Example Based Programming and Ontology Building: A Bioinformatic Application

Quentin Riché-Piotaix[1,2,3(✉)], Patrick Girard[1,3], Frédéric Bilan[1,2], and Ladjel Bellatreche[3]

[1] Université de Poitiers, Poitiers, France
quentin.riche.piotaix@univ-poitiers.fr
[2] CHU de Poitiers, Poitiers, France
[3] LIAS, ISAE-ENSMA, Poitiers, France

Abstract. To find ways to facilitate the querying process of heterogeneous databases reveals a critical research avenue, especially in biology. Making use of ontologies is considered one of the best solutions, which makes the activity of ontology design critical for biologists. However, such design process is not easily attainable by non-experts, issue sublimated by the constant evolution of the domain taxonomies [1]. Moreover, designing ontologies currently requires some expert knowledge of the domain as well as skills in database and ontology modelling. This fact was corroborated by our pilot study involving geneticists from the Poitiers hospital. The specialists did not possess any prior knowledge of conceptual data models. Nevertheless, they were able to build their own mental model of the situation that could later be correlated to the actual database models.

Compared to previous End-User Programming approaches, this experiment shows that End-User Programming techniques permit to build and use conceptual models without any need for specific training. In this poster, we describe the pilot study we conducted using geneticists during the dedicated ontologies design process that allows querying public databases. Several specific constraints were identified, along with their proposed solutions. A complete example of ontology design, built from the genetic field, is then described.

Keywords: Ontology · Database · Human-computer interaction
Genetic · Bioinformatic

1 Introduction

With the advent of new DNA sequencing techniques, a tremendous amount of data is constantly being generated in the field of genetics. They are distributed in many highly specialized databases that geneticists then interrogate successively. Researchers cross the results of their successive requests in order to make

C. Stephanidis (Ed.): HCII Posters 2018, CCIS 850, pp. 101–108, 2018.
https://doi.org/10.1007/978-3-319-92270-6_14

the diagnosis as reliable as possible. It would be very interesting for geneticists to be able to centralise the databases they use rather than to have to make requests on different bases as they currently do. Today's mapping of the field uses various databases containing concepts included in several databases, with semantic differences, and intra and inter-base relationships. This configuration illustrates the need for an ontology spanning the domain. Indeed, within the genetics field, some areas are fully covered, such as genes in Gene Ontology[1] or phenotypes in Human Phenotype Ontology[2], but there is no global ontology. Genetics is a constantly evolving wide and complex field. It is therefore impossible to consider creating and maintaining a complete ontology of the field without a considerable amount of human and material resources. As a rapidly expanding research field, a closed system would become very quickly out of date. Instead, we aim here to automatically build an ontology from the available data. Work already exists in this area, but most of it is aimed at extracting knowledge from unstructured information, such as sets of web pages, text and multimedia files. For genetics, databases already exist, and export files are available for download. However, the structure of the base is not available for perusal. Building an ad hoc tool for each database, if technically possible, is not a good solution due to the domain's constant evolutions. Here the design process will therefore be performed by reusing semi-structured data sources. Our interest is an approach capable of allowing end-users to build the systems that meet their needs. An End-User Programming approach was therefore selected, in line with geneticists' usual lack of expertise in computing. To perform this task, two options were available:

- To train geneticists to acquire the necessary computer skills (as it was done in [2])
- To create a system that disguises computer concepts to the novice user.

Most geneticists have neither the time nor the desire to learn how to code. Moreover, to train them would be as costly as using an expert database engineer, which is out of our scope and lacks adaptability to the domain's constant evolution. The only solution left was to use an approach that does not require any further apprenticeship. Partial answers to this problematic exist in different areas of research, including queries by example, but these systems focus heavily on queries rather than on building an interactive system. Our problematic of reconstruction was modified by this constraint and thus became: how to allow a computer science novice to create an ontology from genetic database exports only? We will present the particular context of this research study in Sect. 2. Our approach to system design will then be described in Sect. 3. To follow, a validation study of the approach will be presented in Sect. 4, and discussed in Sect. 5.

[1] http://geneontology.org.
[2] http://human-phenotype-ontology.github.io/.

2 Analysis

End-user programming (EUP) allows expert users (EU) with no or few programming skills to build programs related to their specific needs [3,4]. Most of the time, EUP is used to automate basic EU interactions, for example for repetitive tasks [5,6]. In our case, the needs to be elicited are situated on the conceptual level. Based on previous observations, our hypothesis is that by using EUP techniques, biology researchers - more specifically geneticists - could build ad-hoc ontologies in a more efficient manner for query generation that would render a quicker diagnosis. We observed common databases usage by geneticists and found that they create complex queries by making multiple simple queries across multiple databases. We concluded that they must already have their own mental model of the field. This allowed us to formulate our initial hypothesis: geneticists possess a mental model of the domain and it is possible to find semantic correlations between the way they represent it and the real model. We are not in a classic EUP case, which represents a computerized automation of repetitive EU tasks. We are at the conceptual level: we use EUP to allow the EU to create a structure that fits his/her mental model, also usable as a database schema. Applied to our situation, a classic EUP case is a system that allows the EU to query various database through their API. While this would work, the situation would not be resolved, as the EU would still have to create an ad hoc program for each API. That is why we focus on the conceptual model.

2.1 A Genetics Case Study

To support our approach statistically, we have defined an example by selecting three databases only: OMIM morbidmap[3] (contains mainly diseases associated with genes), dbSNP[4] (identifies point mutations), and finally a projection of HPO (contains mainly association phenotype-disease). These three bases were chosen because of their diversity in form and substance. Indeed, it presents slightly different structures at the header level, with a visible header (not necessarily formatted in the same way as the data), or without any header (replaced by textual explanations in another file). They were mostly selected due to their content, representing four essential notions within the world of genetics: genes, diseases, phenotypes and variations.

2.2 Case Study Constraints

Most current approaches use complex artificial intelligence to limit the role of the domain expert in the design process. This is a very consistent approach, since the data often come from corpus extracted from the Web, without direct link with such experts. In this project, our configuration elicited another set of constraints, as follow:

[3] https://www.omim.org.
[4] https://www.ncbi.nlm.nih.gov/projects/SNP.

- Accessibility to domain experts, all computer science novices.
- During the design process, these experts will be alone, they will not have access to ontologists or database specialists.
- The experts' confidence in the results must be total, which means that the EU must be able to check the automatic steps. Our goal is therefore to rely as much as possible on the available resources, which are the EU and the data, in order to free ourselves from the heavy approaches of machine learning, or the use of an external ontology engineer. We do not want to only make an ontology, we want to have it done expertly.

2.3 Difficulties Encountered

Current approaches to creating ontologies are facing several pitfalls, at different level of system actions:

Issue 1. Several important notions of databases must be known (such as tables, attributes or relations).
Issue 2. The interfaces are developed to be compatible with the target audience, that is to say engineers specialized in data management and ontologies. The solution often adopted is to rely on a standard domain such as SPARQL [7].
Issue 3. When the data comes from unstructured files, it is difficult to extract the relations [8].
Issue 4. When the data comes from semi-structured files, the format dependency is very important, and it is common to have errors and malformations making the machine interpretation process extremely complex [9].
Issue 5. Most tools are designed following machine learning approaches, they are very dependent on the size of the corpus, and the trust given to each source to avoid conflicts [10].
Issue 6. Since sources are often included iteratively, one after the other, this greatly increases the chances of duplicating an entity that comes from two different sources [11].
Issue 7. Finally, it is complicated to find a good way to evaluate the quality of the ontology produced, and therefore the system's reliability. The ideal solution would be a manual review of the ontology by domain experts able to validate or not the links between the entities. However this would also be too costly. Automation of this phase are under development [12], but will not be detailed here any further.

For the first issue, we may rely on the EU by asking them information, or request a verification of the data structure. The second is a classic EUP issue, we will probably face it, but we cannot use informatic standard, as they are not known by our EU. We must include these EU into the design phase and make sure that the interfaces are clear. We are not concerned with issue 3, but issue 4 will require an intelligent parsing step in order to overcome any potential file malformations. As the EU needs control choices, we cannot use a machine learning approach,

so issue 5 is also discarded. Issue 6 can be solved because of the presence of identifiers in the data. This can be used to avoid duplicating entities. Finally, the last issue is moot here since our ontology was directly developed by the domain expert. However, we can evaluate the result according to a subjective criterion, by checking whether the ontology fits the provided data.

3 Domain Vocabulary

In order to create a system capable of answering our problematic, we must provide three answers. We first need to prove our initial hypothesis: there are relationships between the geneticist's mental representation model and the database concepts used in the model. If this hypothesis is validated, it should be possible to characterise these relations by investigating how EU express them: what vocabulary is used? Which grammatical structures? Finally, based on these results, we must deduce adapted solutions for the EU to inform the concepts that interest us.

3.1 Approach

We first wanted to check that geneticists are able to extract concepts and relationships from data sources by themselves. We also investigated whether other database concepts are described spontaneously by the EU. We then determined how these notions were described, and whether it was possible to find a correspondence between the EU's vocabulary and the database notions. We were particularly attentive at the possible apparition of homonyms or synonyms. To investigate this issue, the following domain vocabulary definition test was proposed to five domain experts: Each participant was asked to verbalise the content of export file from three domain databases. We recorded the think-aloud and searched for possible equivalences between these people's vocabulary choices and ontology notions. From the recordings, we were able to verify our initial hypothesis and note that the geneticists do indeed use databases notions. We then listed the vocabulary and syntactical structures in context and explored their usage:

- Concepts were generally easily isolated when belonging directly to the field of expertise only.
- Attributes were well distributed between well-defined concepts. The notion of identifier was very present, since each concept had at least one identifier per database. Participants found them without problems, even for concepts they did not understand.
- Attribute types were not requested from participants, and never were specified spontaneously. This is not critical, since they can be guessed or requested later from the EU.
- Regarding relations, we found 21 different descriptions to talk about three types of relations: 0:n, 1:n and concept-attribute. The three most frequent descriptions were "have multiple" and "associated with" that described a 0:n relationship, as well as "correspond to" to describe a Concept-Attribute relationship. EU therefore used many synonyms (use of several words to describe

a relationship). Only one EU could use up to 4 synonyms for a relationship. Several homonyms (use of a word to describe several types of relationship) have also been noted. They are related to the presence of several participants, but unlike the synonyms, each participant remained constant, using one word per type of relationship. The two identified homonyms were "associated with" and "have several", used to describe the three types of relations present, however traditionally chosen to describe 0:n relationship.
– Cardinalities were not always detectable orally. When they were specified, this involved the use of modal verbs such as "may" and "must", as well as the use of specific determinants such as "many". This enabled the building of structures such as "a disease must have one or more phenotypes".

4 Prototype Validation

This domain vocabulary definition test proved that it is possible to find a correspondence between the words used by geneticists and the ontological notions necessary for the construction of the system. We have thereafter imagined and developed a prototype that served as a translator between the geneticist and the ontology, to be confronted to our end-users. As previously mentioned, EU are usually highly connected to the data, so we made the data visible throughout the process, which consists of 4 steps:

– Import and parsing of data;
– Creation of the present concepts, with their attributes;
– Creation of relations between concepts, with their cardinalities;
– Visualization of the final ontology.

These different steps were performed using a classic web interface. The first step allowed the EU to load his/her data file and view it directly. He/She could then interact directly with a set of parsing parameters, which allowed him/her to easily find the most useful settings to perform his/her task. When the data display became clear, the EU could enter the concepts and their attributes. A verification step avoided inserting duplicated entries in the ontology, and forced the EU to define a primary key. If a concept already existed, it then ought to be entered as a synonym. Finally, the EU entered relations using a syntax close to the results of the domain vocabulary definition test, relying in particular on drop-down lists of modal such as "may" or "must". Finally, a visualization screen allowed EU to summarize easily concepts and relations present in the ontology. The main objective of this study was to evaluate the usability of the approach. Should the test be a success, even partial, the approach would then be feasible. On the opposite, no conclusion could be drawn as either the approach could be unreliable, or no solution could be computed. In case of favorable results, we would then evaluate the different steps to identify possible blocking point. Finally, we would collect the opinions of EU passing the test to improve the prototype. Four geneticists were recruited for this study. They were asked to create an ontology using the three aforementioned databases with our prototype.

4.1 Results

None of the ontologies created were perfect, because of optimization problems, missing concepts and/or relations. All of them were however fully usable. An example of optimization problem lied in a concept being artificially split into two concepts linked by a 1:1 relation. In addition, several issues concerning the resulting ontologies and the EU's behavior during the tests drew our attention:

- Several synonyms were used at the concept level, such as "phenotypes", sometimes called "symptoms".
- One of the EU created foreign keys before creating any relation. This anticipation can probably be attributed to his personal Access 8 database creation experience.
- Even though some of the attributes were very close semantically, none were misallocated.

5 Discussion

As shown in the previous section, EU have generally managed to provide an implementable result, with no help from the data. However, we can imagine situations that would require more help, asking for clarification from the EU and helping him rely on the provided data. For example, an automatic detection of 1:1 relations could enquire whether it would not be more relevant to group concepts in a single entity. On the one hand, we can imagine the opposite case, where a concept initially included in another must be extracted in the light of new data, in order to create an independent entity. On the other hand, it would be impossible to detect the creation of false concepts, such as the one called "transcript" in our case. However, we can hope the EU would be aware of the problem and seek a more adaptable solution. This issue for example could have been resolved by deleting the concept and adding an attribute to another. Conceptual omissions can traditionally be detected if none of the columns of a database are loaded. However these can also be on purpose, as it was the case with one of the EU. Whilst relational omissions are difficult to detect, except in some special cases. However, one can rely on the presence of concepts in the same file to deduce a probable relationship between the two: the presence of genes and SNP in a single base generally indicates a link between SNP and genes. Relationships can be reflected, and missing cardinalities can be requested from the EU. Cardinality errors would have to be checked in the data itself. It would be impossible to find a definite answer in all cases: impossible for example to contradict a relation 0:n, but a relation 0:1 can be easily verified.

The first issue we had seen in the analysis section can easily be handled. The second has been verified by this validation study, and the debriefing of the test. EU who have passed the study may now be fully involved in future development so we can keep clear interfaces. Our parsing phase shown that file's minor malformations can be handled, we must continue to test it with bigger malformations. Finally, we have not faced the sixth issue yet, we might encounter it with more databases.

6 Conclusion

With this case study, we shown that it is possible for a domain expert, novice in computer science, to build an ontology from existing data. We have evoked the problems faced by users and proposed several solutions. The construction of such as system, however, is not completely solved, since many semantic problems will have to be solved at the data level. Finally, the creation of an adapted query system could allow end-users to find a concrete interest in its use.

References

1. Petasis, G., Karkaletsis, V., Paliouras, G., Krithara, A., Zavitsanos, E.: Ontology population and enrichment: state of the art. In: Paliouras, G., Spyropoulos, C.D., Tsatsaronis, G. (eds.) Knowledge-Driven Multimedia Information Extraction and Ontology Evolution. LNCS (LNAI), vol. 6050, pp. 134–166. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20795-2_6
2. Letondal, C.: Interaction and programming. Ph.D. thesis, Université Paris Sud - Paris XI (2001)
3. Cypher, A., Halbert, D.C.: Watch What I Do: Programming by Demonstration. MIT press, Cambridge (1993)
4. Lieberman, H.: Your Wish is My Command: Programming by Example. Morgan Kaufmann, Burlington (2001)
5. Girard, P.: Bringing programming by demonstration to cad users. In: Your Wish is My Command, pp. 135–VII. Elsevier (2001)
6. Goubali, O., Girard, P., Guittet, L., Bignon, A., Kesraoui, D., Berruet, P., Bouillon, J.-F.: Designing functional specifications for complex systems. In: Kurosu, M. (ed.) HCI 2016. LNCS, vol. 9731, pp. 166–177. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39510-4_16
7. Lefrançois, M., Zimmermann, A., Bakerally, N.: Génération de RDF à partir de sources de données aux formats hétérogènes (2017)
8. Kim, S., Alani, H., Hall, W., Lewis, P., Millard, D., Shadbolt, N., Weal, M.: Artequakt: generating tailored biographies from automatically annotated fragments from the web (2002)
9. O'Connor, M.J., Halaschek-Wiener, C., Musen, M.A.: Mapping master: a flexible approach for mapping spreadsheets to OWL. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010. LNCS, vol. 6497, pp. 194–208. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17749-1_13
10. Brewster, C., Iria, J., Zhang, Z., Ciravegna, F., Guthrie, L., Wilks, Y.: Dynamic iterative ontology learning. In: 6th International Conference on Recent Advances in Natural Language Processing (2007)
11. Dimou, A., Sande, M.V., Colpaert, P., Mannens, E., Van De Walle, R.: Extending R2RML to a source-independent mapping language for RDF. In: Proceedings of the 2013th International Conference on Posters & #38; Demonstrations Track, Aachen, Germany, ISWC-PD 2013, vol. 1035, pp. 237–240. CEUR-WS.org (2013)
12. Drumond, L., Girardi, R.: A survey of ontology learning procedures. In: de Freitas, F.L.G., Stuckenschmidt, H., Pinto, H.S., Malucelli, A., Corcho, O. (eds.) CEUR Workshop Proceedings of WONTO, vol. 427. CEUR-WS.org (2008)