Industrial Data Space: Semantic integration of Enterprise Data with VoCol

Lavdim Halilaj, Niklas Petersen, Irlán Grangel-González, Christoph Lange, Steffen Lohmann, Christian Mader and Sören Auer

Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS)

Introduction

Nowadays, business processes are being digitized across all industries [5]. Just-in-time manufacturing and mass customization generate vast amounts of data at a faster pace than ever. Specialization and outsourcing multiply the number of actors involved in business exchanges. Data management is adapting to these trends: data quality is assured proactively and data is increasingly considered a strategic asset.

Increasing customization, specialization, and outsourcing also lead to increased data heterogeneity, formats, structures, and schemas involved in business processes. At the same time, the volume of unstructured and semi-structured data is growing exponentially. In such settings, it is challenging to:

- guarantee coherent and transparent data management,
- efficiently leverage data lake repositories and avoid data silos,
- establish data sharing agreements with strategic business partners for mutual benefit,
- analyze data in real time in an integrated way.

The problem of integrating data from different systems receives ever-increasing attention. Identifying the main terms across heterogeneous data sources by finding a consensus between the developers and defining a shared vocabulary is an effective approach to tackle this problem. However, this process, which we refer to as distributed vocabulary development, can be quite complex. In fact, the main challenge for vocabulary engineers is to work collaboratively on a shared objective in a harmonic and efficient way, while avoiding misunderstandings, uncertainty, and ambiguity.

The VoCol Approach

We designed VoCol as a holistic approach for realizing a full-featured vocabulary development environment centered around version control systems (VCSs). VoCol is a core component of the Industrial Data Space initiative[1]. It supports a fundamental round-trip model of vocabulary

[1] http://www.industrialdataspace.org/en/

development, consisting of the three core activities *modeling*, *population*, and *testing*, as illustrated in Figure 1.

In the spirit of test-driven software engineering, VoCol allows to formulate queries, which represent competency questions for *testing* the expressivity and applicability of a vocabulary a priori [1]. *Modeling* comprises the analysis and conceptualization of the domain and the specification of the vocabulary terms, such as classes, properties, and the relationships between them (also known as TBox). The creation of this terminology is realized using a logical formalism during the modeling activity [2]. VoCol integrates a number of techniques facilitating the conceptual work, such as automatically generated documentations and visualizations providing different views on the vocabulary as well as an evolution timeline supporting traceability. Once the vocabulary modeling has been completed, the next activity is typically *population*. It includes the addition of actual data in line with the defined classes and properties, also known as ABox [3]. For population, VoCol supports the integration of mappings between data sources and the vocabulary, including R2RML mappings to relational databases.

The governance of distributed vocabulary development is supported by the access control as well as branching and merging mechanisms of the underlying VCS system. As a result, VoCol bridges between the conceptual development of vocabularies and the operational execution in a concrete IT landscape. The implementation of VoCol is based on a loose coupling, leveraging the webhook method provided by many VCSs with tools and techniques focusing on particular aspects of vocabulary development. By providing Vagrant and Docker containers bundling all tools and encapsulating dependencies, VoCol is easily deployable or even usable as-a-service in conjunction with arbitrary VCS installations.

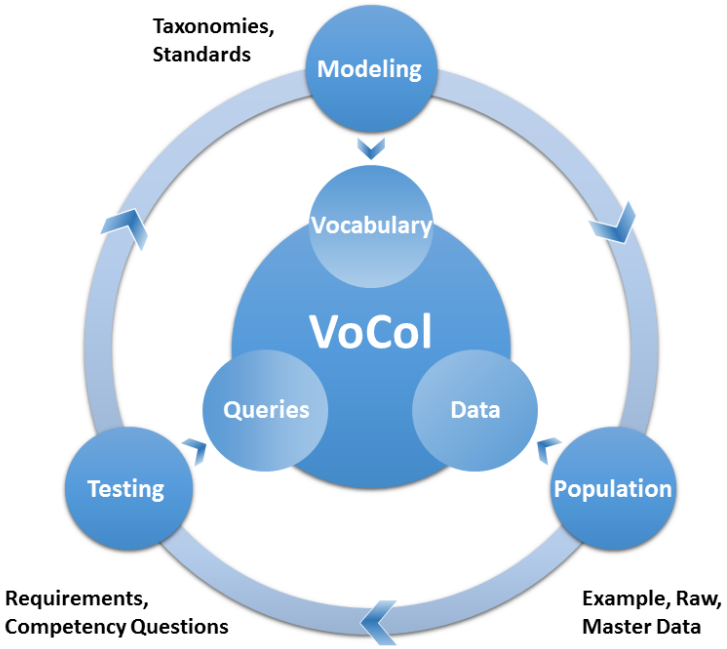

Figure 1. Round-trip vocabulary development supported by VoCol

Architecture

The integrated VoCol system architecture is illustrated in Figure 2. It is based on the Component Based Software Development (CBSD) principles [4], which promote the reuse of (off-the-shelf) components to develop large-scale systems. Each of the VoCol components is exchangeable and can be replaced by alternatives. In the following, these components are described in detail:

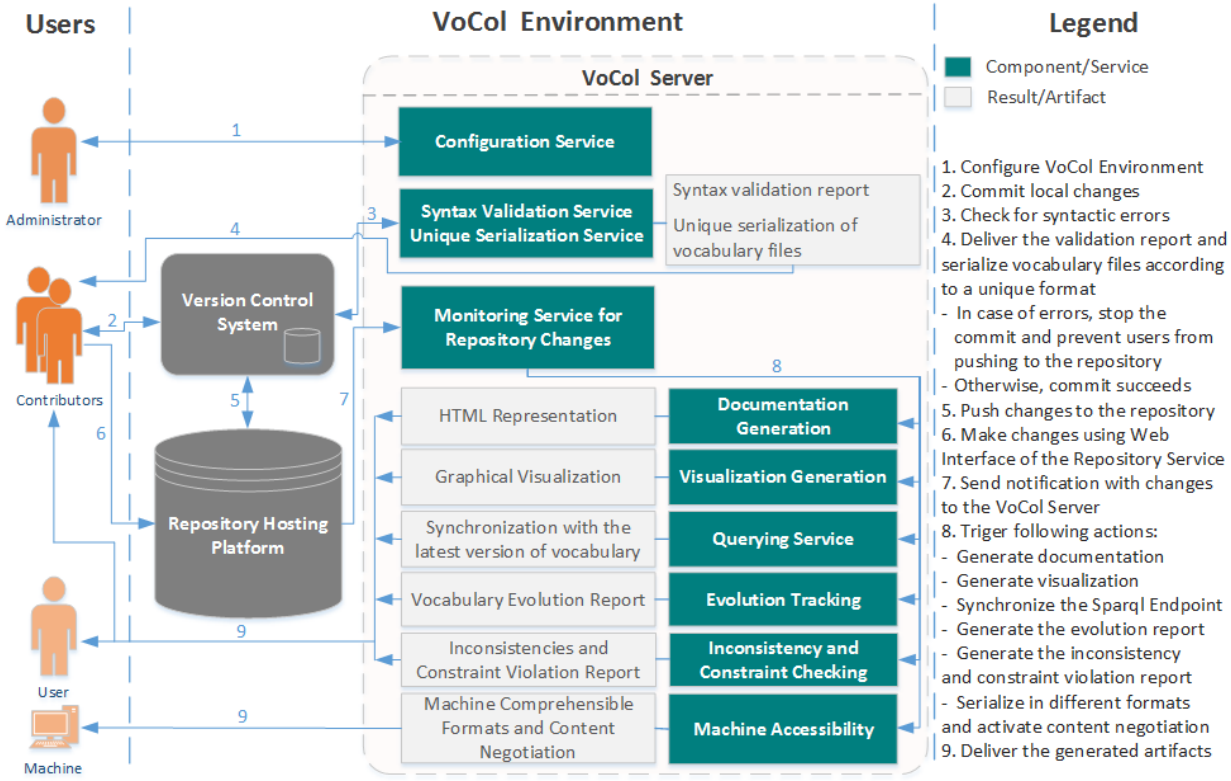

Figure 2. VoCol architecture and workflow.

Version Control System - To support the loosely coupled collaboration between vocabulary engineers without risking to lose data, a VCS component is essential. It is responsible for the management of vocabulary changes, such as change capturing and propagation: By capturing and storing the changes, various revisions of the vocabulary are created. To view the syntactic differences, a changeset must be computed by comparing two different revisions. In order to ensure a consistent development process, any change should be propagated to all other contributors. In addition, conflicts inevitably arise in an environment where multiple contributors are working simultaneously and changing vocabulary terms. The VCS ensures conflict resolution and allows integration of conflicting changes in an effective and easy way.

Since the VCS is the first component that is aware of changes, it is the core component of the VoCol system. Each additional feature that supports vocabulary development needs to be triggered by the VCS. Different *repository hosting platforms*, such as GitHub, GitLab, and Bitbucket, have been integrated into the VoCol environment in order to provide easy user access to the repository. Each of them can act as the repository storage in the distributed

setting where the vocabulary files are saved and accessed. Its access control mechanisms authenticate users with the right permissions. Furthermore, using the integrated issue tracker, contributors are able to discuss the vocabulary elements.

Syntax Validation - ensures that the latest revision in the VCS is always syntactically correct. Syntax validation could be realized at different stages of the overall workflow. However, with the aim to keep the requirements on the client side at a minimum level, we realized a syntax validation service in the backend. As a result, syntactically incorrect commits are rejected and a detailed error report is provided.

Unique Serialization - Different editors may produce different serializations of the vocabulary files. To overcome this problem, VoCol creates a unique serialization of vocabulary files before changes are pushed to the remote repository, thus preventing false-positive merge conflicts.

Documentation Generation - produces an HTML representation of the vocabulary. This permits contributors to easily navigate through the entire vocabulary and provides a concise but still detailed overview (cf. generated documentation for the *ChargingPoint* term in Figure 3).

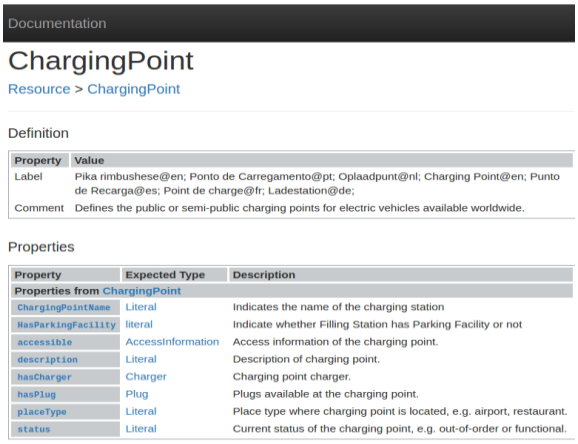

Figure 3. HTML Documentation

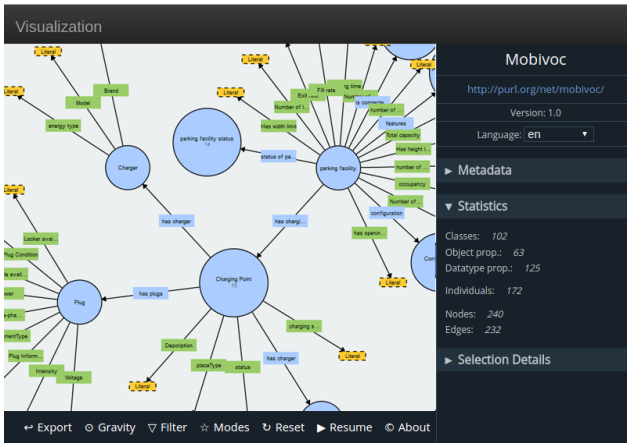

Figure 4. Visualization

Visualization Generation - By visualizing classes, properties and their connections, users are provided with a coherent view of the vocabulary. In addition, this service enables to explore multilingual vocabulary terms (cf. Figure 4).

Querying Service - VoCol integrates a SPARQL endpoint synchronized with the latest version of the vocabulary. During testing, queries derived from competency questions can be used to verify whether the vocabulary fulfills the domain requirements. All defined queries are stored in the repository and pre-loaded in the query user interface.

Inconsistency and Constraint Checking - After the changes have been pushed to the remote repository, checks for semantic inconsistencies and constraint violation are performed and their results are compiled in a detailed report.

Machine Accessibility - Using the content negotiation mechanism and dereferenceable URIs, VoCol delivers various machine-comprehensible representations. By specifying the content type in the HTTP header along with the resource URI, the vocabulary can be accessed by different software agents compliant with the linked data principles.

Evolution Tracking - detects semantic differences between vocabulary versions. It shows which classes and properties have been added, removed or modified, enabling users to see the vocabulary evolution over time (cf. Figure 5).

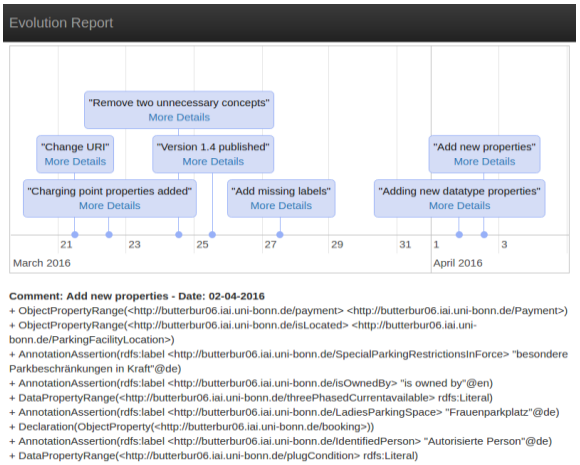

Figure 5. Evolution Tracking

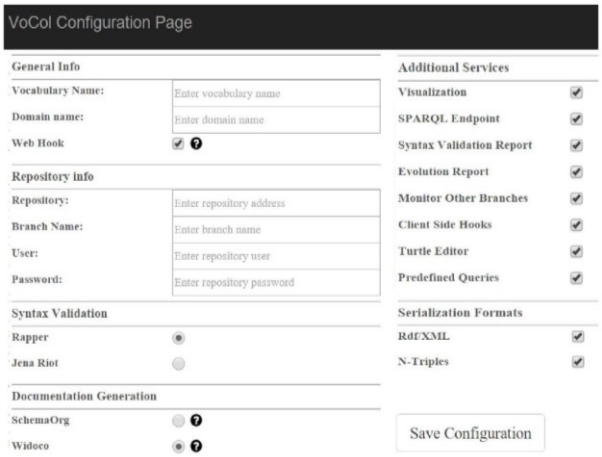

Figure 6. Configuration Page

Configuration Service - provides a Graphical User Interface to facilitate the configuration of VoCol. The VoCol administrator can choose between various alternative tools for syntax validation and documentation generation. Furthermore, other services can be activated or deactivated (cf. VoCol Configuration Page in Figure 6).

Monitoring Service - Repository hosting platforms expose most of their functionality via web service APIs. Any change pushed to the repository is delivered as a payload event to a VoCol monitoring service, which automatically invokes services for documentation generation, visualization, evolution tracking, etc.

Applicability in Industry

VoCol has been successfully applied in industrial use cases to enable semantic data integration over multiple heterogeneous data sources. It facilitates the development and maintenance of vocabularies that are based on standards and the intellectual property of the industrial partners. VoCol's loosely coupled architecture allows for the easy integration of additional features, such as components to support the definition of mappings between the developed vocabularies and legacy data sources of industry systems. Users are thus enabled to execute queries against multiple data sources of the legacy system, and gain new insights from the integrated data.

The semantic integration of heterogeneous data sources, as supported by VoCol, can significantly increase the data quality and ease the data access. Ultimately, it can lead to new business models and applications as well as an enhanced traceability throughout the supply and value chain.

Reference Projects

- Industrial Data Space (IDS): Digital Sovereignty over Data (BMBF funded project and initiative with more than 30 industrial partners).

- bIoTope – Building an IoT Open Innovation Ecosystem for Connected Smart Objects (research project funded by the European Commission).

- Individual RTD projects with different industrial partners in the automation, manufacturing, and consumer electronics domain.

Useful Links

- VoCol Demo: http://butterbur06.iai.uni-bonn.de

- GitHub Repository: https://github.com/vocol/vocol

- Installation and Configuration (screencast): https://drive.google.com/file/d/0By1pR7FDcH8obUV6OEpxeDZ3MFk/view

- Using VoCol (screencast): https://drive.google.com/file/d/0By1pR7FDcH8oTFpIRWxfUHdNS1k/view?usp=sharing

Publications

- Grangel-González, I., Halilaj, L., Coskun, G. Auer, S.: Towards Vocabulary Development by Convention. 7th International Conference on Knowledge Engineering and Ontology Development, pp. 334-343. SciTePress (2015)

- Halilaj, L., Grangel-González, I., Coskun, G. Auer, S.: Git4Voc: Git-based Versioning for Collaborative Vocabulary Development. 10th IEEE International Conference on Semantic Computing, pp. 285-292. IEEE (2016)

- Halilaj, L., Grangel-González, I., Coskun, G., Lohmann, S., Auer, S.: Git4Voc: Collaborative Vocabulary Development based on Git. International Journal on Semantic Computing 10(2):167-191 (2016)

- Halilaj, L., Grangel-González, I., Vidal, M., Lohmann, S., Auer, S.: Proactive Prevention of False-Positive Conflicts in Distributed Ontology Development.; Accepted at 8th International Conference on Knowledge Engineering and Ontology Development (2016)

- Halilaj, L., Petersen, N., Grangel-González, I., Lange, C., Auer, S., Coskun, G., Lohmann, S.: An Integrated Environment to Support Version-Controlled Vocabulary Development. Submitted to 20th International Conference on Knowledge Engineering and Knowledge Management (2016)

References

[1] Ren, Y., Parvizi, A., Mellish, C., Pan, J.Z., van Deemter, K., Stevens, R.: Towards competency question-driven ontology authoring. *11th International Conference Extended Semantic Web Conference (ESWC '14), LNCS 8465, pp. 752-767. Springer (2014)*

[2] Grüninger, M., Fox, M.S.: Methodology for the design and evaluation of ontologies. *IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing. (1995)*

[3] Giuliano, C., Gliozzo, A.M.: Instance-based ontology population exploiting named-entity substitution. *22nd International Conference on Computational Linguistics (COLING '08)*, *pp. 265–272. ACL (2008)*

[4] Kaur, A., Mann, K.S.: Component based software engineering. *International Journal of Computer Applications 2(1):105–108 (2010).*

[5] Otto, B., Auer, S., Cirullies, J., Jürjens, J., Menz, N., Schon, J, Wenzel, S.: Industrial Data Space: Digital Sovereignity Over Data. *Technical Report Fraunhofer-Gesellschaft, February 2016, DOI: 10.13140/RG.2.1.2673.0649*