A Framework for User-Centered Declarative ETL

Vasileios Theodorou[1] Alberto Abelló[1] Maik Thiele[2] Wolfgang Lehner[2]

[1]Universitat Politécnica de Catalunya
Barcelona, Spain
{vasileios,aabello}@essi.upc.edu

[2] Technische Universität Dresden
Dresden, Germany
{maik.thiele,wolfgang.lehner}@tu-dresden.de

ABSTRACT

As business requirements evolve with increasing information density and velocity, there is a growing need for efficiency and automation of Extract-Transform-Load (ETL) processes. Current approaches for the modeling and optimization of ETL processes provide platform-independent optimization solutions for the (semi-)automated transition among different abstraction levels, focusing on cost and performance. However, the suggested representations are not abstract enough to communicate business requirements and the role of the process quality in a user-centered perspective has not yet been adequately examined. In this paper, we introduce a novel methodology for the end-to-end design of ETL processes that takes under consideration both functional and non-functional requirements. Based on existing work, we raise the level of abstraction for the conceptual representation of ETL operations and we show how process quality characteristics can generate specific patterns on the process design.

Keywords

Declarative ETL; user-centered design; goal modeling;

1. INTRODUCTION AND MOTIVATION

Business Intelligence (BI) nowadays involves identifying, extracting, and analysing large amount of business data coming from diverse, distributed sources. In order to facilitate decision-makers, complex IT-systems are assigned with the task of integrating heterogeneous data deriving from operational activities into Data Warehouses, for the purpose of querying and analysis. This integration requires the extraction of data from internal or external sources such as the Web, their transformation to comply with destination syntax and semantics and the loading of the processed data to Warehouses, in a process known as Extract-Transform-Load (ETL).

Like every other business process, ETL processes can be examined using models and techniques from the area of Business Process Management (BPM) [16, 1], which is a holistic approach that aims to optimize business processes with respect to effectiveness and efficiency. A central activity of BPM is Business Process Modeling, which concerns representing the structure and workflow of business processes in a way that is understandable both by business users and IT. One problem that arises with defining appropriate models for ETL processes is relating the process model to fitness to use for the end-user. Therefore, we argue that the modeling approaches defined in [16, 1] are still too low-level to facilitate the evaluation and incorporation of process enhancements reflecting business requirements. This gap between end-user requirements and the low-level activities performed by ETL tools needs to be addressed using appropriate tools and metrics, borrowing techniques from the research area of Requirements Engineering.

In this paper, we introduce our holistic view for a quality-aware design of ETL processes by presenting a framework for user-centered declarative ETL. We raise the level of abstraction for the modeling of ETL activities and we illustrate how business goals can infer the integration of specific patterns to the customizable process model. In addition, we take under consideration the needs and expected skills of both business users (BU) and IT users and define their interaction with our framework in order to foster simplicity, while maximizing efficiency. The contributions of our proposed approach are two-fold — i)definition of an architecture and methodology for the rapid, incremental, qualitative improvement of ETL process models, promoting automation and reducing complexity and ii)clear separation of BU and IT roles where each user is presented with appropriate views and assigned with fitting tasks.

The remainder of this paper is organized as follows: In Section 2 we briefly review the state of the art in the modeling and user-centered automation of ETL processes; in Section 3 we illustrate the architecture and methodology of our proposed framework for quality-aware ETL and raise key issues about each phase of the design process; finally, we provide our conclusions in Section 4.

2. RELATED WORK

Castellanos et al. [5] set the foundation for ETL automation by recognising patterns and defining appropriate generic templates for populating business process data warehouses as a response to business events. They abstract the warehousing process and introduce a process-based high-level

.

analysis of ETL design that is independent of concrete implementations. Regarding conceptual representation in particular, Simitsis et al. [11] adopt a workflow paradigm for the modeling of ETL processes, consider alternative ETL configurations and examine their optimization by reassembling the execution of process activities. In the same direction, Vassiliadis et al. [14] provide a classification of ETL activities, through investigating the particular characteristics of ETL workflows and introducing a formal representation of workflows and activities based on identified patterns.

From the scope of Model Driven Engineering, Muñoz et al. [8] propose an MDA[1] approach for the development of ETL processes, which can enable automatic code generation from the implementation-hiding conceptual model. Likewise, Böhm et al. [4] propose model-driven generation of integration processes and distinct between different levels — conceptual, logical and physical. In order to cope with performance, they make an additional abstraction on the logical level that enables platform-independent optimization of integration processes.

In an attempt to manage ETL processes on a conceptual level that reflects organizational operations, it has been suggested that tools and models from the area of Business Process Management (BPM) [15] should be used. Following this concept, Wilkinson et al. [16] propose business process models for a conceptual view of ETL that can depict their dynamic nature in a real-time angle. In a similar manner, Akkaoui et al. [1] focus on the conceptual level and provide a more specific classification of ETL data flow and control flow by presenting a BPMN-based meta-model for ETL processes.

3. APPROACH

In this section we provide the details of our proposed quality-aware framework for ETL design. As can be seen in Fig. 1, our methodology consists of three phases: design of an ETL process based on functional requirements; instillation of user-defined quality characteristics to the process; and finally deployment and execution. The main drivers of this proposal are the requirements for automation and user-centricity. In addition, one important dimension is the need for communicating business requirements to the design, coming from BU who lack the background to understand technical details of the process. To provide the means for meeting these requirements, we propose a modular architecture that employs reuse of components and patterns to streamline the design. Furthermore, we apply an iterative model where BU are the key participants through well-defined collaborative interfaces. Following is a description of each component in more detail.

3.1 Functionality-Based Design

The *ETL Process Designer* component is responsible for the design of an ETL process model that implements the basic ETL functionality: extraction of data from the original data sources, transformation of data to comply with business rules and finally loading into target repositories. Recently, several approaches have been proposed for the automation of this phase. For example, Romero et al. [10] use an ontology of the data sources and their functional dependencies, together with business queries, to semi-automatically generate the ETL steps and the data warehouse multi-dimentional model at a conceptual level. Similarly, Bellatreche et al. [2] propose an approach where the domain model along with user-requirements are modelled on the ontological level and subsequently, an ETL process is produced, also modelled as an implementation-independent ontology.

Apparently, the required input at this step is an accurate representation of the domain, covering information not only about available data sources, data schemata, entities and interrelations among them but also about business requirements. We argue that naturally, the former can be modelled by IT with technical competencies for data and knowledge representation, while the latter can be introduced on a high level by BU, since they are the experts for the context of use of the resulting data processes.

The output of this step is a conceptual ETL process model, which is described in a high-level representation. This model must be abstract enough to allow for the incorporation of patterns reflecting BU requirements, but at the same time it can be seamlessly translated to a logical, implementation independent model. In addition, this model should be directly translatable to an intuitive visualization for the system user, using for example BPMN. Thus, we suggest that the model at this step could be an ETL-specific extension of the Directed Acyclic Graph, where each node is one high-level ETL operator. Akkaoui et al. [1] provide such a set of high-level ETL operators as part of their proposed BPMN meta-model.

Apart from the process model, domain information about available data sources, entities and their characteristics as well as resource constrains is also passed on to the next phase to allow for design alterations, where needed.

3.2 Quality Enhancement

The second phase regards the infusion of quality parameters to the ETL process. Our choice to segregate the functionality of the process from its qualitative perspective in two discrete phases does not only stem from a need for separation of concerns, but also from the fact that BU are the ones qualified to set quality goals and assess process quality. Nevertheless, the role of IT cannot be neglected, since apart from pro-actively designing all the aspects of this phase, they should constantly oversee the design process and translate technical details to business concepts whenever necessary and vice versa. Our architectural design at this stage is influenced by two paradigms from the areas of Software Development and Business Intelligence: agile methods and self-service BI [3], respectively. The benefits of using agile methods as opposed to the traditional waterfall approach in Data Warehousing activities have recently been recognized [6]. We identify this stage of the ETL process design as a perfect candidate for the application of agile practices because of the complexity and uncertainty of translating quality requirements to design choices. Thus, we adopt the idea of incremental and iterative design with BU in the center of the process. Likewise, we adopt the concept of strategy-driven business process analysis from the area of self-service BI, where BU make decisions in a declarative fashion based on strategies, goals and measures.

Integrating these ideas, we suggest that BU make decisions in stepwise iterations (sprints) that incrementally improve the quality of the ETL process, until they consider it crosses an acceptable quality threshold. Following is a

[1] http://www.omg.org/mda/

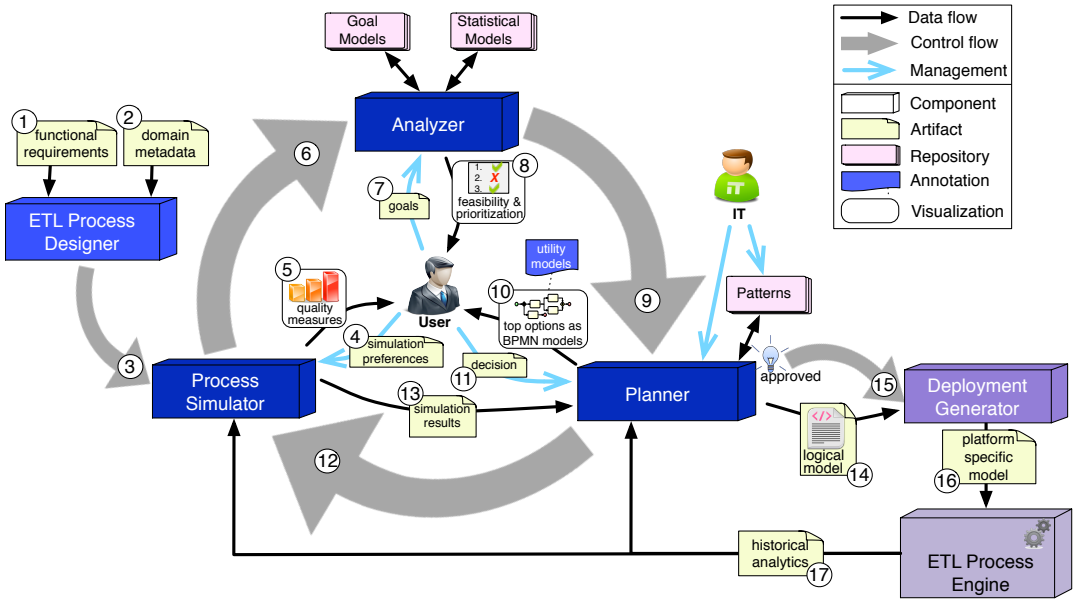

Figure 1: Functional architecture

description of each component that shows the means to facilitate this interaction.

The *Process Simulator* component is assigned with the task of simulating the ETL process and producing primal and complex analytics about both its static structure and its predicted execution behaviour. At this point the BU provide their simulation preferences that reflect quality characteristics of interest and receive as feedback a user-friendly representation of quality measures.To this end, it is important to use representative, realistic input data to test the process, which can be achieved either by "feeding" the simulator with a sample of real input data or by providing an effective test data generation mechanism [9]. One decision that has to be made at this point by the BU is the level of detail of the simulation. For example, the analysis can take place on a process level or on a task level, based on a single process run or on aggregated results of multiple process replications. Additionally, the simulation methodology should also be decided, especially for the case of loops, conditions and events (e.g., probabilistic or deterministic).

One obvious issue is the quantification of high level quality characteristics and their mapping to measures on the process. According to our approach, BU have access to high level metrics and statistics about the process quality, where visualization plays a key role. The metrics as well as the BU input should be applicable for different parts and levels of detail of the process, something that is configured by the IT and used as a service by the BU.

In our previous work [12], we have defined quality characteristics specific to ETL processes and we have identified measures from existing literature. There are measures that derive directly from the process model and their estimation requires no more than plain aggregations of the primal data produced by the Process Simulator component. However, there are also measures that are obtained from analysis of historical traces capturing the runtime behaviour of provenance models and components that the ETL consists of.

Once process measures have been produced, it is time for the *Analyzer* to come into play. The Analyzer takes as input user goals and it is responsible for reasoning about which goals and solution directions are feasible as well as which ones are most fit for use in the specified context. For the first part, it employs goal modeling techniques from the area of Requirements Engineering. Apart from concise visual representation, goal models are used for what-if analysis and reasoning. Selecting which goals are pursued every time, goal models can allow to answer feasibility questions about the set of tasks that can be performed, forming the palette of quality patterns that we will use for the optimization problem.

The second process that can be conducted by the Analyzer is the qualitative evaluation of alternative design patterns application. For this purpose, statistical models can be used that will take as input user goals and quality measures from the simulation of alternative ETL process models and will produce as output the relationships between goals and quality patterns, and thus the prioritization of the patterns that should be used, based on user's goals.

The *Planner* is a core component of the quality enhancement phase, responsible for producing patterns on the ETL process that improve its quality, using as input information about the process structure, current estimated metrics and goals and available patterns prioritization. The available patterns toolset can be predefined and extended on a per case basis by the IT and the resulting model after the integration of patterns is a logical model. This model includes a set of configuration and management operations that are not directly related to the functionality of specific flow components, but are rather external to the process (e.g., security configurations). These operations are necessary to complete the palette of available improvement steps for the satisfaction of quality goals.

Even though the problem space is restricted by estimated (monetary) cost, the optimization problem of selecting an optimal combination of patterns to be applied to the process can be formulated as a multi-objective knapsack prob-

lem [13]. In order to tackle complexity, we propose the use of goal models and statistical models on the previous step on one hand and the application of only one pattern during each iteration, on the other. In this direction, after reasoning, the Planner recommends to the BU a list of the highest ranked potential patterns in a graph-like visualization, together with utility models, which are annotations denoting the estimated affect of each pattern to the quality goals. Obviously, BU are by no means interested in the low level, technical details of the process and thus, similarly to the previous step, this recommendation takes place in the appropriate representation. Judging solely from the BPMN models and the utility models, BU make a selection decision and the Planner implements this decision by integrating a pattern to the existing process flow. These patterns are in the form of process components and the Planner should carefully merge them to the existing process [7]. Subsequently, new iteration cycles commence, until the BU consider that the model adequately satisfies quality goals. The *Planner* receives feedback from the actual runtime of the executed processs as well as from their simulation in order to adjust its heuristics and increase accuracy when selecting top options. In an attempt to assess the feasibility of our approach we have implemented a prototype of the *Planner* and our experimental results so far have been very promising.

3.3 Deployment and Execution

Once the BU observe satisfactory estimations for her measures of interest, she will decide that the quality of the process is acceptable and thus it is ready for deployment and execution. The *Deployment Generator* component processes the logical model and translates it to a platform-specific model. This step can be realized using existing approaches for (semi)automated transition among different abstraction levels, focusing on cost and performance [4, 16]. The *ETL Process Engine* executes the ETL process and as mentioned above, keeps traces to provide related historical analytics to the *Planner* and the *Process Simulator*.

4. SUMMARY AND OUTLOOK

In this paper, we addressed the problem of quality-aware ETL design in multidisciplinary, dynamic business environments. In order to reduce the complexity of deciding optimal process configurations that are in line with business requirements, we raised the level of abstraction of ETL operators and we considered the different roles that BU and IT can play based on their background and goals. Thus, we developed a methodology where BU are assigned with the task of deciding quality goals and evaluating available configurations based on high-level measures. On the other hand, besides development, monitoring and support, IT are responsible for model representation decisions. Based on this concept, we introduced the architecture and methodology of an incremental, iterative framework for end-to-end declarative ETL design and implementation, using goal modeling techniques and keeping BU at the center of quality control.

Acknowledgements. This research has been funded by the European Commission through the Erasmus Mundus Joint Doctorate "Information Technologies for Business Intelligence - Doctoral College" (IT4BI-DC). This work has also been partly supported by the Spanish Ministerio de Ciencia e Innovación under project TIN2011-24747.

References

[1] Akkaoui, Z., Zimányi, E., Mazón, J.N., Trujillo, J.: A BPMN-Based Design and Maintenance Framework for ETL Processes. IJDWM 9(3), 46–72 (2013)

[2] Bellatreche, L., Khouri, S., Berkani, N.: Semantic Data Warehouse Design: From ETL to Deployment á la Carte. In: Database Systems for Advanced Applications, pp. 64–83. Springer (2013)

[3] Berthold, H., Rösch, P., Zöller, S., Wortmann, F., Carenini, A., Campbell, S., Bisson, P., Strohmaier, F.: An architecture for ad-hoc and collaborative business intelligence. In: EDBT. pp. 1–6 (2010)

[4] Böhm, M., Wloka, U., Habich, D., Lehner, W.: Gcip: Exploiting the generation and optimization of integration processes. In: EDBT. pp. 1128–1131 (2009)

[5] Castellanos, M., Simitsis, A., Wilkinson, K., Dayal, U.: Automating the loading of business process data warehouses. In: EDBT. pp. 612–623 (2009)

[6] Golfarelli, M., Rizzi, S., Turricchia, E.: Sprint planning optimization in agile data warehouse design. In: DaWaK. pp. 30–41 (2012)

[7] Jovanovic, P., Romero, O., Simitsis, A., Abelló, A.: Integrating ETL Processes from Information Requirements. In: DaWaK. pp. 65–80 (2012)

[8] Muñoz, L., Mazón, J.N., Trujillo, J.: Automatic generation of ETL processes from conceptual models. In: DOLAP. pp. 33–40 (2009)

[9] Nakuçi, E., Theodorou, V., Jovanovic, P., Abelló, A.: Bijoux: Data generator for evaluating etl process quality. In: DOLAP. In press (2014)

[10] Romero, O., Simitsis, A., Abelló, A.: GEM: Requirement-Driven Generation of ETL and Multidimensional Conceptual Designs. In: Data Warehousing and Knowledge Discovery, pp. 80–95. Springer (2011)

[11] Simitsis, A., Vassiliadis, P., Sellis, T.: Optimizing ETL processes in data warehouses. In: ICDE. pp. 564–575 (2005)

[12] Theodorou, V., Abelló, A., Lehner, W.: Quality Measures for ETL Processes. DaWaK (2014)

[13] Thiele, M., Bader, A., Lehner, W.: Multi-objective scheduling for real-time data warehouses. Computer Science - Research and Development 24(3), 137–151 (2009)

[14] Vassiliadis, P., Simitsis, A., Baikousi, E.: A taxonomy of ETL activities. In: DOLAP. pp. 25–32 (2009)

[15] Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer (2012)

[16] Wilkinson, K., Simitsis, A., Castellanos, M., Dayal, U.: Leveraging business process models for ETL design. In: ER. pp. 15–30 (2010)