Mediation-based Web Services fed Data Warehouse

John Samuel*, Christophe Rey*
Franck Martin **, Lionel Peyron**

*LIMOS, CNRS, Université Blaise Pascal, Aubière, France
samuel@isima.fr, christophe.rey@univ-bpclermont.fr
**Rootsystem, Vichy, France
{franck.martin,lionel.peyron}@rootsystem.fr

Abstract. We present our prototype DaWeS, Data Warehouse fed with Web Services. Its ETL process is grounded on a mediation approach usually used in data integration. This enables DaWeS (i) to be fully configurable in a declarative manner only (XML, XSLT, SQL, datalog) and (ii) to make the warehouse schema dynamic so it can be easily updated. (i) and (ii) aim at making DaWeS scalable and adaptable to smoothly face the ever-changing and growing web services offer. We point out the fact that this also enables DaWeS to be used with the vast majority of actual web services defined with basic technologies only (HTTP, REST, XML,and JSON) and not with more advanced standards (WSDL, WADL, hRESTS or SAWSDL).

1 Introduction

Everyday we use a large number of web services and applications catering to our various requirements. This trend has now caught up even among the enterprises, especially the small and medium scale enterprises, resulting in a situation where their own business data is spread across multiple data centers spanning even across continents, managed and controlled by numerous web service providers. Enterprises need an integrated view of their data in the form of performance dashboards to track the overall growth of their companies and business units. Traditionally enterprises may use a data warehouse (cf Kimball and Ross (2002)) to perform data analysis and compute business performance measures. But with the advent of multiple heterogeneous, autonomous and ever-evolving web services, the task of data integration has become a challenging one. The purpose of our work is to aid enterprises using the web services as their data source with a data warehouse service to track their performance measures. We built DaWeS (Data warehouse fed with Web Services) (cf Samuel and Rey (2014); Samuel (2014)), a multi-enterprise data warehouse service able to fetch interesting data from various web services and expose them in a manner so that the end users can compute their own interesting business measures in a transparent manner hiding from them the various intricacies of the underlying web service application programming interface (API). Recent web services and semantic web technologies (like WSDL, WADL, hRESTS or SAWSDL) allow to solve many integration issues as for instance the automatic generation of wrappers or the automatic discovery of web services. But these technologies are constraining: developers must learn the associated languages, the integrated systems must be fully and precisely described and the

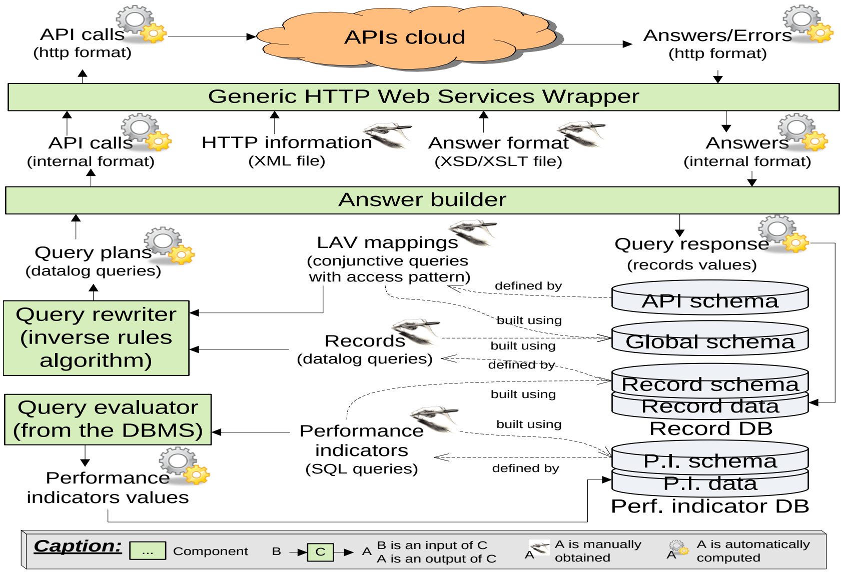

FIG. 1 – *DaWeS Architecture*

surrounding environment must be carefully set up, demanding for instance the building of a services directory or even a services ontology. That's why they are not (yet) used by the majority of companies which provide web services API. In this context, it is not possible to envision a full-fledged automated integration solution based on web services. Thus we propose DaWeS which is a semi-automated platform which aims to be scalable and adaptable using the actually used standards of web services (HTTP, REST, JSON, XML). To achieve scalability, DaWeS implements a full declarative approach reducing the human effort to a minimum while adding new web services, handling the API changes and defining the performance indicators. To achieve adaptability, DaWeS is organized around a mediated schema which is a part of the warehouse schema over which the user queries can be formulated (declaratively) and all data source API operations are defined (declaratively). This enables the warehouse schema to be dynamic which is a key feature towards adaptability. Internally, the system is powered by query rewriting techniques from the mediation approach for data integration (cf Duschka and Genesereth (1997); Duschka et al. (2000)). In the vein of Vassiliadis (2011), the use of mediation in the feeding of a datawarehouse from web services to obtain a scalable and adaptable integration system suited to actually used web services is the contribution brought by DaWeS.

2 Mediation to feed a data warehouse

We use mediation, a well-known approach in data integration systems (cf Wiederhold (1992)) to provide a uniform query interface across multiple heterogenous and autonomous web services API.

Each API operation is modeled as a relation having an access pattern defining inputs and outputs. The set of these operations defines the API schema. These relations are then defined as views over the global schema which is a set of relations that describe some domain (e.g. email marketing, helpdesk,...). This follows the LAV setting of mediation, using conjunctive queries as the view language. The global and API schema are virtual, they do not contain any data. DaWeS users are then able to define records as a second set of views over the global schema. A record is a materialized relation in the warehouse in which data from web services are stored after retrieval. The view language for records is datalog program, ie union of conjunctive queries with recursion. The record schema gathers all record definitions. It is the base schema over which full SQL queries are posed to define complex indicator measures. The set of indicators relations is the performance indicator schema. As the record schema, it is materialized. We don't materialize the complete data warehouse schema since it amounts to duplication of the complete enterprise data, a significant portion of which may not be even useful for business performance computation. Considering the space limitations, we refer the readers to the complete context and settings of DaWeS and the associated examples described in Samuel and Rey (2014).

The feeding process of the warehouse mainly consists of the rewriting of record definitions according to API relations. This happens in the query rewriter where the rewriting is performed by the Inverse Rules algorithm (cf Duschka and Genesereth (1997); Duschka et al. (2000)) which is a classical data integration algorithm. This algorithm is especially interesting in its unique ability to handle recursive record definition which is particularly interesting to define complex records. In addition to this, Inverse rules can be used to specify various tuple generating dependencies (particularly functional and full dependencies) to specify various constraints (like primary key) in the mediated (global) schema. The obtained rewriting is a query plan that retrieves data from the web services. This is achieved thanks to the answer builder and the generic http wrapper. These two modules take as input other characteristics of web service API operations: XSD and XSLT files. The answer builder is also in charge of building the complete answer to be stored in the record relation. The non ETL part of DaWeS is classicaly made up of SQL query answering capabilities to be able to compute and store perfomance indicators.

3 Demonstration Scenarios

There are primarily two types of users for DaWeS: administrators and business executives. Administrators are responsible for adding new web services to the system, managing the global schema relations, formulating record queries over the global schema and providing some default performance indicators. Business executives are in charge of defining their own indicators. The demonstration will illustrate the use of DaWeS through 3 scenarios.

In the first scenario, it will be explained how a DaWeS administrator can connect web services to the system. Real available web services will be used: Basecamp, Teamworkpm and LiquidPlanner for the project management field, MailChimp and CampaignMonitor for the email marketing field, and Zendesk, Freshdesk, Uservoice, Desk and Zoho Support for the support/helpdesk field[1].

In the second scenario, it will be shown how records and performance indicators are defined for the previous web services. Examples of record definitions are "Daily New Projects", "Daily Active Projects",... Examples of performance indicators are "Total High Priority Tickets Registered in a month", "Percentage of High Priority Tickets Registered in a month", ...

[1] http://www.basecamp.com, http://www.liquidplanner.com, http://www.teamworkpm.net, http://www.mailchimp.com, http://www.campaignmonitor.com, http://www.zendesk.com, http://www.freshdesk.com, http://www.uservoice.com, http://www.desk.com, http://www.zoho.com/support,

The third scenario will show how a user can use the platform. For exemple, it will show what a helpdesk user who is interested in the number of tickets solved during the last month or the percentage of high priority tickets registered and solved has to do to get these indicators.

4 Conclusion

With DaWeS, we show that even without defining web services operations with machine readable description languages, the mediation approach defined in data integration (especially the LAV setting) can be valuably used as a data warehouse ETL. We also show that this approach enables a fully declarative way of managing the ETL part of the data warehouse. This study is still in progress concerning its quantitative validation with current benchmark experimentations to assess scalability and adaptability.

Acknowledgement: We thank the Conseil General of the Region of Auvergne (France) and FEDER for funding our research project.

References

Duschka, O. M. and M. R. Genesereth (1997). Answering recursive queries using views. In *PODS*, pp. 109–116.

Duschka, O. M., M. R. Genesereth, and A. Y. Levy (2000). Recursive query plans for data integration. *J. Log. Program. 43*(1), 49–73.

Kimball, R. and M. Ross (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling* (2nd ed.). New York, NY, USA: John Wiley & Sons, Inc.

Samuel, J. (2014). Towards a data warehouse fed with web services. In *ESWC PhD Symposium*.

Samuel, J. and C. Rey (2014). Dawes: Data warehouse fed with web services. In *INFORSID*.

Vassiliadis, P. (2011). A survey of extract-transform-load technology. In *Integrations of Data Warehousing, Data Mining and Database Technologies*, pp. 171–199.

Wiederhold, G. (1992). Mediators in the architecture of future information systems. *Computer 25*(3), 38–49.

Résumé

Nous présentons le prototype DaWeS, un entrepôt de données alimenté par des services web. Son mécanisme d'ETL est basé sur l'approche médiation habituellement utilisée en intégration de données. Cette approche permet à DaWeS (i) d'être entièrement configurable en utilisant des langages déclaratifs seulement (XML, XSLT, SQL, datalog) et (ii) de rendre le schema de l'entrepôt dynamique et donc facilement modifiable. L'objectif est de permettre l'adaptabilité et le passage à l'échelle afin que DaWeS puisse intégrer des services web en constante évolution et en nombre toujours plus grand. Nous mettons l'accent sur le fait que cette approche médiation permet à DaWeS d'être utilisé avec la majorité des services web actuellement disponibles qui sont utilisent des technologies de base uniquement (HTTP, REST, XML,and JSON) et non des standards plus évolués (WSDL, WADL, hRESTS or SAWSDL).