RAMEPS: A GOAL-ONTOLOGY APPROACH TO ANALYSE THE REQUIREMENTS FOR DATA WAREHOUSE SYSTEMS

Azman Taa, Mohd Syazwan Abdullah, Norita Md. Norwawi
Graduate Department of Information Technology
College of Arts and Sciences, Universiti Utara Malaysia
06010 UUM Sintok, Kedah
MALAYSIA
{azman, syazwan, norita}@uum.edu.my

Abstract: - The data warehouse (DW) systems 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 still have limitations on translating the business semantics for DW requirements toward the ETL processes specifications. This paper proposes the Requirement Analysis Method for ETL processes (RAMEPs) that utilize ontology with the goal-driven approach in analysing the requirements of ETL processes. A case study of student affair domain is used to illustrate how the method can be implemented.

Key-Words: - Requirement Analysis, ETL Processes, Data Warehouse, Ontology, Business Intelligence

1 Introduction

DW is a system for gathering, storing, processing, and providing a huge amount of data with analytical tools to present complex and meaningful information for decision makers. These data are collected, stored, and accessed in centralized databases in order to sustain competitiveness in businesses [1]. However, the DW system is dependent on the ETL processes to provide the data [2]. In other words, the success of DW system is dependent on the design of ETL processes. There are many issues in requirement, modeling, and designing the ETL processes due to the non-standardization of methods imposed by the providers through their own DW tools. Moreover, the design tasks need to tackle the complexity of ETL processes from early phases of DW system development. An early phase is important to ensure the satisfaction of information for the DW systems [3].

The complexity of ETL processes always refers to the problem of generating the transformations for data sources toward the DW structure. These transformations involve the reconciliation semantic of user requirements and data source schemas [4]. Generally, an ambiguous definition of user requirements occurs because the users are unable to define their requirements precisely and clearly [1]. Moreover, various meanings of data (i.e. attributes, tables) makes it difficult for integrating the user requirements to the data sources. Thus, reconciliation the appropriate semantic of user terms and data sources are important in generating the transformations accordingly. Generating the transformations are about designing the ETL processes from early phases of DW system development. This should be based on the systematic method for analysing the user requirements toward generating the ETL processes accordingly. However, current method is incomplete due to the limitations and linkages in modeling and designing the DW systems. Clearly, these limitations have contributed to the failure of DW projects [3][20]. Therefore, we propose the RAMEPs, a requirement analysis method based on goal-ontology approaches.

This paper is structured as follows: related work is described in the section 2. Section 3 and 4 explains our approach on RAMEPs, while section 5 discusses a case study on how RAMEPs can be used. Section 6 shows how the case study is implemented on a Jena 2 framework. Finally, section 7 concludes the work and proposes the future research direction.

2 Related Literature

The designing of ETL processes is essential for helping the developer to develop the DW system from the early phases of system development. Due to the heterogeneity problems, the tasks to manage and develop the ETL processes become difficult, tedious and complex. The emergence of ontology as the main artifacts of semantic web technology has been used in resolving the heterogeneity problems in information sharing environments [4]. The ontology has been used to reconcile the semantics

within database integration, especially in DW system environments [5]. Moreover, the database schemas can be modeled as an ontology model with respect of the complexity in ontology construction. Therefore, an effort to simplify these tasks is important through the ETL tools that support the multipurpose data integration platform together with the ontology.

Generally, software design requires unambiguous, complete, verifiable, consistency and usable user requirements that support data analysis and decision-making processes [6]. However, the work of capturing and analysing the user requirements are not an easy task because it involves various levels of users, departments and organizations. Additionally, in DW systems, the tasks should have involved analysing the goals, resources, realities, and rules that affecting the DW structure and ETL processes specifications. The research efforts on developing software requirements [16] and DW requirements [3][6][17] according to the requirements engineering guidelines have been carried out by researchers. In short, their approaches on DW requirements analysis can be classified in Table 1.

Table 1. The DW 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 et al. (2007), Giorgini et al. (2008),	Goal-driven
Mazon et al. (2007)	Model-driven
Romero and Abello (2007), Skoutas and Simitsis (2007)	Ontology-driven

[3] has applied goal oriented approach in designing the DW structure without extended to the ETL processes. Meanwhile, [5] has elaborated the design of ETL processes by using ontology without mentioning how the user requirements are provided. Therefore, this research will fill the gap by developing the method that applied the goal-ontology approaches to design the ETL processes from early phases of DW system development.

3 Goal-Ontology for ETL Processes Requirements

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 analysed from the organization and decision-maker perspectives [3]. We argue an analysing the DW requirements from the abstract of user requirements toward the detail of ETL processes are important in tackling the complexity of DW system design. This widely accepted that the early requirement analysis significantly reduces the possibility misunderstanding of user requirements [7]. The higher understanding among stakeholders possibly increases the agreeable about terms and definitions used during the ETL processes execution. Therefore, our requirement analysis method for ETL processes (RAMEPs) is centered on the organizational and decisional modeling and focuses on the transformation model from the perspective of a developer. By adapting the approach used by [3], the model of our method is presented in Figure 1.

Our extended works in the RAMEPs model are highlighted in the shaded area. 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 by decision makers and related to facts, dimension, and measures. The developer modeling is defined the related actions for the data sources and business rules given.

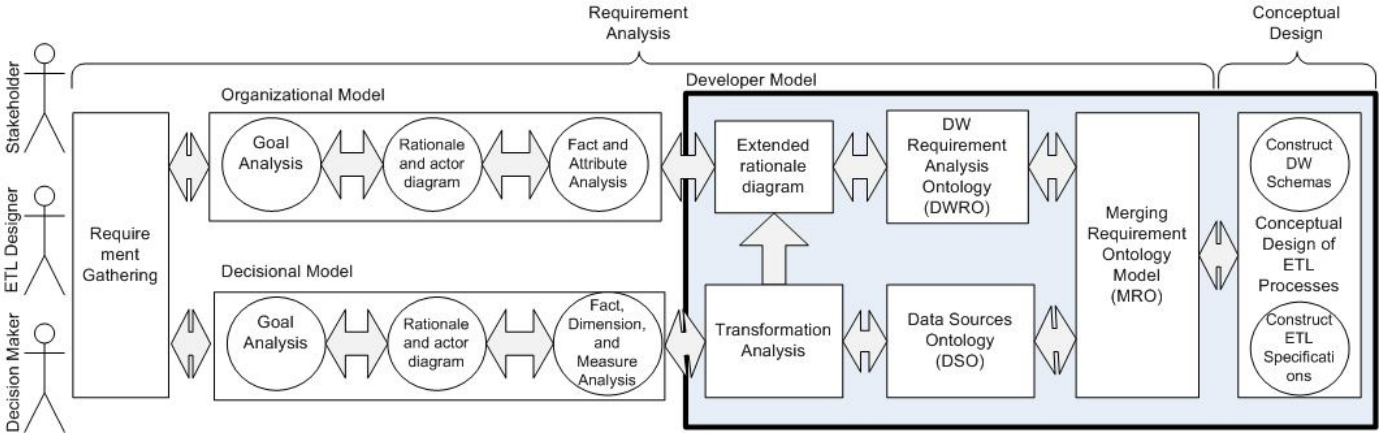

Fig. 1. The RAMEPs

3.1 Organization Modeling

Organizational modeling consists of three (3) different analyses, which are produce in the iterative process. The analyses are: i) goal analysis, which the actor diagrams and rationale diagrams are produced; ii) fact analysis, which the goal rationale diagrams are extended with facts; and iii) attributes analysis, which the fact rationale diagrams are extended with attributes. All goals, facts, and attributes are defined in the context of organization views.

3.2 Decision Modeling

Decision modeling consists of four (4) different analyses, which also produce in the iterative process. However, these analyses are focused on the goal of a decision maker which 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 requires in supporting the decision making.

3.3 Developer Modeling

Developer modeling consists of three (3) different analyses, which also produce in the iterative process. The analysis is focused on the goal of a decision maker which 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 rules analysis, which produces the lists of business rules and constraint for related facts; and iii) transformation analysis, which extends decision-goal diagram with transformation activities and rules involved. The transformation analysis based on plan modeling in Tropos methodology. The Developer modeling explains the facts about actions and rules applied toward the data sources in the perspectives of ETL developers. The Developer modeling will complete the goal-driven analysis of user requirements in order to produce the final informational model for DW system.

4 The RAMEPs Tasks

The RAMEPs is based on the Tropos methodology that was developed from the well-accepted i* conceptual framework of software development [7]. The aim is to provide the decisional information from the perspective of organizational, decision-maker, and developer. The goal oriented requirement analysis will determine the components of DW structure through diagrams. The 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 developer model that related to actions and business rules. These will help the developer to generate the appropriate actions for populating the data sources toward the DW. In summary, all activities in RAMEPs are presented in Table 2.

Table 2. The RAMEPs Tasks

Steps	Activities	Methodology
1	Gather and elicit requirements with stakeholders.	Interview, and document analysis
2	Analyse requirements based on the organization perspectives.	Tropos Goal-oriented
3	Analyse requirements on the decision-maker perspectives.	Tropos Goal-oriented
4	Analyse requirements on the developer perspectives.	Tropos Goal-oriented
5	Ontology construction for requirement analysis.	RDF/OWL Ontology model
6	Ontology construction for data sources.	RDF/OWL Ontology model
7	Map and merge the requirements ontology with the data sources ontology.	RDF/OWL Ontology model
8	Refine the merging ontology to fully satisfy the user requirements.	RDF/OWL Ontology model
9	Construct the required ETL specifications from the merging ontology.	RDF/OWL Ontology model, Jena 2 Framework

An approach proposed by [3] was adopted in order to analyse the requirements from the perspective of decision-makers and organizations. However, the approach was not covering the analysis on data transformation that belongs to the intention of ETL developers. For the next paragraph, we will explain how the Tropos methodology can be used in analysing the requirements of the data transformation needed by the ETL processes as stated in step 4 of RAMEPs.

4.1 Transformation Analysis

Developer perspectives are required beside the organization and decision-maker perspectives to compliment the need of requirements analysis for ETL processes. As comparable, the outcome each of the perspectives can be presented in Table 3.

Table 3. Outcome of the Analysis Perspectives

Perspectives	Outcomes	Notes
Organization	- List of Facts - List of Attributes	Represent the main data in organization and comprises most relevant attributes as exist in data sources.
Decision-Maker	- List of Facts - List of Dimensions - List of Measures	Represent decision-maker needs, summarizing role played in 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.

Developer modeling is about modeling the transformation analysis which is deals with the specification of actors and goals at the low level. The actors and goals which already defined in organizational and decisional modeling will be further explored with transformation analysis to produce the developer modeling. The Plan approach in Tropos methodology is used to present the outcome of the analysis. Metamodel of the plan approach is presented in the Figure 2.

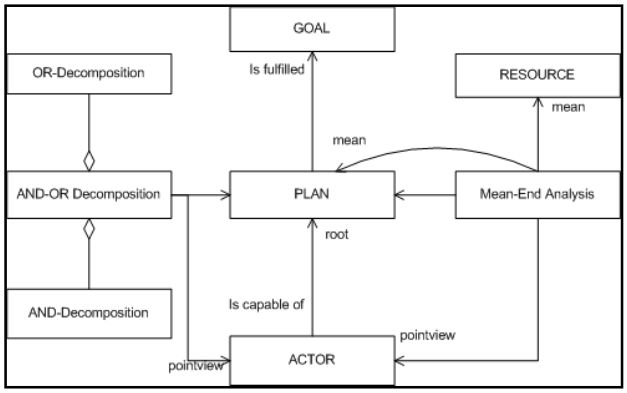

Fig. 2. Plan Modeling Metamodel

In metamodel, for each actor and goal, plans and tasks should determine the goals achievement [11]. The same analysis techniques such as MEAN-END and AND/OR are used to produce the developer actor diagram and extended Developer actor diagram. This task will introduce new actors, tasks, resources that compliment to the goals. Inclusion new actors will contribute positively to fulfill the requirements. A developer diagram of new actors, tasks, and resources that support the goals of each fact will comply with the DW requirements components (i.e. dimension, measure). These new actors can be classified into three common types of transformations namely extract, transform, and loading (ETL).

To begin the transformation analysis, the final goals of facts as defined in the rationale diagram of decision-maker are selected. Then, next tasks to questions are:

- What plans needed to achieve the goals with supporting by the related dimensions and measures?
- Who actors to execute the plans as defined?
- Which plan to execute to achieve the goals?

In order to answer these questions, an understanding of a knowledge domain is necessary to analysis the goals with related dimension and measures. By using MEAN-END and AND/OR analysis techniques, the plan needed to fulfill the goals are $plan_1$, $plan_2$, … $plan_n$. Normally, plan is represented by the hexagon symbol as shown in Figure 3.

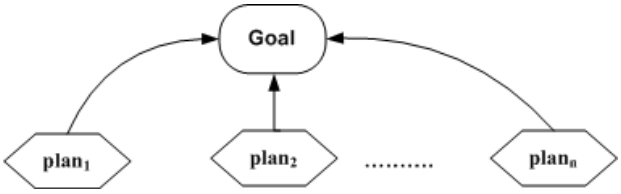

Fig. 3. Plan Modeling

Further analysis is to state the actors who will execute the plan as defined in plan modeling. Obviously, the actors are extract, transform, and loading that represent by the business rules on each of the related plans. The business rules are defined by the users and can have more than one rule to support the plan. However, the plan is unnecessarily containing the business rules. The business rules are represented by the circle symbol as shown in Figure 4.

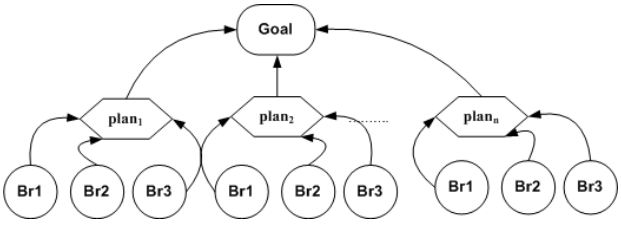

Fig. 4. Plan Modeling with Business Rules

Previously, the business rules and related actions are gathered and determined during the requirement gathering and producing the documentation organized in templates. The template is defined in column form (fact, action, and business rule) and used to record the information. Based on the name of measure in decisional diagrams and supporting by the defined plans and business rules, the ETL

developer can suggest suitable aggregation and population operators to use. For example, if the name of measure is AMOUNT, then the appropriate operator might be SUM or EVERAGE. The business rules will state the conditions within the actions toward applying the aggregation operators for fulfilling the ETL processes execution.

The glossaries of facts, dimensions, measures, business rules, and actions will be produced at the end of the analyses task and used for designing the conceptual of ETL processes. However, these glossaries need to be mapped to the corresponding data sources. The mapping process should be based on a unify model (i.e. ontology) to reduce the uncertainty and clearly the meaning of the glossaries [13]. Indeed, the semantic heterogeneity problems should be resolved along the mapping process take place. This paper provides a case study to evaluate the proposed method.

4.2 Ontology for Requirements Glossaries

The organizational, decisional, and developer models have determined 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 by the users. This will be 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 will be mapped to the data sources in the heterogeneous environments, the semantic heterogeneity problems will remain occurs 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 [4].

In this section, we explain the process for constructing the DW requirements ontology (DWRO) for semantically described 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 transformation and integration process. The strong linkages between requirement glossaries and appropriates data sources through ontology model will possibly produce the ETL specifications automatically. This can be done through invoking an appropriate algorithm and reasoning. In particular, the used of ontology is based on description logic (DL) which is constituting the most commonly used of knowledge representation formalism [15]. This research is using OWL language for knowledge representation that adopts the DL formalism. The Resource Description Framework (RDF) is used together with OWL in presenting the data structure.

The DWRO should be capable to model the following type of information: i) the concepts of the domain; ii) the relationships between concepts to attributes; and iii) the attributes and relationship belong to each attribute. The concepts referred to the facts, whereas the dimensions, measures, business rules, and actions are referred to the attributes. The relationship between concepts and attributes referred to *hasDimension*, *hasMeasure*, *hasAction*, and *hasBusinessRules*. 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 are necessary to specify. However, the RDF/OWL features need to be suited for the high level presentation since all the terms defined are in the abstracted form. For this purpose, we used the RDF/OWL features and ontology notation that also used by [5]. The RDF/OWL features used in our approach are shown in Table 4.

Table 4. OWL features

Notation	Name	Description
C	Class	Classes represent the concepts of the domain being modeled.
$C_1 \equiv C_2$	Equivalent	To state that two classes are equivalent
$C_1 \sqsubseteq C_2$	subClasssof	To create class hierarchies
$C_1 \sqcap C_2$	disjointWith	State that two classes two C1 dan C2 are disjoints
$C_1 \sqcup C_2$	unionOf	The union of two classes
P	Property	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.
$\geq$nP, $\leq$nP	mix/max cardinality	Specifies the min/max cardinality of a property

The DW requirements contain facts (F), dimensions (D), measures (M), business rules (Br), and Actions (Ac). This explains that the DW requirements contain Facts with the set of dimensions, set of measures, set of business rules, and set of actions. In ontology, facts, dimensions, measures, and actions are defined as set of classes, whereas business rules and relationships among them are defined as set of properties. The relationship is the link between class to class, class to property, and property to property. As described in ontology definition, set of axioms used to assert subsumptions between classes are defined from the business rules and actions. The business rules specify the domain and range properties, cardinality constraints, disjointness class, and others. The actions defined a new class for aggregation functions used for each fact. Formally, the DWRO can be defined:

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

F = Facts
D = Set of Dimensions (D_1, D_2, …. D_n)
M = Set of Measures (M_1, M_2, …. M_n)
Br = Set of Business Rules (Br_1, Br_2, .. Br_n)
Ac = Set of Actions (Ac_1, Ac_2, …. Ac_n)

The type of class values is not defined in DWRO because the values were not available yet at this time.

4.3 Ontology for Data Sources

The method of semantic mapping from a relational model to RDF/OWL is adapted to facilitate the transformation of data sources into RDF/OWL based structure [9]. The task to transform the data sources to the data sources ontology (DSO) is known semantic reengineering of the legacy information system. These tasks are as follows:

– Apply the reverse-engineering approach to define the conceptual model of existing data sources system through any modeling tools (e.g. PowerDesigner).
– Construct the ontology structure by using semantic mapping rules. The ontology tuple will consist of concepts, relations, function, axioms, and instances: O = (C, R, func, A, I).
– The ontology structure will be constructed by using Protégé-2000. The Protégé-2000 is used because of its freeware and ability to produce OWL/RDF automatically.

The concepts represent semantic of data sources are established. Since the data sources are heterogeneous, the basic mapping principles applied as follows:

– One or more similar relations R_i is mapped to one related concept C_i.
– Primary-foreign relationship R_i is mapped to property P_i.
– Tuple of a relation R_i is mapped to an instance I_i

Generally, the overall workflow of data transformation can be shown in Figure 5. The transformation process only supports the schemas level of data sources.

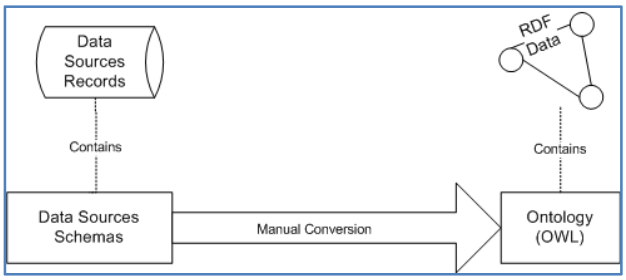

Fig. 5. The Data Sources to RDF/OWL Workflow

Figure 5 shows the transformation of data sources schemas to the ontology structure in manually manner. As far as our knowledge, no procedure or tools available to transform the relational database schemas (i.e. represented in UML) toward the ontology structure (i.e. represented in OWL) automatically. Thus, the manually mapping of data sources schemas to the RDF/OWL structure is applicable to this research since the RDF/OWL corresponding to the UML/OWL profile that can be generated automatically by the protégé-2000.

Formally, the DSO is constructed according to generic ontology model as DSO = (C, P, A, I) which comprising the following:

- $C = C_c \cup C_t \cup C_g$, where 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 is the union of a set of classes that used to be presented different kinds of values for a property that corresponds to an attribute of a concept.
- $P = P_p \cup$ {hasDimension, hasMeasure, hasBusinessRule, hasAction), where P_p is a set of properties that represent attributes of the concepts or relationships.
- A = set of axioms used to assert subsumption relationships between classes, specify domain and range properties, specify cardinality constraints, assert disjointness class, and define a new class.

- I = instance of the concept that presents the values of the ontology tuple.

As mention in Table 4, the RDF/OWL features were adopted for defining and instantiating the ontologies. The RDF/OWL language can be used to represent the meaning and relationship of terms in data sources schemas. These features were adopted from Ontology Definition Metamodel (ODM) document [18] as recommended by World Wide Web Consortium (W3C).

4.4 Mapping the Requirements and Data Sources

The need of mapping the DW requirements toward the associated data sources are important in order to construct the single view of ontology. The different view of the ontology model (i.e. DWRO and DSO) in the same domain is known as heterogeneity in the ontologies [10]. Since the heterogeneity problems in data sources have been tackled via ontology representation of data sources, then the same approach has been applied in mapping mechanism. However, the mapping ontologies are supported from the domain knowledge of user requirements and application knowledge of existing application system.

The DWRO should be able to describe the semantics of the user requirements toward the semantics of data sources in order to establish mapping between classes and properties. Furthermore, the process of mapping is possibly implemented by the appropriate software and tools with the reasoning functionality. The DWRO has modeled the required information according to the following elements:

- The concepts of the domain
- The relationships between the concepts
- The attributes characterizing the concepts
- The different representation format or value for each of the attributes
- The restriction impose by attributes or relationships

These elements can be represented in the ontology structure such as {concept ↔ classes}, {relationship ↔ properties}, {type of format or value ↔ classes in the hierarchy}, {specific element in ETL setting ↔ new classes}, and {restrictions ↔ axioms}. Based on these representations, the characteristics of DWRO and DSO can be mapped as shown in Table 5.

The ontology mapping elements can be described as following: i) fact is defined as a concept of required information; ii) conceptname is defined as a concept of required data; iii) concept is referring to a class; iv) an attribute and a relationship are referring to a property; v) constraint or restriction is referring to an axiom; and vi) individual is referring to an instance. Based on the mapping results, new classes and properties pertaining to the merging requirement ontology (MRO) will be produced. These new classes and properties will be captured the knowledge of ETL processes such as aggregated, aggregation, range, table, formation, and others. The type of knowledge is explained in Table 6.

Table 5. DWRO and DSO Mapping

DWRO elements	**DSO elements**	**Ontology mapping elements**
Fact	-	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 = $Action_1(attr_1, attr_2, … attr_n)$, m_2 = $Action_2(attr_1, attr_2, … attr_n)$ M_n = $Action_n(attr_1, attr_2, … attr_n)$	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, … ca_n)	Behavior/ Constraint	Axiom: ac_1..ac_n ↔ [ConceptName ↔ m_1.. m_n]
-	Data	Instance/Individual

Table 6. Description of New Classes

Type of Knowledge	**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, EVERAGE	Represent the calculation for the measure
Table	RETRIEVE,	Represent the

	LOADING	accessing and pushing the data
Formation	CONVERSION	Represent the transformation of one format to another

These new classes need to be organized accordingly into MRO. Again, this task is done through Protégé-2000. This process will be finished until the ontology structure is reconstructed and rechecking by using reasoner (e.g. Pellet). New RDF/OWL document can be produced to represent the entire specification of ETL processes. The RDF/OWL code will be used to determine the appropriate ETL processes. However, before this can be done, some refinement on the MRO needs to be carried out in order to ensure the ETL processes will fully satisfy the DW formats and structures. Through the reasoning process, the inferred MRO is semantically organized in presenting the knowledge of ETL processes [5]. Therefore, by using semantic web programming (i.e. Jena 2 Framework), the ETL processes specifications can be produced for designing the ETL processes. This will be explained in section 6.

4.5 Refinement the MRO

This step is important to ensure the ontology model presenting the accurate mapping between DW requirements and appropriate data sources. To ensure the semantic mapping is reasonably accurate, the following tasks need to be carried out:

- Recheck the facts whether it can be merged or splitted. In case of merged, the facts should have a common goals and supported by same dimensions. For splitting, the otherwise method is applied.
- Reorganize the *attributes* and *dimensions* whether need to be dropped, added or updated according to applicability within the analysis tasks.
- Recheck the *measures* whether it can be merged or splitted into different kind of aggregation methods (e.g. SUM -> AVERAGE).
- Replan the actions whether it can be merged, splitted, dropped or added according to changes in business rules or goals.

The refinement and adjustment on the MRO is implemented in the particular diagrams of organizational, decisional, or developer model. Then, the changes will be updated into MRO using Protégé-2000.

5 Case Study

The RAMEPs is validated through DW-Tool for goal-oriented analysis and Protégé-2000 with Pellet reasoner for ontology model. However, the explanation on the validation process is not discussed in this paper. We focused on the evaluation process of RAMEPs, which is carried out in the real case study of academic domain.

5.1 Step 1 – Requirement Elicitation

The requirements elicitation process is based on structured interviews with the stakeholder (i.e. Director of Academic Affairs Department (AAD), System Analyst) and study on current system documentations, which focusing on goal-oriented business processes. Based on the results of the interview, the university goals are identified and explore details on the goals of AAD in supporting the university main goal. The university goals can be shown in Figure 6.

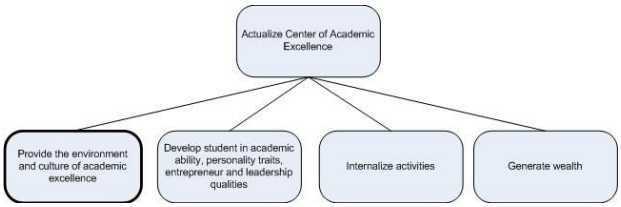

Fig. 6. University Goals

To simplify the process in proposed approach, the reference example will focus on the subject area of student affairs. Based on Figure 6, the sub-goal provides the environment and culture of academic excellence is relevant with the business tasks of AAD. Thus, the next tasks of requirement analysis will be focused on this sub-goal. The scenario of AAD that need the information, which support the said goal can be described as follows:

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

The analysis task commences with 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 represented them by using an actor diagram. An Actor diagram explains about dependencies among actors (i.e. stakeholders such as AAD, student, lecturer) in university as presented in Figure 7. The analysis on the actor diagram produce the requirements documentation

that organized in three difference template namely: main actor (actor, objectives), sub-actor (sub-actor, type, goals), and dependencies (depender, dependee, goal).

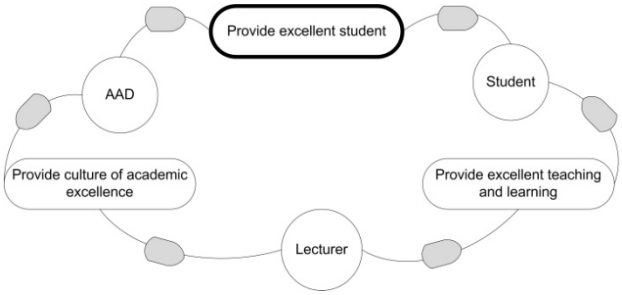

Fig. 7. Actor Diagram for AAD

In Figure 7, the scenario of student affairs in supporting the AAD and university goal is applied for both under-graduate and post-graduate students, even though a few business processes of the post-graduate system are not similar with the under-graduate student system. However, both systems will support the goal of AAD and university.

5.2 Step 2 – Analyse Requirement on the Organizational Perspectives

The next task is to analyses detail about the goals from the organizational perspectives and building the rationale diagram for each actor. The goals are analyses using AND/OR decomposition and contribution analysis for connecting the dependencies among actors. The rationale diagram for the university actors focusing the goal *provide the excellent student* is as shown in Figure 8.

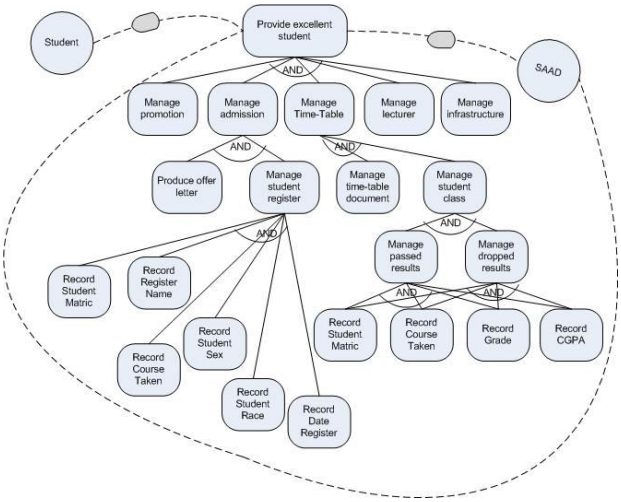

Fig. 8. Rationale Diagram for AAD Actors from Organizational Perspectives

The goal is decomposed into sub-goal *manage promotion, manage admission, manage time-table, manage lecturer, and manage infrastructure* by AND/OR decomposition technique. Further analysis decompose goal *manage student register* into another six (6) sub-goals *record student matric, record register name, record course taken, record student gender, record student race,* and *record date register*. For the goal *manage student class*, it decomposed into *manage passed results* and *manage dropped results*. By using AND/OR decomposition technique, both goals *manage passed results* and *manage dropped results* were decomposed into *record student matric, record course taken, record grade* and *record CGPA*.

The next step is to implement fact analysis that aims to identify all the relevant facts for the AAD. The analysis is carried out by identifying the facts (i.e. information or process) for each goal from top toward the down leaf goals in the goal hierarchy. The analysis of attributes is carried out after the relevant facts have been identified. The aim of attribute analysis is to identify the appropriate attributes that given value when facts are recorded. The possible attributes were explored without specifying their role as dimensions or measures. Importantly, these attributes values should be associated with the goal from the perspective of AAD and university. The facts and possible attributes (e.g. matric, name, course) are shown in Figure 9.

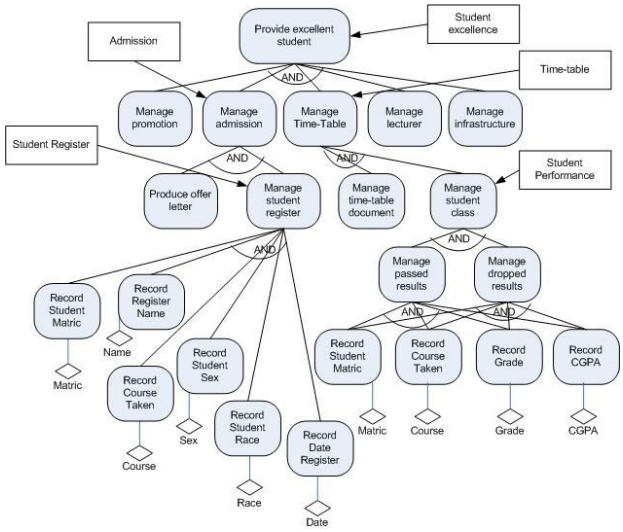

Fig. 9. Extended Rationale Diagram for AAD Actors from Organizational Perspectives

5.3 Step 3 – Analyse Requirement on the Decisional Perspectives

Requirement analysis shifted to decisional modeling, which is focusing on the decision-maker perspectives. The works are surrounding the goal for decision maker and understand how the DW system can support the process of decision making. The analysis process starts with identifying actors in goal analysis, and extended the facts in fact, dimension, and measure analysis. In goal analysis, the actors and their associate dependencies are

defined and presented in actor diagram. Applying the same approach in organization goal analysis, the goal of decision maker decomposed into sub-goal accordingly. The rationale actor diagram of decision-maker can be shown in Figure 10.

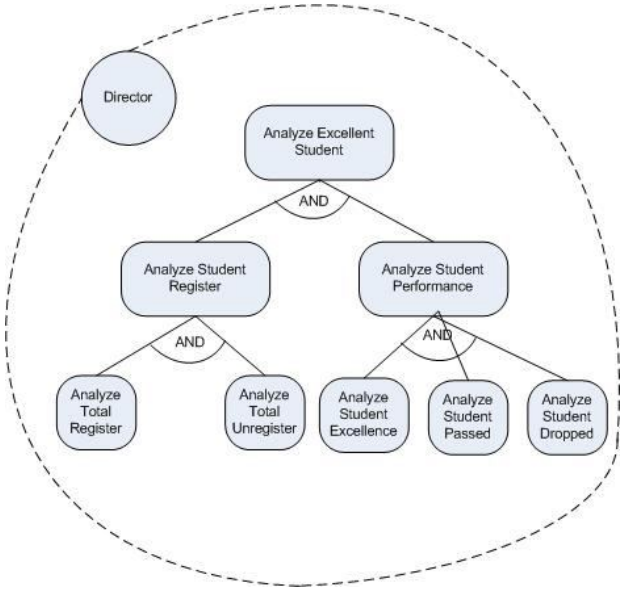

Fig. 10. Rationale actor diagram for AAD from the decisional perspectives

The AAD Director (AADD) is one of the main decision makers that require the information about student registers and performances in each of an academic session. The focus goals associated to AADD (e.g. analysis student register, analysis student performance) is decomposed into sub-goals such as *analysis total register*, *analysis total unregister*, *analysis student excellence*, *analysis student examination.* In fact analysis, the relevant facts are connected to the decision goal. As carried out in organization modeling, facts are defined as business process required by the goal to be fulfilled. Normally, facts are imported from the extended rationale diagrams that produced in organizational modeling. All the relevant facts can be shown in Figure 11.

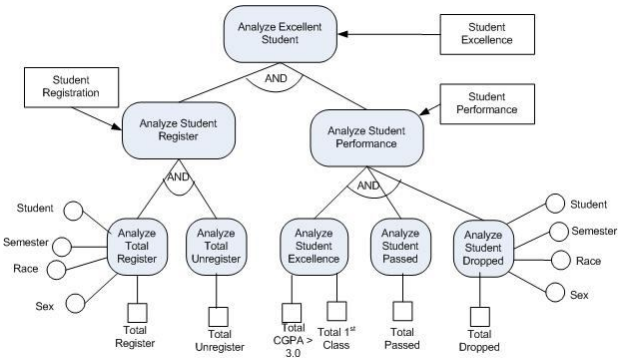

Fig. 11. Extended rationale diagram for AADD from the decisional perspectives

In dimension analysis, each related fact is connected to appropriate dimension. The dimension is defined as required information that supporting the decision goal and it identified by analysing each of goals associated to the upper level goals. In simplify the process, the information gathered in the analysis is recording based on templates (goal, fact, dimensions) and (dimension, description). The list of dimensions is identified as presented in Figure 11. After dimension analysis, the measure analysis is carried out by analysing the goals and facts associated to the upper level goals. However, all the tasks (i.e. dimension and measure analysis) require a clear interaction with decision makers in order to capture the right information for the analysis. The list of measures is also presented in Figure 11.

In [3] method, the requirement analysis process is ended at this stage. The knowledge of facts, dimensions, attributes, and measures will be used in further design of DW. However, the extended analysis on data transformation that related to defined facts, dimensions, and measures need to be carried out in order to ensure the successful implementation of ETL processes. Therefore, the analysis on transformation activities is explained in the developer perspectives.

5.4 Step 4 – Analyse Requirement on the Developer Perspectives

In transformation analysis, the relevant plans are connected to the decision goals. The plans are presented in abstract definition of ETL processes. By using MEANS-END and Contribution Analysis, the abstract of ETL processes can be determined. There are no actions to determine for data source schemas toward the DW structure at this time. However, as the transformation analysis is carried out, the facts, dimensions, attributes, and measures can be used to determine the actions as required by goal to be fulfilled. All tasks in transformation analysis required a clear understanding of decision makers in order to define suitable transformation activities on the ETL processes. The plans for *Student Performances* goal presented in Figure 12, and *Student Registers* goal presented in Figure 13.

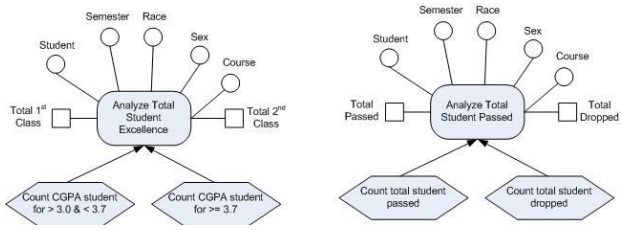

Fig. 12. Transformation analysis for Student Performances

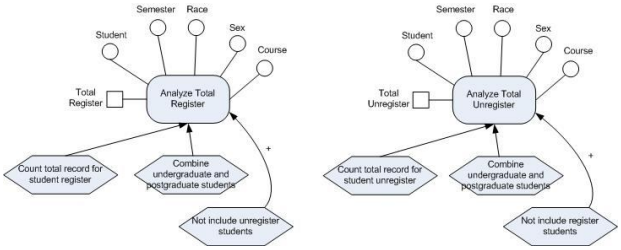

Fig. 13. Transformation analysis for Student Register

The goal of *Analyse Total Student Excellence* is based on the facts about student performances. In order to provide information for the goal, appropriate plans were decomposed into two: i) Count CGPA student between 3.0 and 3.7; and ii) Count CGPA student for 3.7 and above. These plans are proposed to achieve the goal of *Analyse Total Student Excellence*. The rest of the examples explained the analysis for each of the goal. Finally, the extended rationale diagram for AADD will be completed when each of the decision-goal contain plans that support the information required by decision makers. This shows in Figure 14.

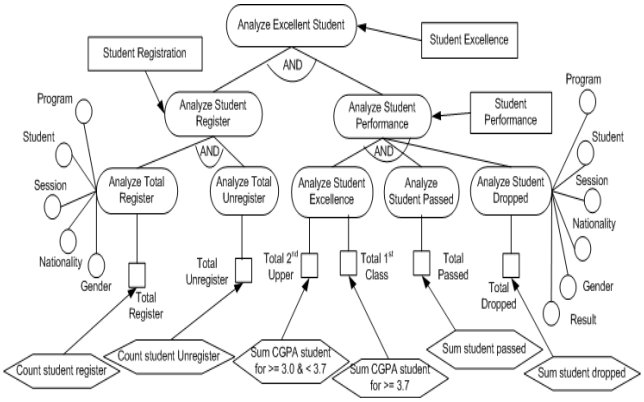

Fig. 14. Final Diagram of DW Requirements

5.5 Step 5 - DW Requirements Ontology

In this step, the DW requirements were transformed into ontology (i.e. DWRO) based on the final diagram of requirement analysis. The DW components or glossaries (i.e. facts, dimensions, measures, attributes, and actions) were modeled according to ontology structure that present the conceptual design of ETL processes. These glossaries are based on the final diagrams of DW requirements model as presented Figure 14.

Based on our ontology model defined as O = (F,D,M,Br,Ac), the DWRO is constructed. Three classes have been identified as *Total Register*, *Total Student Passed*, and *Total Student Excellence*. Then, each of the class contains properties such as *student, semester, race, sex, program, total register, merge for post-under graduate, 1st Class, 2nd Class*. Intuitively, the properties are representing the dimension, measure, or attribute in DW structure. An axiom described the relationship between classes such as hasDimension, hasMeasure, hasAttribute, and hasAction. Moreover, these axioms also included the business rules applied (e.g. "student must be Malaysian") and actions (e.g. aggregation – COUNT for the number of student registered). By using Protégé-2000, the constructed DW requirements ontology is shows in Figure 15.

5.6 Step 6 - Data Sources Ontology

The data sources ontology model defined from two different applications that are Academic Student Information System (ASIS) and Graduate Student Information System (GAIS). The concepts of Student, Gender, Session, Program, Nationality, and Result were introduced to reconcile the agreeable semantics among the data sources. This can be viewed in Figure 16.

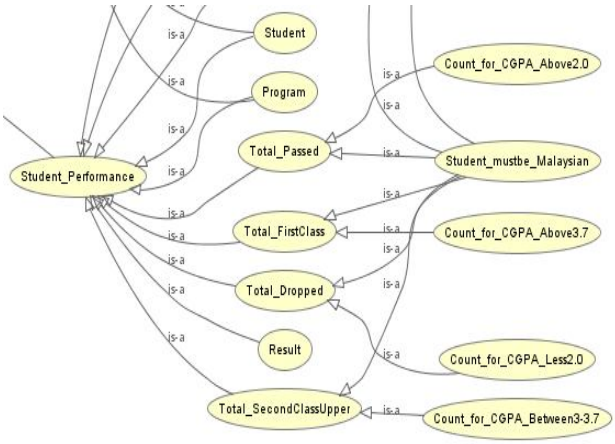

Fig. 15. The DWRO

Fig. 16. The DSO

Consequently, the semantics mapping between data sources to the DW requirements can be established during the mapping process. Therefore, the semantic heterogeneity problems can be resolved prior to the generation of ETL processes specifications.

5.7 Step 7 - Merging Requirement Ontology

The construction of MRO is depended on the mapping between DW requirements and data sources. This involved the identification of similarity and dissimilarity of concepts (i.e. DWRO and DSO) and their associate attributes toward the data sources as follows:

- Concept ↔ Classes (e.g. Student Register, Student Examination)
- Relationship ↔ Properties (e.g. hasDimension, hasMeasure)
- Type of format or value ↔ Classes in the hierarchy (e.g. currency – RM, Dollar)
- Specific element in DW setting ↔ new Classes (e.g. SUM, COUNT, MERGE)
- The restriction ↔ Axioms (e.g. "Student must be Malaysian")

Based on our definition, the mapping between DWRO and DSO is shown in Table 3.

Table 3. DWRO and DSO Mapping

DWRO elements	DSO elements	The mapping elements
Fact (Student Register)	-	Concept: Student Register
Dimension (Student, Semester, Course, Sex, Race, Result)	Concept: Student (t210student, t801studmas) Concept: Gender (t012jantina, t801jantina) …	Student ↔ Concept Student Semester ↔ Concept Session Course ↔ Concept Program Sex ↔ Concept Gender …
Measure (Total student register, Total student Unregister)	Concept: Student (t210student, t801studmas) *- Total student unregister unable to count from Student.	[Total student register] ↔ [Student record]
Business Rule (Student must be Malaysian)	Concept: Student (t210student, t801studmas), Concept: Nationality (t016warga, t016warga)	[Student must be Malaysian] ↔ [Student (t210student, t801studmas), Nationality (t016warga, t016warga)]
Action (COUNT Student Register, SUM Student passed, Student dropped, Student 1st Class, Student 2nd Class, FILTER for Student must be Malaysian)	Concept: Student (t210student, t801studmas), Concept: Nationality (t016warga, t016warga)	[COUNT for Student Register] ↔ [Student (t210student, t801studmas) [SUM for Student passed] ↔ [Result (t312result_exam, t804result) …

Through mapping definition, the mapping setting in Protégé-2000 can be defined as guidelines to develop the MRO. These mapping setting can be written as followed:

MERGE DS_1, DS_2
Classes Student : asis:t210student ∪ gais:t801studmas
Classes Gender : asis:t012jantina ∪ gais:t801jantina
Classes Session : asis:t005term ∪ gais:t005term
…
MergeSources: hasMergeStudent **SOME** Student, hasMergeGender some Gender
…
hasMergeStudent(Domain:Student, Range:t210student, t801studmas)
hasMergeGender(Domain:Gender, Range:t012jantina, t801jantina)
…
AGGREGATE (COUNT) for Student Registered
∀hasMeasureRegister ← Total_Registered
hasMeasureRegister **ONLY** Total_Registered
…
Based on mapping mechanism, the MRO is derived as shown in Figure 17.

Fig. 17. The MRO

5.8 Step 8 – Refinement the MRO

In refinement the MRO, all the classes for fact, dimension, measure, attribute, action, and business rule were recheck for the correctness and consistency. For example, plan for *Total student Unregister* unable to count from the Student concept because of the required attributes was absent from the MRO (refer to Table 3). In this scenario, we have to discard the plan because it is not important to the decision maker. The refinement tasks finished after all of MRO classes was rechecked.

5.9 Step 9 – Construct ETL Specifications

Producing the ETL processes specifications are the main aim for this research. Using ontology as knowledge representation of DW structure and ETL operations create the possibility of producing the ETL processes specifications within the scoping of DW structure. These tasks can be realized through manipulation the semantic annotation of user requirements and data sources. The same approach has been proposed by [5], but our works anticipate on early tasks of ETL processes by setting the data stores (i.e. DW and data sources) from the analysis of user requirements. Thus our method will propose set of ETL processes specifications for transforming the data sources to a DW according to user requirements as determined in the goal-oriented analysis approach.

The ETL processes tasks comprise the process of extract, transform, and loading. Therefore, most of the generic frequently tasks used by the ETL processes shown in Table 4.

Table 4. The Generic ETL Processes

ETL Processes	Actions
RETRIEVE()	Retrieve the data from data sources
EXTRACT()	Extract the data from retrieving data sources
FILTER()	Filters the data from retrieving data sources
MERGE()	Merge two or more set of data sources
CONVERT()	Convert set of data sources to another format or type
AGGREGATE()	Aggregate the data sources into some criteria via some functions
JOIN()	Join two data sources related to each other by some attributes
LOADER()	Loads data into the DW

As stated in MRO, the knowledge about information as required and their related data sources has been defined according to RDF/OWL based language. Thus, the MRO will be processed according to an appropriate reasoning in order to identify and proposed a set of ETL processes specifications for designing the conceptual of ETL processes. The power of reasoning is based on inferencing mechanism in ontology that dealing with the wide range of elaborates processing of information.

Our method is based on the RDF/OWL data model that contains nodes (i.e. subject and object) and arcs (i.e. links between nodes) that represent by OWL visual graph (OWLViz). The nodes and arcs formed statements that comprises of subject, predicate and object that always known as triples. The MRO contains set of RDF/OWL triples which is can be read and manipulated. The procedure to read and manipulate the RDF/OWL statements is developed and achieved the objectives:

- Identify nodes/classes and arcs/properties and listed for their purposes in tabular form which is contains subject, predicate and object.
- Recheck the mapping nodes that represent the DW requirements classes and nodes represent the data sources. These nodes need to hold the following conditions in order to remain applicable:
  - DWRO classes and DSO classes must have a common superclass that explained about the particular records or data. The semantics of both classes are related.
  - DWRO classes and DSO classes are not disjoint that explained about the constraints of both classes are not contradict to each other.
- Identify nodes/classes and arcs/properties and listed for their purposes in tabular form which is contains subject, predicate and object.
- Examine the pair of nodes/classes and their related arcs/properties. This process identified each of class and their respective properties.
- Reasoning will be used on classes and their related properties to derive the ETL processes specifications accordingly.
- The ETL processes specifications will be rearranged according generic ETL processes tasks as shown in Table 4.

Based on these objectives, the formal algorithm is developed for deriving the ETL processes specifications from the MRO. The MRO become an input and the ListOfETL variable becomes an output for the algorithm. In summary, the algorithm works:

- MRO nodes are reading and defined as classes.
- If exist one class subsume of another class and that class is an aggregation class, then determine the relevant aggregator operator.

- If exist two classes are joining, then determine the MERGE or FILTER operator.
- If exist two classes are changes on each other, then determine the CONVERT operator.
- Repeat the above process until the end of the class groups in the MRO hierarchy.
- All the determined operators will be added in the ListOFETL.

The proposed algorithm that adapted from [5] is presented in Figure 18.

```
Input: MRO
Output: ListOfETL
Begin
  Ci ← Class corresponding to MRO nodes i
  Ci+1 ← Class corresponding to MRO nodes i+1
  IF (Ci ⊆ Ci+1)
  { Foreach class Ci to Ci+j  {
   IF (∃Ci: Aggregate (Ci, Ci+1)) {
    ListOfETL ← add AGGREGATE FUNCTIONS
                  (Ci, Ci+1)}
    ELSE { IF (∃Ci: MergeSource (Ci, Ci+1))
           {ListOfETL ← add MERGE (Ci, Ci+1) }
           ELSE
           {ListOfETL ← add FILTER (Ci, Ci+1) }}
    }
    ELSE
      IF (∃ (C1, C2): Ci ⊆ C1 AND Ci+1 ⊆ C2 AND
         ConvertTo (C1, C2)
         { ListOfETL ← add CONVERT (C1, C2)
          Repeat foreach class Ci to Ci+j }
      ELSE {Ci ← for classes Ci ⊆ Ci+1
          Repeat foreach class Ci to Ci+j }
      }
  }
End.
```

Fig. 18. Algorithm for ETL Specifications

```
<!http://www.semanticweb.org/ontologies/2009/1/G
oalRequirementOntology.owl#Student -->
<owl:Class
rdf:about="&GoalRequirementOntology;Student">
    <rdfs:subClassOf
rdf:resource="&GoalRequirementOntology;Student_
Performance"/>
    <rdfs:subClassOf
rdf:resource="&GoalRequirementOntology;Student_
Registration"/>
</owl:Class>
<!http://www.semanticweb.org/ontologies/2009/1/G
oalRequirementOntology.owl#Student_Performance
-->
<owl:Class
rdf:about="&GoalRequirementOntology;Student_Pe
rformance"/>
<!http://www.semanticweb.org/ontologies/2009/1/G
```

Fig. 19. A snippet of MRO

6 Implementation

To generate the ETL processes specifications, a prototype of application for reading, and manipulating MRO has been developed by using Java programming. The MRO structure that representing by RDF/OWL language as shown in Figure 19 is manipulated through Jena 2 Framework that runs by Eclipse platform. By using an algorithm as proposed in Figure 18, the ETL processes specifications can be generated. A part of the results from the prototype application is shown in Figure 20.

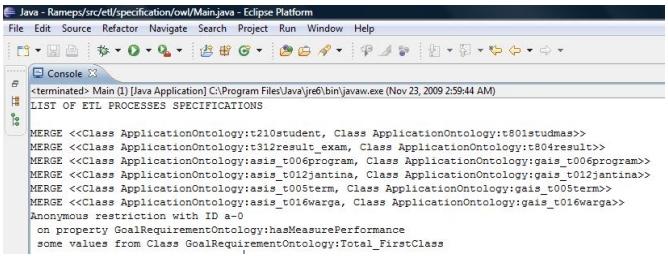

Fig. 20. List of ETL Processes Specifications

The results have showed that the ETL processes specifications can be derived from the ontology model of user requirements. The ETL processes specifications can be further translated into SQL statements or applied directly into any ETL tools for implementing the ETL processes for DW system.

7 Conclusion

The RAMEPs has proven the ETL processes specifications can be derived from the early phases of DW system development. The methodology used in analysing the user requirements has been validated by DW-Tool and Protégé-2000 successfully. Indeed, current work is detailing the process on the evaluation of the RAMEPs. The evaluation approach is carried out by implementing the RAMEPs into various domains of case studies. This will gives the multi views of information in DW systems.

Further works will be completing the application prototype and finalize the validation and evaluation process. We believe the adoption of our method can help developers to define clearly the ETL processes prior to the detail design of DW systems. The ontology model helps a developer to resolve semantic heterogeneity problems during data integration. Moreover, the RDF/OWL language is easy to use and maintain and make the design of ETL processes specifications can be managed easily even the changes in user requirements are frequently occurred.

References:

[1] Inmon, W.H., *Building the Data Warehouse - Third Edition*. 2002: John Wiley & Sons, Inc. 97.

[2] Kimball, R. and J. Caserta, *The Data Warehouse ETL Toolkit. Practical Technique for Extracting, Cleaning, Conforming and Delivering Data*. 2004: Wiley Publishing, Inc., Indianapolis. 491.

[3] Giorgini, P., S. Rizzi, and M. Garzetti, GRAnD: A Goal-Oriented Approach to Requirement Analysis in Data Warehouses. *Decision Support Systems*, 2008. 45: p. 4-21.

[4] Alexiev, V., et al., *Information Integration with Ontologies: Experiences from an Industrial Showcase*. 2005: John Wiley & Son Ltd. 180.

[5] Skoutas, D. and A. Simitsis, Ontology-Based Conceptual Design of ETL Processes for Both Structured and Semi-Structured Data. *Semantic Web & Information Systems*, 2007. 3(4): p. 1-24.

[6] Bruckner, R.M., B. List, and J. Schiefer. *Developing Requirements For Data Warehouse Systems With Use Cases*. in 7th Americas Conference on Information Systems. 2001.

[7] Yu, E., *Modeling Strategic Relationships for Process Reengineering*, in Department of Computer Science. 1995, University of Toronto.

[8] Bresciani, P., et al., Tropos: An Agent-Oriented Software Development Methodology. *Kluwer Academic Publishers*, 2003: p. 1-40.

[9] Shen, G., Huang, Z., Zhu, X., & Zhao, X. *Research on the Rules of Mapping from Relational Model to OWL*. Paper presented at the OWLED'06, Athens, 2006, Georgia (USA).

[10]Aleksovski, Z. *Using Background Knowledge in Ontology Matching*, 2008, Vrije University.

[11]G. Lucian, *Developing Plans for Attaining Goals*, in 8th WSEAS International Conference on E-Activities (E-Activities '09), Puerto De La Cruz, Tenerife, Canary Islands, Spain, 2009, p. 234-236.

[12]L. Niedrite, D. Solodovnikova, M. Treimanis, and A. Niedritis, *Goal-Driven Design of a Data Warehouse-Based Business Process Analysis System*, in 6th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases Corfu Island, Greece, 2007, p. 243-249.

[13]A. Salguero, F. Araque, and C. Delgado, *Spatio-Temporal Ontology Based Model for Data Warehousing*, in 7th WSEAS International Conference on Telecommunications and Informatics (TELE-INFO '08), Istanbul, Turkey, 2008, pp. 125-130.

[14]T. Iqbal and N. Daudpota, *XML Based Framework for ETL Processes for Relational Databases*, in 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, 2006, pp. 481-485.

[15]F. Baader, I. Horrocks, and U. Sattler, Description Logics as Ontology Languages for the Semantic Web, in *Mechanizing Mathematical Reasoning*. vol. 2605/2005: Springer Berlin / Heidelberg, 2005, pp. 228-248.

[16] Nuseibeh, B. and S. Easterbrook. *Requirements Engineering: A Roadmap. in The Future of Software Engineering*. 2000. Limerick, Ireland.

[17] Winter, R. and B. Strauch. *Information Requirements Engineering for Data Warehouse Systems*. in ACM Symposium on Applied Computing. 2004.

[18]Object Management Group, Inc. *Ontology Definition Metamodel*. 2007.

[19]J.-N. Mazon, J. Pardillo, and J. Trujillo, *A Model-Driven Goal-Oriented Requirement Engineering Approach for Data Warehouses*, in RIGiM, Auckland, New Zealand, 2007, pp. 255-264.

[20]S. Lujan-Mora, Data Warehouse Design With UML, in *Software and Computing Systems*: University of Alicante, 2005, p. 291.

[21]O. Romero and A. Abelló, *Automating Multidimensional Design from Ontologies*, in DOLAP'07, Lisboa, Portugal, 2007.