Janus: Automatic Ontology Construction Tool

Ivan Bedini 1, Benjamin Nguyen 2, Georges Gardarin 2

1 Orange Labs,
42 Rue des Coutures,
14000 Caen, France
ivan.bedini@orange-ftgroup.com

2 PRiSM Laboratory, University of Versailles,
45 Avenue des Etats-Unis,
78035 Versailles, France
{benjamin.nguyen, georges.gardarin}@prism.uvsq.fr

Abstract. The construction of an ontology for a large domain still remains a hard human task. The process is sometimes assisted by software tools that facilitate some parts of the construction and evolution life-cycle. But they do not propose a methodology that considers the automation of the entire process. In this paper we present a method for deriving automatically an ontology. Then, we introduce Janus, an implementation of this approach, for deriving automatically a skeleton of an ontology from XML schema files in a given domain. Janus also provides different useful views that can be used for a final revision by an expert.

Keywords. Ontology Creation, XML Mining, Application Integration.

1 Introduction

Over the past ten years, the Semantic Web wave has shown a new vision of ontology use for application integration systems. Researchers have produced several software tools for building ontologies (like Protégé [1] or OntoEdit [2]) and merging them two by two (like FCA Merge [3] or Prompt [4]) or producing alignments (like OLA [5], Mafra [6], S-Match [7], H-MATCH [8]).

As shown in [9] different reasons limit their adoption to the integration of internet and enterprise applications: (i) the lack of tools capable of extracting and acquiring information from a collection of XML files (the “de-facto” format for applications information exchange definition); (ii) the complexity of aligning and merging more than two knowledge sources at a time, which also is a task excessively consuming of computational time; (iii) existing ontology building methodologies, are human centric and are able to assist engineers just automating one or few parts of the entire process.

In this paper, we propose Janus, a tool for semi-automatic derivation of ontologies from XML schemas. It implements a new approach and a new methodology to
ontogeny generation that can provide a solution to the limitations described above.

The aim of this short paper is to introduce Janus, how it works and to show some different views (produced by the tool) of knowledge automatically acquired.

2 Approach and Methodology

In this section is provided a general view of the automation aspect of the ontology generation implemented by Janus.

Several methodologies for building ontologies exist, such as OTK [10], METHONTOLOGY [11] or DILIGENT [12], but they target ontology engineers and not machines. In order to automate completely the ontology construction we define the automatic ontology generation life cycle as a cyclic process composed of five main essential steps to achieve the goal (see Figure 1). The main difference with a human centric methodology is that the "glue" between steps, as well as the integration of the different modules, needed to achieve a step, must be in a machine readable format. Also this approach is more dynamic and permits to acquire new inputs in order to maintain a reusable semantic memory and thus easily update the ontology constantly.

Furthermore, information about the confidence of the learned instances can be displayed with different views and used for a final revision by an expert. More details about the process are shown in [9].

3 XML as Input Source

XML schemas and ontologies in a given domains are somehow related. In general, schemas are built in a domain before ontologies. Consider for example the B2B domain: there exist hundreds of schemas to encode exchanged data but not many ontologies. To benefit from preexisting schemas, we propose a method and a tool to derive an ontology or at least an enriched taxonomy (i.e., a concept hierarchy with concepts properties and main relationships and properties) from a set of XML schemas. Janus currently implements a module for retrieving ontological information from this format, however its architecture is extensible and permits to add new modules retrieving information from other sources, like text documents or the Web.
4 Janus Architecture

Our tool implements an adaptation of several techniques originating from text mining and information retrieval / extraction fields, applied to XML files (that we call XML Mining). XML Mining is used to pre-process simple and compound statements defining XML tags, such as XSD elements and XSD complex types. It includes clustering methods based on a Galois Lattice and Formal Concept Analysis to quickly discovery similarities between names and structures extracted from the source corpora. Figure 2 shows the overall architecture of Janus.

The algorithm generating a high level representation of XML Schema information sources is composed of three main steps.

The first step is the **Extraction** task represented by the Extract arrow and Acquisition rectangle in Figure 2. It provides the knowledge needed to generate the ontology (concepts, properties and relationships). Implemented techniques for knowledge acquisition are a combination of different types, such as: NLP (Natural Language Process) for morphological and lexical analysis, association mining for calculating term frequencies and association rules, semantics for finding synonymy, and clustering for grouping semantic and structural similar concepts.

The second step is **Analysis** represented by the correspondent block in Figure 2. This step focuses on the matching of retrieved information and/or alignment of two or more existing families of concepts issued from different sources.

This step requires techniques already used in the first stage, as syntax and semantic measures, to establish the best similarities; it also requires an analysis of concept structures to determine hierarchical relationships and identify common properties. The output of this task provides enough information for building a semantic network of concepts that will be used in following step to look for similar concepts (it constitutes the basis of the semantic similarity memory of the system).

The last step is **Generation**, represented by the Merging, Generation and Filtering blocks in Figure 2. This step looks for concepts with evident affinites (e.g., concept fully included into another) based on specific rules, to merge or just link them. It generates a final semantic network that can be described in RDFS or OWL, built by the **Transform** module. The tool can derive from the network useful views provided to users by the **Build Views** module. Users can also step into the process to parameterize thresholds for refining results.

![Figure 2 - Janus overall architecture](image-url)
5 Functionalities and Views

The tool currently offers four visualization methods to view the acquired knowledge and a module able to generate a first ontology in OWL format.

The **word view** shows the list of terms composing the "ontology" as tag cloud format. The **detail view** shows all discovered relationships for a specific concept with other concept of the ontology. Between them we can find its properties shared in two main groups, "most common properties" and "other properties". This distinction permits to consider those properties characterizing the concept and the other that we can occasionally find for a concept. The **list view** gives detailed information about each concept like frequencies, family attendance and type (class, data-type or property). The **graph view** displays the semantic network of concepts (see Figure 3). The graph view can show the whole graph or only the part related to selected concepts with different layouts (hierarchical, tree, …).

Furthermore it is possible to select the kind of relationships to display. In fact acquired relationships are of different types: **propertyOf**, **synonym**, **shared terms** (compound tags with common terms like `address` for `tender_address` and `post_box_address`) and **relatedTo** (mainly merged concepts or other of type `owl:sameAs` and `owl:equivalentClass`). This feature permits to analyse in details some parts of the ontology; it is useful when the ontology is too large to be browsed with the global view.

Other views, one called “Concepts Social Network”, and another to identify groups of common occurrences of properties, are under development.

Finally the generated ontology can be exported in OWL format. This is an important feature because permits to transform the Janus generated meta-model in a more generic format that can be used by other tool like Protégé [1].

The tool also offers the possibility to parameterize thresholds for alignment and merging operations.

6 Conclusion

The automatic construction of an ontology is a complex task that requires: i) a specific methodology capable to be executed autonomously by a machine; ii) an extensible semantic memory capable to easily discover concepts similarities and; iii) to be able to extract information from several XML Schema files (at least for our use case). We propose to demonstrate our preliminary results of Janus, a tool that we have developed to provide a first significant return of experience of a complete automation
of the ontology construction. We will show our tool applied to the analysis of several B2B standards, XML based as input source. Differences between the presentation already done in [13, 14] and this one, are that our demo will focus on the automation methodology and, seeing that it is an ongoing work, we show results about new developed algorithms capable to better integrate the structure of input sources.

7 References