EIAW: Towards a Business-friendly Data Warehouse Using Semantic Web Technologies

Guotong Xie[1], Yang Yang[1], Shengping Liu[1], Zhaoming Qiu[1], Yue Pan[1], and Xiongzhi Zhou[2]

[1]IBM China Research Laboratory
Zhongguancun Software Park, Beijing, 100094, China
{xieguot, yangyy, liusp, qiuzhaom, panyue}@cn.ibm.com
[2]Taikang Life Insurance Company
No.156, FuXingMen Street, Beijing, 100032, China
zhouxiongzhi@taikanglife.com

Abstract. Data warehouse is now widely used in business analysis and decision making processes. To adapt the rapidly changing business environment, we develop a tool to make data warehouses more business-friendly by using Semantic Web technologies. The main idea is to make business semantics explicit by uniformly representing the business metadata (i.e. conceptual enterprise data model and multidimensional model) with an extended OWL language. Then a mapping from the business metadata to the schema of the data warehouse is built. When an analysis request is raised, a customized data mart with data populated from the data warehouse can be automatically generated with the help of this built-in knowledge. This tool, called Enterprise Information Asset Workbench (EIAW), is deployed at the Taikang Life Insurance Company, one of the top five insurance companies of China. User feedback shows that OWL provides an excellent basis for the representation of business semantics in data warehouse, but many necessary extensions are also needed in the real application. The user also deemed this tool very helpful because of its flexibility and speeding up data mart deployment in face of business changes.

1 Introduction

Data warehousing and business intelligence (BI) are key technologies for decision making in the industry. A typical BI application deployment process usually requires an existing Enterprise Data Warehouse (EDW), which integrates enterprise-wide data from multiple autonomous heterogeneous data sources and provides a consistent single view of data. A multidimensional model and its corresponding data mart schema can be designed based on business user's analysis requirement. Then, relevant data from EDW are transformed and loaded into the data mart and/or a cube for doing the analysis. The analysis results and reports are finally delivered to the business user.

However, in practice, it proves very difficult to successfully implement the above process [14]. Given the fact that building an enterprise-wide data warehouse is a very time-consuming and expensive activity, to make satisfactory return from this investment is a key issue for the success of BI+EDW system. The typical BI application deployment process is not flexible enough to deal with a fast changing dynamic business environment. For example, about every 2 working days, a new analysis requirement is submitted to the BI department from business units of Taikang Life Insurance Company. For all these analysis requirements IT workers must communicate with business people, understand the business content and redesign the data mart schema and the ETL processes from the EDW to data mart.

This difficulty is mainly due to the fact that the business semantics [6] is only kept in BI designer's mind, then it is hard-coded for physical implementations. The business semantics can be, at least partially, represented by the business metadata that provides a business-oriented description of the data warehouse content and a formal representation of the analysis requirement [8,9]. The two basic types of business metadata are conceptual enterprise data model and multidimensional model. Conceptual enterprise data model is a model to organize business terminology in a semantic way. It is a view of *how the business works* and consists of business concepts, attributes of concepts and relationships among concepts. Multidimensional model is a model to define the analytic requirements for BI application. It is a view of *how the business is measured* and consists of measures and dimensions.

In the typical deployment process, the business metadata becomes technical artifacts which cannot be understood by business users. The conceptual enterprise data model is hidden behind the schema of EDW. The intended meaning of the measures and dimensions are implemented by the ETL process from EDW to the data mart. There is no distinction between transformations needed for the business semantics and transformations which are mainly due to the technical issues.

Fortunately, with the emergence of Semantic Web [2], the formal ontology representation language OWL (Web Ontology Language) [10] has been standardized by W3C. OWL is appropriate for representing business semantics in a formal way [9]. Therefore，we develop a tool to make data warehouses more business-friendly by adopting the Semantic Web technologies. This tool is called Enterprise Information Asset Workbench (EIAW). The main idea is to make business semantics explicit in the data warehouse system by formally representing the business metadata with an extended OWL language. In our tool, the conceptual enterprise data model is expressed by W3C's Web Ontology Language OWL, in particular, OWL-DL, and the multidimensional model is expressed by OWL-DL extended with concrete domain, predefined functions, property path expression, etc.

Based on the explicit business semantics, EIAW supports the deployment of a data warehouse-based BI application with the following steps, assuming the pre-existence of a conceptual enterprise data model and an enterprise data warehouse:

1) Business users build the analysis requirements (multidimensional model) using business terms from the conceptual enterprise data model;
2) IT users only need to build mapping from the business terms involved in the multidimensional model to the data warehouse schema;
3) The system automatically generates a customized data mart with aggregated data and an OLAP cube metadata supported by industry standard.

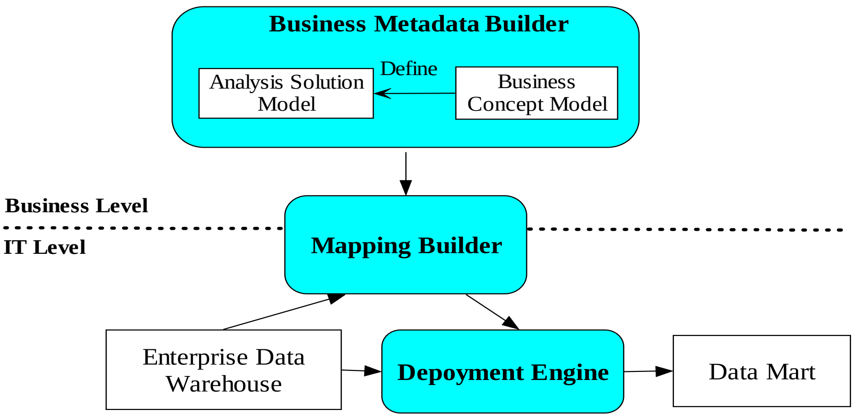
The main advantage of the above deployment process is the separation of concerns of business user and IT user. Business users can organize their business knowledge and express their analysis requirements using business terms familiar to them. IT users focus on resolving the mapping from business terms to EDW schema from the technical view and do not care the business contents of measures and dimensions. In conventional approach, IT users need to fully understand the meanings of measures and dimensions and then design schema and ETL scripts for them. They also need to redo the whole process if the measures and dimensions are changed. In our proposed approach, business users can have a more efficient way to describe what they want to get instead of arranging lots of meetings to communicate with IT people. In addition, in case that analysis requirements are changed, business users can modify the definitions of measures and dimensions, then the data mart can be automatically deployed without IT worker's engagement if the mappings are pre-built.

EIAW is implemented as a plug-in on the Eclipse 3.2 and deployed in Taikang Life Insurance Company as a result of a collaborative project between IBM and Taikang[1]. The positive feedbacks from Taikang are that the explicit business semantics by OWL greatly increases the flexibility, speeds up data mart deployment and improves the quality of multidimensional model. On the other hand the customer also thinks that OWL is not an easy-to-learn language, especially for the property restrictions.

This paper is structured as follows. Section 2 describes the system architecture of EIAW. Section 3 describes the unified representation of business metadata in the EDW system with an extended OWL language. Section 4 shows the mapping from the business terms to the EDW schema. Section 5 presents the approach to automatically deploying the data mart according to business users' requirements. Section 6 discusses the users' evaluation and feedback. Section 7 discusses the related works. Finally, Section 8 draws the conclusions and discusses the future works.

2 System Architecture

The simplified view of the system architecture of EIAW tool is depicted at **Fig. 1**. The system consists of three main modules: business metadata builder, mapping builder and deployment engine.



[1] The press release: http://www-03.ibm.com/press/us/en/pressrelease/19434.wss

Fig. 1. The System Architecture

Business Metadata Builder. The business metadata builder supports business user to create and edit the business metadata in the data warehousing environment. In this tool, the conceptual enterprise data model is called Business Concept Model (BCM) and the conceptual multidimensional model is called Analysis Solution Model (ASM). The ASM is defined using the business terms from the BCM.

Mapping Builder. The mapping builder supports IT users to build the mapping from business terms to the EDW schema. IT users only need to build the mapping for the business terms appeared in the definitions of ASMs. The mapping builder also supports the reuse and incremental building of the mappings. That is to say, the mapping for one business term can be shared if it appears in the definitions of other ASMs. With the accumulation of the mappings for business terms, there will be more and more new ASMs for which all the mappings for their business terms have already been built by others. That means these ASMs can be automatically deployed without IT people's involvement.

Deployment Engine. The deployment engine automatically generates a data mart with aggregated data populated from the EDW, provided the definition of ASM and mappings are given. The deployment engine can also generate cube metadata for this data mart to enable OLAP analysis.

3 Business Metadata

Business metadata plays an important role in the business-friendly data warehouse system. There are two types of business metadata supported in EIAW: BCM and ASM.

3.1 Business Concept Model

Enterprise Data Model plays a critical role in the planning and designing phase and is critical for the future success of the enterprise data warehouse. An Enterprise Data Model is an integrated view of the data produced and consumed across the entire organization. It unifies, formalizes and represents the things important to an organization, as well as the rules governing them. The conceptual enterprise data model is always represented by an Entity-Relationship (ER) model in industry.

Because W3C's Web Ontology Language (OWL), in particular, OWL-DL, is more expressive and has more formal semantics compared to the ER language, EIAW adopts OWL-DL to represent the conceptual enterprise data model (called BCM in EIAW). So, the BCM editor is basically an OWL editor as shown by **Fig. 2**. In the BCM editor, the names used for the terminology are concept (owl:Class), attribute (owl:DatatypeProperty) and relationship (owl:ObjectProperty), which are more familiar to the data warehouse people.

In practices, BCM can be constructed from the scratch or transformed from existing ER models in industry.

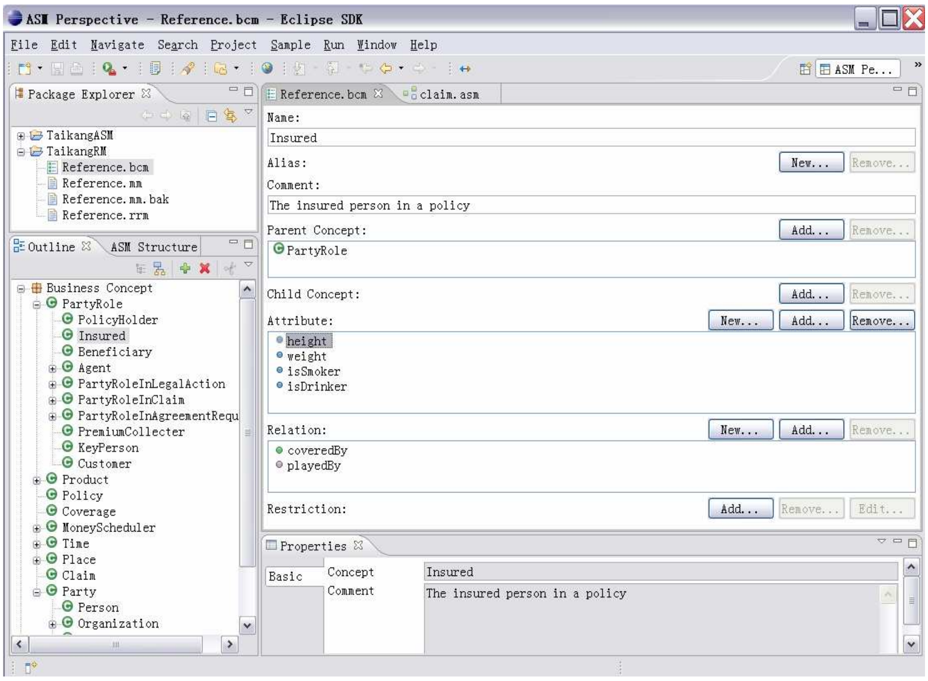


Fig. 2. The BCM Editor.

3.2 Analysis Solution Model

Multidimensional model is a kind of data model widely used for description of the multidimensional and aggregative nature of OLAP applications [3]. The basic notions are the dimension, measure and cube. A dimension represents a business perspective under which data analysis is to be performed and is organized in a hierarchy of dimension levels, which correspond to granularity of the dimension. A dimension can be organized into different hierarchies that correspond to different views of the dimension. A measure represents factual data to be analyzed. A cube associates a set of measures with some defined dimensions. For example, in a claim analysis for insurance company, some of the measures which are of interests are number of fraudulent claims, total amount paid to the claims. These measures can be viewed along several dimensions: actuarial category of the insurance products, professional risk level of the insured, age of policy holders, etc.

In EIAW, the multidimensional model is called Analysis Solution Model (ASM). The most prominent feature of ASM is that the intended meaning for the dimensions and measures are further represented in a semantic way. Different from traditional multidimensional models, all measures and dimensions in ASM are defined by using business terms in BCM. This approach makes automatic deployment of data mart possible. Our business terms include the vocabularies (the named classes and properties) from the BCM and the role expressions on the BCM. The role expression as an extension on the OWL-DL, called *property path* in EIAW, is defined as:

$$R, S \rightarrow P \mid P^{-} \mid R.S \mid C.R \mid R_{[D]}$$

where P denotes the atomic role in OWL-DL, C and D denote the named class in OWL-DL, and R and S denote the property path in EIAW. The property path enables to access an indirect property from the starting class. For example, "Agent's name" is represented in textual form as *Agent.playedBy[Person].name*, which means that: the name of an agent is got by the name of the person that plays the role of *Agent.*

From the semantic point of view, given an interpretation $\mathcal{I}$, class is interpreted as a subset of the domain $\Delta^{\mathcal{I}}$, and roles as binary relations over $\Delta^{\mathcal{I}}$, the semantics for the newly introduced constructs are: $(C.R)^{\mathcal{I}} \rightarrow R^{\mathcal{I}} \cap (C^{\mathcal{I}} \times \Delta^{\mathcal{I}})$, $(R_{[D]})^{\mathcal{I}} \rightarrow R^{\mathcal{I}} \cap (\Delta^{\mathcal{I}} \times D^{\mathcal{I}})$, $(R.S)^{\mathcal{I}} \rightarrow R^{\mathcal{I}} \circ S^{\mathcal{I}}$. It should be note that the property path is not allowed to participate in the class definitions of OWL because it is only used for the definition of ASM.

3.2.1 Representation of Measures

There are two kinds of measure in ASM. One kind is atomic measure, which defines basic variable to be evaluated in business analysis. E.g. number of claims. Another kind is complex measure, which is defined by a function on other atomic or complex measures.

The atomic measure is represented by an aggregate function on an extended expression defined on the ontology. For example, the atomic measure "Number of fraudulent claims" can be defined as:

```
NumberOfFraudulentClaims=
if(Claim.settlementStatus="FRAUDULENT")COUNT(Claim)
```

which means that the value of this atomic measure is calculated by counting the instances of the class *Claim* that satisfy the condition that the settlement status is fraudulent. Another atomic measure example is "the overall days of settlement for Individual Insurance", which can be defined as:

```
TotalDaysOfSettlement=
if(Claim.ofProduct.category="INDIVIDUAL")
SUM(Claim.settlementTime.date - Claim.requestTime.date)
```

The complex measure can be defined by a function on other measures: i.e. $m = f(m_1, ..., m_n)$, where m_i is an atomic measure or complex measure. The function expression supported in EIAW is the four arithmetic operations: addition, subtraction, multiplication and division. For example, the complex measure "Fraudulent Claims Ratio" can be defined as:

```
FraudulentRatio=NumberOfFraudulentClaims/NumberofClaims
```

The grammar of the script language to define measures is shown in **Table 1**. The formal semantics of the language is out of the scope of this paper.

Table 1. Parts of the BNF grammar for measure definition

```
Literal := String |  Number
PExpr := PropertyPathExpr | OWLClassExpr
BOper(Boolean Operator):= "and" |  "or" | "not"
COper(Compare Operator ):= "<"|"<="|"="|">="|">"
AOper(Arithmetic Operator):= "+" | "-" | "*" | "/"
Func (Predefined Functions):= currentYear() etc.
// currently only support functions in SQL
Expr ::= Expr AOper Expr | Func "(" ArgLst ")"
        | PExpr | Literal
ArgLst ::= Expr ("," Expr)*
CondExpr(Condition Expression)::=
               CondExpr(BOper  CondExpr)* | Expr COper Expr
AggFunc(Aggregation Function):= "SUM"| "COUNT"| "AVG"
Statement ::= AggFunc "(" Expr ")"
MsrName ::= String
AmDef(atomic measure) := MsrName "=" "if" CondExpr
Statement ("else if" CondExpr Statement)*["else" Statement]
CmDef(Complex measure) :=
                       MsrName "=" MsrName (AOper MsrName)*
```

3.2.2 Representation of Dimensions

In conventional approach, dimension in multidimensional model consists of a set of dimension levels with a partial order on the dimension levels to support roll-up and drill-down analysis, called level-based hierarchy. To support the OLAP analysis on the data mart, the multidimensional model is strictly adhered to the schema of the data mart. For example, every dimension level is correspondent to a column in the dimension table of the data mart, and the values of every dimension level are stored as values in the correspondent column. The strong dependency between multidimensional model and the data mart make it difficult to modify the multidimensional model to meet the changed analysis requirements. If business user wants to re-organize the dimension levels, the whole deployment process need to be redone manually, including the re-design the data mart schema and the ETL processes from EDW to data mart.

To make the data warehouse more adaptable to the business change, EIAW also supports the user-definable value-based hierarchy. In EIAW, a dimension is a virtual property, whose range is an enumeration of nodes organized as a tree-like hierarchy (the multiple hierarchy is not supported yet), called value-based hierarchy. For each node in the value-based hierarchy, it is denoted by a string literal and attached with a condition expression that defines the meaning of the node. The condition expression has the same expressivity as in the measure definition.

For example, the dimension “the professional risk level of the insured person” is defined as in the **Table 2** and **Table 3**.

Table 2. The node and its definition for the dimension

ProfessionalRiskLevelOfInsured=

Value	Condition Expression
"Level 1"	Insured.playedBy[Person].profRiskLevel=1
"Level 2"	Insured.playedBy[Person].profRiskLevel=2
"Level 3"	Insured.playedBy[Person].profRiskLevel=3
"Level 4"	Insured.playedBy[Person].profRiskLevel=4
"Level 5"	Insured.playedBy[Person].profRiskLevel=5
"High"	Insured.playedBy[Person].profRiskLevel<3
"Medium"	Insured.playedBy[Person].profRiskLevel=3
"Low"	Insured.playedBy[Person].profRiskLevel>3

Table 3. The value-based hierarchy for the dimension

The root node	The first level	The second level
ALL	"Low"	"Level 1"
		"Level 2"
	"Medium"	"Level 3"
	"High"	"Level 4"
		"Level 5"

where the *Insured.playedBy[Person].profRiskLevel* is a property path expression that denotes the professional risk level of the insured person.

The main advantage for user-definable value-based hierarchy is the decoupling of business semantics and technical implementation. Business user can modify the value-based hierarchy by business terms without consideration of the physical storage of data mart.

The UI for editing ASM is shown in **Fig. 3**. Business user can define measures by formulas, and define value-based hierarchy for dimensions using the business terms from BCM.

4 Mapping

To enable automatic deployment of data mart for an ASM, information must be provided on where to retrieve the data and what calculations should be done for the dimensions and the measures. Because the business semantics for measures and dimensions are formally defined by the business terms (including property path expressions) from BCM, IT users only need to build the mappings from the business terms to the EDW schema to resolve the technical issues, and then the deployment engine will interpret the business semantics and automatically generate the data mart schema and loading data into the data mart.

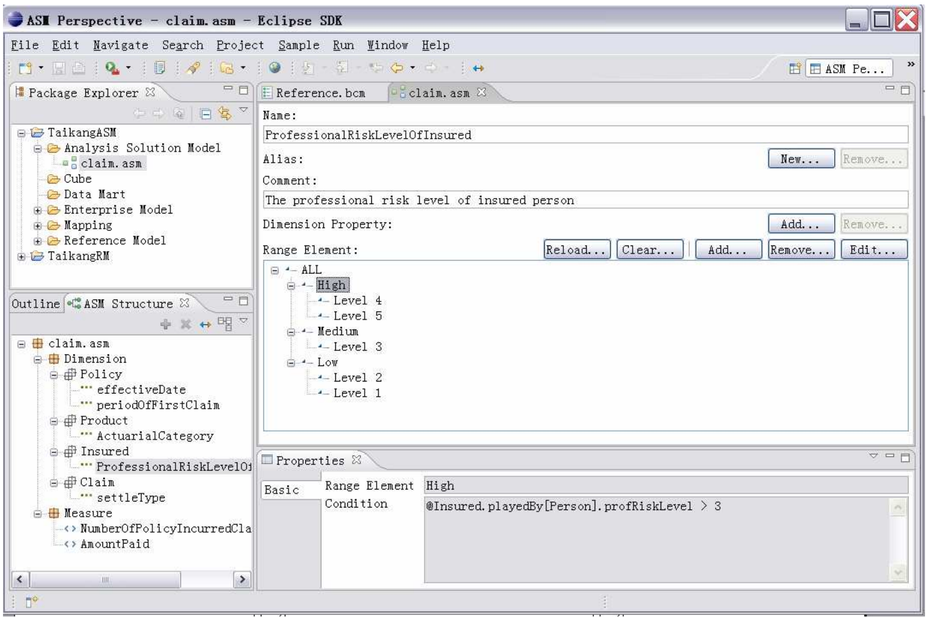


Fig. 3. The ASM Editor

The mapping from business terms to EDW schema is a set of mapping rules with 4-ary tuples as: <SOURCE, TARGET, CONDITION, TRANSLATION >, and can be written in a textual form as:

```
SOURCE :- TARGET WHERE CONDITION WITH TRANSLATION.
```

The SOURCE can be a single class or a property path. The TARGET is the schema path expression in the EDW schema. A schema path expression is a column in the logical table connected by the join operator. It can be written as:

$$Table_1.\,fk_1[Table_2],\ \ldots,\ fk_{n-1}[Table_n].column$$

The CONDITION is the definition of under what condition for the target the mapping is correct, similar to the WHERE clause in SQL, and the TRANSLATION is a function that translate the values for ontology property to the data values for columns in the EDW.

For example, the property path *Insured.playedBy[Person].profRiskLevel* can be mapped to:

```
Insured.playedBy[Person].profRiskLevel:-
F_PLCY_EVT.PLCY_PTCP_ID[D_CUST].CUST_PROF_ID[D_PROFESSI
ON].RISK_LVL
```

The mapping between EDW schema and ontology make the semantics of data in EDW explicit. The intended meaning of the mapping rule is to explain how to retrieve the data for the business term. For example, the above mapping tells the deployment engine to get the data about the professional risk level of insured person by starting from the fact table F_PLCY_EVT, following the foreign keys PLCY_PTCP_ID

(Policy Participant ID) and CUST_PROF_ID (Customer's Professional ID) to the column RISK_LVL to get the required data.

The mapping editor is shown in **Fig. 4**, which supports IT users to map the business terms appeared in ASM definitions to the EDW schema.

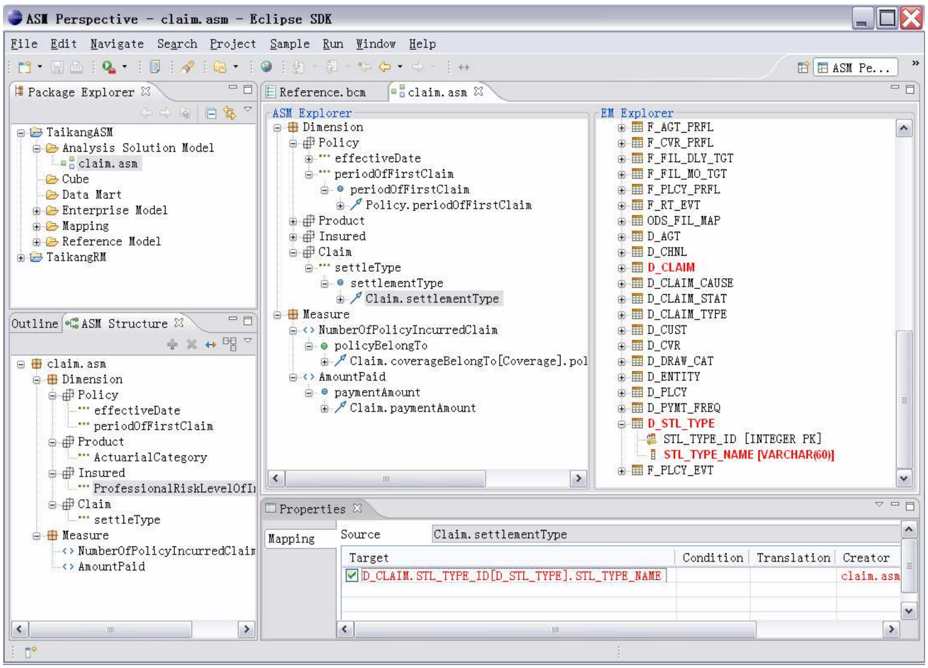


Fig. 4. The Mapping Editor. It consists of three views: the *ASM Explorer* displays the business terms for the definition of the ASM, the *EM Explorer* displays the EDW schema, the *Mapping properties view* enables the editing of the mapping formula.

5 Deployment Engine

The deployment engine automatically generates a data mart from the EDW, given the definition of the ASM and the mappings. The data mart is a customization of the EDW in that it is a subset of data from the EDW according to the analysis requirements defined in ASM. The deployment procedure includes two steps: firstly generate schema of target data mart derived from original data warehouse according to the ASM and mapping. The schema will be organized in form of star schema. Secondly, all the needed data will be loaded into the target data mart by the Date Engine. Here we only introduce the basic idea (shown in Fig. 5) and ignore the technical details due to the limitation of paper length.

Currently, we assume that the EDW schema is organized as a star schema or snowflake schema. In such case, the data loading problem for target data mart can be simplified as a problem of pruning unused data, instead of re-structuring data in the

EDW. It is a strict assumption because some EDW schemas are 3-NF schemas. However, it is a good starting point for a quick validation for the whole idea.

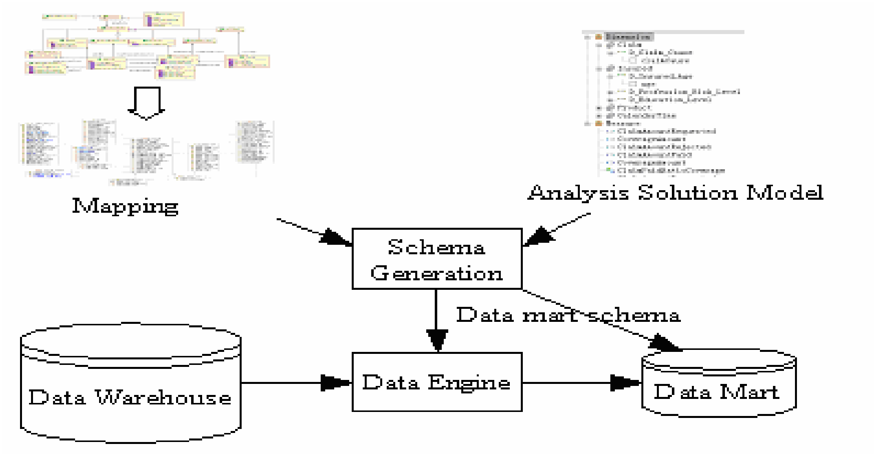


Fig. 5. The deployment of data mart

Step 1: Generating data mart schema. The goal of this step is to generate a physical star-schema for the target data mart. For every dimension in the ASM, the system will create a dimension table whose columns are the columns in EDW schema that appeared in the mapping for the business terms that defines the value-based hierarchy for the dimension. The system will create a fact table with one column as the surrogate primary key, some columns for every measure and some foreign key columns to the dimension tables.

Step 2: Loading data into the data mart. Once the target data mart schema is generated, given the mapping, the data engine will interpret the definition of ASM and extract relevant data into the dimension tables, aggregate data for the atomic measures and make calculations for the complex measures based on their definitions.

6 Tool Deployment and User's Feedback

EIAW is implemented as a plug-in on the Eclipse 3.2 and its internal operation on the BCM and ASM is based on the Integrated Ontology Development Toolkit (IODT)[2]. EIAW was deployed to the Taikang Life Insurance Company as a result of a collaborative project between IBM and Taikang Life. The work on this tool is also invited to present on the IBM Financial Services Solutions Symposium (FS3 2007)[3].

The experimental data warehouse is organized as a star-schema, consisting of 5 fact tables and 38 dimension tables. The biggest table is the fact table about policy event with about 53 millions rows of records. The other largest tables are the

[2] IODT, http://www.alphaworks.ibm.com/tech/semanticstk
[3] http://www.ibm.com/financialservices/symposium

dimension tables about coverage, customer and policy, with 30, 17 and 8 millions rows of records respectively. The overall data size is about 250 gigabytes.

Currently the built-in BCM is a customization to the business model in the industry reference model: Insurance Information Warehouse (IIW)[4]. The business model in IIW is originally an ER model with 430 entities, 437 relationships and 884 attributes. We manually selected the elements covered by the data in the data warehouse and transformed them to an OWL ontology, which consists of 76 classes, 48 object properties and 67 datatype properties. We created one ASM for claim analysis, agent performance analysis and financial analysis respectively with about average 6 dimensions and 11 measures.

In general, users are very impressed by the definitions of the measures and dimensions using business terms from OWL ontology. They also find that the mapping efforts can be dramatically reduced because only the mapping for resolving technical issues are needed and others are represented as the definitions of measures and definitions. For example, in a Financial Analysis Project, there are above 30 measures defined using only three property paths, so the only mappings are for the three property paths, other than for the 30 measures in the conventional approach. Though they need to define the 30 measures, rather than provide a simple textual name as in the conventional approach, but they indicated that the formal definitions help to clarify the meaning of measures and can improve the quality of multidimensional model. They even further request an approach for consistency checking for multidimensional model, which needs a complete formalization of the extended language and complex reasoning technologies. Another encouraging feedback is that the formal definition with less ambiguity helps for the reuse. Since the purpose of ontology is to be shared and reused for multiple applications, the measures and dimensions defined on the ontology, and the mappings for vocabularies in ontology can also be reused for multiple BI applications.

They also indicate that the tool provides an amazing solution for adapting to analysis changes for data mart deployment. After the mapping is built, they can further modify the multidimensional model, such as adjusting the value hierarchy for dimension and modifying the definitions of measures, and then re-deploy the data mart. They also suggest that the data mart can be generated incrementally if just a minor modification to the multidimensional model is made.

7 Related Works

There has been continuous works on designing and modeling of multi-dimensional mode from conceptual level [3,5,7,11]. They represent measure as a function from a set of dimension names to a data value, while dimension as a set of dimension levels with a partial order on the dimension levels to support roll-up and drill-down analysis. However, they do not further define the intended meaning of measures and dimensions in terms of business terms. Actually, the measures and dimension in conventional modeling approach are just textual names with descriptive information

[4] http://www-03.ibm.com/industries/financialservices/doc/content/solution/278652303.html

in natural language. Instead, measures and dimensions in our tool are formally defined by the business terms from the ontology using an expressive language.

The idea of describing measures and dimensions using vocabularies in conceptual model is not completely new. Muller et al. [8] proposed an approach to use UML as a uniform language for all the business metadata, including the conceptual model, the multidimensional model and the dependencies between these two models. However, the representation of business metadata is too coarse-grained to enable the automatic deployment of data mart. There are also attempts to extend the Description Logics language with multidimensional aggregation. Baader and Sattler [1] explored the extension of different Description Logics (DL) languages by concrete domains and aggregation functions over these domains and studied the decidability of satisfiability problem in these extended languages. Franconi and Sattler [4] further proposed a Data Warehouse Conceptual Data Model which allows for the description of both the relevant aggregated entities of the domain and the relevant dimensions involved in building the aggregated entities, based on DL. In EIAW, the language is much less complex and more application-specific. For example, the property path expression and the measures are not allowed to participate in the definition of class expression. But their works provide a good reference on the formal grounding for our proposed language. We also noticed that "property chain" is introduced in the recent OWL 1.1 proposal, but the property path expression is more expressive by introducing the notion of constraints on domain and range of properties, i.e. $C.R$ and $R_{[D]}$.

There are also growing interests in the introducing of Semantic Web technologies into the area of Data Warehousing and Business Intelligence. Skoutas[13] showed the usage of ontologies to enable a high degree of automation regarding the construction of an ETL design for data warehousing. Sell et al.[12] proposed a Semantic Web based architecture for analytic tools, in which the domain ontology are used to rewrite the conditions of the query to the data warehouse, in order to broaden the results of a query and to support inferences over the results of the queries. In addition, there will be a special issue on Semantic Web and data warehousing published in the International Journal of Semantic Web and Information Systems[5].

8 Conclusions

Similar to the trend that business rules and business processes are isolated from the programming codes; business semantics for data warehousing is also needed to be isolated from the data warehouse based BI application implementation and explicitly represented using a formal language. The explicit business semantics enables business users organize the business knowledge and express their analysis requirements, and enable IT users only build the mappings due to the technical issues.

Based our practices, we think OWL extended by some constructs needed for real applications, such as concrete domain, predefined functions and property path expression, is a good candidate language for expressing the business semantics for data warehousing system. However, a complete formalization of the extended

[5] The Call For Paper link: http://www.ijswis.org/cfp/semwebandwarehousing.html

language is still an open problem due to the complexity of measure and dimension definitions.

References

1. Franz Baader and Ulrike Sattler, "Description logics with concrete domains and aggregation," In Proceeding of *the 13th European Conference on Artificial Intelligence (ECAI-98)*, pp. 336-340, 1998.
2. Tim Berners-Lee, Jame Handler, and O. Lassila, "The Semantic Web," in *Scientific American*, vol. 184, no.5, 2001, pp. 34-43.
3. L. Cabbibo and R. Torlone, "A Logical Approach to Multidimensional Databases," In Proceeding of *the 6th International Conference on Extending Database Technology*, pp. 183-197, 1998.
4. E. Franconi and U. Sattler, "A Data Warehouse Conceptual Data Model for Multidimensional Aggregation," In Proceeding of *Workshop on Design and Management of Data Warehouses (DMDW'99)*, Heidelberg, Germany, 1999.
5. M. Golfarelli, D. Maio, and S. Rizzi, "The Dimensional Fact Model: a Conceptual Model for Data Warehouses," *International Journal of Cooperative Information Systems*, vol. 7, no. 2,3, pp. 215-247, 1998.
6. Dave McComb, *Semantics in Business Systems: The Savvy Manager's Guide*: Morgan Kaufmann, 2003.
7. D. L. Moody and M. A. R. Kortink, "From enterprise models to dimensional models: A methodology for data warehouse and data mart design," In Proceeding of *the 2nd International Workshop on Design and Management of Data Warehouses (DMDW)*, 2000.
8. Robert Muller, Thomas Stohr, and Erhard Rahm, "An Integrative and Uniform Model for Metadata Management in Data Warehousing Environments," In Proceeding of *International Workshop on Design and Management of Data Warehouses*, pp. 12-28, 1999.
9. Bonnie O'Neil, "Semantics and business metadata," in *The Data Administration Newsletter(TDAN.com)* 2007.
10. Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks, "OWL Web Ontology Language Reference, W3C Recommendation 10 February 2004", http://www.w3.org/TR/owl-ref/, 2004.
11. C. Sapia, M. Blaschka, G. Höfling, et al., "Extending the E/R Model for the Multidimensional Paradigm," *LNCS on Advances in Database Technologies*, vol. 1552, 1999.
12. Denilson Sell, Liliana Cabral, Enrico Motta, et al., "A Semantic Web based Architecture for Analytical Tools," In Proceeding of *the Seventh IEEE International Conference on E-Commerce Technology (CEC'05)* pp. 347-354 2005.
13. Dimitrios Skoutas and Alkis Simitsis, "Designing ETL processes using semantic web technologies," In Proceeding of *the 9th ACM international workshop on Data warehousing and OLAP* pp. 67-74, 2006.
14. Ad Stam, "A Successful Enterprise Data Warehouse? The Four Biggest Reasons for Failure," in *BI Review Newsletter* 2007.