LDtogo: A Data Querying and Mapping Framework for Linked Data Applications

Niels Ockeloen, Victor de Boer, and Lora Aroyo

Web and Media Group, Department of Computer Science, The Network Institute,
VU University Amsterdam, Amsterdam, The Netherlands
{niels.ockeloen,v.de.boer,lora.aroyo}@vu.nl

Abstract. Despite its rising popularity, using and managing Linked Data remains a challenge for developers of mainstream web and mobile applications. In this demo we present "LDtogo", a framework that makes it easy for application administrators to integrate and maintain Linked Data when building new applications or when re-using existing ones. LDtogo does this by 1) supporting data processing by means of plug-ins, 2) providing an easy-to-use interface to create a customized API wrapper for applications and 3) using only technologies available on common web hosting platforms (e.g. LAMP hosting environments). Its modular structure and support for standards are important properties.

Keywords: RDF, Linked Data, Framework, Data Mapping, WSDL, Web Services, LAMP (Linux, Apache, MySQL, PHP), Shared Hosting.

1 Introduction

Despite the growing possibilities for the creation of Linked Data applications, the integration of Linked Data into applications has not yet become mainstream. For Linked Data applications to become mainstream, we need solutions that make it easier to develop, deploy and maintain Linked Data based applications.

In this demo, we present a framework that facilitates Linked Data based applications for mainstream developers.

2 Related Linked Data Solutions

There are several solutions that aim to increase or improve the usage of Linked Data in applications, such as The Callimachus project [1], the Large Knowledge Collider (LarKC) [2], ARC2 [3], the RAP Toolkit [4], the Linked Data API [5], ActiveRDF [6], Drupal [7] and others. However none of these provide a solutions that a) targets main stream web application developers involved in small or medium size projects; b) is usable and installable alongside the targeted application on a standard LAMP hosting environment; c) is easy to deploy; d) acts as a middle layer between Linked Data sources and the targeted application; e) provides a graphical user interface (GUI) that allows for easy setup and the division of roles; e) provides a configurable API

P. Cimiano et al. (Eds.): ESWC 2013, LNCS 7955, pp. 199–203, 2013.

interface towards the application; f) provides a generic plug-in system. General 'data mash-up' solutions such as Yahoo Pipes[1] and especially DERI Pipes[2] also provide support for Linked Data, however, they are not deployable on standard LAMP hosting environments as a standalone system. Yahoo pipes is offered as a web service while DERI pipes can be installed but requires Tomcat.

3 Description of the LDtogo Framework

We have developed 'LDtogo': A framework that facilitates easy data management for Linked Data applications. LDtogo is a (web) application in its own right, which manifests itself as a 'control panel' - type application that can be installed on any PHP/MySQL hosting environment, in many cases being the hosting environment of the 'target application'. By 'target application' we mean the application that will be using LDtogo to gather data. The target application can be of several types, e.g. a web application, a mobile app, or a native application that relies on HTTP requests to gather data. An example instance of LDtogo can be found at `http://www.amstertour.com/LDtogo/` with the target application AmsterTour[3] at `http://www.amstertour.com`.

Extracting the Data Selection Process. Using LDtogo, the data selection process no longer takes place within the target application, but in the framework. LDtogo has a modular setup, to allow for easy editing of the data selection process. The framework uses SPARQL to select data from remote sources, and has a plug-in system to allow for manipulation of the selected data. The devised queries can be tested immediately from within the framework, and the sequence of queries and plug-ins can easily be edited using a visual 'plug-in chain'. The target application interacts with the framework using an API interface. The target application can do a 'data request', i.e. a request to the framework to provide it with data, optionally providing input parameters. LDtogo will handle the request by selecting data using the chain of queries and plug-ins as setup for the specific request.

Flexibility in Application Maintenance and Reuse. With the data selection process handled by LDtogo, the way in which data is selected can be edited without the need to make changes to the application. This can include simple modifications e.g. to select additional data, as well as more drastic changes; By altering settings in the framework such as the used selection query and/or the used plug-ins, the main application (or modules thereof) can be reused with data from other Linked Data sources. These 'application modules' can be shared among users of the framework, while the same holds for plug-ins. To make the distinction between these two terms more clear, an application module is a piece of code that is part of the target application and does something with the data received from the framework (after the complete request has

[1] `http://pipes.yahoo.com/pipes/`

[2] `http://pipes.deri.org/`

[3] A mobile city tour guide application. See [8] for details.

been handled), while a plug-in does something with the data as part of the request handling within the framework. .

Components within LDtogo. LDtogo can be used to specify a collection of settings that together determine how data is selected from the Linked Data Cloud to fulfill the target applications' data needs. In order to be flexible and allow for re-use, we have chosen a modular setup for LDtogo. The framework consists of 'Linked Data Sources' (i.e., sources that provide an accessible endpoint), 'Queries' to select data from specified Linked Data Sources, 'Plug-ins' to alter selected data, 'Data Requests' that contain the settings to define how a specific request should be handled, and 'Application Modules' that group together multiple types of requests that can be made by the same target application. In short, these components in LDtogo together define how to transform the request input of data requests done by the target application into the desired request output.

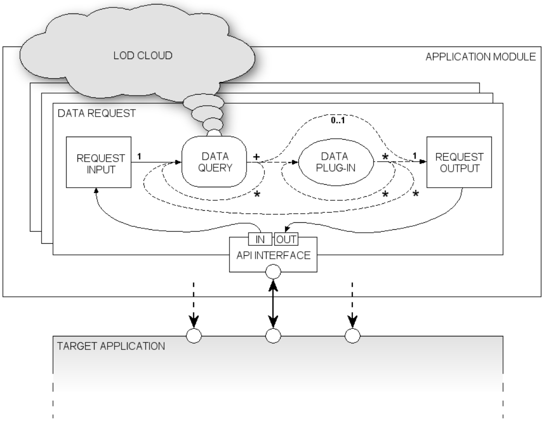


Fig. 1. Conceptual model of LDtogo. An Application Module groups together multiple Data Requests. The 'Data Chain' of a Data Request can consist of multiple queries and plug-ins.

Interplay of Components. These components together form the configurable set of tools that allow an application administrator to define and adjust the data selection strategy for an application without the need to modify the target application. The target application uses an API interface provided by the framework to request data needed for its operation. Figure 1 provides an overview of the various components and how they work together to fulfill a data request made by the target application.

The target application can perform a data request by making an API call to the framework. This can either be a HTTP request or a direct function call in PHP. When LDtogo receives a data request, it loads the settings for the particular request and executes the Data Queries and Plug-ins that together form the data chain for that request. The queries and plug-ins are executed in the order that was specified using the GUI (figure 2) of LDtogo. The end result of the data chain is used as response to the original data request made by the target application.

4 Usage Scenario

An advantage of using the LDtogo GUI is that duties can be split over multiple parties/persons. A developer can focus on developing the core functionality of the application, while a data administrator (e.g the data owner or someone familiar with SPARQL) can focus on the data selection process. The application developer can assume an API interface as suits him/her best, and together parties can use the GUI interface to map selected data onto the assumed API interface. Many organizations, for example in the cultural heritage sector, now have mature interesting data, but the number of applications is still not overwhelming. Using LDtogo, these organizations can cooperate with any 'random' developer to create interesting applications, where the organization fulfills the role of data manager within LDtogo. The development party can specify the desired API interface needed to implement the functionality requested, and together they can create the needed mappings and plug-in chain. Vice versa, a web application developer not familiar with Linked Data can contact a third party with knowledge in this area (possibly the publisher of the data) to assist with the data selection process.

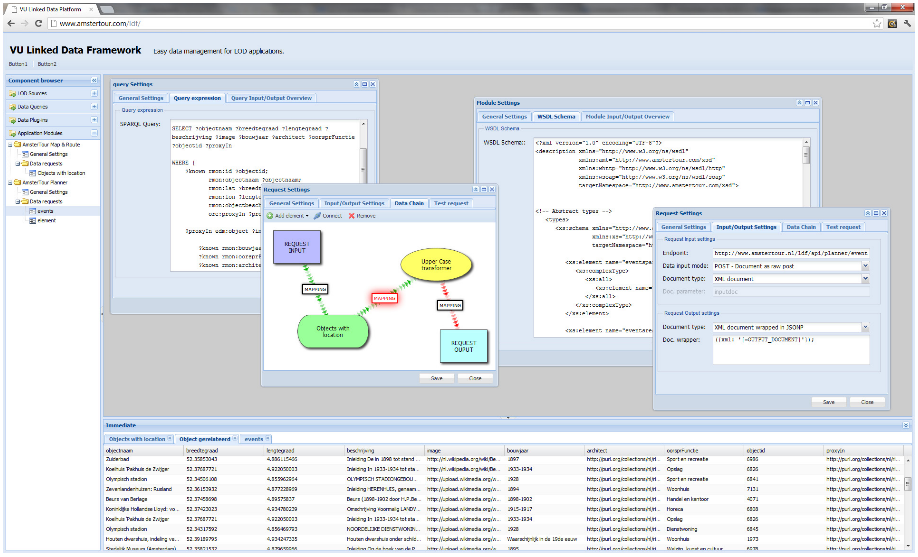
5 Demonstration

During the demonstration we will show the different components within the LDtogo framework, explain their purpose and how they interact with each other. We will demonstrate how to define Linked Data sources, Data Requests and Plug-ins within LDtogo and how to combine them to fulfill a data request made by a target application. To this end, we will use the data mapping solution in the GUI of LDtogo.

We will demonstrate LDtogo in combination with two applications; the aforementioned AmsterTour application and the Agora Touch Demonstrator of the Agora project[4]. Using the AmsterTour application, we will explain the steps involved in fulfilling a data request. We will show how to map the data selected for the application to the defined API interface using the visual data chain. Thereafter we will demonstrate how to alter the data using plug-ins. Along with that we will explain how the plug-in system works.

Next, we will demonstrate how existing applications can be re-used with LDtogo by looking at the Agora Touch demonstrator. We will take a brief look at the API interface of this application, developed before LDtogo was conceived, and use LDtogo to mimic the expected API interface and re-use the Agora Touch demonstrator with Linked Data from other sources.

[4] http://agora.cs.vu.nl/



SPARQL Query:

```
SELECT ?objectnaam ?breedtegraad ?lengtegraad ?
beschrijving ?image ?bouwjaar ?architect ?oorsprFunctie
?objectid ?proxyIn

WHERE {
    ?known rmon:id ?objectid;
           rmon:objectnaam ?objectnaam;
           rmon:lat ?breedt
           rmon:lon ?lengte
           rmon:objectbesch
           ore:proxyIn ?pro

    ?proxyIn edm:object ?im

       ?known rmon:bouwjaa
       ?known rmon:oorsprF
       ?known rmon:archite
```

WSDL Schema::

```
<?xml version="1.0" encoding="UTF-8"?>
<description xmlns="http://www.w3.org/ns/wsdl"
             xmlns:amt="http://www.amstertour.com/xsd"
             xmlns:whttp="http://www.w3.org/ns/wsdl/http"
             xmlns:wsoap="http://www.w3.org/ns/wsdl/soap"
             targetNamespace="http://www.amstertour.com/xsd">


<!-- Abstract types -->
  <types>
    <xs:schema xmlns="http://www.
               xmlns:xs="http://w
               targetNamespace="h

      <xs:element name="eventspa
        <xs:complexType>
          <xs:all>
            <xs:element name="
          </xs:all>
        </xs:complexType>
      </xs:element>

      <xs:element name="eventsre
```

Endpoint: http://www.amstertour.nl/ldf/api/planner/event

Data input mode: POST - Document as raw post

Document type: XML document

Doc. parameter: inputdoc

Document type: XML document wrapped in JSONP

Doc. wrapper: `({xml: '[=OUTPUT_DOCUMENT]'});`

objectnaam	breedtegraad	lengtegraad	beschrijving	image	bouwjaar	architect	oorsprFunctie	objectid	proxyIn
Zuiderbad	52.35853043	4.886115466	Inleiding De in 1898 tot stand ...	http://nl.wikipedia.org/wiki/Be...	1897	http://purl.org/collections/nl/ri...	Sport en recreatie	6986	http://purl.org/collections/nl/ri...
Koelhuis 'Pakhuis de Zwijger	52.37687721	4.922050003	Inleiding In 1933-1934 tot sta...	http://nl.wikipedia.org/wiki/Be...	1933-1934	http://purl.org/collections/nl/ri...	Opslag	6826	http://purl.org/collections/nl/ri...
Olympisch stadion	52.34506108	4.855962964	OLYMPISCH STADIONGEBOU...	http://upload.wikimedia.org/w...	1928	http://purl.org/collections/nl/ri...	Sport en recreatie	6841	http://purl.org/collections/nl/ri...
Zevenlandenhuizen: Rusland	52.36153932	4.877228969	Inleiding HERENHUIS, genaam...	http://upload.wikimedia.org/w...	1894	http://purl.org/collections/nl/ri...	Woonhuis	7131	http://purl.org/collections/nl/ri...
Beurs van Berlage	52.37458698	4.89575837	Beurs (1898-1902 door H.P.Be...	http://upload.wikimedia.org/w...	1898-1902	http://purl.org/collections/nl/ri...	Handel en kantoor	4071	http://purl.org/collections/nl/ri...
Koninklijke Hollandse Lloyd: vo...	52.37423023	4.934780239	Omschrijving Voormalig LANDV...	http://upload.wikimedia.org/w...	1915-1917	http://purl.org/collections/nl/ri...	Horeca	6808	http://purl.org/collections/nl/ri...
Koelhuis 'Pakhuis de Zwijger	52.37687721	4.922050003	Inleiding In 1933-1934 tot sta...	http://upload.wikimedia.org/w...	1933-1934	http://purl.org/collections/nl/ri...	Opslag	6826	http://purl.org/collections/nl/ri...
Olympisch stadion	52.34317592	4.856469793	NOORDELIJKE DIENSTWONIN...	http://upload.wikimedia.org/w...	1928	http://purl.org/collections/nl/ri...	Dienstwoning	6845	http://purl.org/collections/nl/ri...
Houten dwarshuis, indeling ve...	52.39189795	4.934247335	Houten dwarshuis onder schild...	http://upload.wikimedia.org/w...	Waarschijnlijk in de 19de eeuw	http://purl.org/collections/nl/ri...	Woonhuis	1973	http://purl.org/collections/nl/ri...
Stedelijk Museum (Amsterdam)	52.35821532	4.879659966	Inleiding Op de hoek van de P...	http://upload.wikimedia.org/w...	1895	http://purl.org/collections/nl/ri...	Welzijn, kunst en cultuur	6978	http://purl.org/collections/nl/ri...

Fig. 2. An overview screenshot of the LDtogo GUI, showing the component browser, the immediate testing panel, and several setting windows for a data request such as the data chain

References

1. The Callimachus Project, http://www.callimachusproject.org (accessed on January 15)
2. Fensel, D., Van Harmelen, F., Andersson, B., Brennan, P., Cunningham, H., Della Valle, E., Fischer, F., et al.: Towards LarKC: A Platform for Web-Scale Reasoning. IEEE International Conference on Semantic Computing 12(12), 524–529 (2008)
3. Nowack, B.: ARC: appmosphere RDF Classes for PHP Developers (2008)
4. Oldakowski, R., Bizer, C., Westphal, D.: RAP: RDF API for PHP. In: Proceedings of the Workshop Scripting for the Semantic Web (2005)
5. The Linked Data API, http://code.google.com/p/linked-data-api/wiki/Specification (accessed on January 15)
6. Oren, E., Delbru, R., Gerke, S., Haller, A., Decker, S.: Object-oriented semantic web programming. In: WWW ACM, pp. 817–824 (2007)
7. Corlosquet, S., Delbru, R., Clark, T., Polleres, A., Decker, S.: Produce and Consume Linked Data with Drupal! In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 763–778. Springer, Heidelberg (2009)
8. Ockeloen, C.J.: A Data Querying and Mapping Framework for Linked Data Applications. Master Thesis, VU University Amsterdam (2012), http://www.amstertour.com/docs/Master_Thesis.pdf (accessed on January 15)