

# Bottom-Up Ontology Construction with Contento

Enrico Daga<sup>1</sup>, Mathieu d'Aquin<sup>1</sup>, Enrico Motta<sup>1</sup>, and Aldo Gangemi<sup>2</sup>

<sup>1</sup> Knowledge Media Institute, The Open University  
Walton Hall, Milton Keynes, United Kingdom  
{enrico.daga,mathieu.daquin,enrico.motta}@open.ac.uk  
<http://kmi.open.ac.uk>

<sup>2</sup> Université Paris13, Sorbonne Cité  
CNRS UMR7030, France, and  
Istitute of Cognitive Sciences and Technologies - CNR  
Via S. Martino della Battaglia 44, 00185 Rome (RM), Italy  
aldo.gangemi@{univ-paris13.fr,cnr.it}  
<http://www.univ-paris13.fr>  
<http://istc.cnr.it>

**Abstract.** In this demo paper we show an approach to build Semantic Web ontologies from sample linked data with a tool named Contento. Contento is a data driven ontology construction kit, based on Formal Concept Analysis (FCA). We show the exploration and analysis functionalities of Contento, as well as the method to generate, annotate and prune concept hierarchies. Moreover, we describe a procedure to go from sample data - extracted from SPARQL endpoints - to a new OWL ontology.

**Keywords:** Linked Data, Formal Concept Analysis, Ontology Design

## 1 Introduction

In this demo paper we show an approach to build Semantic Web ontologies from datasets in the Linked Data with a tool named Contento<sup>3</sup>. Contento is a data driven ontology construction kit, based on Formal Concept Analysis (FCA). With Contento, users can build the input data (objects and attributes) from scratch or import them from existing datasets, for example querying Linked Data endpoints. We show how to go from sample data to a complete OWL ontology in four steps: (1) extract data from one (or more) SPARQL endpoints; (2) generate a FCA lattice; (3) annotate and prune the conceptual lattice; (4) generate the OWL ontology.

---

<sup>3</sup> <http://bit.ly/contento-tool>

## 2 The Contento Way

Contento<sup>4</sup> has been developed to create, populate and curate FCA formal contexts and associated lattices, also interpreted as taxonomies of concepts.

### Step 1. Extract data from SPARQL endpoints.

Formal contexts can be created and populated from scratch. With this interface, the binary matrix can be populated or supervised to constitute a proper input for a FCA algorithm. In many cases, however, a ready made binary matrix can be imported from pre-existing data, for example as output of a query to a Linked Data SPARQL endpoint like `http://data.open.ac.uk/sparql`:

```
SELECT distinct ?object ?attribute ("1" as ?holds)
FROM <http://data.open.ac.uk/context/xcri>
WHERE {
?object a <http://data.open.ac.uk/ontology/Qualification> ;
    <http://purl.org/dc/elements/1.1/subject> ?attribute
}
```

The above query will report about the topics of qualifications under presentation at The Open University. The output of this query, requested as CSV file, can be used to feed Contento using the context import procedure. Figure 1 shows the facilities offered by the context browser. In this case the formal context is

Relation	Objects	Attributes	Holds
<input type="checkbox"/>	#3548219	#14532	<input checked="" type="checkbox"/>
<input type="checkbox"/>	#3548220	#14499	<input checked="" type="checkbox"/>
<input type="checkbox"/>	#3549818	#14482	<input type="checkbox"/>
<input type="checkbox"/>	#3549819	#14483	<input checked="" type="checkbox"/>

**Fig. 1.** Contento: formal context browser and editor. The tool supports various methods for filtering the data. In this figure, we have fixed the object in order to review its relations with the attributes. Similarly, the user can inspect subset of the data and perform bulk actions to change boolean status (the Holds value in the table).

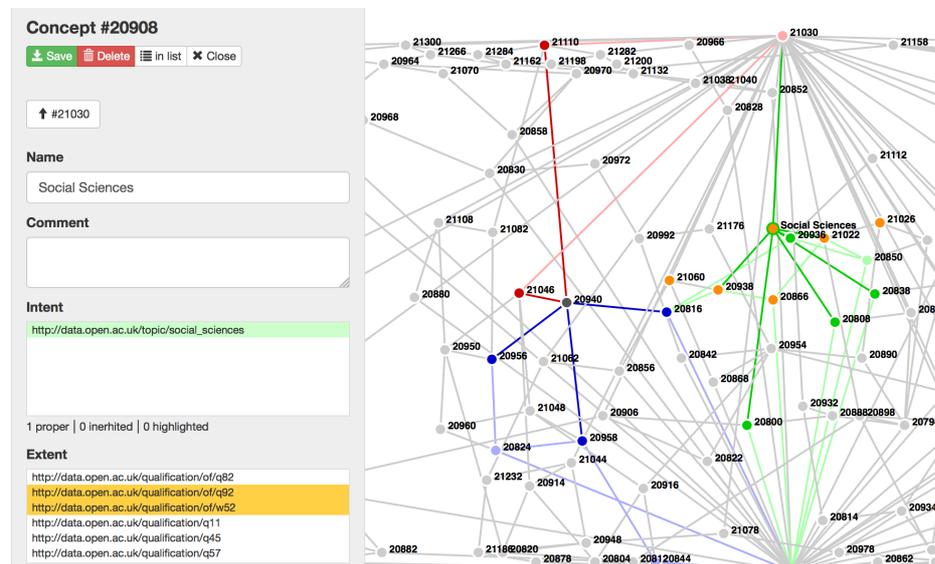
created directly from that, ready to be used to generate the concept lattice with the procedure provided.

<sup>4</sup> <http://bit.ly/contento-tool>

## Step 2. Generate a FCA lattice.

Contento implements the Chein algorithm [1] to compute concept lattices. The result of the algorithm is stored as a *taxonomy*. A taxonomy can be navigated as an ordered list of concepts, from the top to the bottom, each of them including the *extent*, the *intent* and links to upper and lower concept bounds in the hierarchy. In addition, the tool shows which objects and attributes are *proper* to the concept, i.e. do not exist in any of the upper (for attributes) or lower (for objects) concepts.

Moreover, it can be visualized and explored as a concept lattice (Figure 2). The lattice can be navigated by clicking the nodes. Focusing on a single node, the respective upper and lower branches are highlighted, to facilitate the navigation to the user. Similarly, objects and attributes from the focused node can be selected, thus highlighting all nodes in the hierarchy sharing all of the selected features (in orange in Figure 2).



**Fig. 2.** Contento: the lattice explorer for annotation and pruning. The branching of the current concept is presented in the lattice in green (on the left side of the picture). The user can still point to other nodes to inspect the branching of other concepts (on the right side of the picture, the lower branch being displayed in blue and the upper in red). By selecting one or more items in the extent or intent of the concept, all the nodes sharing the same are bolded in orange.

## Step 3. Annotate and prune the conceptual lattice

Contento supports the user on the curation of the concept hierarchy, supporting annotation of concepts with label and comment, and the pruning of unwanted concepts. This last operation implies an adjustment of the hierarchy, by building

links between lower and upper bounds of the deleted node (only if no other path to the counterpart exists). As a result, relevant concepts can be qualified, and concepts that are not relevant for the task at end can be removed.

#### Step 4. Generate the OWL ontology

The data of the FCA lattice can then be translated into OWL using predefined or custom profiles. The user can decide how to represent the taxonomy in RDF, what terms to use to link concepts, objects and attributes, and whether items need to be represented as URIs or literals. For example, *Contento* offers a default profile, using example terms, or a SKOS profile. Ultimately, these export configurations can be shared and reused.

### 3 Related work

Bottom-up approaches for ontology design have been commonly applied in knowledge engineering [2] and we use here one particular method based on Formal Concept Analysis (FCA) [3]. FCA has been proposed in the past to support ontology design and other ontology engineering tasks [4, 5]. Recently, we used *Contento* to support the design of the License Picker Ontology (LiPiO) [6]. In this demo we show how to use FCA as a learning technique to boost the early stage of the ontology design.

### References

1. Michel Chein. Algorithme de recherche des sous-matrices premieres d'une matrice. *Bull. Math. Soc. Sci. Math. RS Roumanie*, 13(61):21–25, 1969.
2. Paul E Van Der Vet and Nicolaas JI Mars. Bottom-up construction of ontologies. *Knowledge and Data Engineering, IEEE Transactions on*, 10(4):513–526, 1998.
3. Bernhard Ganter, Gerd Stumme, and Rudolf Wille, editors. *Formal Concept Analysis, Foundations and Applications*, volume 3626 of *Lecture Notes in Computer Science*. Springer, 2005.
4. Philipp Cimiano, Andreas Hotho, Gerd Stumme, and Julien Tane. Conceptual knowledge processing with formal concept analysis and ontologies. In *Concept Lattices*, pages 189–207. Springer, 2004.
5. Marek Obitko, Vaclav Snasel, Jan Smid, and V Snasel. Ontology design with formal concept analysis. In *CLA*, volume 110, 2004.
6. Enrico Daga, Mathieu d’Aquin, Aldo Gangemi, and Enrico Motta. A bottom-up approach for licences classification and selection. In Serena Villata and Silvio Peroni, editors, *Proc. of the International Workshop on Legal Domain And Semantic Web Applications (LeDA-SWAn) held during the 12th Extended Semantic Web Conference (ESWC 2015)*, pages 33–40. ACM, 2015.