Logical, graph based knowledge representation with CoGui

Jean-François Baget, Michel Chein, Madalina Croitoru, Alain Gutierrez, Michel Leclère, Marie-Laure Mugnier

Logical, graph based knowledge representation with CoGui

Jean Francois Baget*, Michel Chein*, Madalina Croitoru*, Alain Gutierrez*, Michel Leclere*, Marie-Laure Mugnier*

*LIRMM (CNRS and University Montpellier II, France)
«lastname»@lirmm.fr

Abstract. This paper reports on the ongoing effort in building an RDF ontology for the de-facto standard conceptual model for library catalogs. We motivate our work by a concrete real world application and demonstrate how using the CoGui Conceptual Graphs ontology editor will highly benefit the task.

1 Introduction

RDF [1](Resource Description Framework) is a language standardized by W3C (World Wide Web Consortium) dedicated to the representation of knowledge on the Web. Its popularity made it an important backbone of knowledge base related applications. A querying language for RDF has also been proposed (SPARQL) which allows to uniformly access different RDF data sources. To facilitate cross-questioning of such repositories, many class hierarchies (ontologies) have been proposed as a standard allowing the modeling of different areas: see for example FOAF dedicated to description of persons and relationships they have among themselves, or BIBO dedicated to bibliographic information etc. The use of ontologies standards ensures that we can link "in house" knowledge bases to many other existing databases via the Web. Moreover, extracting the RDF knowledge from a data repository is an indispensable tool for ensuring the quality of the database at hand.

To conclude, by the use of RDF modelling we provide (1) a standardized syntax for interoperability purposes and (2) a consistent vocabulary.

This paper reports on the ongoing effort in building an RDF ontology for the de-facto standard conceptual model for library catalogs. We motivate our work by a concrete real world application and demonstrate how using the CoGui Conceptual Graphs ontology editor highly benefits the task.

The paper is organised as follows: Section 2 details the motivation of our work in great detail and shows how using CoGui is advancing the state of the art. Section 3 puts down the theoretical foundations behind the visual representation of CoGui: Conceptual Graphs. Finally Section 4 presents current and future work within this project.

1. `http://www.w3.org/`

2 Motivating example

2.1 Scenario Description

Our motivating example is a real case scenario of representing library catalogue data. This work is being carried out in the context of a collaboration with ABES within the TGE-ADONIS [2] project. ABES [3] is a French public institution managing library catalogue records created in 1994.

Let us begin by briefing on the work of ABES before our collaboration started. ABES has designed and implemented SUDOC (the university documentation system), Calames (an on-line catalogue of higher education archives and manuscripts) and Star (a tool for posting and permanently archiving theses). Since this paper is mainly concerned with SUDOC, from here onwards we will only focus on this system. SUDOC encompasses more than 1100 public or private libraries, and its knowledge base consists of more than 8 million bibliographical data referencing more than 25 million documents. The main aim of SUDOC is to design a coherent set of referentials identifying people, collectivities, works, domains etc. for libraries records.

SUDOC has been initially encoded in UNIMARC [4]: a standard for assigning labels to catalogue records. A snapshot of a UNIMARC file is shown in Figure 1. All the constructs needed for interoperability (either with other libraries, or simply ensuring coherence of the catalogue records amongst each other) are represented using well defined functional blocks identified by three-character numeric tags. These blocks organise the data according to its function in a traditional catalogue record. Unfortunatelly, the semantics of these blocks is not made explicit in the language itself. Obviously, this will impose certain limitations when either (1) checking for consistency (e.g. identifying multiple entries of the same entity) or (2) for search optimisation. To this end a translation of UNIMARC in XML [5] was performed.

However, at this point - SUDOC records expressed in XML - the problem of semantics for the above mentioned problems was far from being solved. Moreover, an opening towards data available on the Web (that could benefit the system) was not at all possible. The decision was then taken to create an ontology for SUDOC. The language of choice was RDF(S) [6] given its standard status and the expressivity of the records. This is the moment when the collaboration with our group begun.

The motivation of our paper is modelling RDF(S) ontologies in a practical setting. To this end we will present a visual tool for representing and reasoning with ontologies: CoGui. We will fully detail CoGui further on in the paper. For the remainder of this section we will put our work in context and present the representational needs of the application scenario.

The question of the ABES joint project was how to design an RDF(S) ontology for SUDOC. Obviously, the design of such ontology has to be done in close collaboration with the ABES domain experts and needs to contain all the information already present in SUDOC. Basically, from a representation view point, the envisaged SUDOC ontology had to:

– Model already existing catalogue records based on the information already available in the XML file

2. http://www.tge-adonis.fr/
3. http://www.abes.fr/abes/index.html
4. http://www.ifla.org/en/unimarc
5. http://www.w3.org/XML/
6. http://www.w3.org/RDF/

```
001 0192122622@
010##$a0-19-212262-2$d£12.95@
020##$aUS$b59-12784@
020##$aGB$bb5920618@
100##$a19590202d1959####|||y0engy0103####ba@
1011#$aeng$cfre@
102##$aGB$ben@
105##$aac######000ay@
2001#$a{NSB}The {NSE}lost domain$fAlain-Fournier$gtranslated from the French by Frank
Davison$gafterword by John Fowles$gillustrated by Ian Beck@
210##$aOxford$cOxford University Press$d1959@
215##$aix,298p,10 leaves of plates$cill, col.port$d23cm@
311##$aTranslation of: Le Grand Meaulnes. Paris : Emile-Paul, 1913@
454#1$1001db140203$150010$a{NSB}Le {NSE}Grand Meaulnes$1700#0$aAlain-Fournier$f1886- 1914$1210##$aParis$cEmile-Paul$d1913@
50010$a{NSB}Le {NSE}Grand Meaulnes$mEnglish@
606##$aFrench fiction$2lc@
676##$a843/.912$v19@
680##$aPQ2611.O85@
700#0$aAlain-Fournier,$f1886-1914@
702#1$aDavison,$bFrank@
801#0$aUK$bWE/N0A$c19590202$gAACR2@
98700$aNov.1959/209@
```


FIG. 1 – *Example of a UNIMARC representation of the book "Le Grand Meaulnes" by Alain-Fournier*

– Ensure interoperability with data (e.g. other catalogue records, existing ontologies etc.) that could potentially be of interest in SUDOC in the course of its future development

A first direction of work was to take the SUDOC document expressed in XML and, based on a XSLT, to create the "corresponding" RDF(S) file. Of course, the output file has to be further enriched with a lot of information based on the implicit semantics of the SUDOC-XML but not explicitly expressed due to the limitations in the choice of language. Naturally, making explicit all the implicit rules in the XML file boils down to understanding the basis of the interchange formats of library catalogs.

It is at this point that we decided to study in detail the FRBR model. The FRBR model [7] is the de-facto standard conceptual model developed by an IFLA (International Federation of Library Associations and Institutions) group of experts. The model standardizes the functions a library catalogue should perform and lays the foundations for innovative catalogs that will use state-of-the-art techniques. A 142 pages document available online [8] makes explicit all the constructs that should be present when modeling a catalogue library record. Of course this document will subsume the information already present in SUDOC. The decision was then taken to use FRBR as the basis for the SUDOC ontology.

2.2 Limitations of existing work

Now that we set the context of the considered application scenario, let us look at existing work and how it compares to our approach. As mentioned in the previous section the problem at hand was how to represent the FRBR document in RDF(S).

7. http://www.bnf.fr/pages/zNavigat/frame/version_anglaise.htm?ancre=normes/no-acFRBR_gb.htm
8. http://archive.ifla.org/VII/s13/frbr/frbr_current_toc.htm

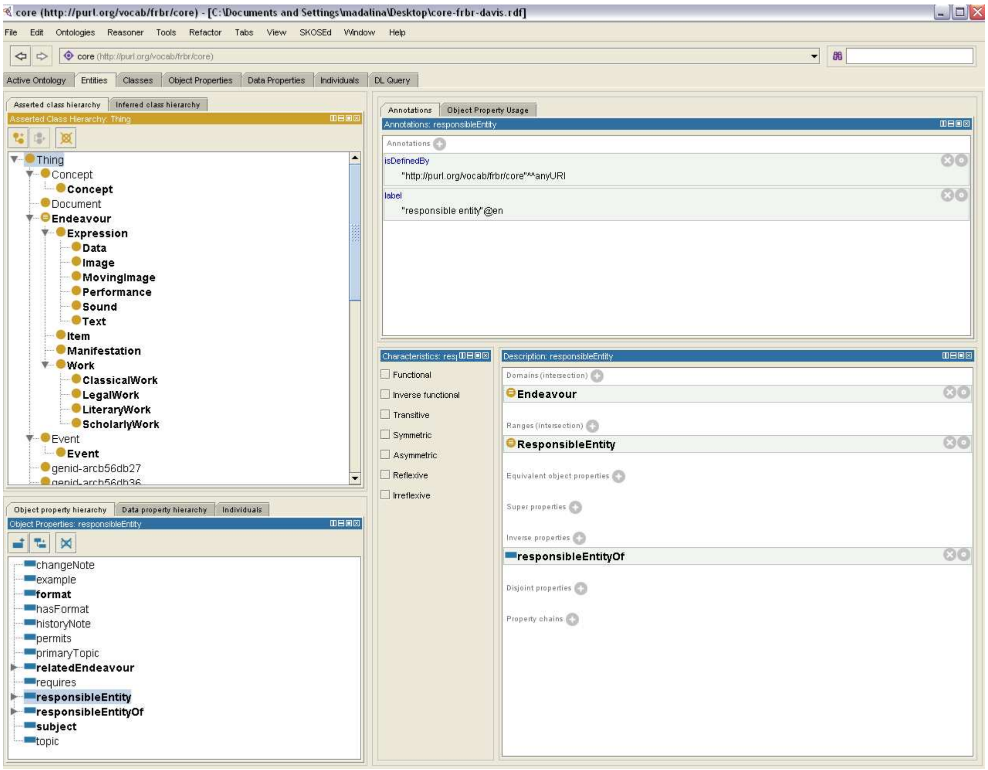

FIG. 2 – *Visualisation of* `http://vocab.org/frbr/core.rdf` *in Protégé*

The literature search revealed an existing RDF(S) file expressing FRBR. This file is publicly available on the web [9] and has been created by Ian Davis and Richard Newman. As no scientific paper on its creation exists (to the knowledge extend of the authors) the document had to be examined by simply opening it with an RDF(S) ontology editor and trying to understand the modelling choices behind it.

The de-facto standard for ontology editors is Protégé [10]. Of course, other tools for visualising RDF(S) exist [11] but their purpose is at most visualising RDF(S) (sometimes using hierarchical graphs, clustering techniques etc.) and not ontology (1) building and (2) reasoning using RDF(S).

Figure 2 shows a screen-shot of the above mentioned FRBR file in Protégé. Please note that there are no suitable visualisations for Protégé 4 which render information more intuitive (OWL Viz [12] is incompatible, Jambalaya [13] is not suitable for knowledge modelling etc.).

9. `http://vocab.org/frbr/core.rdf`
10. `http://protege.stanford.edu/`
11. `http://planetrdf.com/guide/`
12. `http://protegewiki.stanford.edu/index.php/OntoViz`
13. `http://protegewiki.stanford.edu/index.php/Jambalaya`

Unfortunately, as pointed out by the domain experts, in the context of our application the visualisation in Figure 2 has a number of drawbacks:

– Difficulty to visualise / explore concept / relation hierarchies
– Impossibility to represent any poset (and not only trees)
– No visual distinction between the model and the instances
– Impossibility to visualise relevant knowledge about instances

In this paper we present a tool for visually representing knowledge: CoGui [14]. CoGui is a Conceptual Graphs editor compatible with RDF(S). Conceptual Graphs are a logical graph-based knowledge representation language equivalent to the positive existential fragment of first order logic further detailed in Section 3. In the reminder of this section we will focus on how using CoGui will overcome all of the above mentioned drawbacks of Protégé.

2.3 Approach

Let us begin by presenting at a glance how the RDF(S) file available at `http://vocab.org/frbr/core.rdf` will be visualised and manipulated when being opened in CoGui. Figure 3 presents the concept type hierarchy, while Figure 4 presents the relation type hierarchy. Finally Figure 5 presents a fact in the ontology.

An ontology is composed by background knowledge and factual knowledge. In the context of this example we consider the background knowledge composed only by a hierarchy of concept types and a hierarchy of relation types with arity greater or equal to 1. For the general case of Conceptual Graphs (and not just Conceptual Graphs rendering of RDF(S)) please see Section 3.

In Figure 3 and 4 the concept / relation type hierarchies are displayed on the left hand side of the screen using a type list view and on the right hand side rendered graphically (types are displayed as vertices). Please note that type items may appear more than once hence allowing for the edition of posets. Types can be dragged from type views to graph views and vice-versa.

The edition of the concept hierarchy is being assisted by the control button (top menu, center) which allows to detect poset circuits or redundant edges in the concept types. As for relation types they have associated a signature (an ordered list of concept types). In this context the control button will forbid circuits and ensure that the signatures compatibility is respected.

All of the above demonstrates how CoGui syntactically and semantically overcomes the two first drawbacks of Protégé: visualising / exploring concept / relation hierarchies and representing posets in general.

In the main project window the background knowledge (consisting of concept type hierarchy, relation type hierarchy etc.) is presented alongside with the factual knowledge. Each fact is visualised in the editor graphically. Let us consider Figure 5 depicting a labeled bipartite graph where one class of nodes is the concept nodes and the other the relation nodes. A concept node is labeled by a concept type (e.g. *cc:Work*) and, possibly, by an individual (e.g. *file:///var/www/vocab.org/www/htdocs/frbr*). Incident edges nodes have to have compatible signatures with the concept nodes. Similarly, the control button will highlight any incompatibility. It is clear how CoGui allows for the representation of relevant knowledge about instances hence addressing the last two drawbacks mentioned above of Protégé.

14. `http://www.lirmm.fr/cogui/`

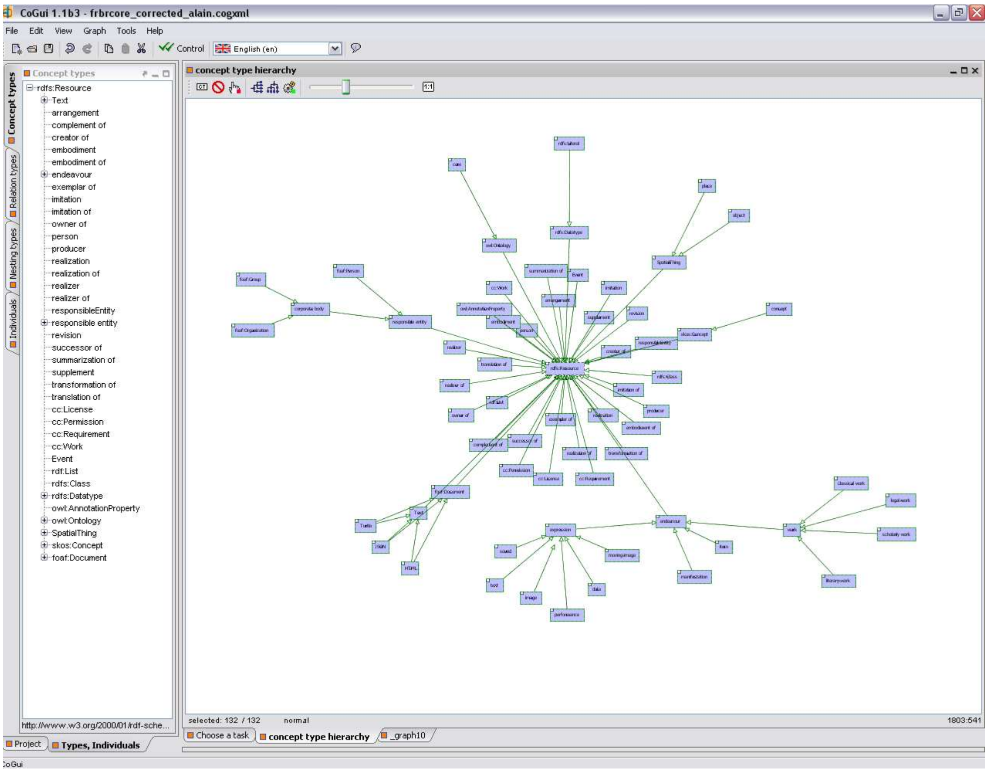

FIG. 3 – *Visualisation of concept types in* `http://vocab.org/frbr/core.rdf` *using CoGui*

2.4 Discussion

Please note that CoGui is not just a simple edition tool for ontologies but it also allows for reasoning (querying). This is done by calling the COGITANT library [15]. The approach of the querying mechanism is to search for homomorphisms between the query graph and the "knowledge base". The founded pieces of graphs (called projections) are specializations (in the logical sense) of the query graph. More details on homomorphism and Conceptual Graphs are given in Section 3.

It is also important to note here that CoGui offers additional constructs to assist with the edition of ontologies as well as with reasoning. An important number of expressive constructs is provided to the user, namely:

– **Rules**: used to represent implicit (common sense) knowledge. For instance, let us assume that "Eve is the mother of Abel" is a fact graph. If the ontology contains a rule saying that if x is the mother of y then y is a child of x then the system can automatically add the information that Abel is a child of Eve.

15. `http://cogitant.sourceforge.net/`

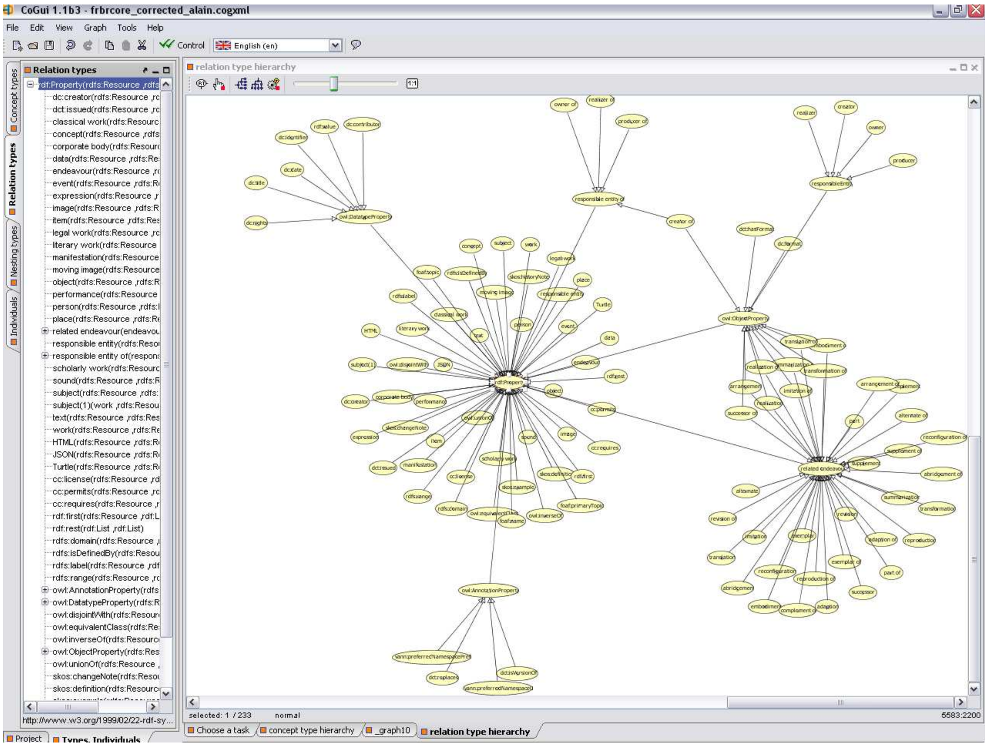

FIG. 4 – *Visualisation of relation types in* `http://vocab.org/frbr/core.rdf` *using CoGui*

- **Nesting types**: used to facilitate the construction of (well-structured) nested graphs (graphs detailing a certain concept). For instance, in RDF(S) this will correspond to easily editing *rdf:Statement*.
- **Individual graphs**: a simple conceptual graph having a special concept node (the head) which represents factual knowledge about an individual.
- **Pattern graphs**: usually used in an annotation scenario, representing a starting point when describing a document with respect to the chosen annotation type.

All of the above functionalities are fully described in the online [16] manual of CoGui.

3 Conceptual Graphs

The CoGui editor described above is a free graph-based visual tool, developed in Java, for building Conceptual Graph knowledge bases. CoGui allows the import / export of RDF(S) files.

16. `http://www.lirmm.fr/cogui/userguide.php`

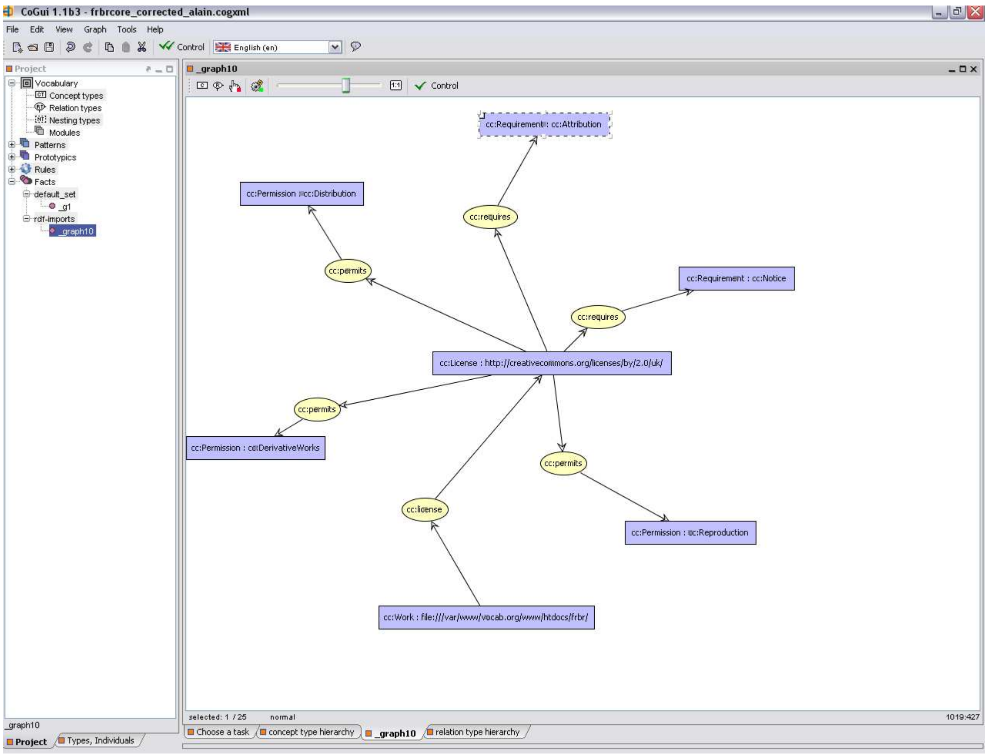

FIG. 5 – *Visualisation of individuals in* `http://vocab.org/frbr/core.rdf` *using CoGui*

In this section we will describe Conceptual Graphs, hence laying the theoretical foundations of the applicative work described in the Section 2. It is important to mention at this point that our choice of Conceptual Graphs is not only based upon visualisation and expressivity, but also on the potential for optimisation induced by using graphs for reasoning (see Chein and Mugnier (2009) for an in depth analysis). This is an important aspect to be considered in the context of this application and further work in this direction is detailed in Section 4.

Conceptual Graphs were introduced by Sowa (cf. Sowa (1976, 1984)) as a diagrammatic system of logic with the purpose "to express meaning in a form that is logically precise, humanly readable, and computationally tractable". In this paper we use the term "Conceptual Graphs" to denote the *family of formalisms* rooted in Sowa's work and then enriched and further developed with a graph-based approach in Chein and Mugnier (2009).

Conceptual Graphs encoded knowledge as graphs and thus can be visualized in a natural way:

- The vocabulary, which can be seen as a basic ontology, is composed of hierarchies of concepts and relations. These hierarchies can be visualized by their Hasse diagram, the usual way of drawing a partial order.
- All other kinds of knowledge are based on the representation of entities and their rela-

tionships. This representation is encoded by a labeled graph, with two kinds of nodes, respectively corresponding to entities and relations. Edges link an entity node to a relation node. These nodes are labeled by elements of the vocabulary.

The **vocabulary** is composed of two partially ordered sets: a set of concepts and a set of relations of any arity (the arity is the number of arguments of the relation). The partial order represents a specialization relation: $t' \leq t$ is read as "t' is a specialization of t". If t and t' are concepts, $t' \leq t$ means that "every instance of the concept t' is also an instance of the concept t". If t and t' are relations, then these relations have the same arity, say k, and $t' \leq t$ means that "if t' holds between k entities, then t also holds between these k entities").

A **basic graph** (BG) is a bipartite graph: one class of nodes, called *concept* nodes, represents entities and the other, called *relation* nodes represents relationships between these entities or properties of them. A concept node is labeled by a couple $t : m$ where t is a concept (and more generally, a list of concepts) and m is called the marker of this node: this marker is either the generic marker, denoted by $*$, if the node refers to an unspecified entity, otherwise this marker is a specific individual name. BGs are used to represent assertions called *facts*. They are also building blocks for more complex kinds of knowledge (such as rules, or nested graphs). In this paper we only detail rules as they are of direct interest to the framework we are proposing.

A **rule** expresses implicit knowledge of form "if *hypothesis* then *conclusion*", where hypothesis and conclusion are both basic graphs. Using such a rule consists of adding the conclusion graph (to some fact) when the hypothesis graph is present (in this fact). There is a one to one correspondence between some concept nodes of the hypothesis with concept nodes of the conclusion. Two nodes in correspondence refer to the same entity. These nodes are said to be *connection nodes*. The knowledge encoded in rules can be made explicit by applying the rules to specific facts.

These graphical objects are provided with a **semantics in first-order-logic**, defined by a mapping classically denoted by Φ in conceptual graphs (see Sowa (1984)). First, a FOL language corresponding to the elements of a vocabulary $\mathcal{V}$ is defined: concepts are translated into unary predicates and relations of arity k into predicates of arity k. Individual names become constants. Then, a set of formulas $\Phi(\mathcal{V})$ is assigned to the vocabulary. These formulas translate the partial orders on concepts and relations: if t and t' are concepts, with $t' < t$, one has the formula $\forall x(t'(x) \rightarrow t(x))$; similarly, if r and r' are k-ary relations, with $r' < r$, one has the formula $\forall x_1 \ldots x_k(r'(x_1 \ldots x_k) \rightarrow r(x_1 \ldots x_k))$. A fact G is naturally translated into a positive, conjunctive and existentially closed formula $\Phi(G)$, with each concept node being translated into a variable or a constant: a new variable if it is a generic node, and otherwise the constant assigned to its individual marker. The logical formula assigned to a rule R is of form $\Phi(R) = \forall x_1 \ldots x_p \; ((hyp) \rightarrow \exists y_1 \ldots y_q \; (conc))$, where: hyp et $conc$ are conjunctions of atoms respectively translating the hypothesis and the conclusion, with the same variable being assigned to corresponding connection nodes; $x_1 \ldots x_p$ are the variables assigned to the concept nodes of the hypothesis; $y_1 \ldots y_q$ are the variables assigned to the concept nodes of the conclusion except for the connection nodes.

More importantly, first order logic subsumption can be translated in a graphical operation: homomorphism. A homomorphism from G to H is a mapping between the node sets of G to the node sets of H, preserving the adjacency between nodes of G and decreasing the node labels. If there is a homomorphism (say π) from G to H, we say that G *maps* to H (by π).

4 Current and future work

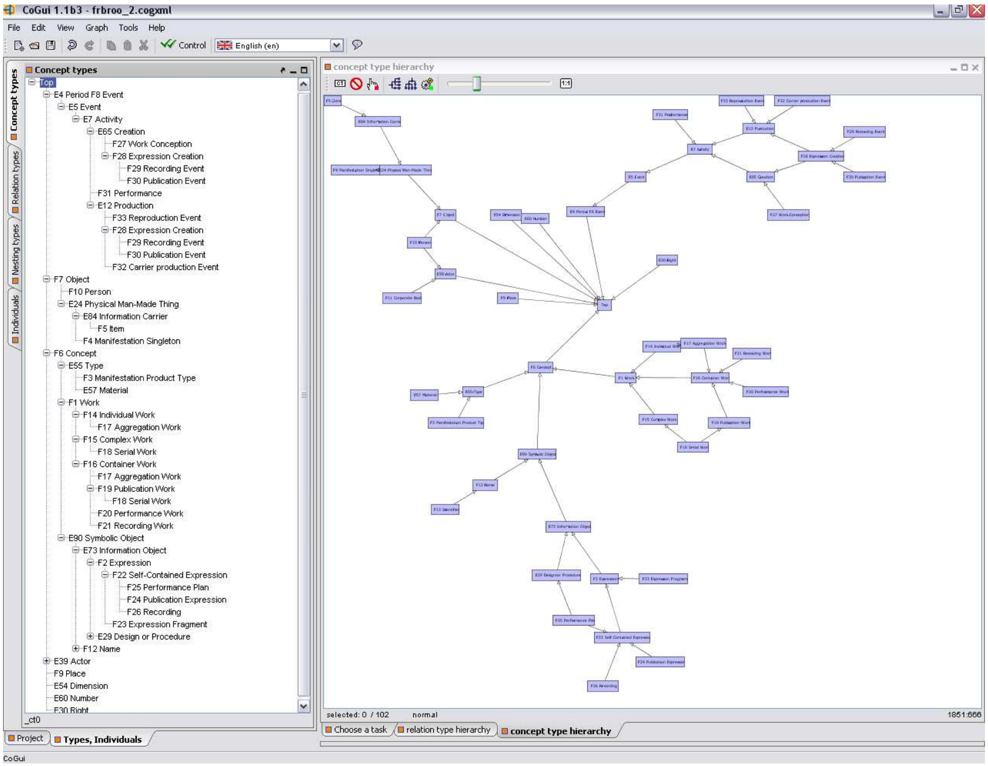

FIG. 6 – *Visualisation of the current state of the proposed RDF FRBR ontology using CoGui*

Let us now get back to the example presented in Section 2. We have shown our need for an FRBR RDF ontology and why we chose CoGui as the ontology editor for this task. Moreover, CoGui is now the de-facto tool used for our modelling collaboration with the ABES domain experts (which are not trained computing scientists).

Let us now further analyse the RDF(S) file presented in `http://vocab.org/frbr/core.rdf`. Following discussions with ABES it became obvious that this existing RDF file is not suitable for the application at hand. This is due to:

- The lack of important structures present in FRBR indispensable for incorporating innovative catalogs. Moreover, the missing structures cannot be easily integrated in the existing ontology given the different level of granularity chosen by Ian Davis.
- The lack of semantic relations between certain essential concepts
- The semantic ambiguity and overlapping of certain ontological constructs

While the first two items are self explanatory, we will detail the third. The RDF ontology at `http://vocab.org/frbr/core.rdf` aims at interoperability within the LinkedData [17]

17. `http://linkeddata.org/`

cloud by employing different but semantically overlapping constructs from different ontologies already in the cloud (for instance foaf:Author [18] and dc:Author [19]). It is not clear how these constructs will be directly used for reasoning in the ontology.

In the light of the above mentioned reasons we have thus decided to use the 142 pages FRBR reference document as a common base between the experts and our group and to build, using CoGui, the complete [20] FRBR RDF ontology. Figure 6 shows this endeavor at the current date.

We plan to extend this work in a number of ways. First, an obvious extension will be the finalisation of the ontology. To this end we are investigating how using Conceptual Graphs rules (formally defined in Section 3) can speed up the ontology building process. Second we are interested in how to achieve interoperability within the LinkedData cloud without the above mentioned drawbacks of semantic ambiguity. Third we are interested in different important theoretical problems arising from this applicative scenario. More precisely we are investigating not only how to manipulate, but also store, in an intelligent way, the large number of graphs this application will output.

References

Chein, M. and M. Mugnier (2009). *Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs*. Springer.

Sowa, J. F. (1976). Conceptual Graphs. *IBM Journal of Research and Development 20*(4), 336–375.

Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley.

Résumé

Cet article décrit le travail de modélisation d'une ontologie RDF pour des notices bibliographiques. Nous motivons notre approche par une application concrète du monde réel et démontrons comment l'utilisation d'un éditeur visuel à base de graphes améliorera cette tâche.

18. From the FOAF ontology: `http://www.foaf-project.org/`
19. From the DUBLIN CORE ontology: `http://dublincore.org/`
20. Regarding the needs of the SUDOC