On Analyzing and Developing Data Contracts in Cloud-based Data Marketplaces

Hong-Linh Truong*, G.R. Gangadharan†, Marco Comerio‡, Schahram Dustdar*, Flavio De Paoli‡
*Distributed Systems Group, Vienna University of Technology, {truong,dustdar}@infosys.tuwien.ac.at
†IBM Research India, gangadharan@in.ibm.com
‡ Department of Informatics, Systems and Communication, University of Milano - Bicocca, {comerio,depaoli}@disco.unimib.it

***Abstract*—Currently, rich and diverse data types have been increasingly provided using the Data-as-a-Service (DaaS) model, a form of cloud computing services. However, data offered by DaaS are constrained by several data concerns that, if not automatically being reasoned properly, will lead to a wrong way of using them. In this paper, we support the assumption that data concerns should be explicitly modeled and specified in data contracts to support concern-aware data selection and utilization. Instead of relying on a specific definition of data contracts, we analyze contemporary data contracts and we present an abstract model for data contracts. Based on the abstract model, we propose several techniques for evaluating data contracts that can be integrated into data service selection and composition frameworks. We also illustrate our approach with some real-world scenarios.**

I. Introduction

Recently, delivering data based on service-oriented and Cloud computing techniques is becoming popular. In such a delivery model, data are typically made available for retrieving from Web services, mostly implemented using SOAP or REST technologies, deployed in the Internet and Cloud environments. On the one hand, this model offers extensible and interoperable delivery means in which data can be easily retrieved and business supporting can be easily implemented. On the other hand, this model allows incorporating data constraints, such as free or commercial usage. This model is in fact a form of the so-called read-only Data-as-a-Service (DaaS) [1] which is the core element of cloud-based data marketplaces. Unlike the conventional view on services in which the service provider is the only responsible for all its functions and deliveries, in DaaS, data provider and DaaS provider are considered separated. DaaS providers offer backbone for delivering data while data providers offer data.

While techniques for making data available through DaaS are well-developed, we are interested in the specification of data contractual terms and in the relationship between data contracts and service contracts in the ecosystem of DaaS, which have been neglected in current research. In fact, when a DaaS provides rich types of data, then service contracts cannot be used to specify data contracts as (i) a DaaS offers facilities for multiple data providers, (ii) a data provider has multiple types of data, and (iii) each type of data can be associated with multiple data contracts. In this paper, we argue that it is required to define data contracts that can be used separate from service contracts or in combination with service contracts. In particular, we concentrate on data contracts that can support (automatic) data selection and composition.

Currently, there is a lack of understanding and techniques to deal with data contracts, although data delivered via DaaS is typically associated with human-readable data contracts (often called data agreements or data licenses). Our contributions in this paper are (i) the analysis of current data contracts in order to identify the properties that are relevant in the perspective of contracts for DaaS, (ii) the analysis of the relationship between service contracts and data contracts, and (iii) the definition of an abstract data contract model for developing data contracts in order to facilitate the right selection and utilization of data assets in data marketplaces. To demonstrate the feasibility of our approach, we provide an implementation of data contracts related to real-world scenarios using the Resource Description Framework (RDF).

The rest of the paper is organized as follows: Section II presents the motivation and related work. Section III presents our analysis of current data contracts. In Section IV we present techniques for developing data contracts. We describe our experiments in Section V, followed by conclusions and future works in Section VI.

II. Motivation and Related Work

A. Background and Motivation

The DaaS model is based on the concept that the data can be provided on-demand to the user at anytime and from anywhere, encapsulating the actual platform where data resides. DaaS plays a vital role in emerging data marketplaces in cloud computing environments, such as Microsoft Azure Data Marketplace[1] and Infochimps[2], as well as in Open Data Initiative[3]. In DaaS and data marketplaces, data contracts are used to:

- define the extent to which the data can be used, on the basis that any use outside the terms of the contract would constitute an infringement;
- have a remedy against the consumer where the circumstances are such that the acts complained of do not constitute an infringement of the contract;

[1] https://datamarket.azure.com/
[2] http://www.infochimps.com/
[3] http://www.data.gov/opendatasites

978-0-7695-4624-7/11 $26.00 © 2011 IEEE
DOI 10.1109/APSCC.2011.65

- limit the liability of data providers in case of failure of the provided data;
- specify information on data delivery, acceptance, and payment.

Currently, most real DaaSs and data marketplaces present data contracts for their offered data assets, often called data agreements or data licenses, in human-readable forms. Typically, data contracts consisting of constraints on data concerns are diverse, rich, and contextual (e.g., depending on geographical regions and publishing purposes).

The lack of well-formed data contract models hinders the data selection and utilization with respect to data contractual terms, such as data rights, quality of data, and law enforcements. This triggers calls for consideration of data contracts in data mashup [2] and data provisioning [3]. Our main motivation is that an analogy to a well-researched service contracts but for data assets in DaaS and data marketplaces should be conducted. By doing so, we can answer several questions, e.g., : Are we allowed to use these data? Do the qualities of data delivered via DaaS meet the agreement between data providers and data consumers? Are we allowed to republish the results built based on these data sources? However, such questions require extensible models that are able to capture contractual terms for data contracts and to represent them in a form to be reasoned by automatic techniques. Moreover, certain domain-specific properties of data, such as quality and compliance, make the definition of the methodology to be used for developing data contracts more complicated.

B. Related Work

ODRL [4] allows specifying data terms but it is not designed for data assets in data marketplaces. ONIX-PL [5] is another XML-based licenses for digital resources. Our abstract data model is more flexible as we do not fix contractual terms; we do not think that fixed terms in a specification will be suitable for rich data assets in cloud data marketplaces. In SOA, QoS models for Web services have been well-researched and various techniques, methods and tools to support QoS modeling for Web services have been proposed [6], [7], [8]. However, they mainly focus on operational aspects of services like performance, reliability, availability, and security, while the data aspects related to data publishing are largely ignored. On the other hand, much effort has been spent on data quality from database perspectives and many metrics characterizing data quality have been proposed [9], [10]. Nevertheless, there is a lack of integration between data contract terms and service contract terms. In fact, no standard model of data contracts that could serve as a basis for the DaaS specification is available so far. Similarly, existing service licensing and service level agreements (SLAs), see e.g, [11], [12], are mainly for "operational" service APIs and they do not include mechanisms to deal with data contract terms. In specific domains, some data licensing models exist but they are not standards (e.g., see [13]), so they cannot be used in the DaaS model.

To support the composition of data sources in the Internet, especially in the recent Web 2.0 phenomenon, many data composition tools have been developed [14], [15]. However, existing techniques mainly focus on selecting data sources based on data structures and on dealing with syntax and semantics of the data, but neglecting data terms. Existing concepts, such as ad-hoc flows [16] and Web mash-up [17], are not integrated with data contracts. Contemporary service selection and combination techniques are built around the QoS, cost, and the semantics of service operations [7], [8], [18] without paying attention to data quality and data contracts. Our work does not focus on data composition taking into account data contracts but we support the development of data contracts that can be integrated into existing data discovery and composition tools.

III. Analysis of Data Contracts

A. Main Data Contract Terms

Although data include variety of properties, in this paper, we investigate some of the properties that are considered relevant in the perspective of contracts for DaaS. Our analysis is conducted based on studying of existing data licenses and agreements as well as service contracts. Some of the key properties of data that are significant in making a data contract in DaaS are elucidated as follows:

1) Data Rights: are the rights that the provider authorizes the consumer to exercise for data in DaaS. They are important for clarifying and assuring intellectual property rights. The set of common data right terms for data assets offered by existing DaaS and data marketplaces are the following:

- *Derivation*: any translation, adaptation, or any other alteration of a data asset or of a substantial part of the data makes a derivative data asset. This derivation includes, but is not limited to, extracting or re-utilizing the whole or a substantial part of the data in a new data asset.
- *Collection*: a collective data asset refers to a data asset in unmodified form as part of a collection of independent works in themselves that together are assembled into a collective whole.
- *Reproduction*: from a given data asset, temporary or permanent reproductions can be created by any means and in any form, in whole or in part, including of any derivative data assets or as a part of collective data assets.
- *Attribution:* the data provider may expect attribution (a kind of moral right) for the use of its data.
- *Noncommercial Use:* a data asset could be allowed/denied either for non-commercial purposes or for commercial purposes.

2) Quality of Data (QoD): multiple metrics can be used to describe data quality, such as completeness, reliability, accuracy, consistency, and interpretability [10]. In existing DaaS, QoD data certification is mentioned, e.g., in certain data assets in data.gov. However, it is not clear how to establish data quality certification. In our view, there exist several QoD metrics, each can have a unique name. The interpretation of a QoD metric for a data asset should be based on common agreements established in the domain in which the data asset is created and used. Usually, a QoD term specifies a range of possible values associated with a QoD metric.

3) Regulatory Compliance: it is important to protect privacy and confidentiality of information published, thus data assets are typically associated with many regulatory compliance. For example, in certain data assets in data.gov, data compliance is mentioned[4]. Some of the common regulatory compliance laws include the Healthcare Insurance Portability and Accountability Act (requiring the securing of patient information), Sarbanes-Oxley (SOX) Act (requiring company financial executives to be culpable for financial reporting), the European Union Data Protection Directive (protecting data privacy for citizens throughout the European Union), and so on. Most of the DaaS providers define specifications on data compliance terms. Most data compliance laws and regulation assume that the liable party controls the infrastructure and the location where the data is stored [19]. In our view, a compliance term can be specified as a term name and a set of values where values relate to respective compliance specifications.

4) Pricing Model: data consumers pay data providers for the right to use the data asset subject to the contract by the financial terms. The most common models for data pricing in DaaS and data marketplaces are transaction and subscription based model. The *Transaction model* allows DaaS providers to charge for each use. The *Subscription model* allows consumers to purchase data for a fixed term, during which time they automatically receive full support from providers including any upgrades or feature enhancements. For both models, pricing can be applied to the whole DaaS (e.g., Gnip[5] supports subscription) or specific data assets (e.g., the pricing model in Microsoft Azure Data Marketplace and Infochimps). In our view, pricing model is typically specified as a set of values per pricing plan which includes cost, usage time and/or maximum number of transactions to be applied to the whole DaaS or a particular data asset.

5) Control and Relationship: the control and relationship terms consist of evolution terms, support terms, indemnification, limitation of liability, and audits of contract compliance. Existing data contracts indicate control and relationship terms using similar ways in service contracts. Therefore, in our opinion, control and relationship terms in data contracts could reuse the similar ways of control and relationship terms in service contracts. From the modeling perspective, control and relationship terms can be specified as a set of $tuple(name, value)$ in which $name$ and $value$ have corresponding interpretations. For example, tuples $(LawandJurisdiction, US)$ and $(LawandJurisdiction, Austria)$ can be used to describe two different laws, US and $Austria$ to be enforced for data contracts. Here, all terms $LawandJurisdiction, US, Austria$ require concrete interpretation rules in order to understand their semantics.

B. Examples of Data Contracts

As mentioned above, the most popular form of data contracts is human-readable textual description of data agreements/licensing. Table I presents our analysis of data contracts in real-world data services in which all data contracts are in textual description for human beings, thus they do not foster the incorporation of data contracts in data discovery and composition. Overall, we have not seen a relevant difference between current data contracts/licensing and existing service contract/licensing with respect to the specification of scope of rights, control and relationships (e.g., warranty and liability). As shown in Table I, studied data contracts do not cover many aspects of contractual terms related to data. For example, most of the current DaaS contracts do not provide information about quality of data, which in fact should be one of the main terms in data contracts.

The analysis of data contracts heralds the requirement of new research directions for data contracts because data assets provided by DaaS have different properties, compared to software services. For example, data contract composition is needed when mashup of data from different data providers are performed. This composition consists in (i) retrieving comparable contractual terms from the different data contracts and (ii) evaluating the new contractual terms for the data mashup applying proper composition rules. Another example is data contract compatibility evaluation. This activity must be performed, e.g., before conducting a data mashup, to check if terms are compatible or not.

IV. Developing Data Contracts

A. Community View on Data Contract Development

As we discuss in the previous section, categories of data contract terms are limited. However, contract terms are diverse. In particular, data contract terms are contextual (e.g., based on laws of geographical regions and the domain of data assets). Furthermore, in many cases, data contract term values and their measurement units are also complex and contextual, e.g., one needs to make sure that the value “Austria” can be interpreted as a sub element of “EU” (European Union) in some specific contexts. Therefore, we do not expect that a unified specification for data contracts, with pre-defined term names, will be available and sufficient. In order to deal with data contracts in data marketplaces, we propose a different approach centered on a combination of community and people-centric collaboration.

First, we propose to enable community users to participate in defining (i) fundamental elements in data contracts, such as term categories, term names, term values and term units, (ii) rules for data contracts, such as syntax validation and evaluation rules, and (iii) common contracts and contract fragments (see Figure 1). The combination of community and people-centric collaboration is required to solve the heterogeneity of data contract terms, their values, and their measurement units. Such terms and units are contextual since different terminologies can be used in different domains. In our

[4] http://explore.data.gov/Law-Enforcement-Courts-and-Prisons/2008-Crime-in-the-United-States/bds9-jrca

[5] www.gnip.com

Contracts	Data Rights					Quality of Data		Compliance	Pricing Model		Control and Relationship			
	Derivation	Collection	Reproduction	Attribution	Non-commercial Use	Completeness	Accuracy		Transaction	Subscription	Warranty	Indemnity	Liability	Laws, Jurisdiction
AvianKnowledge.net (AKN) [20]				+	+				+	+	+	+	+	
Building Model Products [21]		+			+			+		+	+	+	+	+
Creative Common – Attribution-ShareAlike 2.0 Generic [22]	+	+	+	+							+	+	+	+
Consumer Expenditure Data [23]		+			+			+		+	+	+	+	+
Freebase data dump [24]	+	+	+	+							+	+	+	+
GBIF Data Usage Agreement [25]	+	+	+	+				+			+	+	+	+
Infochimps Twitter Census: Stock Twittes [26]	+	+						+	+		+		+	+
Open Data Commons Attribution License[27]	+	+	+	+							+	+	+	
Open Government License [28]	+	+	+	+									+	
U.S. Consumer Price Index - 1913 to Current [29]		+			+			+		+	+	+	+	+

TABLE I
EXAMPLE OF DATA CONTRACTS IN REAL-WORLD DAAS

view, common terminologies and domain-specific knowledge are used by domain experts to define term categories, term names, term values and term units that characterize a particular domain. Then, domain experts utilize these definitions and domain-specific knowledge to provide common contracts and contract fragments as well as customized validation and evaluation rules. This way is similar to the approach carried out in developing the Dublin core, which results in several fundamental and well-understood terms.

Second, by employing people-centric approach in establishing and developing data contracts, we propose that data providers and consumers can utilize fundamental elements to define their own contracts and evaluation techniques.

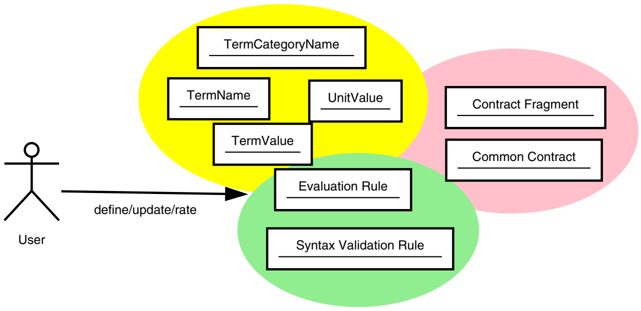

Fig. 1. Community contributions in data contracts

B. Representing Data Contract Terms

From our analysis, Table II presents possible ways to model data contract terms for different categories. Overall, we can represent a data contract term as a tuple of $(termName, termValue)$ in which $termName$ is either common terms established via standards/communities or user-specific and $termValue$ are the assigned values for $termName$. As shown in Table II, $termValue$s of a $termName$ can be a set, a single value, or a range.

C. Structuring Data Contracts

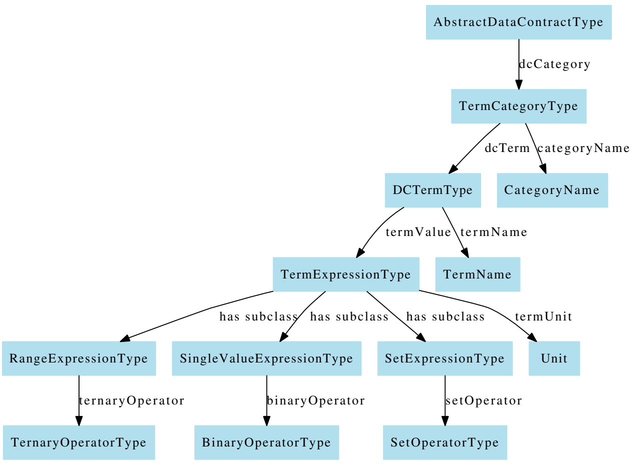

Fig. 2. (Simplified) abstract structure of data contracts

Category	Term Representation	Examples
Data Rights	$termName = \{val_1, val_2, \cdots, val_n\}$	$termName$={Derivation, Collection, Reproduction, Attribution, Noncommercialuse}, val_i = {Undefined, Null, Allowed, Required, True, False}
Quality of Data	$val_l \leq termName \leq val_u$	$termName$={Accuracy, Completeness, Uptodateness}, val_l and $val_u \in [0, 1]$
Compliance	$termName = \{val_1, val_2, \cdots, val_n\}$	$termName$ and val_i are any string, e.g., $termName$={PrivacyCompliance} and $termValue$={Sarbanes-Oxley (SOX) Act}
Pricing Model	$termName = (cost = val_1, usagetime = val_2, maximumuse = val_3)$	$termName$ is any string, e.g., MonthlyPayment; $val_1 \in R$, e.g., $cost = 50$ €, $val_2 = \{(end_t - start_t); UNLIMITED\}$ where $end_t, start_t \in datetime$, e.g., $usagetime = 30$ days; $val_3 \in N$, e.g, $maximumuse = 1000$ calls
Control & Relationship	$termName = val$	$termName$ and val are any string, e.g., $termName$={Liability, LawandJurisdiction} and $val = \{US, Austria\}$

TABLE II
DATA CONTRACT TERMS AND VALUES

Figure 2 presents our abstract data contract structure. By "abstract" we mean two aspects. First, this structure is not the final form of a data contract as it represents only contractual conditions without specifying on which data assets it is applied to. Second, the structure is not a proposal for final and concrete data contract specification, which can be seen and obtained by data consumers in data marketplaces, but it can be used to make abstract contracts from which concrete specifications will be generated for data consumers and providers. In the following, we explain the proposed structure.

TermCategoryType is used to specify one or more elements that categorize the terms specified in the contract. CategoryName is used to identify the category. From the analysis of data contract terms in the previous section, we identify that five CategoryName – (DataRight, QoD, Compliance, PricingModel and ControlRelationship) – are available. In principle, a new category name can be defined. A contract consists of a set of TermCategoryType elements, each includes a set of data contract terms described by DCTermType which covers specific-aspects and it is specified by means of a termName and a termValue. Each termValue is defined by means of a TermExpressionType that is specified by an operator and by a set of attributes that depend on the constraint operator used. Both the specification of qualitative and quantitative terms is supported. The former are terms defined through single and set expressions of qualitative values with well-defined meanings; the latter use expressions of numeric values, whose measurement units is specified by Unit.

All the above-mentioned types are associated with an identifier and set of tags, specified by Identifier and Tag, respective. Using Identifier we follow the Dublin core model to distinguish well-defined, agreed categories, terms, and values. Using Tag, we allow community users to specify tags to support searching terms, values and contracts.

D. Rules for Data Contracts

Due to the heterogeneity of data contract terms, a wide set of rules must be defined. We propose to have two different types of rules: syntax validation rules and evaluation rules. Syntax validation rules are used to verify if the syntax of fundamental elements is correct or not. Evaluation rules (examples are in Table III) are used to evaluate and interpret terms defined in data contracts specified by data providers, data consumers and DaaS service providers. Here, we do not define how many terms and which types of terms could be evaluated by a rule. In our work, we focus on defining a general way for developing rules to support the evaluation of contracts based on our model.

Category	Evaluation Rule	Examples
Data Right	Subsumption	(Derivation=allowed) $\supset$ (Collection=allowed)
QoD	Binary, Ternary	$Accuracy = 0.9$ satisfies a request for $Accurary \in [0.7, 0.95]$

TABLE III
EXAMPLE OF DATA CONTRACT CATEGORIES AND EVALUATION RULE MAPPING

Rules, however, are tightly coupled to the syntax for representing contract elements. From the implementation point of view, rules can be implemented using different techniques, such as RuleXML, OWL rules, or specific rule languages.

E. Guidelines for the Development of Applications

Several applications for the management of data contracts can be developed by utilizing our proposed data contract model and approach. Examples of such applications are:

- *data contract compatibility evaluation*: when we intend to combine multiple data assets, each governed by a data contract, we need to check whether data terms associated with these data assets are compatible. As an example, compliance terms must be checked in order to avoid violation in privacy of the provided data.
- *data contract composition*: when we produce a new data asset from multiple data assets, each governed by a data contract, we need to associate the new data asset with a new data contract. This new data contract can be composed from existing data contracts. As an example, Data Rights of the new data asset can be composed from Data Rights terms of existing data contracts.

Describing these applications in detail is out of the scope of this paper. However, in general the development of these applications can be performed based on the following principles:

- For each `DCTermType` t_j in each `TermCategoryType` tc_i, we can extract the comparable terms from all the data contracts. For example, in the category of `DataRight`, comparable terms can be `Derivation`, `Composition` and `Reproduction`
- Then we can retrieve from the *rule repository* the evaluation rule associated with the `DCTermType` t_j. In cases, such a rule does not exist, one needs to develop it.
- Finally, we can execute the rule by passing the list of comparable terms extracted from the contracts.

These basic steps show how applications can utilize rules in order to implement the applications. After the execution of rules, the results must be collected and aggregated based on the functionality of the application.

V. Prototype and Experiments

A. Prototype

We choose to use the Resource Description Framework (RDF) to represent term categories, term names, term values and term units. As a consequence, we have rules developed atop RDF. Figure 3 describes our prototype. Our community-based term categories, names, values and units can be defined, edited and rated by community users (such as owners of data assets) via different processes. We use Allegro Graph[6] as our *Data Contract Knowledge Service*. By utilizing the RDF knowledge, data providers and consumers can edit and evaluate data contracts. The resulting contracts can be extracted into different formats, such as XML, JSON and RDF. These contracts can be associated with data assets, managed by DaaS, stored in other services (such as a data agreement exchange service for data marketplaces), or stored into *Data Contract Knowledge Service* as common, shared data contracts. In our current prototype, *Data Contract Knowledge Service* includes common terms, categories, and contracts (based on data contracts in Table I).

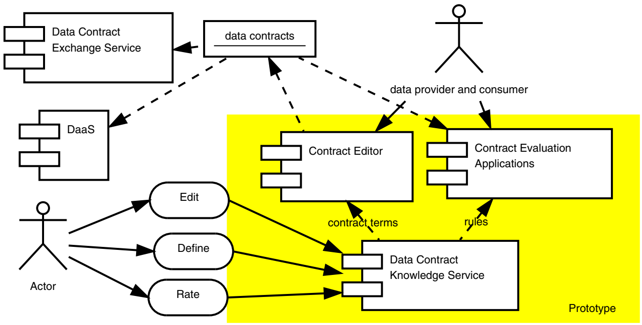
Data Contract Exchange Service
data contracts
DaaS
data provider and consumer
Contract Editor
Contract Evaluation Applications
Edit
Define
Rate
Actor
contract terms
rules
Data Contract Knowledge Service
Prototype

Fig. 3. Our prototype for data contract management

B. Illustrating Examples

Let us consider a cloud sustainability governance platform that manages very large sustainability monitoring data, such as the Galaxy platform [30]. Using the data and analysis capability in this platform, several summarized data could be provided. In our example, the platform provider would like to combine the real-time total and per capita of CO_2 emission of monitored buildings with an open government data asset about the CO_2 emission per capita in the national level[7] to show how green these buildings are.

In the first step, the provider decides to utilize Open Data Common terms for building CO_2 emission data but the provider wants to include certain quality of data and to prevent any derivation of the emission data. Thus, the provider first checks existing common terms in *Data Contract Knowledge Service* in order to reuse these terms. Figure 4 shows examples of existing common categories, term names, operators, units, and expressions as well as Open Data Commons (ODC) -based terms. By utilizing this existing knowledge, the provider defines a new data contract named `OpenBuildingCO2`. For this contract, the provider takes all ODC terms except `odcDerivation` (for derivation in data rights) and defines a category `obcQoD` (for quality of data) and a new term `obcDerivation` (for derivation). The new `obcDerivation` is defined by combining the common existing `Derivation` term and `NotAllowed` expression in the service. Listing 1 shows an excerpt of `OpenBuildingCO2` with respect to the `DataRight` category and the `Derivation` term. From this abstract data contract, concrete forms of the data contract can be generated in XML, RDF or JSON and then associated with appropriate data and DaaS.

The next step is to combine building CO_2 emission data with an open government data asset and an open map data[8]. Because the resulting data is a combination of different data assets controlled by different data contracts, the provider has to check the compatibility and even propose a new data contract for the combined data. In this experiment we assume that the open government data is based on the Open Government License [28] and we create an abstract contract – named `OpenGovernment` – for the open government data.

Listing 2 shows the rule used for composing an `Accuracy` term under `QoD` category from two inputs – `varAcc1` and `varAcc2`. This rule considers that `varAcc1` has `SingleValueExpressionType` with `atLeast` operator and `varAcc2` has `RangeExpresionType` with `interval` operator. Due to the operators and expression types, the composite accuracy, denoted by `compositeAccuracy`, will have `RangeExpressionType` and its lower bound value must be $max(varAcc1, varAcc2.lowerBound)$, while its upper bound will be the upper bound of $varAcc2$. Note that

[6] http://www.franz.com/agraph/allegrograph/

[7] such as http://www.apho.org.uk/resource/view.aspx?RID=91904

[8] e.g., The data in http://www.openstreetmap.org/ is governed by Creative Commons Attribution-ShareAlike 2.0

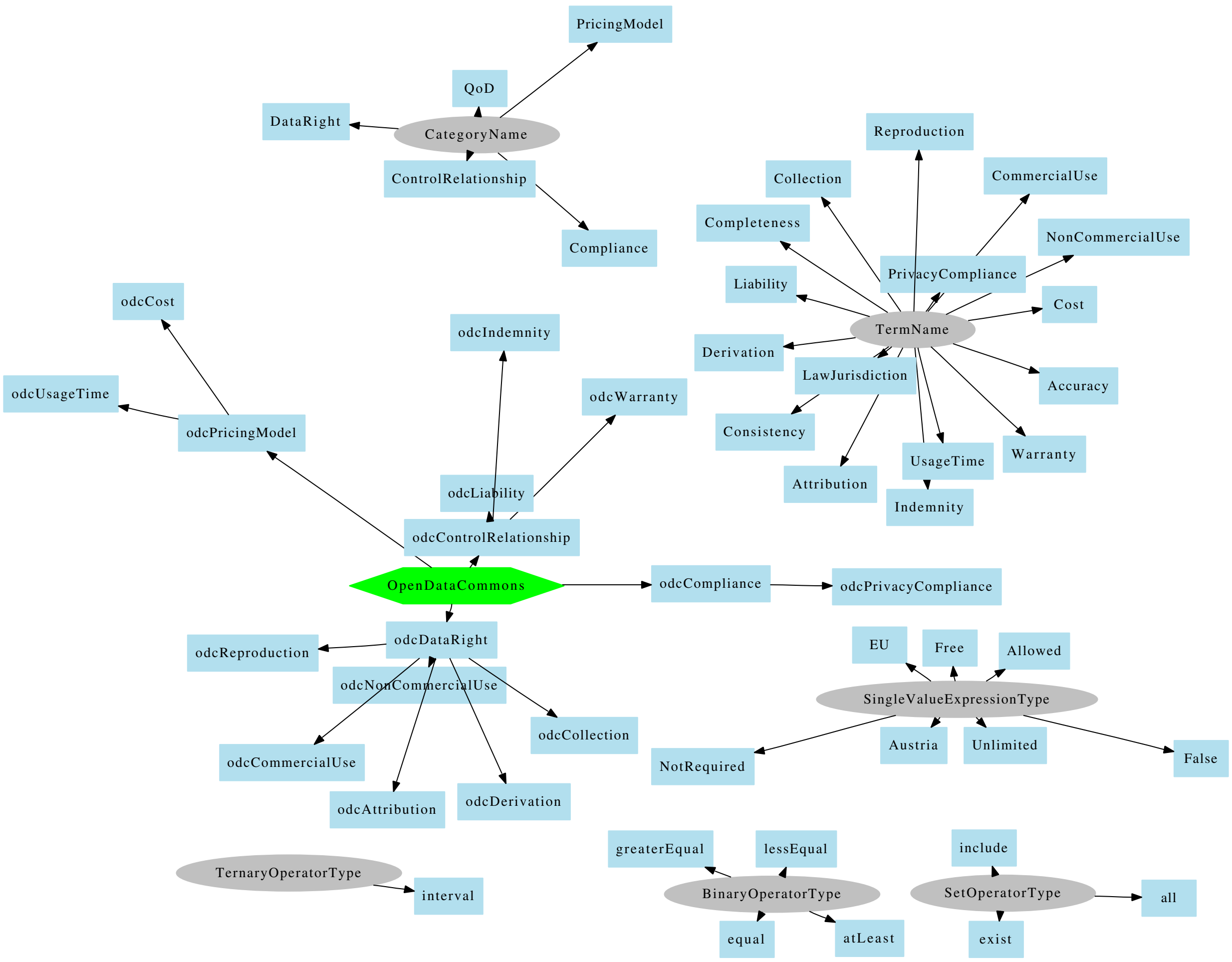

Fig. 4. Example of exploring common categories, terms, expressions, operators and values in *Data Contract Knowledge Service*, visualized by our prototype which utilizes GraphViz

depending on different `TermExpressionType` of input variables, we could have different rules for composing two terms under `QoD`. Thus, in principle, several rules can be developed and data contract applications can utilize these rules based on their needs. In our case, since `OpenGovernment` has no QoD term, the rule can take the QoD terms from `OpenBuildingCO2`.

Overall, our experiments illustrate the usefulness of having abstract data contracts being defined by utilizing existing categories and terms. The concrete data contracts in XML, JSON or RDF will facilitate the search and composition of data assets.

VI. Conclusion and Future Work

Although various data marketplaces and DaaS emerge and provide multitude sets of data, data contracts associated with these data so far are mainly written in textual form for human beings. Furthermore, what constitutes data contracts has not been deeply investigated. In this paper, we analyze data contracts in DaaS and data marketplaces in detail. Our approach for supporting the definition of data contracts that takes into account diverse types of data terms is based on the community model.

Specifying and evaluating data contracts surely are just at an early stage. Our future plan is to continue with our prototype and start to test it in a larger setting. Furthermore, we plan to integrate our work into a data agreement exchange service for data marketplaces and into data selection and composition framework.

Acknowledgments

This work is partially supported by the Vienna Science and Technology Fund (WWTF), project ICT08-032, by the Pacific Controls Cloud Computing Lab (PCCCL), and by the SAS Institute srl (Grant Carlo Grandi).

References

[1] H. L. Truong and S. Dustdar, “On analyzing and specifying concerns for data as a service,” in *APSCC*, M. Kirchberg, P. C. K. Hung, B. Carminati, C.-H. Chi, R. Kanagasabai, E. D. Valle, K.-C. Lan, and L.-J. Chen, Eds. IEEE, 2009, pp. 87–94.

```
<Description rdf:about=".../adcm#
    OpenBuildingCO2">
 <ns1:dcCategory rdf:resource=".../adcm#
     odcDataRight"/>
     ...
</Description>
<Description rdf:about=".../adcm#odcDataRight"
    >
 <ns1:dcTerm rdf:resource=".../adcm#
     odcDerivation"/>
     ...
 <rdf:type rdf:resource=".../adcm#
     TermCategoryType"/>
</Description>
<Description rdf:about=".../adcm#odcDerivation
    ">
 <ns1:termName rdf:resource=".../adcm#
     Derivation"/>
 <ns1:termValue rdf:resource=".../adcm#
     NotAllowed"/>
 <rdf:type rdf:resource=".../adcm#DCTermType"/
     >
     ...
</Description>
```

Listing 1. Simplified excerpt of OpenBuildingCO2

```
PREFIX
  adcm: <http://www.infosys.tuwien.ac.at/SOD1/
       adcm#>
CONSTRUCT {
  adcm:compositeAccuracy adcm:lowerBound ?
       compositeLowerBound .
  adcm:compositeAccuracy adcm:upperBound ?
       compositeUpperBound .
}
WHERE {
  ?varAcc1 rdf:type adcm:
       SingleValueExpressionType .
  ?varAcc1 adcm:numericValue ?value .
  ?varAcc1 adcm:binaryOperator adcm:atLeast .
  ?varAcc2 rdf:type adcm:RangeExpressionType .
  ?varAcc2 adcm:lowerBound   ?lowerBound .
  ?varAcc2 adcm:upperBound   ?upperBound .
FILTER (?value <= ?upperBound) .
LET (?compositeLowerBound := afn:max(?value, ?
    lowerBound)) .
LET (?compositeUpperBound :=?upperBound) .
}
```

Listing 2. An example of a composition rule for QoD

[2] B. C. M. Fung, T. Trojer, P. C. K. Hung, L. Xiong, K. Al-Hussaeni, and R. Dssouli, "Service-oriented architecture for high-dimensional private data mashup," *IEEE Transactions on Services Computing*, no. PrePrints, 2011.

[3] P. Miller, R. Styles, and T. Heath, "Open data commons, a license for open data," April 2008, copyright is held by the author/owner(s).LDOW2008, April 22, 2008, Beijing, China. [Online]. Available: http://events.linkeddata.org/ldow2008/papers/08-miller-styles-open-data-commons.pdf

[4] R. Iannella, "Open digital rights language (odrl) version 1.1," World Wide Web Consortium (W3C), 2002. [Online]. Available: http://www.w3.org/TR/odrl/

[5] "Onix-pl," http://www.editeur.org/21/ONIX-PL/.

[6] K. Lee, J. Jeon, W. Lee, S.-H. Jeong, and S.-W. P. (eds.), "QoS for Web Services: Requirements and Possible Approaches," Nov. 2003, w3C Technical Report. [Online]. Available: http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/

[7] S. Ran, "A model for web services discovery with qos," *SIGecom Exch.*, vol. 4, no. 1, pp. 1–10, 2003.

[8] X. Wang, T. Vitvar, M. Kerrigan, and I. Toma, "A qos-aware selection model for semantic web services," in *Proc. ICSOC 2006*, ser. Lecture Notes in Computer Science, vol. 4294. Springer, 2006, pp. 390–401.

[9] L. L. Pipino, Y. W. Lee, and R. Y. Wang, "Data quality assessment," *Communications of the ACM*, vol. 45, pp. 211–218, 2002.

[10] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino, "Methodologies for data quality assessment and improvement," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–52, 2009.

[11] G. R. Gangadharan and V. D'Andrea, "Licensing services: Formal analysis and implementation," in *Proc. ICSOC 2006*, ser. Lecture Notes in Computer Science, vol. 4294. Springer, 2006, pp. 365–377.

[12] A. Keller and H. Ludwig, "The wsla framework: Specifying and monitoring service level agreements for web services," *J. Network Syst. Manage.*, vol. 11, no. 1, 2003.

[13] Committee on Licensing Geographic Data and Services, National Research Council, Ed., *Licensing Geographic Data and Services*. The National Academies Press, 2004.

[14] G. Di Lorenzo, H. Hacid, H.-y. Paik, and B. Benatallah, "Data integration in mashups," *SIGMOD Rec.*, vol. 38, no. 1, pp. 59–66, 2009.

[15] V. Hoyer and M. Fischer, "Market overview of enterprise mashup tools," in *ICSOC '08: Proceedings of the 6th International Conference on Service-Oriented Computing*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 708–721.

[16] M. Voorhoeve and W. M. P. van der Aalst, "Ad-hoc workflow: Problems and solutions," in *Procc. DEXA Workshop*, 1997, pp. 36–40.

[17] X. Liu, Y. Hui, W. Sun, and H. Liang, "Towards service composition based on mashup," in *Proc. IEEE SCW 2007*. IEEE Computer Society, 2007, pp. 332–339.

[18] B. Blau, W. Michalk, D. Neumann, and C. Weinhardt, "Provisioning of Service Mashup Topologies," in *Proceedings of the 16th European Conference on Information Systems (ECIS)*, Galway, Ireland, June 2008.

[19] C. Wang, S. Balaouras, J. Staten, O. King, and L. Nelson, "Compliance with clouds: Caveat emptor," Forrester Research Report, Tech. Rep., 2010.

[20] "Akn data sharing policy," http://www.avianknowledge.net/content/about/akn-data-sharing-policy, last access: 25 July, 2011.

[21] "Building model products," Available in Microsoft Azure – https://datamarket.azure.com/dataset/bfa417be-be79-4915-82c7-efae9ced5cb7, last access: 21 Aug, 2011.

[22] "Creative common – attribution-sharealike 2.0 generic (cc by-sa 2.0)," http://creativecommons.org/licenses/by-sa/2.0/.

[23] "Consumer expenditure data," Available in Microsoft Azure – https://datamarket.azure.com/dataset/1a89a286-6ff2-4cc1-a215-ea4370259049, last access: 21 Aug, 2011.

[24] "Freebase data dump," Available in Amazon Public Dataset – http://aws.amazon.com/datasets/2320?_encoding=UTF8{\{\&}}jiveRedirect=1, last access: 21 Aug, 2011.

[25] "Data usage agreement – gbif (global biodiversity information facility)," http://data.gbif.org/terms.htm.

[26] "Twitter census: Stock twittes," http://www.infochimps.com/datasets/twitter-census-stock-tweets.

[27] "Open data commons attribution license," http://opendatacommons.org/licenses/by/, last access: 25 July 2011.

[28] "Open government license," http://www.nationalarchives.gov.uk/doc/open-government-licence/, last access: 25 July, 2011.

[29] "U.s. consumer price index - 1913 to current," Available in Microsoft Azure – https://datamarket.azure.com/dataset/26058d69-5cad-4a7c-9a14-a21a0c40de86, last access: 21 Aug, 2011.

[30] "The pacific controls galaxy," http://pacificcontrols.net/products/galaxy.html, Last access: 8 August 2011.