Requirement-Driven Creation and Deployment of Multidimensional and ETL Designs

Petar Jovanovic[1], Oscar Romero[1], Alkis Simitsis[2], and Alberto Abelló[1]

[1] Universitat Politècnica de Catalunya, BarcelonaTech, Barcelona, Spain
{petar,oromero,aabello}@essi.upc.edu
[2] HP Labs, Palo Alto, CA, USA
alkis@hp.com

Abstract. We present our tool, GEM, for assisting designers in the error-prone and time-consuming tasks carried out at the early stages of a data warehousing project. Our tool semi-automatically produces multidimensional (MD) and ETL conceptual designs from a given set of business requirements (like SLAs) and data source descriptions. Subsequently, our tool translates both the MD and ETL conceptual designs produced into physical designs, so they can be further deployed on a DBMS and an ETL engine. In this paper, we describe the system architecture and present our demonstration proposal by means of an example.

1 Introduction

At the early phases of a data warehouse (DW) project, we create conceptual designs for the multidimensional (MD) schema of the DW and the extract-transform-load (ETL) process that would populate this MD schema from the data sources. These labor-intensive tasks are typically performed manually and are known to consume 60% of the time of the overall DW project [12]. Automating these tasks would speed up the designer's work both at the early stages of the project and also, later on, when evolution events may change the DW ecosystem.

Several works have dealt with MD schema modeling and they either focus on incorporating business requirements (e.g., [6,7]) or on overcoming the heterogeneity of the data sources (e.g., [8,11]). Furthermore, it has been noticed that while trying to automate this process, people tend to overlook business requirements or introduce strong constraints (e.g., focus only on relational sources [7]) that typicaly cannot be assumed. On the other hand, several approaches have dealt with ETL design using various techniques like MDA and QVT (e.g., [5]), semantic web technologies (e.g. [10]), and schema mapping (e.g., Clio [4] and Orchid [2]). However, these works do not address the problem of automating the inclusion of the business requirements into the ETL design. To the best of our knowledge, our work is the first toward the synchronous, semi-automatic generation of MD and ETL designs.

Our tool, called GEM, incorporates the business requirements into the design, all the way from the conceptual to the physical levels. The fundamentals behind GEM are described elsewhere [9]. Here, we focus on our system internals and discuss GEM's functionality through an example. In the demonstration, we will show GEM through a number of pre-configured use cases and the conference attendees would be able to interact either by changing the input requirements or by creating designs from scratch.

S. Castano et al.(Eds.): ER Workshops 2012, LNCS 7518, pp. 391–395, 2012.

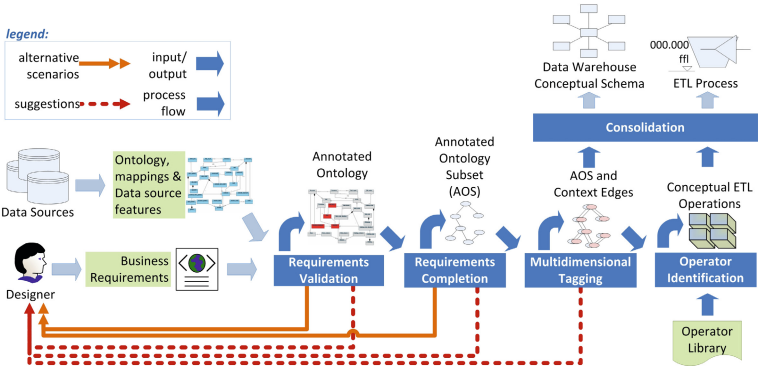

Fig. 1. System overview

2 GEM in a Nutshell

GEM uses an ontology to boost the automation of the design process and produces a conceptual MD design fulfilling the given set of business requirements. At the same time, unlike previous approaches, GEM benefits from the knowledge inferred when producing the MD schema and along with information about the data sources, it automates the production of conceptual ETL design. A high level view of how GEM operates is shown in Figure 1.

Inputs. GEM starts with the *data sources* and *requirements* representing business needs; e.g., service-level agreements (SLAs). First, it maps the data sources onto a domain OWL *ontology* that captures common business domain vocabulary and uses XML to encode the source mappings. It has been shown in [10] that a variety of structured and unstructured data sources can be elegantly represented with an ontology. In addition, the requirements expressing some business needs (e.g., "Revenue for each nation of North Europe region") are formalized by means of an extensible and generic XML structure (see Figure 3).

Stages. GEM maps each requirement to ontology concepts and further, through the source mappings, to the corresponding data sources (*requirements validation*).Then, by exploring the ontology topology, it identifies the ontology subset needed to retrieve the data concerning the requirement in hand (*requirements completion*). Next, the system produces the complete MD interpretation of the ontology subset (i.e., concepts are either *dimensional* or *factual*), validates the subset respecting MD paradigm and generates the conceptual design of the output MD schema (*multidimensional tagging and validation*). Finally, by considering the MD schema knowledge inferred during the previous stage and how the concepts are mapped to the sources, it identifies the ETL operations needed to populate the MD schema from the sources (*operation identification*).

Physical designs. After having produced the MD and ETL designs, we translate each design to a physical model. Due to space limitation, we omit the technical details (see [13]), but in the demonstration we will show how GEM connects to a DBMS for creating and accessing the MD schema and to an ETL engine for creating an ETL flow.

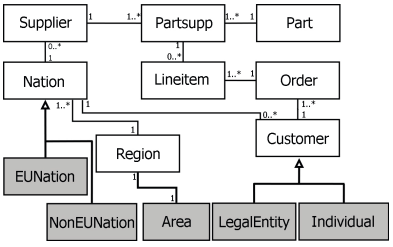

Fig. 2. Example ontology for the TPC-H schema

```
<measures>
  <concept id="revenue">
    <function>
l_extendedprice*(1-l_discount)
    </function>
  </concept>
</measures>
<slicers>
  <comparison>
    <concept id="r_name" />
    <operator>=</operator>
    <value>North Europe</value>
  </comparison>
</slicers>
<dimensions>
  <concept id="n_name" />
</dimensions>
```

Fig. 3. Example requirement in XML

Implementation. GEM is implemented in Java 1.6. We use JAXP - SAX API for parsing XML files and JENA for parsing OWL ontologies. The interface is implemented using Java Swing library. In its current implementation, GEM connects to a DBMS (like PostgreSQL) for storing and accessing database constructs and uses Pentaho Data Integration (PDI) as an ETL execution engine.

3 Demonstration Scenario

Our on-site demonstration will involve several use cases. Each case is pre-configured so that would help us demonstrate individual characteristics (e.g., variety and complexity of MD and ETL designs, variety of business requirements, and so on). However, here due to space considerations, we limit ourselves into a single use case that represents two problems typically encountered in real-world DW projects: (P1) the information at hand for data sources is incomplete and (P2) the business requirements are ambiguous.

Our example is based on the *TPC-H benchmark* [1]. First, the domain ontology (Figure 2), describing the TPC-H sources is enriched with the business domain vocabulary (shown as shaded elements in Figure 2). Then, we consider the mappings of the ontology concepts to the data sources in an iterative fashion. For (P1), we consider a mapping where the concept *nation* is not mapped to any source. In addition, we create the input XML representing business requirements. For (P2), we consider an ambiguous requirement as shown in the snippet depicted in Figure 3: *"Revenue for each nation of North Europe region"*.

During the requirements validation stage, GEM identifies requirement concepts (i.e., *lineitem*, *nation*, and *region*) as MD concepts and checks how they map to the sources. Since, the concept *nation* is not mapped to any data source the system tries to map it by looking for synonyms (1-1 relationships) and exploring concept's taxonomies inside the ontology. In this case, GEM suggests mapping *nation* through its mapped subclasses (i.e., *EUNation* and *NonEUNation*).

In the requirements completion stage, GEM identifies that due to ambiguous business requirements the concepts *nation* and *lineitem* may be related either through the concept *customer* or through *supplier*; i.e., the revenue of *customers* or the revenue of *suppliers* may be of interest to the business user. The designer is informed about this ambiguity and is asked to identify the appropriate semantics. After a path is chosen (e.g., through *supplier*), GEM produces the suitable ontology subset.

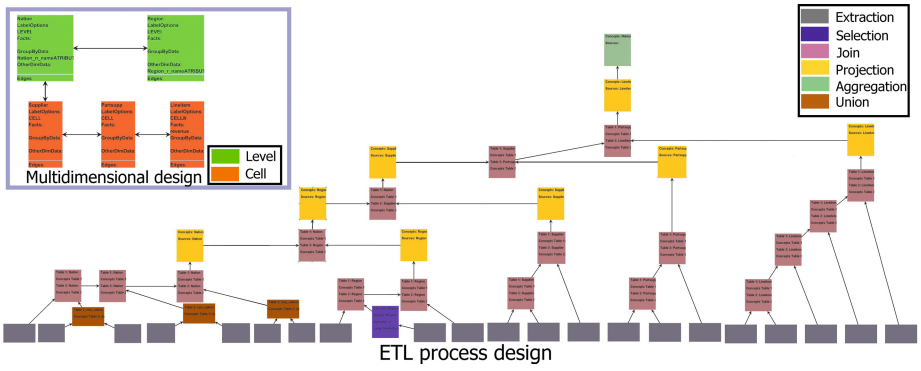

Fig. 4. GEM output designs

Next, GEM checks for a sound MD interpretation of the produced subset and eventually, produces a UML-like conceptual MD schema design fulfilling the input requirement (as shown in the top left part of Figure 4). Finally, for each mapped concept, GEM produces an *extraction* operation and, in case of derived mappings, such as *nation*, the proper operator (e.g., *union*) over the corresponding *extraction* operators (e.g., *EUNation* and *NonEUNation*) is added. Similarly, the slicer on *region* is translated as a *selection* operation and the remaining ETL operators (e.g., *joins*, *projections*, and *aggregations*) needed to produce the data cube described by the ontology subset are also added. Figure 4 shows the ETL design for this case.

The interested reader may see a detailed walkthrough of this use case with snapshots of the tool in a web page we have set up (see [3]). In the web page, we also show the corresponding physical designs for both MD and ETL designs.

References

1. TPC-H, `http://www.tpc.org/tpch/spec/tpch2.14.0.pdf`
2. Dessloch, S., Hernández, M.A., Wisnesky, R., Radwan, A., Zhou, J.: Orchid: Integrating schema mapping and etl. In: ICDE, pp. 1307–1316 (2008)
3. GEM snapshots, `http://www.essi.upc.edu/~petar/demo.html`
4. Haas, L.M., Hernández, M.A., Ho, H., Popa, L., Roth, M.: Clio grows up: from research prototype to industrial tool. In: SIGMOD Conference, pp. 805–810 (2005)
5. Muñoz, L., Mazón, J.N., Trujillo, J.: Automatic generation of etl processes from conceptual models. In: DOLAP, pp. 33–40 (2009)
6. Nabli, A., Feki, J., Gargouri, F.: Automatic construction of multidimensional schema from olap requirements. In: AICCSA, p. 28 (2005)
7. Romero, O., Abelló, A.: Automatic Validation of Requirements to Support Multidimensional Design. Data Knowl. Eng. 69(9), 917–942 (2010)
8. Romero, O., Abelló, A.: A framework for multidimensional design of data warehouses from ontologies. Data Knowl. Eng. 69(11), 1138–1157 (2010)
9. Romero, O., Simitsis, A., Abelló, A.: *GEM*: Requirement-Driven Generation of ETL and Multidimensional Conceptual Designs. In: Cuzzocrea, A., Dayal, U. (eds.) DaWaK 2011. LNCS, vol. 6862, pp. 80–95. Springer, Heidelberg (2011)
10. Skoutas, D., Simitsis, A.: Ontology-based conceptual design of etl processes for both structured and semi-structured data. Int. J. Semantic Web Inf. Syst. 3(4), 1–24 (2007)

11. Song, I., Khare, R., Dai, B.: SAMSTAR: A Semi-Automated Lexical Method for Generating STAR Schemas from an ER Diagram. In: DOLAP, pp. 9–16 (2007)
12. Vassiliadis, P., Simitsis, A.: Extraction, transformation, and loading. In: Encyclopedia of Database Systems, pp. 1095–1101 (2009)
13. Wilkinson, K., Simitsis, A., Castellanos, M., Dayal, U.: Leveraging Business Process Models for ETL Design. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 15–30. Springer, Heidelberg (2010)