The Information Workbench as a Self-Service Platform for Linked Data Applications

Peter Haase, Michael Schmidt, Andreas Schwarte

fluid Operations AG, Walldorf, Germany

Abstract. Pursuing the goal to lower the entry barrier into the world of Linked Data, we present the Information Workbench as a platform to support self-service Linked Data application development. Targeting the full life-cycle of Linked Data applications, it offers support in discovery and exploration of Linked Data sources, facilitates the integration and processing of Linked Data following a Data-as-a-Service paradigm (where remote data sources can be virtually integrated through a federation layer), and eases self-service UI development based on Semantic Wiki technologies, combined with a large set of widgets for interacting with the data. Coming with all these features, the Information Workbench can be used to rapidly build industrial-strength Linked Data applications.

1 Introduction

In recent years, a large amount of Linked Data (LOD) has been published on the Web [2]. Growing in size and domain coverage, this data becomes more and more interesting for building innovative applications that integrate heterogeneous data from different sources, in order to overcome the limitations of traditional data management systems. Apart from a new era of Web applications that exploit the LOD corpus, this development also offers opportunities in building novel applications for the enterprise by bringing together company-internal data with external data, to augment and contextualize internal knowledge bases [3].

The development of specific applications that benefit from Linked Data, though, comes with a variety of new challenges. First, at the data management side, developers are faced with a variety of new data formats and query languages (such as RDF, OWL, and SPARQL), but also struggle with heterogeneity at data level (facing Linked Data available via HTTP lookups, RDF dumps, and SPARQL endpoints) and new database systems and tools to store, process, and access this data. Second, once the relevant data has been identified and integrated into the system, Linked Data applications require new data interaction paradigms to deal with the specific challenges – and opportunities – of the underlying data formats, such as schema flexibility and data semantics. In particular, to leverage the benefits of Linked Data, aspects such as dynamic discovery of data sources, seamless integration of Linked Data from multiple sources, provenance, application development environments, and – last but not least – end-user interfaces that implement generic interaction paradigms with Linked Data are important aspects when building Linked Data applications.

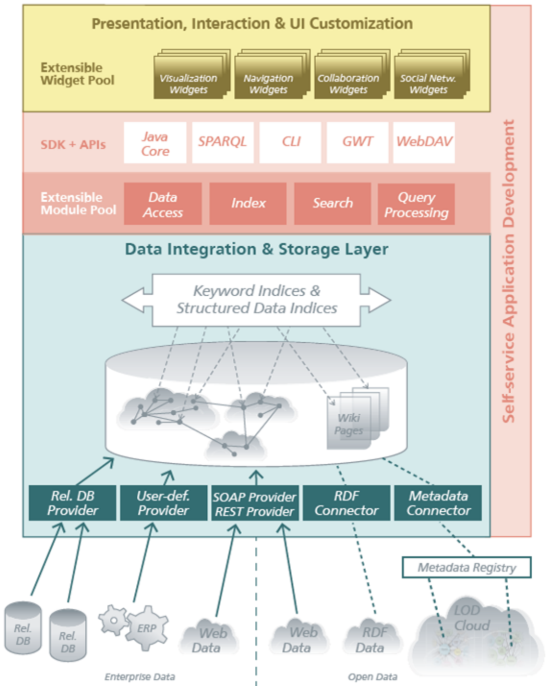


Fig. 1: Information Workbench Architecture

In this paper, we present the *Information Workbench* as a platform to support self-service Linked Data application development. We start with a discussion of the Information Workbench architecture in Section 2. In Section 3, we discuss how the self-service idea empowers all stages of the application development process, in particular (1) the deployment of the Information Workbench as a *virtual appliance* based on cloud technologies, (2) the *interactive exploration of public Linked Data*, (3) the ad hoc *virtual integration of data sets* by means of a federation layer, and (4) *self-service User Interface customization.*

2 The Information Workbench

Figure 1 shows a high-level architecture diagram of the Information Workbench platform. Starting at the bottom, data integration is supported by three complementary concepts.

- For every data source, a **data provider** can be instantiated. Such providers gather information from the source, convert it into the RDF data format, and materialize the RDF output in the central triple store. Apart from built-in mechanisms to integrate semantic data formats such as RDF dumps or data behind SPARQL endpoints, there are several generic providers supporting the integration of legacy and Web data sources.

- Data can be manually loaded in the platform using **predefined UIs and APIs** for data import. Supporting the integration of tabular data and spreadsheets, the Information Workbench also offers interfaces to Google Refine.
- Finally, our platform supports **virtualized data integration**, where local or public Linked Data sources (such as SPARQL endpoints) can be connected through a federation layer without materializing the data in the central store.

Once the data has been integrated, every resource in the data graph is automatically associated with a Semantic Wiki page, making it possible to bring together the structured data of the resources with unstructured information. Overcoming the limitations of traditional wikis, Semantic Wikis offer built-in mechanisms to access the underlying data, namely to extract and display information from the underlying data graph (e.g., by means of SPARQL queries) and to write directly to the store (e.g., by semantic links that connect resources at data level). Accounting for the coexistence of structured and unstructured data, the platform implements *advanced search and information access paradigms*, ranging from keyword search to complex graph pattern-based search, supporting the user in constructing expressive search queries by use of forms and state-of-the-art semantic query auto-completion techniques.

All the core system functionality is extensible and exposed to the outside by different APIs, including Java interfaces, an integrated SPARQL endpoint, or an interactive CLI. The Information Workbench also comes with an extensible AJAX-based Web frontend, which supports *mashups on both data and widget level*, making it possible to interlink data from multiple sources using different visualization and exploration widgets.

3 Self-Service Linked Data Application Development

The Information Workbench supports the whole Linked Data application development process. This process is aligned with the self-service idea, thus hiding the technical details behind data integration, data management, as well as the complexity of UI building. In particular, the platform offers support for (a) self-service data integration, (b) self-service analytics (comprising aspects like user-defined, mashed-up interactive dashboards, e.g., in the form of charts, time-series diagrams, geo-mappings, etc.), and (c) the ability to explore data and problems collaboratively and interactively in real time. In the following we describe how the Information Workbench supports self-service application development along the whole application development process. We illustrate the process using an example application in the media domain that intends to provide an end-user oriented music portal built on top of different public Linked Data sources.

(1) Provisioning the Platform as a Service

As an alternative to the download from the Information Workbench website, a fully equipped Information Workbench is available as a virtual appliance for immediate deployment from our self-service portal. We employ virtualization

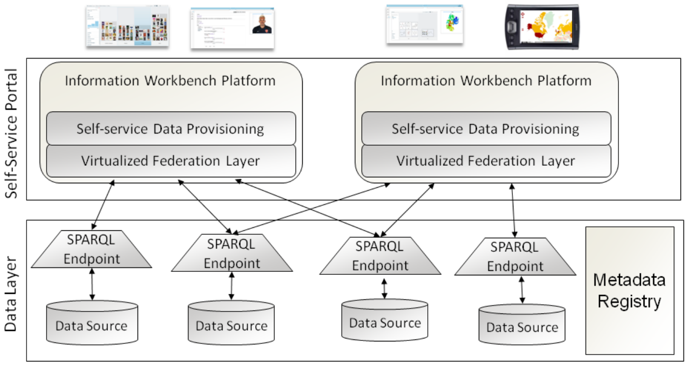

Self-Service Portal
Information Workbench Platform
Self-service Data Provisioning
Virtualized Federation Layer
Information Workbench Platform
Self-service Data Provisioning
Virtualized Federation Layer
Data Layer
SPARQL Endpoint
SPARQL Endpoint
SPARQL Endpoint
SPARQL Endpoint
Data Source
Data Source
Data Source
Data Source
Metadata Registry

Fig. 2: Architecture of the Self-Service DaaS Platform

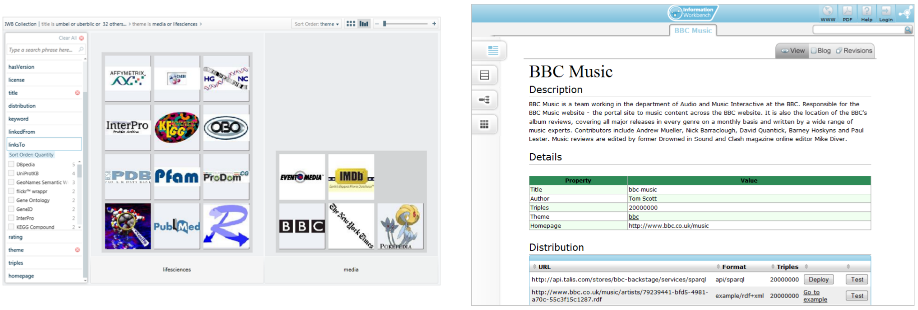
techniques to enable wizard-based self-service provisioning of both the application and user-selected data sets. Through our self-service portal users can choose an available Information Workbench template and deploy an instance of the system. We offer different templates that are pre-populated with data from different domains such as life science, governmental data, or media data. This data is either connected via public SPARQL endpoints (possibly through a federation), or as local RDF databases. Upon completion of the wizard, the Information Workbench instance is deployed in the hosting landscape of the service provider and henceforth publicly accessible from any computer with Internet access.

In our example application, we select a template designed for the media domain, which is configured to use the data from the LinkedBrainz project (a Linked Data version of MusicBrainz.org data) as well as the DBpedia dataset. The template-based provisioning of the Information Workbench (including the local data repositories) is completed in about 5–10 minutes.

(2) Data Source Discovery

The prerequisite of a self-service Linked Data platform is a rigorous implementation of the Data-as-a-Service (DaaS) paradigm, where users are able to discover, integrate, and consume available Linked Data ad hoc and on demand. DaaS relies on (1) the availability of individual data sets that can be deployed independently (yet may be interlinked with each other) and (2) the availability of meta information about the content of and access mechanisms to the sources.

Building upon these prerequisites, Figure 2 shows how the Information Workbench implements the DaaS paradigm. At the bottom is the data layer, which includes public data sources from different publishers as well as local data sources (typically in the form of SPARQL endpoints). An integral component of the Information Workbench is a metadata registry, which provides a unified view on metadata and statistics about these data sets by integrating information from different data registries and markets such as `http://ckan.org`, `http://data.gov`, and others. As standard vocabulary for representing the metadata we use DCAT and VoID [1]. The user can explore the metadata catalog from within the Information Workbench along all meta information that is available, including domain and size of the data sets, origin, licenses, information about interlinked sources, etc. Figure 3(a) shows a screenshot of the PivotViewer interface for visual exploration of available data sources. In our example, we are interested in an additional dataset from the media domain: The BBC Music data set. Selecting the data set takes us to the detail page for BBC Music (cf. Figure 3(b)), with a description, statistical data, as well as information about the distribution of the data set.



BBC Music

Description

BBC Music is a team working in the department of Audio and Music Interactive at the BBC. Responsible for the BBC Music website - the portal site to music content across the BBC website. It is also the location of the BBC's album reviews, covering all major releases in every genre on a monthly basis and written by a wide range of music experts. Contributors include Andrew Mueller, Nick Barraclough, David Quantick, Barney Hoskyns and Paul Lester. Music reviews are edited by former Drowned in Sound and Clash magazine online editor Mike Diver.

Details

Property	Value
Title	bbc-music
Author	Tom Scott
Triples	20000000
Theme	bbc
Homepage	http://www.bbc.co.uk/music

Distribution

URL	Format	Triples		
http://api.talis.com/stores/bbc-backstage/services/sparql	api/sparql	20000000	Deploy	Test
http://www.bbc.co.uk/music/artists/79239441-bfd5-4981-a70c-55c3f15c1287.rdf	example/rdf+xml	20000000	Go to example	Test

Fig. 3: (a) Data Set Search and Discovery Interface, (b) Ad Hoc Deployment

(3) Data Integration and Deployment

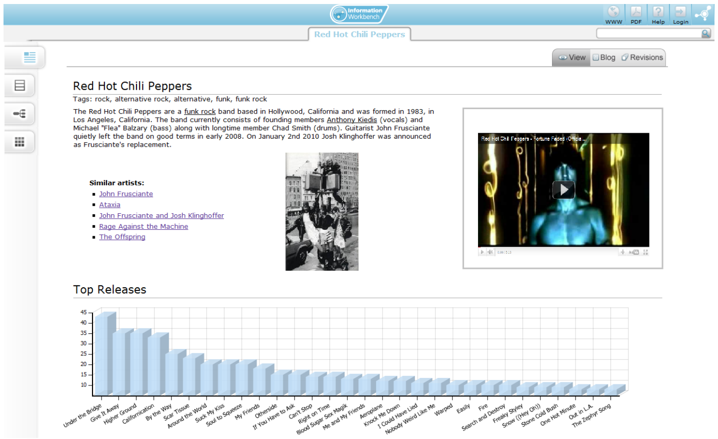
Once a relevant data source has been identified, its data can be integrated into the system by the click of a button. Depending on the access mechanisms that are supported for the data set, the Information Workbench offers options to either load data into the local repository (if an RDF dump is available) or to connect the data virtually through a federation layer (whenever there exists an open SPARQL endpoint). The integration approach is transparent to the end user, which means that (i) within the deployment process the user does not need to be concerned with aspects of physical distribution, access protocols and interfaces, underlying data models etc., and (ii) the details of the integration are hidden at runtime, so both local data and virtually integrated data sources can be queried and accessed in an integrated way.

Virtualized data integration is realized by the use of FedX, a federation layer for Linked Data, which we developed specifically for the transparent access to Linked Data [4]. FedX is a practical framework that incorporates novel optimization techniques for efficient federated query processing in a distributed setting and supports the ad-hoc integration of data sources at runtime.

(4) Customization of the User Interface

The Information Workbench provides a rich UI out of the box, which enables basic interactions with the data as soon as the data has been integrated into the platform. The basic interaction components include tabular and graph-based visualization and exploration widgets, Semantic Wiki pages for editing and annotating the data, as well as components for semantic and faceted search.

The user interface can be customized using a rich pool of widgets that are shipped with the Information Workbench, targeting different data interaction



Red Hot Chili Peppers

Tags: rock, alternative rock, alternative, funk, funk rock

The Red Hot Chili Peppers are a funk rock band based in Hollywood, California and was formed in 1983, in Los Angeles, California. The band currently consists of founding members Anthony Kiedis (vocals) and Michael "Flea" Balzary (bass) along with longtime member Chad Smith (drums). Guitarist John Frusciante quietly left the band on good terms in early 2008. On January 2nd 2010 Josh Klinghoffer was announced as Frusciante's replacement.

Similar artists:
- John Frusciante
- Ataxia
- John Frusciante and Josh Klinghoffer
- Rage Against the Machine
- The Offspring

Fig. 4: Media Scenario Showing a Mash-Up for the Red Hot Chili Peppers

paradigms such as semantic search, data visualization (e.g., as tabular results, graphs, charts, timelines, maps, heatmaps, etc.), navigation and exploration of the data (e.g., a graph-based data browser or a PivotViewer), collaborative editing, knowledge acquisition, as well as mashups with external data sources (e.g., Youtube, NY Times data, Facebook, and Twitter). All widgets can be easily embedded into Semantic Wiki pages and are specified using a declarative wiki-based syntax. With very little effort, the standard views can be customized to create domain- and application specific interfaces. Figure 4 shows a screenshot of the page of the Red Hot Chili Peppers. It is based on a wiki-based template definition that describes how resources of type musical artist are presented. The chart at the bottom, for instance, is generated by the following widget declaration:

```
{{#widget: Chart |
query ='SELECT (COUNT(?release) AS ?count) ?label WHERE {
  ?? foaf:made ?release .
  ?release rdfs:label ?label .
} GROUP BY ?label ORDER BY  DESC(?count) LIMIT 30' |
chart = 'BAR_VERTICAL' | input = 'label' | output = 'count' }}
```

In addition, the page includes a description of the artist taken from a remote Web service (Last.FM) and an embedded live video from YouTube for this artist.

References

1. Keith Alexander, Richard Cyganiak, Michael Hausenblas, and Jun Zhao. Describing linked datasets - on the design and usage of void. In *In Linked Data on the Web Workshop (LDOW 09), in conjunction with WWW '09*, 2009.
2. Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.
3. Michael Hausenblas. Exploiting linked data to build web applications. *IEEE Internet Computing*, 13(4):68–73, 2009.
4. Andreas Schwarte, Peter Haase, Katja Hose, Ralf Schenkel, and Michael Schmidt. FedX: Optimization Techniques for Federated Query Processing on Linked Data. In *ISWC*, 2011.