A CLOUD-BASED SOLUTION FOR DATA QUALITY IMPROVEMENT

Marco Comerio
University of Milano - Bicocca, Milano, Italy

Keywords: Service Selection, Service Contract, Data Contract, Data Quality, Cloud Computing.

Abstract: The application of techniques to improve the data quality of an organization is traditionally costly since different specific tools are required. Potentially, cloud computing models could offer powerful solutions to reduce costs. However, some challenges remain in the widespread acceptance of cloud computing models because they require the sharing of business critical data. Therefore, services for data quality improvements in the cloud should act in compliance with predefined contracts. This paper extends previous works on the specification, selection and evaluation of service and data contracts. Moreover, a cloud-based architecture for data quality improvement that supports contract-based service selection is proposed. Experimental activities on a real scenario demonstrate the feasibility of the proposed solution.

1 INTRODUCTION

In the last decade, the research has focused on defining *data quality methodologies* (Batini and Scannapieco, 2006) providing a set of guidelines that defines a rational process to select, customize, and apply data quality techniques for improving the quality of data of an organization. Techniques for data quality improvement can be divided into three different categories: (i) *data cleansing* techniques (e.g., *standardization*, *normalization* and *record linkage*), which are applied to correct errors, standardize information, and validate data; (ii) *data integration* techniques (e.g., *data and schema integration*), which support the linking of data elements about the same item available in different data sources and the consolidation of these data elements into a single view; (iii) *data enrichment* techniques (e.g., *data verification* and *data validation*), which support the incorporation of additional data from external sources to complete and verify existing records. The application of data cleansing, integration and enrichment techniques inside an organization is traditionally costly since different specific tools are required.

The emerging cloud computing paradigm supports the Software-as-a-Service (SaaS) and Data-as-a-Service (DaaS) distribution models in which software and data are made available as on-demand services to the customers. Potentially, these distribution models could offer powerful solutions to reduce costs in performing data quality improvement inside an organization. The SaaS model (Viega, 2009) can be used to provide data quality improvement tools and the DaaS model (Truong et al., 2011) can be used e.g., for data enrichment activities. In this paper, SaaS and DaaS dealing with data quality improvement are referred to *data intensive services*.

The possibility to use data intensive services in the cloud should be deeply investigated. By moving from tools and data sources in an organization to data intensive services in the cloud, traditional data quality improvement activities must be re-formulated (Comerio et al., 2010). Moreover, some challenges remain in the widespread acceptance of SaaS and DaaS delivery models because they require the sharing of business critical data (Li et al., 2009). Due to this fact, data intensive services should act in compliance with predefined agreements with the customer. The generic term *contract* is used to refer to these agreements, the specific term *service contract* is used when the contract deals with Quality of Service (QoS), business, contextual and legal terms related to a service and the term *data contract* is used when the contract deals with data managed by a service. Typically, service and data contracts are composed of one or more contractual terms that specify conditions established on the basis of non-functional parameters.

Current data intensive services are typically associated with human-readable contracts that hinder their automatic evaluation and selection. In the literature,

Comerio M..
A CLOUD-BASED SOLUTION FOR DATA QUALITY IMPROVEMENT.
DOI: 10.5220/0003902802220227
In *Proceedings of the 2nd International Conference on Cloud Computing and Services Science* (CLOSER-2012), pages 222-227
ISBN: 978-989-8565-05-1

different approaches (e.g., policies, licenses, service level agreements) for the specification of structured contracts have been proposed. These approaches define contractual terms by means of <attribute,value> clauses that prevent the inclusion of articulated contractual terms specifying technological and business interdependencies among non-functional parameters (e.g., a higher price for the service that guarantees a certain bandwidth). Moreover, these approaches support only the specification of service contracts without any reference to the data managed by the services (Comerio et al., 2009b).

Contract specification is not the only impediment to the use of data intensive services in the cloud. Current tools for data quality improvements (e.g., *DataFlux dfPower Studio*[1]) provide a limited support for the integration of third-party services and do not support automatic contract-based service selection. Basically, even if they allow for the usage of external tools for specific data quality activities (e.g., geocoding services for data quality enrichment), the selection of these tools is performed manually.

Previous works provide contributions to automate the evaluation and selection of contracts (namely, the *matchmaking* process) w.r.t. a set of user requirements. In (De Paoli et al., 2008), a semantic metamodel, namely *Policy-Centered Metamodel* (PCM), has been defined to support expressive descriptions of contractual terms according to a language-independent ontology. The PCM was exploited as intermediate format to develop a hybrid matchmaking process (Comerio et al., 2009a), where hybrid indicates the capability of addressing symbolic and numeric values and expressions. The matchmaking process has been implemented by the *Policy Matchmaker and Ranker* (PoliMaR) framework[2].

This paper extends previous works providing new contributions on modeling non-functional parameters that are relevant when performing data quality improvement activities in the cloud. Moreover, a cloud-based architecture that supports contract-based service selection is proposed. Experimental activities on a real scenario demonstrates the feasibility of the proposed solutions.

The rest of this paper is organized as follows. Section 2 reports the new contribution on modeling contracts for data intensive services. The proposed cloud-based architecture is described in Section 3 and tested in Section 4 on a real scenario. Related works are described in Section 5. Section 6 concludes the paper and outlines future works.

[1]http://www.dataflux.com/Downloads/Documentation/8.2.1/dfPowerGuide.pdf

[2]http://sourceforge.net/projects/polimar/

2 MODELING CONTRACTS FOR DATA INTENSIVE SERVICES

The analysis of existing contracts offered by real data intensive services (e.g., *Strikerion*[3], *Postcode Anywhere*[4] and *Jigsaw*[5]) allows the identification of non-functional parameters that must be considered when modeling contracts. In order to allow the semantic specification of these parameters in PCM-based contracts for data intensive services, a reference ontology (namely, *Data Contract Ontology (DCOnto)*) has been produced. In this section, the attention is focused on *Data Rights* terms (Truong et al., 2011) related to data handling and anonymity.

2.1 Modeling Data Handling

Several different authorizations on data handling can regulate the exchange of data between the data owner and the data intensive service provider. Examples of rights to be observed when a data owner sends data to the data intensive service and asks for a particular request (e.g., data cleansing) are:

- *no-reuse*: the service can use the data only to process the request and after that the data must be deleted. Re-use is not allowed.
- *reuse for commercial use (reuse4CU)*: the data are used to process the request and, possible, to process third-party requests. Re-use is allowed.
- data must be managed according to a particular *Open Data Commons license*[6]:
  - *Public Domain Dedication and License (PDDL)*: the data can be copied and distributed. Moreover, the data can be modified and elaborated to produce new data assets. Distribution and re-use are allowed.
  - *Attribution License (ODC-By)*: as the PDDL but with the restriction that any distribution and re-use of the data (or of new data assets produced from them) must be attributed to the original license.
  - *Open Database License (ODC-ODbL)*: as the ODC-By with the additional restriction that the data (or of new data assets produced from them) must be distributed under the same license. This hinders the commercial use of the data.

Data handling contractual terms are characterized by *inclusion relations* ($\sqsubseteq$). As an example, if a data

[3]http://www.strikeiron.com

[4]http://www.postcodeanywhere.co.uk

[5]http://www.jigsaw.com/

[6]http://www.opendatacommons.org/licenses

owner requests a data cleansing activity on data covered by a ODC-By license, a data intensive service that offers the same license for data handling or licenses that include it (i.e., the more restrictive licenses ODC-ODbL, no-reuse or reuse4CU) must be selected. The complete set of inclusion relations on data handling terms is the following: $ODC-ODbL \sqsubseteq noReuse$, $PDDL \sqsubseteq ODC-By$, $ODC-By \sqsubseteq ODC-ODbL$, $ODC-By \sqsubseteq reuse4CU$ and $reuse4CU \sqsubseteq noReuse$.

A fragment of DCOnto that models the data handling terms ODC-By and ODC-ODbL is shown in Figure 1. To be noticed that the terms are modelled as instances of the *DataHandlingType* concept. The modeling of the inclusion relations is performed by means of relationInstances *pcm#composedOf*.

```
concept DataHandlingType

instance odcBy memberOf DataHandlingType

instance odcOdbl memberOf DataHandlingType

relationInstance pcm#composedOf(odcOdbl, odcBy)
```

Figure 1: Data handling terms modeling.

2.2 Modeling Anonymity

Anonymity is an important requirement when dealing with privacy-aware data intensive services. In general, data can be categorized into different classes. Among them, one class includes data, referred to as *sensitive data*, concerning the private life, political or religious creed and so on, while another class contains data that describes the identity of individuals (e.g., first name, family name, etc.) (Coen-Porisini et al., 2010). A privacy-aware data intensive service must assure that only authorized users can view the existing relationships between sensitive data and the identity of the individuals.

Different data modification methods (e.g., *perturbation* and *blocking*) can be used when dealing with the anonymity of data that needs to be released to the public. To measure the anonymity, several metrics characterized by different levels of applicability, complexity and generality are available. Similar to data handling terms, the anonymity metrics present inclusion relations. For example, let us consider the metrics *k-anonymity* and *l-diversity*.

A dataset provides k-anonymity protection if the information for each person contained in the dataset cannot be distinguished from at least k-1 individuals whose information also appears in the dataset. As shown in (Machanavajjhala et al., 2007), a k-anonymized dataset has some subtle, but severe privacy problems (namely, the *homogeneity attack* and the *background knowledge attack*). The metric l-diversity solves these privacy problems introducing the restriction that the k-anonymazed individuals are characterized by at least l-diverse sensitive information. Therefore, l-diversity introduces privacy beyond k-anonymity and so when a data owner requests for k-anonymity data management, a data intensive service offering the k-anonymity or l-diversity (with k=l) must be selected.

Since k-anonymity and l-diversity assume numeric values, they must be modelled as quantitative terms. The PCM allows for the specification of *single value* terms and *range* terms by means of *pcm#SingleValueExpression* and *pcm#RangeExpression*. The relation $k-anonymity \sqsubseteq l-diversity$ cannot be directly modelled into *DCOnto* since it is valid only when *k=l*. However, it must be considered and evaluated along the service matchmaking process by means of predefined matching rules. Figure 2 shows a matching rules stating that a request for a single value k-anonymity must be evaluated with offered terms on single value k-anonymity and l-diversity.

```
axiom anonymityMatching definedBy

    matchCouple(?request,?offer,anonymity) :-
    (?request memberOf dcOnto#singleValueKAnonymity
        ) and
    (?offer memberOf dcOnto#singleValueKAnonymity)
        or
    (?offer memberOf dcOnto#singleValueLDiversity).
```

Figure 2: Matching rule for single value anonymity terms.

3 AN ARCHITECTURE FOR DATA QUALITY IMPROVEMENT IN THE CLOUD

In (Comerio et al., 2010), a preliminary cloud-based conceptual architecture for data quality improvement has been proposed. Basically, this architecture allows a data owner to invoke a data quality SaaS providing data cleansing, data integration and data enrichment services. The data quality SaaS can directly offer the services or it can act as a broker that selects and invokes the best SaaS/DaaS over the cloud able to satisfy the data owner request. The cloud-based conceptual architecture in (Comerio et al., 2010) does not provide support for the evaluation of service and

data contracts. This limitation can be overcome with the integration of a new component for contract-based service selection.

In this paper, the resulting conceptual architecture is applied to extend the architecture of *DataFlux dfPower Studio*[7], a commercial solution for data quality engineering activities. Within the dfPower Studio, the *dfPower Architect* supports the definition of workflows (namely, *jobs*) that are composed by different nodes (namely, *steps*) configurable via metadata properties. Furthermore, DataFlux also offers the *Enterprise Integration Server* (EIS) to support the execution of dfPower Architect jobs and their conversion to Web services. Through tight integration with the dfPower Architect, the EIS enables the complete reuse of jobs and services. However, the automatic selection and invocation of external data intensive services in the cloud is not supported.

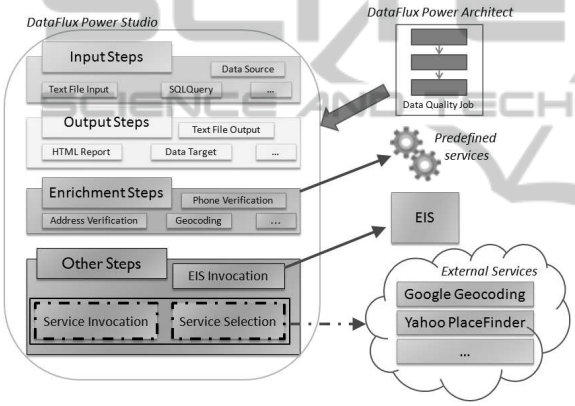

Figure 3: DataFlux dfPower Studio extended architecture.

Figure 3 shows the extensions proposed for the architecture of DataFlux dfPower Studio. Two new steps are introduced: *service selection* and *service invocation*. The **Service Selection** step receives a *Request* specifying the requested contractual terms of the service to be selected and realizes the selection process on PCM-based contracts. The matchmaking process is realized through the following phases: a *Term Matching* phase, which identifies couples of comparable requested and offered contractual terms; a *Local Evaluation*, that computes a local degree (LD) of match between each couple; a *Global Evaluation*, which exploits LDs to compute a global degree (GD) of match for each offered contract; and a *Contract Selection* that performs a sorting based on the GDs and returns the best matching contract. As mentioned in Section 1, the matchmaking process has been implemented in the PoliMaR framework. Therefore, the usage of a service selection step in dfPower Architect jobs basically consists in a invocation of PoliMaR, seen as an external service.

The **Service Invocation** step receives the ID of the service to be invoked and the data to be sent to the service (e.g., the URI of an external data enrichment service and the data to be enriched). The service invocation step invokes a *wrapper* that supports the following three-phase process: (i) the mapping between the input of the service invocation step and the input of the selected service; (ii) the invocation of the selected service and (iii) the mapping between the output of the selected service and the output of the service invocation step. The mapping rules used along phases (i) and (iii) are predefined and available into a *rule repository*.

4 EXPERIMENTS

In order to verify the feasibility of the architecture proposed in Section 3, a test on a real scenario has been performed.

The Academic Patenting in Europe (APE-INV)[8] project has the objective to produce a freely available database on *academic patenting in Europe*, that will contain reliable and comparable information on the contribution of European academic scientists to technology transfer via patenting. The reference source of raw patent data is the PATSTAT database, issued for statistical purposes by the European Patent Office (EPO)[9]. The raw data of the PATSTAT database presents data quality issues (e.g., the addresses of the inventors in the database are often inaccurate causing the presence of duplicated records).

To execute the test, the following process has been realized: (i) **Geocoding Step**: usage of a geocoding service to verify/correct the addresses of the inventors; (ii) **De-duplication Step**: usage of a de-duplication service to identify and delete duplicated records; (iii) **Enrichment Step**: usage of an enrichment service to extend data of the inventor's city of residence with information about the number of inhabitants. This information is useful for statistical purposes.

The process has been designed as a *dfPower Architect job*, where the geocoding and enrichment steps are realized by means of the selection and invocation steps described in Section 3. The objective is to show that, changing the requested contractual terms (namely, *Request*), different functional-equivalent services can be selected and invoked at

[7] DataFlux has recently released a new version of the tool changing its name into *Data Management Studio*.

[8] http://www.esf-ape-inv.eu

[9] http://www.epo.org

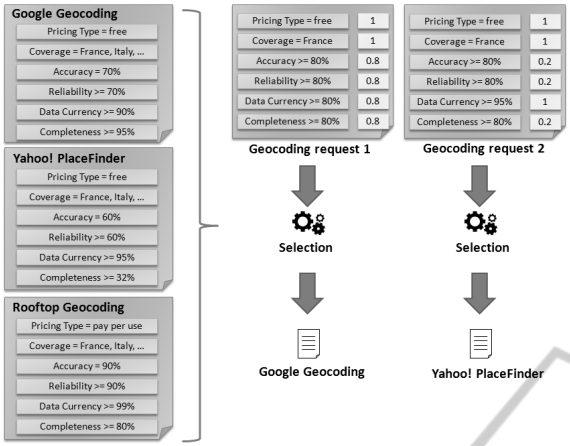

Figure 4: Examples of the selection performed for the geocoding step.

run-time.

Due to space limit, the description of the test proceeds focusing only on the geocoding step. The enrichment step is performed similarly. Figure 4 shows how the selection for the geocoding step is performed. Tree geocoding services are available; two of them are services in the cloud (namely, *Google Geocoding*[10] and *Yahoo! PlaceFinder*[11]) and the last one (*Rooftop Geocoding*) is an internal service offered by DataFlux dfPower Studio. Each service is characterized by a contract including contractual terms on the *pricing type*, the *service coverage* and typical data quality metrics (i.e., *accuracy*, *reliability*, *data currency* and *completeness*). The selection has been performed considering two different Requests containing contractual terms associated with different *relevance value* in [0..1] (i.e., the importance for the requestor to have the terms satisfied). The former contains strictly-required (relevance=1) contractual terms on pricing type and service coverage and preferred terms (relevance=0.8) on data quality metrics. The latter contains strictly-required contractual terms on pricing type, service coverage and data currency and optional terms (relevance=0.2) on accuracy, reliability and completeness. Figure 4 shows that: (i) the selection step for the first Request returns *Google Geocoding* since this service better satisfies the strictly-required and preferred contractual terms and (ii) the selection for the second Request returns *Yahoo! PlaceFinder* since this service better satisfies the additional strictly-required constraint on data currency.

Figure 5 shows how the invocation of the Google

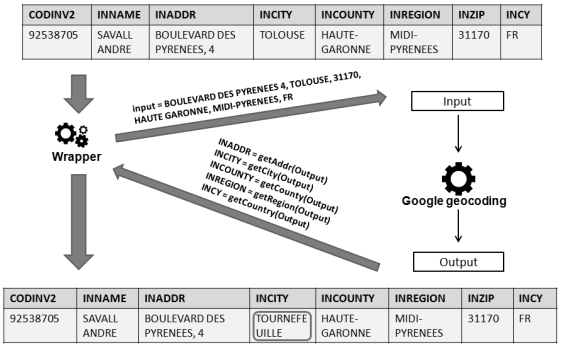

Figure 5: Examples of invocation performed for the geocoding step.

Geocoding service is performed. For each record of the table, the wrapper composes the input for the geocoding service merging the six fields (i.e., INADDR, INCITY, INCOUNTRY, INREGION, INZIP, INCY) that specify the address of an inventor. Then, the wrapper invokes the Google Geocoding service and receives, when available, a string with the corrected address. Finally, the wrapper, by means of a pre-defined mapping rule, splits the string and substitutes the address in the table.

The usage of external service in the cloud provides positive results for the data quality improvement. As an example, Google Geocoding has verified/corrected the 75% of the addresses in the table. This results outperforms the one obtained using the internal Rooftop Geocoding service. This is due to the fact that the completeness (95%) of the data provided by Google is higher. A limitation of the proposed solution is represented by the semantic solution adopted for contract selection. The response time of the PoliMaR framework is unsatisfactory due to the slowness of semantic matching activities; moreover, as already tested in (Comerio et al., 2009a), the performance decreased dramatically with the growing of contracts in size and number. However, this limitation is not crucial since the approach in this paper is not proposed for a time-critical application.

5 RELATED WORK

Only few papers (Zhou et al., 2009; Tsai et al., 2007) propose a SOA-based solution to perform data quality improvement activities. A SOA-based semantic data quality assessment framework which supports automatically searching for proper data quality assessment services which fulfill user requirements is proposed in (Zhou et al., 2009). A dynamic framework for data provenance (i.e., the origins and routes of data) classification in a SOA system is described

[10] http://code.google.com/intl/it-IT/apis/maps/documentation/geocoding/

[11] http://developer.yahoo.com/geo/placefinder/

in (Tsai et al., 2007). Respect to the proposed cloud-based architecture, these SOA-based solutions cover only a particular aspect (i.e., data quality assessment (Zhou et al., 2009) and data provenance (Tsai et al., 2007)) of a data quality improvement process.

A very limited number of papers (Faruquie et al., 2010; Dani et al., 2010) proposes a cloud-based solution to perform data quality improvement activities. A cloud infrastructure to offer virtualized data cleansing that can be accessed as a transient service is presented in (Faruquie et al., 2010). Setting up data cleansing as a transient service gives rise to several challenges such as (i) defining a dynamic infrastructure for the cleansing on demand based on customer requirements and (ii) defining data transfer and access that meet required service level agreements in terms of data privacy, security, network bandwidth and throughput. Moreover, as further discussed in (Dani et al., 2010), offering data cleansing as a service is a challenge because of the need to customize the rules to be applied for different datasets. The Ripple Down Rules (RDR) framework is proposed in (Dani et al., 2010) to lower the manual effort required in rewriting the rules from one source to another. The solutions in (Faruquie et al., 2010; Dani et al., 2010) face challenges similar to the ones tackled in this paper but they focused only on data cleansing and not to a complete data quality improvement process. Moreover, the contract-based service selection is not addressed.

6 CONCLUSIONS AND FUTURE WORKS

Cloud computing models offer powerful solutions to reduce costs when performing data quality improvements by using software and data offered as services on-demand. However, since data quality improvements potentially require the sharing of business critical data, these services should act in compliance with predefined contracts. This paper has extended previous works on the definition of methods and techniques for the specification, selection and evaluation of service and data contracts. Moreover, this paper has proposed an extension for the DataFlux dfPower Studio architecture that supports contract-based service selection for data quality improvement activities in the cloud. Experimental activities on the PATSTAT database have demonstrated the feasibility of the proposed solutions.

Future works deal with some open issues concerning data transfer and resource allocation for data processing services over the cloud.

ACKNOWLEDGEMENTS

This work is supported by the SAS Institute srl (Grant Carlo Grandi). The author wants to thank Andrea Scrivanti for his precious contribution in this work.

REFERENCES

Batini, C. and Scannapieco, M. (2006). *Data Quality: Concepts, Methodologies and Techniques (Data-Centric Systems and Applications)*. Springer-Verlag.

Coen-Porisini, A., Colombo, P., and Sicari, S. (2010). Dealing with anonymity in wireless sensor networks. In *Proc. of SAC 2010*, pages 2216–2223. ACM.

Comerio, M., De Paoli, F., and Palmonari, M. (2009a). Effective and flexible nfp-based ranking of web services. In *Proc. of ICSOC/ServiceWave 2009*, pages 546–560.

Comerio, M., Truong, H.-L., Batini, C., and Dustdar, S. (2010). Service-oriented data quality engineering and data publishing in the cloud. In *Proc. of SOCA 2010*, pages 1–6.

Comerio, M., Truong, H.-L., De Paoli, F., and Dustdar, S. (2009b). Evaluating contract compatibility for service composition in the seco2 framework. In *Proc. of ICSOC/ServiceWave 2009*, pages 221–236.

Dani, M. N., Faruquie, T. A., Garg, R., Kothari, G., Mohania, M. K., Prasad, K. H., Subramaniam, L. V., and Swamy, V. N. (2010). A knowledge acquisition method for improving data quality in services engagements. In *Proc. of SCC 2010*, pages 346–353.

De Paoli, F., Palmonari, M., Comerio, M., and Maurino, A. (2008). A Meta-Model for Non-Functional Property Descriptions of Web Services. In *Proc. of ICWS 2008*, pages 393–400.

Faruquie, T. A., Prasad, K. H., Subramaniam, L. V., Mohania, M. K., Venkatachaliah, G., Kulkarni, S., and Basu, P. (2010). Data cleansing as a transient service. In *Proc. of ICDE 2010*, pages 1025–1036.

Li, J., Stephenson, B., and Singhal, S. (2009). A policy framework for data management in services marketplaces. In *Proc. of ARES 2009*, pages 560–565.

Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M. (2007). L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1.

Truong, H.-L., Gangadharan, G., Comerio, M., Dustdar, S., and De Paoli, F. (2011). On analyzing and developing data contracts in cloud-based data marketplaces. In *Proc. of APSCC 2011*, pages 174–181.

Tsai, W.-T., Wei, X., Zhang, D., Paul, R., Chen, Y., and Chung, J.-Y. (2007). A new soa data-provenance framework. In *Proc. of ISADS 2007*, pages 105–112.

Viega, J. (2009). Cloud computing and the common man. *Computer*, 42:106–108.

Zhou, Y., Hanß, S., Cornils, M., Hahn, C., Niepage, S., and Schrader, T. (2009). A soa-based data quality assessment framework in a medical science center. In *Proc. of ICIQ 2009*, pages 149–160.