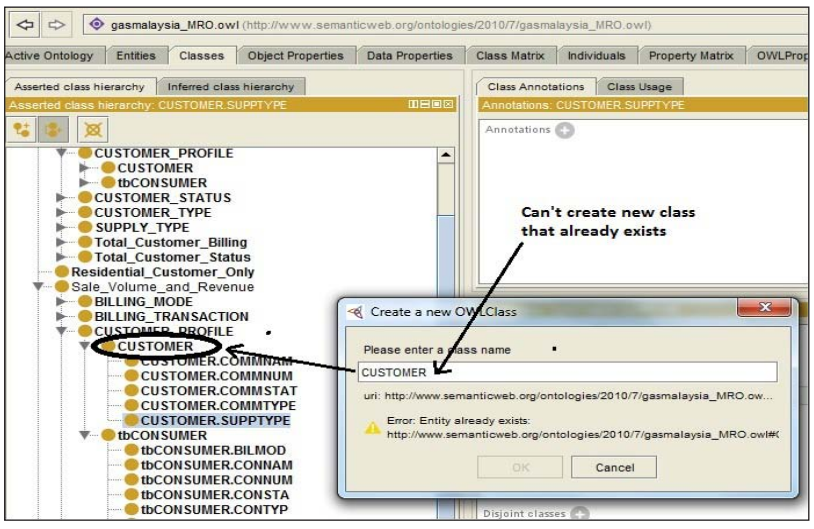

Figure 15. Consistency Checking for New Class

Model Checking Example for Ontology Modelling

The ontology model is validated by using the Pellet reasoner. The Pellet reasoner is a complete protege-OWL checker, which is based on DL formalism. The current Pellet reasoner, which comes together with the protégé-OWL has an ability to validate the RDF/OWL-based ontology model by executing the following functions (Sirin, Parsia, Grau, Kalyanpur & Katz, 2007):

- **Consistency checking** – ensures the ontologyis free from any contradictory facts such as type property-value, equality and inequality assertion.
- **Concept satisfiability** – checks whether the classes should have any instance or not.
- **Classification** – computes the subclasses' relations between every named class to create the complete class hierarchy.
- **Realization** – find the most specific classes that belong to a specific individual.

In protégé-OWL, the Pellet reasoner has integrated with the OWL editor for easy use by the developer. By invoking the Pellet reasoner, the ontology model is checked for correctness and consistency. As a result, the representation of the ETL processes semantic through ontology is validated and shown as an *inferred* ontology. Any incorrect classes, properties, and axioms are highlighted in red color and can be fixed instantly. Therefore, the usefulness of the RAMEPs approach depend on the correctness of the ETL processes specifications that are produced from the correctness of the ontology structure.

Additionally, consistency checking is used to ensure that the ontology class, properties, relationships, and formal ontology definitions are free from any contradictory facts. For example, the consistency for the existing class name while creating a newly class *CUSTOMER* is checked as shown in Figure 15.

Expert Reviews

The Expert reviews were conducted to clarify the strengths and weaknesses of the RAMEPs. The review process is based on the Gas Malaysia case study and has adopted the method that focuses on the evaluation of the requirement engineering methodology. This method is known as an *exemplar* and has been proven in evaluating the software requirement engineering approach(Cysneiros, Werneck, & Yu, 2004). A set of questionnaires together with the case study was disseminated to seven DW developers, of whom four

are from government agencies, and the others from DW companies. Their experiences are within the range of three to seventeen years in developing and implementing the DW systems in various organizations.

The questionnaires were designed and accommodated within the scope of the RAMEPs and aimed to highlight the issues of the abstraction level, participants in a domain, understanding terminology, requirement elicitation and analysis, the DW and the ETL design decision, DW evaluation and evolution, the tool used and the learning curve. The questionnaire was designed in order to capture feedback about the RAMEPs processes within the knowledge scales of *yes, no,* and *neutral*. This scale was enough to probe the capabilities and limitations of the RAMEPs methodology by supporting of the open-ended real-world exemplar. Additionally, briefings and explanations about the RAMEPs were given to the experts on separate occasions. Based on their knowledge and experiences, the experts responded to the questionnaires as shown in Table 8.

Table 8

The Summary of DW Expert Reviews

	The Questions	Yes	No	Neutral
1.	Different levels of abstraction	7	0	0
2.	Identifying participants in the domain	6	0	1
3.	Capturing, understanding, and registering terminology	6	1	0
4.	Domain analysis	6	0	1
5.	Finding requirements	7	0	0
6.	DW design	5	0	2
7.	DW evolution	5	1	1
8.	ETL processes design	5	1	1
9.	Eliciting non-functional aspects	5	2	0
10.	Formal verification and validation	6	0	1
11.	Tool supports	3	0	4
12.	Learning curve	1	4	2
13.	Methodology for simpler problem	5	0	2

Based on the feedback, the experts generally agree that the RAMEPs can be implemented by using proper tools and going through proper learning exercises.

This finding is indicated by the higher number of *Yes* for questions 1 to 9 that explain about the ETL processes design and the lower number of *No* for question 11 to 13 that explain the RAMEPs learning process. Specifically, the feedbacks for finding requirements, the ETL processes design, tool supports, and the learning curve are illustrated in Figure 16, Figure 17, Figure 18, and Figure 19 respectively.

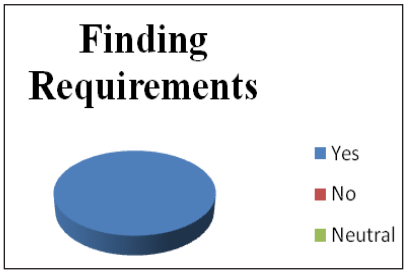

Figure 16. Finding Requirements

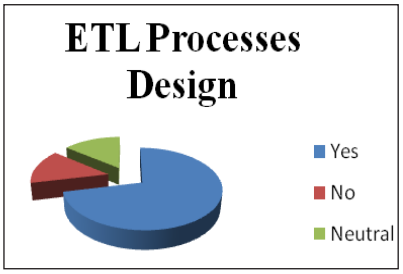

Figure 17. ETL Processes Design

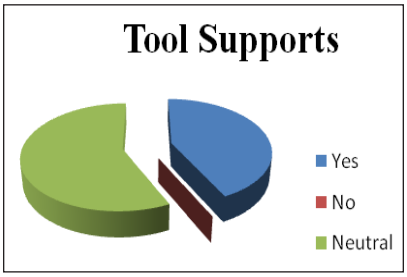

Figure 18. Tool Supports

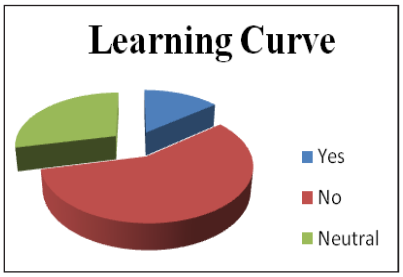

Figure 19. Learning Curve

The implementation in the real environment will be challenging because of the complexity of the DW model and requires longer time for learning the RAMEPs. Nevertheless, the RAMEPs approach enables the DW developers to model the DW system from the early phases to the generation of the ETL processes, which currently no specific tools have supported.

However, the RAMEPs has facilitated most of the important activities in the DW system development, especially in the ETL processes design.

ANALYSIS OF RESULTS

The results have shown that the ETL processes specifications can be derived from the early stages of user requirements. The ETL processes specification represents the data transformation of utility billing for producing the information of *Sale Volume and Revenue* and *Billing and Customer Status*. The ETL processes specification can be further translated into SQL statements or applied to any ETL tool for the DW system implementation. However, it is out of the scope of this paper. The sequence of the ETL processes executions follow the results as produced in the generation process. Therefore, the execution order may not necessarily follow the sequences of the ETL processes activities. However, the best practices still depend to the developer efforts, experiences, and knowledge.

Importantly, by properly analysing the requirements within the organization, decision-maker, and developer perspectives, the main components of the ETL processes were successfully captured. This is shown in the Gas Malaysia case study. The DW experts have reviewed the RAMEPs and positively support the method to be implemented in the real DW environment. Most of the experts believe that the adoption of this method can help developers to define the initial requirements prior to the detailed design of the ETL processes, and accelerate the development of the DW systems. Furthermore, the use of ontology has helped developers to resolve semantic heterogeneity problems during data integration and transformation.

Obviously, there are no easy ways to map high-level user requirements to the DW design model, especially on the design of the ETL processes. Most of the previous approaches such as the ERD-based (Kimball and Ross, 2002), UML-based (Lujan-Mora, 2005), and adhoc-based (Rizzi, 2007) were not provided the adequate formalisms and techniques to derive the ETL processes specification from the design model that was built from the users' goals. The RAMEPs is not only capable of deriving the DW schemas and the ETL processes from the users' goals, but also it has been resolving the design-related problems during the designing and generating of the ETL processes specification. This new approach can help an organization to reduce the DW project failure and support the advancement of the requirement analysis tools for the ETL processes.

CONCLUSION

The goal-ontology approach founded on the RAMEPs has proven that the ETL processes specifications can be designed from the early phases of the DW

system development. The methodology used in analysing the user requirements has been validated by the compliant tools (i.e. DW-Tool and Protégé-OWL) successfully. The evaluation approach was carried out by implementing the RAMEPs into the Gas Malaysia DW system. The DW experts have reviewed the RAMEPs and certainly support the method to be implemented in the real DW environment by using proper tools and training. By adopting this method, the developers can define clearly the ETL processes activities prior to the detailed design of the DW system. This can accelerate the implementation of the DW systems. Furthermore, the goal and ontology paradigm had helped a developer to resolve user requirements ambiguity and semantic heterogeneity problems during data integration and transformation. Thus, the ETL processes specifications can be easily generated and implemented.

REFERENCES

Abdullah, R., Selamat, M. H., Ibrahim, H., Chulan, U. A. U., & Nasharuddin, N. A. (2009). Semantics representation in a sentence with concept relational model (CRM*). Journal of Information and Communication Technology, 8*, 55–65.

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 & Sons.

Allemang, D., & Hendler, J. (2008). *Semantic web for the working ontologist*: Elsevier Science.

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.

Cysneiros, L. M., Werneck, V., & Yu, E. (2004). *Evaluating methodologies: A requirements engineering approach through the use of an exemplar*. 7th Workshop on Autonomous Agents.

Giorgini, P., Rizzi, S., & Garzetti, M. (2008). Grand: A Goal-oriented approach to requirement analysis in data warehouses. *Decision Support Systems, 45*, 4–21.

Hutter, D., Stephan, W., Baader, F., Horrocks, I., & Sattler, U. (2005). Description logics as ontology languages for the semantic web.

Mechanizing Mathematical Reasoning, 2605, 228–248: Springer Berlin / Heidelberg.

Inmon, W. H. (2002). *Building the data warehouse* – (3rd ed.). John Wiley & Sons.

Kimball, R., & Caserta, J. (2004). *The data warehouse ETL toolkit. Practical technique for extracting, cleaning, conforming and delivering data*: Indianapolis.: John Wiley Publishing.

Kimball, R., & Ross, M. (2002). *The data warehouse toolkit - The complete guide to dimensional modeling* (2nd ed.). John Wiley & Sons.

Lamsweerde, A. V. (2010). *Requirements engineering - From system goals to UML model to software specifications*: John Wiley & Sons.

Lujan-Mora, S. (2005). *Data warehouse design with UML* (Unpublished doctoral dissertation). University of Alicante.

Ogawa, H., Kumeno, F., & Honiden, S. (2008). *Model checking process with goal oriented requirements analysis*. 15th Asia-Pacific Software Engineering Conference.

Rizzi, S. (2007*). Conceptual modeling solutions for the data warehouse.* Idea Group Inc, 1–26.

Shen, G., Huang, Z., Zhu, X., & Zhao, X. (2006). *Research on the rules of mapping from relational model to OWL*. OWLED'06, Athens, Georgia (USA).

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. (2007). Ontology-based conceptual design of ETL processes for both structured and semi-structured data. *Semantic Web & Information Systems*, *3*(4), 1–24.

Ta'a, A., Abdullah, M. S., & Norwawi, N. M. (2010). RAMEPs: A goal-ontology approach to analyze the requirements for data warehouse systems. *WSEAS Transactions on Information Science and Applications*, *7*(2), 295–309.

Yu, E. (1995). *Modeling strategic relationships for process reengineering* (Unpublished doctoral dissertation). University of Toronto.