User Feedback Framework for Dataspace Improvement over Linked Data

Rene Sánchez
fernando.sanchezserrano@postgrad.manchester.ac.uk

The University of Manchester
Oxford Road, Manchester M13 9PL, UK

Abstract. The amount of data across the Web has grown considerably; locations, formats and technologies in which data is stored and retrieved presume heterogeneous sources where data itself is not interchangeably and thus incompatible among these sources. Emerging technologies such as the semantic web and linked data have led to this important growth. Pay-as-you-go data integration has proposed several methods and techniques to deal with this problem in an automatic and incremental fashion. User feedback is a prominent condition in pay-as-you-go data integration since it allows refining and boosting results from an integrated data schema. This work will propose different types of feedback as well as the required actions to assimilate such feedback in pay-as-you-go data integration approach.

1 Introduction

The amount of data across the Web has grown considerably; locations, formats and technologies in which data is stored and retrieved presume heterogeneous sources where data itself is not interchangeably and thus incompatible among these sources. Linked Data is a great opportunity to publish data across the web regardless its structure and domain; Linked Data and more specifically the Semantic Web are emerging technologies which have lead a significant growth and consequently increased the number of these sources where users can retrieve or search for data.

Retrieving information from disparate and heterogeneous sources has been the focus of recent research which has aided to give the illusion that a single and uniform data source is being accessed. Classical data integration approach involves a high front cost effort and high time consuming tasks when integrating data because human and expert intervention is highly needed. Pay-as-you-go data integration offers a great opportunity to integrate data in an automatic and incremental fashion and by following the Dataspace vision. Pay-as-you-go data integration encourages the process by requesting users to provide feedback in the light of improving the results and quality of data. Most of the current proposals, on integrating feedback, collect feedback in isolated mode and utilizes it in certain phase of the integration cycle [1]–[3]. There is evidence that a richer type of feedback can be used for acquiring and assimilating feedback onto a pay-as-you-go data integration [4] and that can be applied in a variety of applications on the whole pipeline of the data integration life cycle.

Feedback plays a decisive role in data integration with the aim to boost the quality of data that can be retrieved from a dataspace. The present work investigates on techniques to solicit and assimilate feedback on pay-as-you-go data integration approach over a Dataspace system for Linked Data sources.

1.1 Aim and Objectives

1.1.1 Aim

To investigate pay-as-you-go data integration for Linked Data based on user feedback.

1.1.2 Objectives

1) To put a plan an end-to-end pay-as-you-go test for Linked Data Integration.
2) To design, implement and evaluate techniques for improvement through the pay-as-you-go pipeline managing the feedback in Dataspace systems.
3) To evaluate such improvement techniques for real-world examples of open government data.

2 Overview

2.1 Linked Data

Linked Data is a publishing practice which allows to link, expose and connect pieces of data or information across the Web by implementing URIs (Uniform Resource Identifier) and RDF (Resource Description Framework) [5]. RDF offers the possibility to make statements between data and derive a meaning [6], it also provides a way to describe objects and connections between them. An increasing number of publishers have adopted RDF and has generated a huge volume of interlinked data on the web and has help to shape the Semantic Web.

Uniform Resources Identifier (URI) provides a simple and extensible means for identifying a resource, the concept is derived from the World Wide Web information initiative [7], an URI is a sequence of characters referred as the schema, authority, path, query and fragment. Lind Data and specifically RDF utilizes URIs to locate resources thru the HTTP protocol along the Web and as a generic means to assign names to things [8].

According to the above definition the HTTP URI http://dbtune.org/magnatune/track/3151 describes a music track of an artist call *Kperl* which title is *Suite No 5 in G major BWV 816*.

2.2 Resource Description Format (RDF)

RDF is composed by triples in the form of *subject, predicate, object* where the subject is an URI which identify the resource. The object is a literal value like string, number, or date; it can also be the URI of another resource, e.g., the name of a person or a date of birth. The predicate indicates the relation between a subject and an object, it can be in the form or "is a", "has a", "belongs to". A triple can be seen as the basic structure of a sentence:

Alan Turing	Was born	June 23, 1912
Subject	Predicate	Object

The possible values for a triple are URIs, strings or blank nodes; subject can take values from URIs and blank nodes, predicates take values from URIs exclusively and objects take values from any of URIs, strings or blank nodes.

2.3 Semantic web

The Semantic Web or Web of Data (WoD) is the result of a significant volume of Linked Data published across the Web in which information is intended to provide well-defined meaning and to allow humans and computers to work in cooperation [8], [9].

A significant contribution has been made by the World Wide Web Consortium (W3C) to enable the Sematic Web. As a part of this effort W3C has published several Web standards to publish, managed and retrieve information from the Semantic Web. These standards include a *common data model* for representing data in a structured form, *syntax* to allow machines parse text, *formal languages* to define the semantics of data as well as *declarative query languages* such as SPARQL [8].

The ability to connect data through links is of great significance since it brings out the idea to provide interlinked data to users from different sources and locations which in turn leverages the quality of data that users can consume.

Four principles for publishing and interlinking data on the Web have been proposed in [10]. These principles are as follows:

1. Use URIs as names to things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards.
4. Include links to other URIs so that they can discover more things.

The use and the adoption of the principles for publishing Linked Data have resulted in the bootstrapping of the LOD cloud [8], containing several interlinked RDF data sources, by March 2015[1] the Web of Data contains 3842 datasets which comprises more than 8.8 billion of triples.

2.4 Data Integration

The enormous number of published Linked Data along different sources and locations has opened the opportunity to consume data from more than a single point; however, every source describes its data in many different ways although they implement publishing standards which provide a homogeneous language for describing data.

Even though publishing best practices, interlinked data across the Web may result in poor or inaccurate data quality when searching, browsing and linking data from Linked Data sources, this represents a challenge to improve data integration over Linked Data and has encouraged the necessity to develop and investigate techniques to explore data integration approaches.

Related problems regarding data integration for Linked Data have been revealed in a previous work [11]. Some of the obstacles are dead-links, classes without formal definition, misuse of ontologies, incorrect data types and incoherent or contradictory data. Best practices in publishing and consuming Linked Data are not straightforward when providing a uniform and well-structured data result to users and moreover when trying the illusion that a single data sources is being accessed [12].

[1] http://stats.lod2.eu

2.5 Pay-as-you-go data integration

Pay-as-you-go data integration offers the opportunity to integrate data in an automatic and incremental fashion and by following the Dataspace foundation. Pay-as-you-go data integration approach envisions the idea that the benefits of classical data integration can be obtained at a lower cost over the time [1] by refining and improving the integration process over the principles of Dataspaces and supporting integration on demand and the integration of multiple and changing resources [12].

2.6 Dataspace

Dataspace systems support the idea and principles of pay-as-you-go data integration. Dataspaces rely on a framework which involves a life cycle that focuses its functionality on automation for bootstrapping and on feedback for improvement [13].

This framework comprises four main stages: initialization, use, improvement and maintenance.

Initialization: In this stage the system must be bootstrapped and this process is intended to be automatic, no human expertise involvement. Sources must be identified, that is, collecting information about their structure, technology and relationships between resources (schemas). This stage is also responsible to produce matches between schemas which yield associations and similarities between them. These associations are then used as input to generate a set of schematic correspondences which offer additional information about the relationships; this information brings evidence whereas a concept of a schema is associated with other concept from another schema by using the same name or by partitioning it horizontally or vertically along concepts.

Most of the methods used in a dataspace system are implementations of the Model Management operators, specially, the Model Independent Schema Management (MISM) [14], such operators are enacted in algebraic programs which derive into functional methods to perform operations between schemas, some operators are: match, merge, compose, extract and difference.

Finally, the set of schematic correspondences are used to set up mappings between schemas which describes one schema *s* in terms of another schema *r.*

Use: A dataspace system might include functionality to pose queries against the source schemas or the integrated schema, where a query q1 is rewritten, using the mappings, in terms of an integrated schema or in terms of the source schemas. This stage also includes methods for performing translations between mediated query languages and platform specific languages for example SQL, XML and SPARQL.

Improvement: This is stage is of great significance, it materializes one of the main claim of pay-as-you-go data integration, refinement. Here, users play an important role due to the feedback they provide. A dataspace system caters to acquire and assimilate feedback which must enhance the quality of the integration and must increase the quality of data.

Maintenance: This stage is responsible to propagate changes on sources to the integrated or merged schema; these changes include updating schematic correspondences, matches and mappings. Every product or artefact produced by the system must be informed in order to change or update its parameters and settings.

3 DSToolkit End-to-End Dataspace Platform

DSToolkit is the first dataspace management system which ensures the vision of dataspaces, it combines opportunities from incremental refinement and supports the pay-as-you-go data integration approach [2].

DSToolkit provides functionality for every stage in a dataspace life cycle; it includes operators for bootstrapping and refinement and it builds on the foundations of Model Management Systems.

The methods and operators are not restricted to those proposed by the model management; DSToolkit extends various kinds of morphisms which represent associations at different levels of the integration, e.g. matches, schematic correspondences and mappings.

3.1 DSToolkit Operators

3.1.1 Match

The match operator implements some matching algorithms proposed by [15] namely nGram and EditDistance (Levenshtein); the operator perform the calculation of every matcher and combines the score results into a similarity matrix where further calculation (average) can be made to get more confident results; in this sense, the operator also allows to set a threshold to avoid such results that may appear inconsistent. Figure 1 shows scores of a running match example between two schemas.

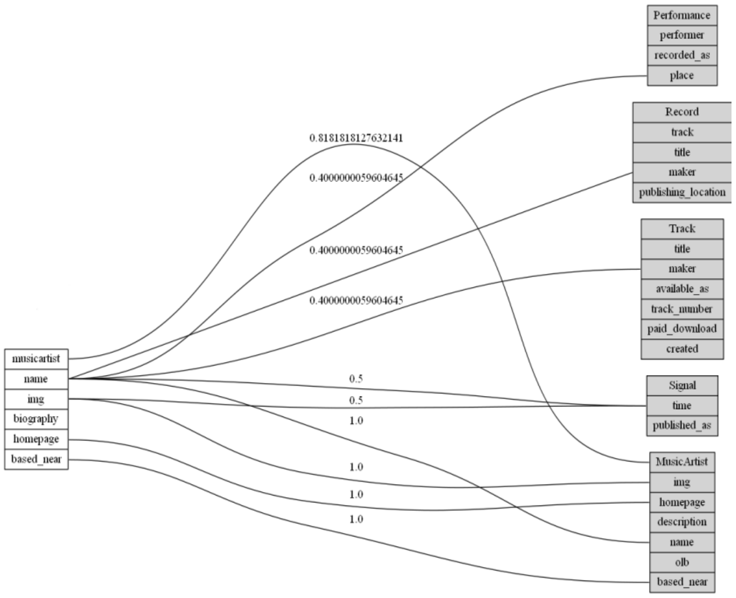


Figure 1 Match scores

3.1.2 InferCorrespondences

InferCorrespondences takes as input the matches between the schemas and produce a set of correspondences between them. The inferCorrespondence operator relies on an evolutionary algorithm proposed by [16] where four types of relationships (correspondences) are introduced. These correspondences are as follow:

Same Name Same Construct (SNSC). This type of correspondence outlines that schema objects keep the same name but also refers to the same piece of information, e.g. the attribute name of a person in two tables.

Different Name Same Construct (DNSC). The object's name are different one another but are related to the same instance of a domain.

Horizontal Partitioning (HP). An entity is partitioned into new entities where the attributes of the original entity remain in every new entity [16]. The original entity can be obtained by the union of every new entity.

Vertical Partitioning (VP). An original entity is partitioned into new entities which contain a subset of the attributes from the original one. The original entity can be composed by the join of the attributes of every new entity.

Figure 2 illustrates the types of schematic correspondences between two schemas.

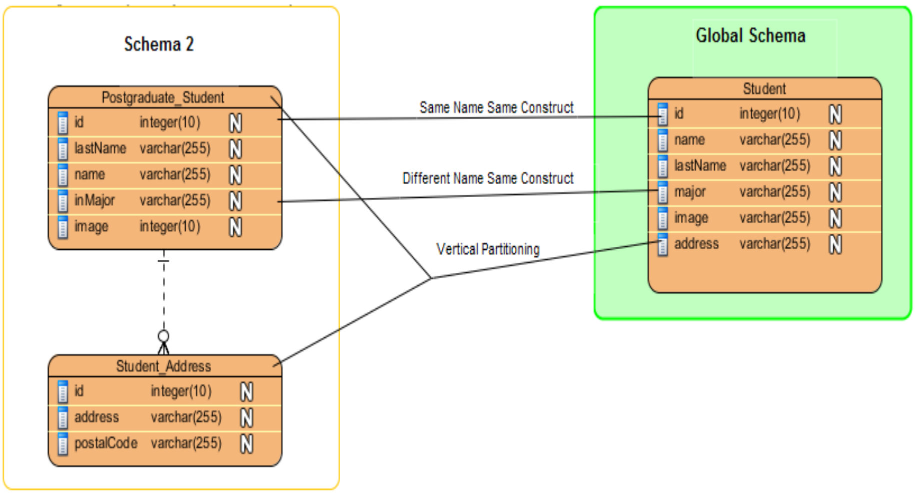


Figure 2 Schematic Correspondences

3.1.3 ViewGen

ViewGen creates the mappings derived from the schematic correspondences; these mappings are represented in a mediated language SMql, an independent query language over the model Model Independent Schema Management (MISM); this language is an extension of SQL and provides support for query evaluation across different data sources [14]. These mappings produce a representation of one model into an equivalent model and fulfil the requirements to perform translations over platform-specific data sources (SQL, XLS, SPARQL).

3.1.4 EvalIQ

This method is capable to evaluate a set of mappings and translated them into executable queries which can be posed over SQL and SPARQL end points to produce a result.

3.2 DSToolkit Evaluation and Extension

Previous investigations [8], [14], [16] have been focused on proposing approaches, methods and techniques to run through the bootstrapping phase whose results have been exposed by implementing functionality into the DSToolkit. Improvement, especially user feedback, represents an opportunity to extend such investigations and promote new ones in the light of completing the pipe line of pay-as-you-go data integration. My research will explore such approaches and techniques to gather and assimilate user feedback following the idea of pay-as-you-go data integration over dataspaces and consequently will extend the functionality of DSToolkit so it will become the first Dataspace system for pay-as-you-go data integration [2].

Introduction of different types of feedback and required actions to assimilate such feedback are key factors to enhance dataspaces over pay-as-you-go data integration process. These new methods and techniques must be implemented in DSToolkit to provide functionality over the whole life cycle of data integration.

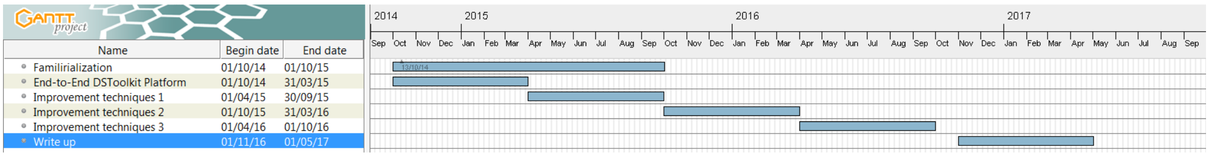
4 Conclusions

Pay-as-you-go data integration over linked data sources seem to be largely unexplored [1] since current approaches covers only certain types of feedback and its assimilation covers only specific phases of the integration. Moreover, these approaches relay on relational and xml models.

User feedback is a key factor to improve data integration because it helps to confirm inconsistency and to reduce the cost when adding new sources. Indeed, pay-as-you-go data integration as well as user feedback provide an affordable form to incrementally improve dataspaces over the time.

4 Produce Plan

Figure 3 shows the work plan of the research.


GANTT project
2014
2015
2016
2017
Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep
13/10/14

Name	Begin date	End date
Familirialization	01/10/14	01/10/15
End-to-End DSToolkit Platform	01/10/14	31/03/15
Improvement techniques 1	01/04/15	30/09/15
Improvement techniques 2	01/10/15	31/03/16
Improvement techniques 3	01/04/16	01/10/16
Write up	01/11/16	01/05/17

Figure 3 Work plan

References

[1] N. W. Paton, K. Christodoulou, A. a. a. Fernandes, B. Parsia, and C. Hedeler, “Pay-as-you-go data integration for linked data: opportunities, challenges and architectures,” *Proc. 4th Int. Work. Semant. Web Inf. Manag. - Swim ’12*, pp. 1–8, 2012.

[2] C. Hedeler, K. Belhajjame, L. Mao, C. Guo, I. Arundale, B. F. Lóscio, N. W. Paton, A. a a Fernandes, and S. M. Embury, “DSToolkit: An architecture for flexible dataspace management,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7100 LNCS, pp. 126–157, 2012.

[3] K. Belhajjame, N. W. Paton, S. M. Embury, A. A. Fernandes, and C. Hedeler, “Feedback-based annotation, selection and refinement of schema mappings for dataspaces,” *EDBT ’10 Proc. 13th Int. Conf. Extending Database Technol.*, pp. 573–584, 2010.

[4] K. Belhajjame, N. W. Paton, a a a Fernandes, C. Hedeler, and S. M. Embury, “User Feedback as a First Class Citizen in Information Integration Systems,” *Syst. Res.*, pp. 175–183, 2011.

[5] “Linked Data - Connect Distributed Data across the Web.” [Online]. Available: http://linkeddata.org. [Accessed: 23-Mar-2015].

[6] D. Anhai, H. Alon, and Z. G. Ives, *Principles of Data Integration*. 2012.

[7] T. Berners-Lee, R. Fielding, and L. Masinter, “Uniform resource identifiers (URI): generic syntax.” RFC 2396, August, 1998.

[8] P. Sciences and K. Christodoulou, “PAY-AS-YOU-GO DATA INTEGRATION OF LINKED DATA,” 2014.

[9] T. Berners-Lee, J. Hendler, O. Lassila, and others, “The semantic web,” *Sci. Am.*, vol. 284, no. 5, pp. 28–37, 2001.

[10] T. Berners-Lee, “Desing issues: Linked data,” 2006. [Online]. Available: http://www.w3.org/DesignIssues/LinkedData.html.

[11] A. Hogan, A. Harth, A. Passant, S. Decker, and A. Polleres, “Weaving the pedantic web,” 2010.

[12] C. Hedeler, K. Belhajjame, A. a a Fernandes, S. M. Embury, and N. W. Paton, “Dimensions of dataspaces,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5588 LNCS, pp. 55–66, 2009.

[13] K. Belhajjame, N. W. Paton, C. Hedeler, and A. a. a. Fernandes, “Enabling community-driven information integration through clustering,” *Distrib. Parallel Databases*, vol. 33, no. 1, pp. 33–67, 2014.

[14] C. Hedeler and N. W. Paton, “Utilising the MISM model independent schema management platform for query evaluation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7051 LNCS, pp. 108–117, 2011.

[15] H.-H. Do and E. Rahm, “COMA: a system for flexible combination of schema matching approaches,” *Proc. 28th Int. Conf. Very Large Data Bases*, pp. 610–621, 2002.

[16] C. Guo, C. Hedeler, N. W. Paton, and A. a a Fernandes, “EvoMatch: An evolutionary algorithm for inferring schematic correspondences,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8320 LNCS, pp. 1–26, 2013.