Linked Data Interface, Semantics and a T-Box Triple
Store for Microsoft SharePoint

Christian Fillies and Frauke Weichhardt

1 Semtation GmbH, Geschw.-Scholl-Str. 38,
14771 Potsdam, Germany
{cfillies, fweichhardt}@semtalk.com

Abstract. In the last years collaboration platforms became an integral part of IT infrastructure in enterprises. Beyond document management for unstructured data, the main focus is now on collaborative work with structured data, which is an important part of the vision of the Semantic Web. Many local database solutions and a lot of spread sheets are being migrated to the new distributed information infrastructure, where end users can easily relate data from virtually any source. Data can be located in corporate and/or cloud-based systems. Schema information and Meta data are available but (as always) not very popular. There is a strong demand for more integrity and a huge potential in adding more semantics to structured data. Merging schema information and data with ontologies allow applying reasoning techniques to this kind of data. The paper presents first experiences in using the collaboration platform Microsoft SharePoint together with semantic technologies for analysis, matching and classification. OWL mappings for content types and list elements enable an exchange of any kind of SharePoint data with semantic tools. Based on the Open Source application SemWeb we gave access to applications to ask SPARQL queries against data located in SharePoint. It was straight forward to setup Linked Data endpoints retrieving their data from SharePoint. Our main intent is to give SharePoint users access to semantic technology and vice versa let semantic tools access the large amount of structured data existing in corporate platforms. Glossaries maintained in lists and wikis are usually the starting point for terminological work. It was found to be quite useful for end users to edit ontologies directly in SharePoint. A simple triple store for classes, instances, and data and object properties was implemented, which can be mapped bidirectional to OWL and more sophisticated ontology editors. Visio connected to SharePoint is used as a graphical editor for data, Meta data and ontologies. Finally we present a practical use case where we applied this very pragmatic approach to a library of services in the E-Government context.

Keywords: SharePoint, Linked Data, OWL

1 Overview

This article covers many aspects of working with SharePoint [1] and ontologies. It starts with a discussion of those areas of SharePoint which are relevant for the
representation of structured knowledge in SharePoint lists. The next chapter describes why and how a triple-store for ontologies is used if the ontologies are supposed to be maintained in SharePoint. We shortly introduce our object engine and the Visio based OWL editor. This engine is being used as a gluing component between SharePoint and SemWeb [2], a Dot Net based Open Source framework which we are using for OWL and SPARQL. We will introduce SemTalk Services as caching application server which is needed in all scenarios related to SharePoint if fast answers are needed. SiteBuilder is a graphical tool which helps to setup SharePoint list for a specific ontology. Finally we give a summary how the proposed scenario is being used to streamline service descriptions for the implementation of the important EU service directive.

2 SharePoint and Structured Data

SharePoint is a popular and well documented product, so that we do not have to describe it here in detail. For our use case the most important aspect is, that it can be used by technically unskilled users as a weakly typed object oriented database. Users can create sites which contain document libraries for unstructured data and lists for structured data. Lists may have multiple columns which can be typed. Lookup columns get their values from other lists. Lists can retrieve their values from external sources such as ERP systems. In a typical enterprise scenario one will find thousands of lists such as issue lists, meeting records, customers and products, SOX Controls and anything else that can be listed, structured and collaboratively edited. The main idea of SharePoint is to avoid documents and to break information down into statements. In the SharePoint world these are list items.

There are a couple of additional functionalities such as views, permissions, versioning, alerts, tagging, forms and workflow available on items.

Schema information can be added using so called ContentTypes. ContentTypes are made of ColumnTypes and act like a blueprint for new entities made from such a ContentType. ColumnTypes are specified similar to individual columns, which has the implication that Lookup ColumnTypes are defined on concrete lists.

There is the option of inheritance which allows specializing ContentTypes. Multiple inheritance is not supported so far. Changes to ContentTypes can be propagated to their elements. A list may contain elements of different ContentTypes, but each element can have only one ContentType.

ContentTypes improve the reusability and consistency of data in the SharePoint landscape, but they are not intended to be used as ontology. They are comparable to the schema of an object-oriented database. Nevertheless lists with elements, Lookups, ContentTypes etc can be mapped straight forward to Classes, DataProperties, ObjectProperties and Instances in OWL. ContentTypes are the Classes; entities will become Instances and statements.

SharePoint is completely programmable via web services, but these web services are batch oriented (Language named CAML [3]) and very complex. The latest version of SharePoint comes with LINQ and REST Interfaces. There is no native support for Semantic web protocols such as RDF or OWL.
The Windows Live storage system is based on SharePoint. Windows Live moves a huge amount of end user data to the Cloud. This data will be in the first wave mainly unstructured data such as pictures or videos with some Meta data. The deep integration with mobile devices and their location based services is expected to increase the amount of structured data in SharePoint systems dramatically. It also shows that Linked (Open) Data from various sources should be available e.g. as lookup values in SharePoint lists.

3 Triple Stores for Ontologies

In order to analyze, match, classify or map elements stored in SharePoint lists one needs ontologies. ContentTypes alone cannot be regarded as a real ontology. They are not sufficient for reasoning because of the weak representation of ObjectProperties and the missing option of restrictions or rules. The other reason is that SharePoint is not designed for managing a large number of ContentTypes which are frequently changing.

For reasoning we merged and mapped the ContentTypes with external ontologies initially created with Protégé and the OWL version of our tool SemTalk. Even if the owl files created by both of the tools can be stored in SharePoint, using external tools which are designed for knowledge engineers rather than end users was not a successful approach. Learning about OWL and Description Logics seems to be too much effort in a commercial scenario. Users could only be motivated to maintain their vocabularies with structures lists. Using external editors – including ours - still is possible, but should not be required.

Creating and managing ontologies in a SharePoint list environment [Fig.1] is not much more comfortable than using a spread sheet, but it can be done by anyone without being aware that all his entries will be mapped to OWL. We choose a T-Box implementation with:

- a list of classes with a multi value lookup for super classes, DataProperties, verbs (“methods”) and a definition column
- a list of triples for Object Properties
- a list of triples for Data Properties
- lists for the definition of Object Properties and Data Properties

This looks pretty simple and lacks any comfort while editing, but it was strongly preferred by our target audience compared to tools like Protégé or our own graphical OWL notation. We still have the option to merge the resulting ontologies with additional knowledge such as restrictions entered with those tools. The following screenshot shows an example of an ontology editing site in SharePoint.
The main advantage of using lists as a class editor is, that a user may add additional columns such as a status or domain and apply views and filtering to those class libraries. There are many built-in features of SharePoint which make the collaborative building of ontologies easier:

- Versioning
- Discussion and Issues
- Document workspaces
- Attachment of documents

The same approach could be taken obviously for A-Box elements, but in our case facts are represented usually as column values of list elements in separate lists. RDF statements are made by the selection of other objects in lookup columns or by entering literal values in ordinary columns. The main idea is to map any SharePoint list to RDF rather than forcing people to enter triples in a triple list.

Default-Classes of list elements are their ContentTypes, even if we had scenarios, where a “ClassOf” column as a LookUp column into an ontology library made sense. This is always the case, if you have more classes than ContentTypes and you are aware of those classes as a SharePoint user. A typical case is a list of IT-Systems with a type column but not with a SharePoint ContentType for each IT-System type (class). In case we have both: ContentTypes and a column “ClassOf”, the classes for the items should be subclasses of the ContentType. A “ClassOf” column is supposed to be a Lookup column with an ontology list as its range.
4 SemTalk Engine as a Mediator

Instead of directly connecting SharePoint data to OWL we have chosen to build a bidirectional interface to our existing Dot Net based SemTalk engine. It is an in memory framework, which can represent classes, objects, attributes and relations dynamically and offers some reasoning functionality.

We already have a graphical editor based on Visio for knowledge structures in OWL and for Business Process Modeling. It offers the option of graphical visualization and editing of semantic nets and other graphs in Visio. This following screenshot shows a graphical representation of an ontology in our tool SemTalk [Fig.2].

![Fig. 2. Graphical OWL Editor](image)

For more difficult modeling situations e.g. for complex restrictions we recommend to do this graphically in Visio.

SemTalk Classes can be mapped to SharePoint lists bidirectional. If one has e.g. a list of wines in SharePoint with the columns name, year, vintage and grape (last ones might be LookUp columns of other lists), she may connect it to a class “Wine” in SemTalk. The SemTalk SharePoint interface makes sure, that all instances and their properties are taken from the SharePoint lists. New objects will be added to the SharePoint lists etc. Basically SharePoint is used a repository for facts or instances.

In our “home” scenario of business process modeling, typical candidates for such kind of lists are roles, business-objects and IT-systems.

Classes might also be connected to a triple stores implemented in SharePoint as mentioned above. In this case T-Box knowledge such as class definitions, subclassing or properties are gathered from or written to a specific SharePoint site, which follows the specific Triple-Store structure for ontologies.
5 SemWeb

Joshua Tauber’s Open Source SemWeb implementation was chosen as a RDF parser, generator and SPARQL endpoint. Once each SharePoint Site with or without arbitrary content types or SemTalk specific ContentTypes can be imported to SemTalk, it can be exported to SemWeb. Internally it is using the SemWeb API, which does the main job as a Dot Net based RDF parser/generator and easily helps us to setup SPARQL endpoints working on the SharePoint data. Users and applications can run queries in SPARQL syntax against data residing in SharePoint lists. List items are returned as SPARQL result sets.

Having the option to publish an SPARQL endpoint for any SharePoint list unleashes a huge amount of structured data to Semantic Web (tools). Even if much of this data is not intended to be public, it might enable the use of semantic tools in a business context. We are using this interface just for semantic matching of process models / flow charts, but there might be many more use cases.

In the opposite direction SemWeb is used to access 3rd party OWL and SPARQL sources. The gathered data is mapped to SemTalk classes and instances and finally exported to SharePoint lists.

6 SemTalk Services

In order to run queries against SharePoint data in the scenario described above the data would be fetched via web service and afterwards converted to RDF for each call of a query. Since the transport is not performing really well, we implemented a web application which serves as a cache. The advantages of this architecture are: No installation of code on the SharePoint server and the option to implement some components as web services on the server [Fig. 3]. These components are OWL conversion, search, merging, computation of navigational structure e.g. class hierarchies.
The SemTalk Service Server is implemented as a Windows Communication Foundation server with a HTTP web interface including some REST parameters. In our solution the results of the web services are being rendered using a set of plain ASPX pages or using SharePoint web parts which are more flexible to configure. A screen shot of some of these pages including Tag Cloud style navigation is shown later at the E-Government use case.

7 Creating SharePoint ContentTypes and Lists from Ontologies

SharePoint lists are not always given. In many enterprise solutions data is found in spread sheets often exported from ERP systems. In some cases it comes as an OWL xml file or it can be retrieved from an SPARQL endpoint. Once it was decided to use or maintain this data in the future via SharePoint lists new list structures have to be created in SharePoint.

The trivial implementation of RDF in SharePoint would be a (A-Box) triple store. But this is not convenient for end users. They need lists for products, customers and other concrete entities with columns where they can pick related entities. Given an ontology with classes and instances the task is to setup a set of lists with Data Columns, Lookup Columns and Content Types. If the structure of the classes in the ontology is not purely relational, it is often required to setup also a triple store for T-Box data.
We have build a graphical tool named SemTalk SiteBuilder which supports users in transforming given class structures or ontologies to SharePoint sites [Fig.4]. It allows marking some classes as ContentTypes, specifying lists for their instances and then specifying the domains and ranges for DataProperties and ObjectProperties as SharePoint Columns. Finally the site is generated in SharePoint and it can be linked to a concrete ontology. The opposite direction is also possible. In this case the tool can be used to redocument existing SharePoint sites. The approach of generating sites makes it easy to create even complex list structures for an ontology so that end users can edit instance data in a distributed fashion using intranet or internet and can make use of all collaboration functionality.

**8 Use Case EU-DLR**

SharePoint as a shared repository for modeling tasks related to Visio using the architecture described above has been implemented in various cases in the extremely fast growing market of SharePoint in the last couple of month.

We want to focus here on one specific solution for the EU service directive, since it really has a strong focus on semantics. The EU service directive says that everyone should be able to start a new business anywhere in the EU via internet using any EU language and there must be a single point of contact to clear out any arising problems. This was (and still is) a serious challenge for state wide, regional and local authorities. There are hundreds of different kinds of businesses a user might try to apply for. Each of them requires specific documents to be uploaded and approved. There are lots of scenario specific forms to be filled out. Many legacy IT-Systems are involved to actually fulfill the requested demand, which have never been integrated by web
service before. To make it more complicated there are various synonyms a user might use to search for a certain kind of business in the database.

Our partners Knowlogy Solutions AG and CIT GmbH came up with domain specific ontologies maintained in SharePoint lists. They have setup the appropriate structures using SemTalk Sitebuilder for the problem and most importantly, they brought shared teams of several German federal states to work jointly on elaborating a common knowledge base of services. CIT offers a JBOSS based engine to gather data in forms and execute service applications. The library of processes itself is based on ontologies and was build on a SharePoint infrastructure [Fig. 5].

![Fig. 5. EU-DLR Library](image)

The main issue for semantic technology in this case is the need to manage thousands of synonyms for kinds of businesses, activities and documents. Use of ontologies supports the disambiguation of terms and the consolidation of service descriptions.

The following screenshot [Fig. 6] gives an impression how to mash up the (SemTalk.) services with a graphical specification of the dependencies between processes made with Visio. All Visio elements in the drawing are connected to corresponding elements in SharePoint Libraries. On the left side is a tag-cloud style navigation, a hierarchy of process chain and a property detail service frame. On the right side we have inheritance hierarchy, dynamic execution of queries and an ontology based search component. All of these frames are connected via REST interfaces and the content is dynamically computed by SemTalk Services on the elements in the repository.
In a current research project we are working on ontology based automatic matching of process models. The outcome of this project will help to compare and find process models in large repositories.

9 Related Work

Other approaches we are aware of trying to combine SharePoint and ontologies are IntelliDimension’s RDF Gateway [5] and Ontoprise’s SemanticMiner for SharePoint [6]. RDF Gateway is definitely a great professional alternative in the Microsoft sphere to SemWeb, but it does not seem to work natively on SharePoint (anymore). Semantic Miner has its focus on semantically searching existing SharePoint data. Essentially this is realized by adding an ontology and Meta data for individual SharePoint items using OntoStudio.
Both tools might be combined with our approach of using SharePoint directly as a repository for ontologies.

10 Summary

We have discussed how structured data in the collaboration platform SharePoint can be extended by semantic technology and how it can be used as a repository for semantic web data. The driving idea for the using SharePoint is that it is already established in enterprises, SharePoint Foundation is perceived to be free of charge,
and no new infrastructure needs to be setup. There are no Open Source alternatives in sight and the SharePoint UI is significantly easy to use that anybody can contribute data and ontological knowledge. Combining SharePoint and SemWeb can make all the content accessible through Linked Data from Semantic Web and in the other direction it makes Linked (Open) Data accessible in SharePoint.

References