A Conceptual-KDD tool for ontology construction from a database schema

Renzo Stanley and Hernán Astudillo

Universidad Técnica Federico Santa María, Av. España 1680, Valparaíso, Chile
{rstanley,hernan}@inf.utfsm.cl

Abstract. The UNESCO convention on Intangible Cultural Heritage (ICH) requires countries to document their oral traditions, performing arts, traditional festivities, and so forth. Several institutions gather ICH, traditionally by hand, and record and disseminate it through conventional information systems (static knowledge in relational databases, RDB). Two difficulties are that (1) review/refinement of their underlying database schemata by domain experts becomes disruptive, and (2) contribution from community, non-expert users becomes hard, even impossible. This article presents an interactive tool that implements a recent technique to perform Knowledge Discovery in Databases (KDD) guided by Formal Concept Analysis (FCA). The tool takes an RDB schema (in SQL), translates it into a formal context and later in a concept lattice using the CORON platform, allows domain experts to manipulate it and produces a formal ontology (in RDFS). Later, the ontology can be used to instantiate a semantic wiki as community collaboration tool, for example. The technique and tool are illustrated with an example from the ICH domain, using Chile's Culture Ministry online data. The tool is also available online.

Keywords: Formal Concept Analysis, Knowledge Discovery in Databases, Ontologies, Intangible Cultural Heritage

1 Introduction

The Chilean National Council of Culture and Arts[1] (CNCA) has undergone the mission of documenting the ICH of different areas of the country in the context of a world-wide UNESCO[2] convention to incentive the *states parties*[3] and NGOs to properly maintain their cultural knowledge. Considering the dynamic structure (data, concepts and relations) of this domain, the conventional information management systems should be sufficiently flexible in order to support changes and community collaboration such as well-known wikis [5]. For these reasons, CNCA needs a tool that allows to simplify the process of refinement of their current relational database model. KDD emerged as a tool to support humans

[1] http://cultura.gob.cl
[2] United Nations Educational, Scientific, and Cultural Organization
[3] http://whc.unesco.org/en/statesparties/

in the discovery and extraction of knowledge from large collections of data (usually stored in databases) where a manual approach for such task is very difficult (or nearly impossible) [3]. Thus, the *human-centered* nature of the approach is a key factor in any KDD process [1] since it has to ensure that knowledge is not only successfully found, but also understood by the final user. For this reason, FCA proved to be a good support for a KDD process given its two-folded manner of representing knowledge, i.e. as concepts containing an extent (instances of the concept) and an intent (the attributes of the concept) [8]. To stress this fact, we quote [7] in the relation of FCA and KDD: "the process of concept formation in FCA is a KDD *par excellence*". FCA has been used to support KDD in several tasks for different domains. For example, [4] states that nearly 20% of the papers in the FCA domain consist on knowledge discovery related approaches. Furthermore, in [2] FCA is presented as the cornerstone of *Conceptual Knowledge Discovery in Databases (CKDD)* described as a human-centered process supporting the visual analysis of a conceptual structure of data for a given context of information. Since the principal difficulty of CNCA (reviewing and refinement of ICH model) are rooted in a database schema analysis and amelioration which heavily requires human domain expertise, we rely on a CKDD tool to redesign the data schema already in use and to elicit an ontological schema from it.

In this article, we show a tool that implements an iterative and human-centred approach based on KDD and FCA. This method uses the concept lattice generated as a support for guiding the redesign process, considering the relevant knowledge of experts. This approach was proposed in an earlier work [6], however applies it in a web-based tool that allows any user work with his own schema.

The reminder of this article is organized as follows: Section 2 resumes the method proposed, Section 3 describes in detail the principal functionalities of the tool developed, Section 4 outlines an example for validating the tool with a domain expert. Finally, Section 5 presents a discussion on future work and concludes the paper.

2 Method

Figure 1 presents a 3-step CKDD process designed to take a database schema and translating it into an ontological schema. In the following, we provide a general view of the tasks at each step.

2.1 First step: Data Preprocessing

The first step starts by extracting the database schema and ends when it is converted to a formal context. This step consists of three tasks: (1) Schema processing, (2) Attribute integration and (3) Relational attribute scaling. However, this process is fully automatized by the tool, and does not require expert intervention.

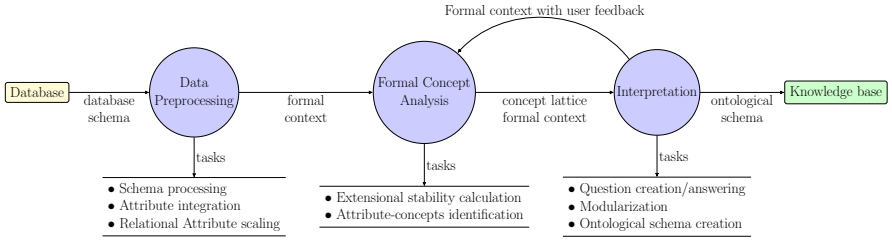

Fig. 1: FCA-based KDD process

2.2 Second step: Formal Concept Analysis

This step receives a formal context and ends when a concept lattice is constructed. The tasks performed are: (1) Extensional stability calculation and (2) Attribute-concepts identification and these are calculated using the Coron Platform. The extensional stability value and the attribute-concept calculated are shown to a domain expert for the next step.

2.3 Third step: Interpretation

The final step receives a formal context and its associated concept lattice where each attribute concept has been identified and each formal concept contains an *extensional stability* value. The tasks performed for this step are: (1) Question creation/answering, (2) Modularization, (3) Ontological schema creation. The options (1) and (2) allow the user to make another iteration sending a modified version of the formal concept received according to user feedback, but option (3) allows the user to end the process, an "ontological schema" will be created and it will be downloaded by the user in RDF file format.

2.4 Ontological schema creation

The final task of the process converts the concept lattice into an ontological schema which can be used for data integration and linked data publication. This schema is obtained by creating a set of RDF triples for the elements of the concept lattice. Table 1 shows a overview of the rules used to create the ontological schema. This table is based on an adapted definition of the relational data schema model.

Relational data schema model: A relational schema $S = \{R_1, R_2, ..., R_{|S|}\}$ is defined as a set of tables or "relation schemas" $R_i(A_1, A_2, ..., A_n)$ consisting of a table name R_i and a list of fields A_j which are value assignments of the domain $dom(A_j)$ to an *entry* in the table. The notation $R_i.A_j$ stands for the field A_j in table R_i.

Table 1: Formal concepts translation into an ontological schema [6].

Concept	Element	Actions
$\top = (S, S')$	$R_i \in S$	R_i `rdf:type rdfs:Class` *e.g. cnca:*Agent *rdf:type rdfs:Class*
$\bot = (A', A)$	$A_j \in A$	A_j `rdf:type rdfs:Property` A_j `rdfs:range rdfs:Literal` *e.g. cnca:establishment rdf:type rdfs:Property* *cnca:establishment rdfs:range rdfs:Literal*
$\bot = (A', A)$	`related_to:`$R_i \in A$	`related_to:`R_i `rdf:type rdfs:Property` `related_to:`R_i `rdf:range rdfs:`R_i *e.g. cnca:participant rdf:type rdfs:Property* *cnca:participant rdfs:range cnca:Agent*

$\bot = (A', A)$	`domain:Label` $\in A$	`cnca:Label rdf:type cnca:Domain`
		`cnca:Domain rdf:type rdfs:Class`
		`cnca:in_domain rdf:type rdfs:Property`
		e.g. cnca:People rdf:type cnca:Domain

$\mu A_j = (A'_j, A''_j)$	$R_i \in A'_j$	`cnca:`A_j `rdfs:domain cnca:`R_i *e.g. cnca:participant rdfs:domain cnca:Ritual*
$\mu A_j = (A'_j, A''_j)$	($A_j =$`domain:Label` $\wedge$ $R_i \in A'_j$)	`cnca:`R_i `cnca:in_domain cnca:Label` *e.g. cnca:Agent cnca:in_domain cnca:People*

This task is also interactive allowing the user to take most of the decisions w.r.t. how the ontological schema should be created. In the following, we refer to *cnca:* as the prefix used for the schema to be created.

Top Concept $\top = (S, S')$: All tables are modelled using the resource description framework schema (RDFS) element *rdfs:Class* by default (e.g. *cnca:Agent a rdfs:Class*). The user may choose to annotate some of them with the element *rdfs:Resource*. For the set of attributes in S', we provide a list of properties from RDFS and the *dublin core* ontology [4] where the user can select mappings going from the attributes to the ontology. For example, the attribute *name* is mapped to the property rdfs:label. The special attribute *id* is disregarded as its value in each entry is only considered to create a unique an valid URI [5].

Bottom Concept $\bot = (A', A)$**:** All fields in A are modelled according to their nature: *relational*, *non-relational attributes* or *special attributes*.

- *Regular attributes* are modelled by default using the *rdfs:Property* while the *cnca:* prefix is added to its name (e.g. *cnca:establishment a rdfs:Property*). In addition, the range of the property is set to rdfs:Literal (e.g. *cnca:establishment* rdfs:range rdfs:Literal).
- *Relational attributes* of the form *related_to:table* are modelled with *rdfs:Property* and the range is set to the table they refer to. Additionally, the user is asked to rename the relation (e.g *related_to:Agent* is modelled as *cnca:participant*

[4] `http://www.w3.org/wiki/Good_Ontologies#The_Dublin_Core_.28DC.29_ontology`

[5] Universal resource identifier.

a rdfs:Property; cnca:participant rdfs:range cnca:Agent). While the user may also be requested to create the inverse property, this feature is not available in RDFS and for the sake of simplicity we have disregard the use of OWL for now.
- *Special attributes* of the form *domain:Label* are modelled differently. For each different *domain:Label* we create a resource *cnca:Label a cnca:Domain* where *cnca:Domain a rdfs:Class* (e.g. *cnca:People a cnca:Domain*). A single property *cnca:in_domain a rdfs:Property; rdfs:range cnca:Domain; rdfs:domain rdfs:Class* is created to annotate classes created from tables.

Attribute concepts $\mu A_i = (A_i', A_i'')$: For each attribute concept, we use its extent to set the domain of the already modelled properties in its intent creating *cnca:*A_i *rdfs:domain cnca:R* for all $R \in A_i'$ (e.g. *cnca:participant rdfs:domain (cnca:Festive_Event,cnca:Ritual)*). For the special attributes of the form *domain:Label*, objects are annotated using *cnca:R cnca:in_domain cnca:Label* for all $R \in A_i'$ (e.g. *cnca:Agent cnca:in_domain cnca:People*).

There are some other actions taken during modelling, however for the sake of space and simplicity we do not discuss these in here.

3 Tool

The web-based tool intended to construct an ontological schema for a specific SQL relational database schema is compound of two principals components: (1) the CORON platform to calculate concept lattices and the stabilities values of each attribute-concept, and (2) the python backend application connecting user interface with CORON in order to execute functions that manage formal contexts, attribute-concept detections and ontology generation. Thus, the tool allows domain experts obtain an ontology in RDF file format.

3.1 Technology

This tool was developed on Python 2.7 and Flask micro-framework[6]. For developing the following technologies are used, namely: SQLAlchemy ORM[7] to connect Python to the DB schema, *python concepts*[8] to translate the DB schema to a formal context for the first time. Also we used the Coron Platform[9] to calculate the concept lattices and their extensional stabilities in order to identify the attribute concept in each iteration. RDFLib[10] was used for working with RDF files in Python. At this moment, the tool is available in `http://dev.toeska.cl/rstanley/rdb2ontology`. Once there, you can create a user account and connect it with your own MySQL DB schema.

[6] Flask `http://flask.pocoo.org/`
[7] Python Object Relational Mapper (ORM) `http://www.sqlalchemy.org/`
[8] Concepts: a python library for Formal Concept Analysis `https://pypi.python.org/pypi/concepts`
[9] Coron System: a symbolic data-mining platform `http://coron.loria.fr/site/index.php`
[10] RDFLib `https://github.com/RDFLib/rdflib`

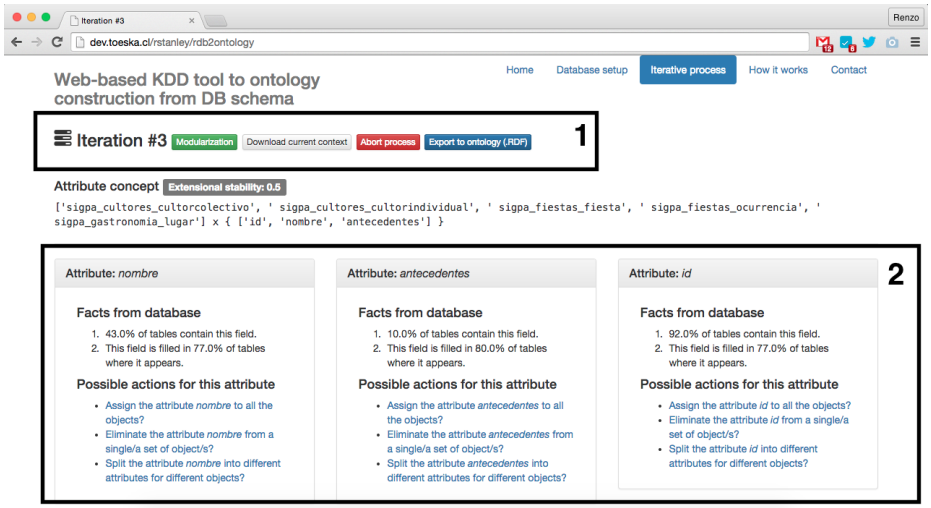

Fig. 2: Screen capture of an iteration

3.2 Functionalities

To provide a way to modify the underlying formal context for the domain expert we implemented some functionalities that can be looked at in figure 2. These actions are divided in two groups named *general options group* marked with #1 and *attribute-specific options* marked with #2. They are available for the domain expert in each iteration. Firstly, the *general options group* is composed by (1) Modularization, (2) Download current context, (3) Abort process, (4) Export to RDFS ontology. Secondly, the *attribute-specific options group* contains a set of actions to modify each attribute depending of a expert decision, namely: (1) Assign the attribute to all the objects?, (2) Eliminate the attribute from a single/a set of objects?, (3) Split the attribute into different attributes for different objects? For the sake of space and simplicity, we have left out the explanation of each of these options as it can be found in depth in our previous work [6].

4 Example

The database schema of CNCA[11] includes nearly 100 tables, however, for this example we have selected only 24 tables representing multi-disciplinary knowledge. These tables contain 24 objects, 53 attributes, and 13 relational attributes. The database schema for this example represents descriptions of *agents*, *collective agents*, *festive events*, *culinary manifestations*, *geolocations* and more. Figure 3 depicts the concept lattice obtained from the formal context generated by the

[11] Chilean National Council of Culture and Arts

database schema. Table 2 shows the decisions taken by the domain expert during 14 iterations. These decisions are based on question answering, domain labeling (modularization) or stoping the iterations.

Table 2: Iterations made by the domain expert

Iteration number	**Attribute**	**Action**
1	`name`	Assign to all tables
2	`background`	Split the attribute
3	`background`	Split the attribute
4	`views`	Eliminate from some tables
5	`published`	Eliminate from some tables
6	`description`	Assign to all tables
7	`founding_date`	Split the attribute
8	`related_to Agent`	Eliminate from *Ritual* table
9	-	Domain labelling: Culinary descriptors
10	`domain:culinary`	Eliminate from *CulinaryPlace* table
11	-	Domain labelling: ICH
12	-	Domain labelling: Agent descriptors
13	-	Domain labelling: Festive descriptors
14	-	Domain labelling Geo descriptors

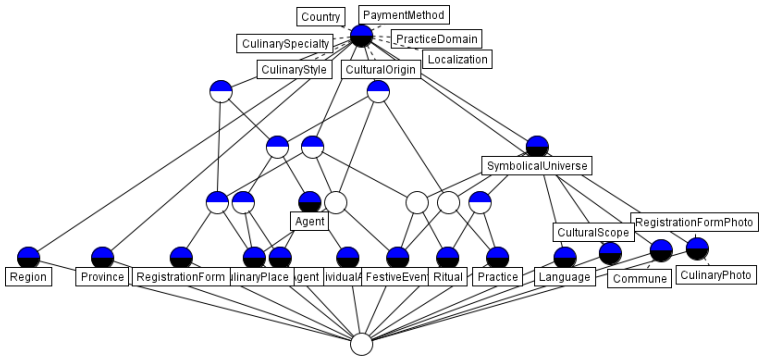

Fig. 3: Initial lattice obtained automatically from database schema

Figure 4 illustrates the final concept lattice presenting the refined structure after 14 iterations of the domain expert. We can distinguish several modules of information that have been marked. The expert called these modules as *ICH subdomains* identified from left to right, namely: *Festive Event descriptors sub-*

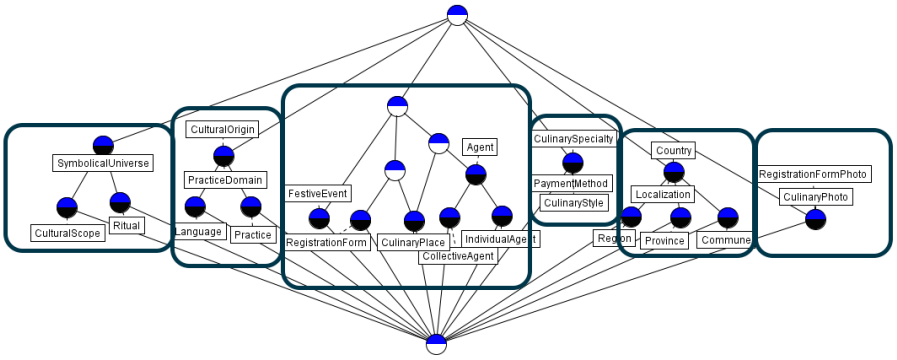

Fig. 4: Final lattice obtained after 14 iterations. Each ICH subdomain found have been marked.

domain, *Agent descriptors subdomain*, *ICH inventory subdomain*, *Culinary descriptors subdomain*, *Geographical subdomain*, *Photo subdomain*.

5 Conclusions and Future Work

To conclude, in this article we have presented a web-based tool fully functional based on an approach published in a previous work [6]. In this earlier work a case study was exposed obtaining interesting results, however these results were obtained executing calls to CORON platform in a manual way with the intervention of a knowledge engineer. The difference between the previous work and this work is that the tool allows a domain expert to get an ontological schema himself in RDFS. In the example showed in section 4 we obtained 14 iterations from a similar excerpt of a database schema, however in the previous case study executed in [6] we obtained 9 iterations, so the resulting concept lattices were very similar. In each lattice the same modules were found, however, the time to reach the same result was higher. We have to consider that the expert used the tool without the assistance of a knowledge engineer. Currently, we are implementing the next step of this tool related to construct a semantic wiki based on the ontological schema. So even though the ontology obtained was simple, the domain expert could enrich it by using annotations in a semantic wiki. Also, this wiki could aid a domain expert in order to collaborate in the documenting process.

References

1. Ronald J. Brachman and Tej Anand. The process of knowledge discovery in databases. In *Advances in Knowledge Discovery and Data Mining*, pages 37–57.

1996.
2. Joachim Hereth Correia, Gerd Stumme, Rudolf Wille, and Uta Wille. Conceptual knowledge discovery–a human-centered approach. *Applied Artificial Intelligence*, 17(3):281–302, March 2003.
3. Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery: an overview. In *Advances in knowledge discovery and data mining*, pages 1–34. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.
4. Jonas Poelmans, Paul Elzinga, Stijn Viaene, and Guido Dedene. Formal concept analysis in knowledge discovery: a survey. In *Proceedings of the 18th international conference on Conceptual structures: from information to intelligence*, ICCS'10, pages 139–153, Berlin, Heidelberg, 2010. Springer-Verlag.
5. Renzo Stanley and Hernan Astudillo. Ontology and semantic wiki for an intangible cultural heritage inventory. In *Computing Conference (CLEI), 2013 XXXIX Latin American*, pages 1–12, Oct 2013.
6. Renzo Stanley, Hernan Astudillo, Victor Codocedo, and Amedeo Napoli. A conceptual-kdd approach and its application to cultural heritage. In Manuel Ojeda-Aciego and Jan Outrata, editors, *Concept Lattices and their Applications*, pages 163–174, La Rochelle, France, October 2013. L3i laboratory, University of La Rochelle.
7. Petko Valtchev, Rokia Missaoui, and Robert Godin. Formal concept analysis for knowledge discovery and data mining: The new challenges. In Peter W. Eklund, editor, *ICFCA*, volume 2961 of *Lecture Notes in Computer Science*, pages 352–371. Springer, 2004.
8. Rudolf Wille. Why can concept lattices support knowledge discovery in databases? *Journal of Experimental and Theoretical Artificial Intelligence*, 14(2-3):81–92, 2002.