A Conceptual-KDD approach and its application to cultural heritage

Renzo Stanley[1], Hernán Astudillo[1], Víctor Codocedo[2], and Amedeo Napoli[2]

[1]Universidad Técnica Federico Santa María, Av. España 1680. Valparaíso, Chile,
{rstanley,hernan}@inf.utfsm.cl
[2]LORIA, BP 70239, F-54506 Vandoeuvre-lès-Nancy, France.
{victor.codocedo,amedeo.napoli}@loria.fr

Abstract. Several governmental and non-governmental organizations (NGOs), motivated by the UNESCO have undertaken the task of documenting the intangible cultural heritage of their communities. However, this has proven to be a difficult task. In this work we present a conceptual knowledge discovery in databases (CKDD) approach to aid a particular organization in this task (which has already started). Because of the dynamism of the cultural heritage domain, the design of the database used to store the documentation data has become obsolete. We propose to redesign the database (actually, its schema) to unveil independent modules of information collaboratively created by different domain experts. Finally, we present a straightforward method to convert the redesigned data schema into an ontological model which can be used for integration and publication purposes.

Keywords: Formal concept analysis, Knowledge discovery in databases, Conceptual knowledge discovery, Intangible Cultural Heritage, Documentation

1 Introduction

The Chilean National Council of Culture and Arts[1] (CNCA) has undergone the mission of documenting the intangible cultural heritage (ICH) of different small zones of the country in the context of a world-wide UNESCO[2] crusade to incentivize governments and NGOs to properly maintain our cultural knowledge. The ICH, as drafted in the *Convention for the Safeguarding of the Intangible Cultural Heritage*[3] refers to *"traditions or living expressions inherited from our ancestors and passed on to our descendants, such as oral traditions, performing arts, social practices, rituals, festive events, knowledge and practices concerning nature and the universe or the knowledge and skills to produce traditional crafts"*.

[1] http://www.consejodelacultura.cl
[2] United nations educational, scientific and cultural organization.
[3] http://en.wikipedia.org/wiki/Convention_for_the_Safeguarding_of_Intangible_Cultural_Heritage

To survey the ICH, the CNCA maintains several "documenters" who meet artists, artisans and other actors of the folkloric stage. Each documenter fills a "file" containing several semi-structured and text-free fields which are later registered in a relational database (DB). This database is later consulted by curators (usually domain experts) who fix problems in data definitions or storage. The CNCA has requested help in two specific aspects. Given the difficulty of documenting very different domains in ICH, the design of a data schema to support its documentation is a hard task which leads to inconsistencies in its model. Currently, the data schema used by the CNCA was designed by a computer engineer with some knowledge on the domain. However, as the ICH documentation process expanded to other cultural domains ("music", "folk festivals", "crafts", etc.) the data schema became too general and some modifications were requested to enable a higher level of details. The CNCA requested to analyse the data schema in order to find "modular" partitions on the model corresponding to different domains in ICH. Thus, each module can be maintained by a related domain-expert avoiding the risk of information conflict.

The second request derives from the fact that, since the ICH documentation is a multi-organizational endeavour motivated by UNESCO, it is expected that data collected by CNCA will be integrated with other databases later on. Because of this, CNCA has asked aid to obtain an ontological schema from its current data schema which can be used for linked data publication.

In this work we propose a novel knowledge discovery in databases process guided by formal concept analysis (also, referred to as *conceptual knowledge discovery in databases CKDD* [1]) approach to deal with both requirements. We analyse the current data schema definition to construct a concept lattice through FCA in which we apply different techniques to find the modules representing *ICH domains*. The contributions of this work are firstly, to present a real-world experience using FCA and concept lattices to achieve a task as important as heritage documentation and secondly, the description of a process to elicit an ontological schema from a relational DB schema which can be of benefit in different applications.

The remainder of this article is organised as follows: Section 2 presents the CKDD process while Sections 3, 4 and 5 detail each of its steps. Section 6 describes the case study of applying the CKDD process over the CNCA database for ICH documentation. Finally, Section 7 presents a discussion on related work and concludes the paper.

2 Formal concept analysis and knowledge discovery in databases.

We propose an iterative and human-centred approach to overcome both requirements of the CNCA based on Conceptual Knowledge Discovery in Databases (CKDD) [1]. CKDD is a tool to support humans in the discovery and extraction of knowledge from large collections of data where the conceptual representation of knowledge is a key aspect [12].

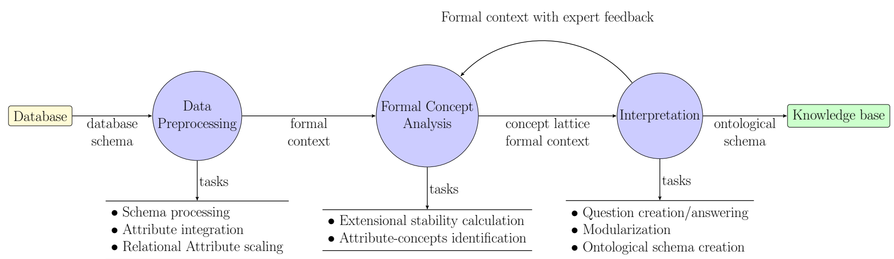


Fig. 1: FCA-based KDD process for CNCA

Figure 1 presents a 3-step CKDD process designed to take a database schema and convert it into a knowledge base. The process contains a loop between the interpretation and the FCA step in which domain expert knowledge is included. Notice that to avoid conflicts with the CNCA data, we only extract information from the database and never modify it. In the following sections we provide a detailed description of each step and sub-tasks developed.

3 First Step of the CKKD process: Data preprocessing

The first step starts by extracting the database schema and ends when it is converted to a formal context. Currently, most of the relational database management systems are based on the relational model constituted by tables and their relations (there are actually more elements in a relational model which are not of first interest here). We provide an adapted definition of the relational schema model as described in [2].

3.1 Relational data schema model

A relational schema $S = \{R_1, R_2, ..., R_{|S|}\}$ is defined as a set of tables or "relation schemas" $R_i(A_1, A_2, ..., A_{n_i})$ consisting of a table name R_i and a list of n_i fields A_j which define 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. An entry in a table is defined as an ordered *n-tuple* of values denoted by $t[R_i] =< v_1, v_2, ..., v_{n_i} >$ where each value is denoted as $t[A_j] = v_j \in dom(A_j) \cup \{NULL\}$. $NULL$ indicates that the value is unknown for the field in the entry. The *relation state* $r(R_i) = \{t_1, t_2, ..., t_{r_i}\}$ of table R_i denotes its total set of entries. In a table R_i, the set of fields $SK \subset R_i$ denotes a *superkey* which identifies a tuple as unique. A *primary key* PK is defined as a *superkey* where $|SK| = 1$ and $|.|$ denotes set cardinality, i.e. PK is a single field and we say that $R_i.PK$ is the *primary key* of table R_i the value of which unequivocally identifies an entry in table R_i. Finally, a field $R_1.FK$ is called a *foreign key* iff $R_1.FK = R_2.PK$ which indicates a *relation* between R_1 and R_2. In the particular case of the CNCA database for ICH documentation, a notion of inheritance of tables is supported. We define this as

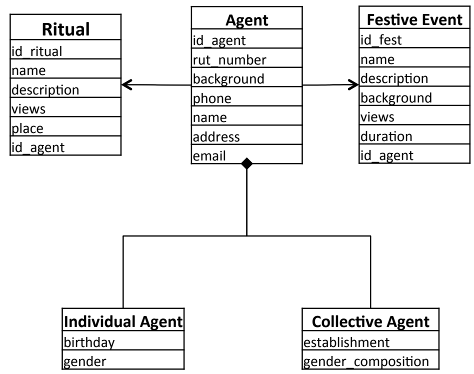


Fig. 2: Data schema example from the ICH domain

follows: table R_1 inherits from table R_2 iff $R_2 \subset R_1$ and $R_1.PK = R_2.PK$, i.e. table R_1 contains all the fields in table R_2 and they share the same primary key.

Figure 2 illustrates a data schema example $S = \{$*Festive Event*, *Ritual*, *Agent*, *Individual Agent*, *Collective Agent*$\}$ where arrows represent relation between two tables and a line with a rhombus at the end represents inheritance. Table *Ritual* can be represented as *Ritual(id_ritual, name, description, views, place, id_agent)* where *dom*(*name*) = *string*, *Ritual.id_ritual* is the primary key of table *Ritual* and *Ritual.id_agent* is a *foreign key* relating *Ritual* with *Agent*. The example also shows that *Individual Agent* and *Collective Agent* inherit from table *Agent*, meaning that all the attributes from table *Agent* are also present in tables *Individual Agent* and *Collective Agent*.

3.2 Formal context definition

Considering data schema $S = \{R_1, R_2, ..., R_{|S|}\}$ and the tables of the form $R_i(A_1, A_2, ..., A_{n_i})$ composed by the fields A_j, we define the formal context $\mathcal{K} = (S, A, I)$ where

$$A = \bigcup_{R_i \in S} R_i.A_j$$
$$I = \{(R_i, A_j) \,/\, \forall R_i \in S, \forall A_j \in R_i\}$$

Formal context $\mathcal{K}$ is composed by tables in the set S (all the tables or a sub-set of them), the set of fields A (composed by the set of all fields from all tables considered in S) and the relation set I (the relations between tables in S and their fields in A). Notice that we define a formal context by making the correspondences object-table and attribute-field. To avoid confusions, in this work we differentiate between these correspondences by using different font faces for `objects or attributes` and for *tables or fields*. Along with the formal context, we also define the following rules of **attribute integration** based on the characteristics of the fields:

– The special attribute `id` represents all *primary keys*:
  $id \in A\,,\ (R_i, id) \in I\ \forall R_i \in S$.
– Fields with the same name are integrated into a single attribute (e.g. attribute `name` represents $Agent.name$ and $Ritual.name$)

3.3 Relational attribute scaling

In the context of FCA, *foreign keys* correspond to relational attributes. For example, in Figure 2 we do not say "table *Ritual* contains a *foreign key*" (as in the case of *Ritual* contains a *primary key*), but rather "*Ritual* is related to *Agent* by a *foreign key*". Such kind of attributes cannot be included in a binary formal context. To deal with relational attributes, we scale them and treat them as normal attributes in the lines of [11]. We do so by prepending the prefix `related_to:` to the name of the table where the *foreign key* directs to (e.g. in the formal context in Table 1, we say that the object `Ritual` contains the attribute `related_to:Agent`). This is formalized as

$$R_j.FK = R_i.PK;\ R_i, R_j \in S \Rightarrow \texttt{related\_to:}R_i \in A \wedge (R_j,\texttt{related\_to:}R_i) \in I$$

In the case the table R_i pointed to by a foreign key do not exist in S, then we simply do not take into consideration that table nor we create the scaled relation $\texttt{related\_to:}R_i$.

	id	rut_number	email	phone	name	address	background	description	views	place	related_to:Agent	birthday	gender	establishment	gender_composition	duration
Agent	×	×	×	×	×	×	×									
IndividualAgent	×	×	×	×	×	×	×					×	×			
CollectiveAgent	×	×	×	×	×	×	×							×	×	
FestiveEvent	×				×		×	×	×		×					×
Ritual	×				×			×	×	×	×					

Table 1: Formal context created from the data schema example

4 Second Step of the CKDD process: Formal Concept Analysis

This step receives a formal context and ends when a concept lattice is constructed. Figure 3(a) shows the concept lattice calculated for the formal context in Table 1. Along with the concept lattice, the original formal context is sent to step 3 to allow modifications in it for further iterations of the process. We develop two tasks related to the identification of elements which would help in the redesign of the data schema.

4.1 Attribute concept identification

The *attribute concept* of attribute A_j is defined by $\mu A_j = (A'_j, A''_j)$, where $()'$ is the derivation operator in FCA [3]. The identification of attribute concepts is rather simple and we do it by navigating from the top to the bottom of the lattice. Attribute concepts are the most general concepts in the lattice containing a given attribute. In our case, the attribute concept of a given field contains the largest set of tables related to it. This is important since, as we will describe in the next step of interpretation, we process each field separately in order to create the ontological schema. Attribute concepts make this simpler as for each field we only need one concept and not the whole lattice.

4.2 Extensional stability calculation:

Since we are looking to enhance the design of a data schema in the database, within the lattice we would like to have only those concepts which group together in their extents tables of the same domain. This would allow us to address the modularization of information per domain issue which is one of the request of the CNCA, assuming that tables in the same domain are likely to share similar fields. In order to do so, we use the notion of *extensional stability* as firstly described in [6] and later in [10] as a way to measure "the probability of a concept to preserve its extent after leaving out an arbitrary number of attributes".

5 Third Step of the CKDD process: Interpretation

The final step receives a formal context and its associated concept lattice where each attribute concept is identified and each formal concept contains an *extensional stability* value. Since CKDD is an iterative process, this step has two possible outputs. If the expert decides to make another iteration, the process goes back to step 2 sending a modified version of the formal context received including feedback of the expert. If the expert decides to end the process, this will create an "ontological schema" which will be stored in a knowledge base and the process ends. For the iteration, there are two tasks to perform: *Question creation/answering* and *Modularization*. The task which transforms the concept lattice into an *ontological schema* ends the process. A further step of *annotation* which converts the entries in tables into linked data using the ontological schema can be considered, but for the sake of space we have left it out of this paper.

5.1 Question creation and answering

We use *extensional stability* as an indicator of how related are tables within a given concept extent. A lower stability indicates that those tables are grouped more as an accident (for example, because of the misuse of a single (or several) field name(s)) rather than because they belong to the same domain. We look for unstable attribute concepts because they are the most general concepts containing a given attribute, so they are those that relate more tables together. Moreover, we can create questions for the expert to answer in the hope they provide information to "break" the attribute concept and separate domains.

Consider the example in Figure 3(a) in which two domains are illustrated. The first relates events while the second relates people and communities. The most unstable attribute concepts (extensional stability of 0.5) correspond to those labelled with the attributes `background` and `place`. In the case of `place`, it only contains one table meaning that "breaking" this concept does not help in separating domains[4]. The attribute concept of `background` contains four objects, namely `FestiveEvent`, `Agent`, `IndividualAgent` and `CollectiveAgent` for which `FestiveEvent` belongs to a different domain than the others.

Regarding the questions posed to the expert, we have:

1. Would you like to assign `attribute` to all the objects?
2. Would you like to eliminate the `attribute` from a single/a set of objects/s?
3. Would you like to split `attribute` into different attributes for different objects?

If the expert selects option 1, the attribute will be shared by every object and hence, it will be placed as part of the intent of the top of the lattice (like `id` and `name` in Figure 3(a)). In the case of option 2, the attribute is erased from the formal context and hence, its attribute concept is removed. With option 3, we can recommend the partition of the attribute by looking at the sub-concepts of the attribute concept. Then, a new attribute is created for each partition made while the original attribute disappears. Consider A_i to be the attribute for which the expert has to answer a question. For each option we can define a new formal context as follows:

1. Option 1: $\mathcal{K}_A = (S, A, I \cup \{(R, A_i); \forall R \in S\})$
2. Option 2: $\mathcal{K}_A = (S, A \backslash \{A_i\}, I \backslash \{(R, A_i;) \forall R \in S\})$
3. Option 3: Let A^j be the splits of attribute A_i assigned to objects R^j, then $\mathcal{K}_A = (S, (A \backslash \{A_i\}) \cup A^j, I \backslash \{(R, A_i); \forall R \in S\} \cup \{(R^j, A^j)\})$

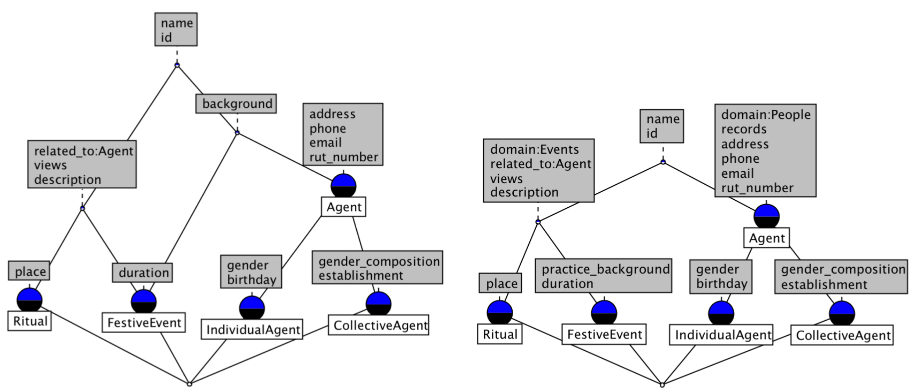
In the example, the expert selected option 3 splitting `background` into an attribute for `FestiveEvent` (`practice_background`) and another for the `Agents` (`records`). This example is actually an extract of a real case study where the

[4] A low stability in a singled-object attribute concept may indicate that the table should be split in two or more different tables. We do not address this issue in this work

expert realized that the term "background" was used with multiple meanings and that it should be separated into an attribute registering the history of past events and another for the official records of people. Figure 3(b) presents the concept lattice created from the formal context yielded by this decision.

5.2 Modularization

The expert may choose to perform this task disregarding answering questions. We start from the already processed concept lattice L and find its sublattices such as $L_i \cap L_j = \{\top, \bot\}$ where L_i, L_j are sublattices of L and $\top, \bot$ represent the top and the bottom of L respectively. We achieve this by obtaining the connected graphs from the lattice once $\top$ and $\bot$ are removed. Each sublattice is a candidate to represent a domain which must be labelled by the expert. The label is included into the formal context as a special attribute of the form `domain:Label` with relations to all the objects in the sublattice. The concept lattice is later recalculated. Figure 3(b) depicts the final form of the lattice for the running example. Finally, the expert may also be interested in merging more than one sublattice into a single domain. In that case, the special attribute is added to all objects in the set of sublattices selected by the expert.



(a) Concept lattice created from the data schema example

(b) Concept lattice after the expert's decision (attribute `background` split into `practice_background` and `records`) including domain labels

Fig. 3: Concept lattices for CNCA (before and after the expert's decision)

5.3 Ontological schema creation

The final task of the process derives an ontological model from the concept lattice that can be used for data integration and linked data publication. This is done

Concept	Element	Triples created
$\top = (S, \emptyset)$	$R_i \in S$	R_i `rdf:type rdfs:Class` *e.g. cnca:*Agent *rdf:type rdfs:Class*
$\bot = (\emptyset, A)$	$A_j \in A$	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*
$\bot = (\emptyset, A)$	`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*

$\bot = (\emptyset, A)$	`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` *and* $R_i \in A'_j$)	`cnca:`R_i `cnca:in_domain cnca:Label` *e.g. cnca:Agent cnca:in_domain cnca:People*

Table 2: Formal concept translation into RDF triples. Triples in the third column are created for the elements described in the second column within the formal concepts in the first column

by creating a set of RDF triples[5] for given elements in the formal concepts of the lattice. Table 2 shows this conversion where for example, in the first row it is shown that for the top concept, all the tables in its extent are modelled as RDFS classes[6] [7]. For simplicity purposes, Table 2 only presents a part of the total set of triples created considering the top $\top$, the bottom $\bot$ and attribute μA_j concepts.

6 Case study: CNCA Intangible Cultural Heritage Database

The database schema of the CNCA for ICH documentation consists of nearly 300 tables, however only 27 tables were selected by the experts on the basis of their representative and multi-disciplinary knowledge. Selected tables consider descriptions of agents, collective agents, festive events, culinary manifestations, geolocations and more. The formal context derived from the database contains 27 objects, 56 attributes, and 25 relational attributes. Figure 4 depicts the concept lattice built from this formal context.

[5] Resource Description Framework (RDF) is a standard specification for semantic web and linked data based on triples `http://www.w3.org/RDF/`.

[6] Resource Description Framework Schema (RDFS) is an extension of RDF which supports classes, properties and more complex definitions.

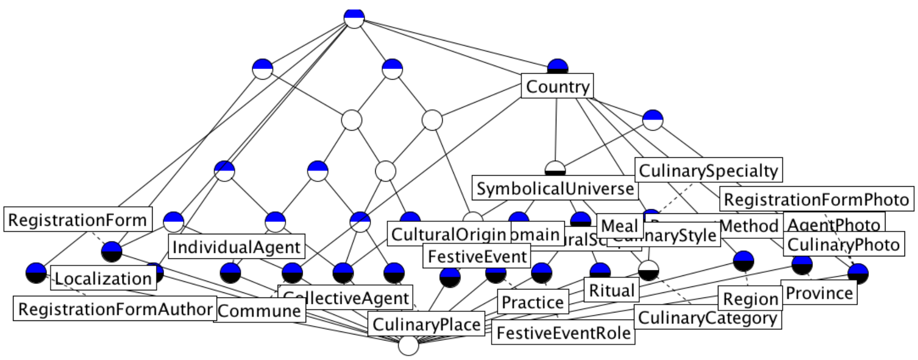


Fig. 4: Concept lattice from the ICH database schema

Iteration	**Attribute**	**Action**
1	`name`	Assign to all tables
2	`description`	Assign to all tables
3	`background`	Split the attribute
4	`background`	Split the attribute
5	`related_to:Commune`	Eliminate from some tables
6	`related_to:Localization`	Eliminate from some tables
7	-	Domain labelling
8	-	Domain labelling
9	-	Domain labelling

Table 3: Iterations made for the case study

Table 3 shows the actions taken by the domain expert during 9 iterations of the CKDD process. We distinguished between facts, questions and actions. Facts represent database assertions which are displayed for helping the expert in decision taking. For example, in iteration 1 the expert is presented with the facts: "71% of the objects contain the attribute `name`" and "The attribute `name` has a value in 100% of the entries in the objects where it appears". The first fact helps the expert to understand that `name` is actually very common among the objects and can easily be extended to the whole object set. The second fact combines information from the DB and the concept lattice to tell the expert that in all the entries t which contain the field *name* (represented by the attribute `name`), the value $t[name]$ is different than $NULL$. This indicates him that the attribute should not be removed from any object. Questions and actions were already detailed in Section 5.1.

In iterations 3 and 4 the expert split the attribute `background` ("antecedentes" in Spanish). This attribute appears in several objects (46% of object set) through all the domains, however with different semantics. Finally, the expert created the attributes `historical_background`, `records`, `practice_background` and `culinary_background`. In iteration 5 and 6, the expert decided to eliminate some attributes from the object set. For example, the relation `related_to:Commune` which represented a *foreign key* in the database, was not being used in 3 tables from which it was removed. Finally, in iterations 7, 8 and 9 the expert labelled the modules as subdomains of ICH `FestiveFeatures`, `Geo` and `AgentFeatures`. Figure 5 illustrates the final concept lattice presenting the different subdomains identified from left to right, namely; *Agent descriptors subdomain*, *Festive Event descriptors subdomain*, *Culinary descriptors subdomain*, *Geographical subdomain*, *Content creators subdomain*, *Photo subdomain*, *ICH subdomain*. The last subdomain includes all the manifestations about ICH in this database schema.

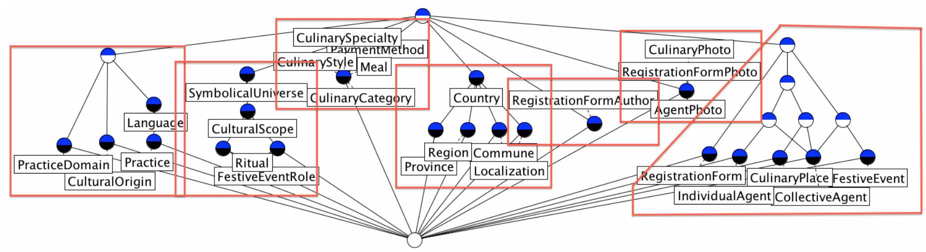


Fig. 5: Concept lattice after 9 iterations

7 Related Work, Discussion and Conclusion

Modelling semantic relations in DBs is proposed in [8] where the author presents a framework to formalize semantics in a lexical database. A lexical database maintains information about concepts and their semantic position w.r.t. each other (*hypernymy* or *meronymy*). In general, relational DBs do not share the same characteristics nor structure as lexical databases, allowing to store more heterogeneous data. In [9] the same author proposes an algebra for relational DBs which can be interpreted also in terms of a formal context in FCA. A similar idea was proposed by [5] where the author uses the algebra to translate a relational database into a family of formal contexts to benefit from both, the simplicity of the relational model description and the power of FCA in analysis. By contrast, our approach is more straightforward since we are not interested in data operations using the concept lattice, but rather in a direct translation of the data schema. Thus, we do not work with the entries in the tables of the database. In our approach we use the concept lattice as a support for guiding the redesign process in which a domain expert is the main provider of knowledge.

To conclude, in this article we have presented an application of a conceptual knowledge discovery in databases process designed to redesign and convert a database schema into an ontological model. The process is heavily *human centred* as it considers a domain expert as the main source of knowledge to guide the process. To support him, we use formal concept analysis with a formal context created from the set of tables and fields extracted from the database schema. The concept lattice calculated from this formal context is used to analyse the schema and create questions which the user should answer. Each question has an associated set of actions aimed at redesigning the database schema model.

The application is implemented over an excerpt of the Chilean National Council of Culture and Arts (CNCA) database for intangible cultural heritage documentation. Currently, we are implementing the process in full scale including more domain experts. Future work include the annotation process of data using the ontological schema created which has already been considered, however not described in this article. As pointed out by one of the reviewers of this work, the approach presented in this article may also be applied in the context of software engineering, specifically in code re-factoring and object-model re-engineering [4].

References

1. 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.
2. Ramez A Elmasri and Shankrant B Navathe. *Fundamentals of Database Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd edition, 1999.
3. Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, December 1999.
4. Robert Godin and Petko Valtchev. Formal concept analysis-based class hierarchy design in object-oriented software development. In *Formal Concept Analysis*, volume 3626 of *Lecture Notes in Computer Science*. 2005.
5. Joachim Hereth. Relational scaling and databases. In *Conceptual Structures: Integration and Interfaces*, volume 2393 of *Lecture Notes in Computer Science*, pages 62–76. Springer Berlin Heidelberg, 2002.
6. Sergei O Kuznetsov. On stability of a formal concept. *Annals of Mathematics and Artificial Intelligence*, 49(1-4):101–115, 2007.
7. Brian McBride. The resource description framework (rdf) and its vocabulary description language rdfs. In *Handbook on Ontologies*, International Handbooks on Information Systems, pages 51–65. Springer Berlin Heidelberg, 2004.
8. Uta Priss. Efficient implementation of semantic relations in lexical databases. *Computational Intelligence*, 15:79–87, 1999.
9. Uta Priss. Establishing connections between formal concept analysis and relational databases. In *Common Semantics for Sharing Knowledge: Contributions to ICCS 2005*, pages 132–145, 2005.
10. Camille Roth, Sergei Obiedkov, and Derrick Kourie. Towards concise representation for taxonomies of epistemic communities. *Concept Lattices and Their Applications*, 2008.
11. Mohamed Rouane-Hacene, Marianne Huchard, Amedeo Napoli, and Petko Valtchev. A proposal for combining Formal Concept Analysis and description Logics for mining relational data. In *Proceedings of ICFCA 2007*, 2007.
12. Rudolf Wille. Why can concept lattices support knowledge discovery in databases? *Journal of Experimental and Theoretical Artificial Intelligence*, 14(2-3):81–92, 2002.