Semantic Techniques for Enabling Knowledge Reuse in Conceptual Modelling

Jorge Gracia[1], Jochem Liem[2], Esther Lozano[1], Oscar Corcho[1], Michal Trna[1], and Asunción Gómez-Pérez[1], and Bert Bredeweg[2]

[1] Ontology Engineering Group, Universidad Politécnica de Madrid, Spain
{jgracia, elozano, ocorcho, mtrna, asun}@fi.upm.es
[2] Informatics Institute, University of Amsterdam, the Netherlands
{j.liem, b.bredeweg}@uva.nl

Abstract. Conceptual modelling tools allow users to construct formal representations of their conceptualisations. These models are typically developed in isolation, unrelated to other user models, thus losing the opportunity of incorporating knowledge from other existing models or ontologies that might enrich the modelling process. We propose to apply Semantic Web techniques to the context of conceptual modelling (more particularly to the domain of qualitative reasoning), to smoothly interconnect conceptual models created by different users, thus facilitating the global sharing of scientific data contained in such models and creating new learning opportunities for people who start modelling. This paper describes how semantic grounding techniques can be used during the creation of qualitative reasoning models, to bridge the gap between the imprecise user terminology and a well defined external common vocabulary. We also explore the application of ontology matching techniques between models, which can provide valuable feedback during the model construction process.

Keywords: Qualitative Reasoning, Semantic Grounding, Ontology Matching

1 Introduction

The Qualitative Reasoning (QR) area of Artificial Intelligence (AI) researches conceptual representation of systems, and the prediction of their behaviour through reasoning. QR has been successfully applied in a variety of domains, e.g., environmental science [6, 27], autonomous spacecraft support, failure analysis and on-board diagnosis of vehicle systems, automated generation of control software for photocopiers [7], etc.

Of particular relevance to this paper is the use of QR in science and education. QR models can be used as a means for learners to formally express and test their conceptual knowledge about systems in an educational context [3]. A desirable feature would be the possibility of uploading expert and learner models to a shared learning environment, and receiving feedback from the common knowledge contained in such a resource. This paper addresses the issue of how this environment can be created and used effectively.

In the current state of the art in qualitative modelling and simulation tools [5, 4, 28, 17], modellers are free to choose their own domain vocabulary. However, this results in different modellers using different terms to denote the same concept (e.g. *death rate* and *mortality*). Different languages and spelling variations further exacerbate the issue. This

makes generating feedback based on a large set of models difficult, since the consensus and disagreement between models cannot be easily determined. We hypothesise that the application of Semantic Web techniques to describe and interlink QR models will be beneficial.

We call *grounding* the process of linking terms in models to concepts in a common vocabulary. Grounding transforms the set of models into a semantically enabled networked resource of scientific data that can be exploited both in the scientific and educational contexts. By allowing comparison between models, algorithms can be written to make modelling suggestions based on other models. Furthermore, when reusing model parts, knowledge can be more gracefully integrated, as equivalent knowledge already existing in a model can be reused. For finding these pieces of common information, *ontology matching* techniques can be applied to explore the similarities among models, with the purpose of getting valuable feedback during the model construction process.

The approach presented in this paper consists of the following steps (see Figure 1):
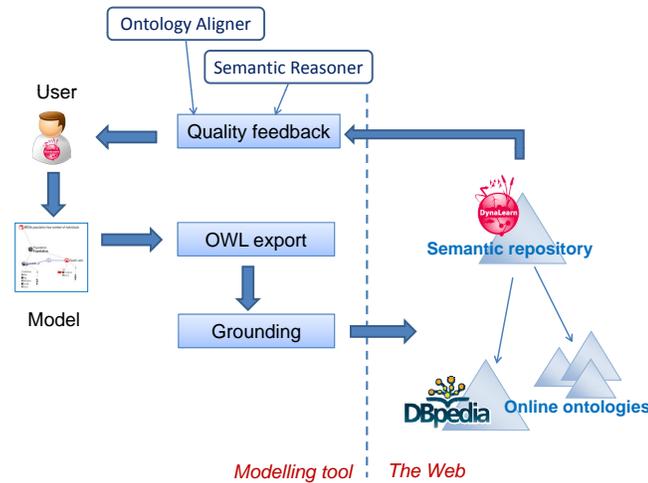
Fig. 1. Overview of the approach.

1. *OWL Export*. The conceptual knowledge contained in QR models is extracted and expressed in OWL [20], to facilitate their ontology-based description.
2. *Grounding*. The terms from the QR models are linked into external vocabularies (e.g., DBpedia [2]). These grounded models are stored in a semantic repository.
3. *Quality feedback*. Alignment and reasoning techniques are applied to discover similarities and dissimilarities among models, and based on that to enrich the modelling process with adequate feedback and suggestions for knowledge reuse.

The output of this process is a networked pull of online aligned conceptual models (expressed as ontologies) anchored to common vocabularies, and representing specific scientific domains. This has the potential of being a valuable Web resource for scientific progress in general and for semantic guided learning in particular.

The rest of the paper is organised as follows. Section 2 introduces the topic of qualitative reasoning. In Section 3 a method for expressing QR models in OWL is presented. Section 4 describes the semantic grounding process. Quality feedback from stored models is described in Section 5. Section 6 describes our experimental results. In Section 7 some related work is presented. Conclusions and future work are discussed in Section 8.

2   Qualitative Modelling and Simulation

The functionality presented in this paper is implemented in the DynaLearn[3] Interactive Learning Environment (ILE) [4] (an evolution of Garp3[4] [5]), which implements a diagrammatic approach to modelling and simulating qualitative models.

DynaLearn allows modellers to capture their knowledge about the structure and the important processes governing their system of interest. Generic knowledge about processes, such as how a process causally affects quantities and when it is active, are represented in *Model Fragments* (MFs). MFs incorporate other model ingredients as either conditions or consequences, and thus form a rule that, for example, indicates that if a population has a biomass above zero, the production will increase the biomass, while the mortality will decrease the biomass (Figure 2(b)).

QR models can be simulated based on a *scenario*, which represents an initial situation of the system (i.e. a particular variant of the system and a set of initial values for its quantities). The result of the simulation is a state graph in which each state represents a qualitatively unique state of behaviour (i.e. the current structure of the system and quantities with particular values). The transitions represent how the system can change from one state of behaviour to others. To perform the simulation, MFs are sought that match the scenario (i.e. the model ingredients fulfil the conditions of the MF). The consequences of matching MFs are merged with the scenario to create an augmented state from which the next states of behaviour can be determined.

Model ingredient definitions, or *domain building blocks*, are instantiated in MFs and scenarios, and are of particular importance for this paper. These definitions include entities, agents, assumptions, configurations, quantities, and quantity spaces. *Entities* define the concepts with which the structure of the system is described, e.g. environment and population. Entities are organized in a taxonomy. Figure 2(a) shows an entity hierarchy. *Agents and assumptions* are also defined in taxonomies. Agents represent influences from outside the system (when a modeller decides these are not part of the system). Assumptions represent simplifying or operating assumptions about the system, such as the assumption that resources for primary producers is considered constant. *Configurations* define relationships with which the structural relations between entities are described. They are defined by their name (e.g. part of, contains, lives in). *Quantities* represent the features of entities and agents that may change during simulation, and are defined by

---

[3] http://www.dynalearn.eu

[4] http://www.garp3.org

their name and a set of possible quantity spaces. *Quantity spaces* represent the possible values a magnitude (or derivative) of a quantity can have, and are defined by their name and an ordered set of possible values.

Next to the model ingredients defined by the modeller, there is also a set of predefined model ingredients called *generic building blocks*. These include causal relationships, correspondences, the operator relations plus and minus, value assignments, and inequalities.
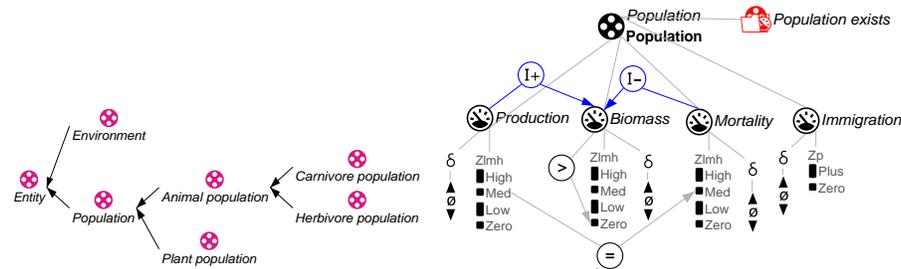
(a) The entity hierarchy of the plant growth resource model.

(b) The Population growth model fragment (from the plant growth model) incorporates the Population Exists model fragment (indicated by the folder with content icon) describing the population, its four quantities, and the inequalities. The model fragment introduces the production (I+) and mortality (I-) influences.

Fig. 2. The entity hierarchy and a model fragment of the *model of plant growth based on exploitation of resources* [27].

3 Export of QR models into OWL

To ease the ontology-based definition of QR models and its later semantic grounding, they are exported [24] to the Web Ontology Language (OWL) [20]. To determine how the QR models can be formalised as ontologies, an ontological perspective on QR is taken. Previous research distinguishes different types of ontologies based on the type of ontological commitments they make [29]. For example, the ontological commitments of a knowledge representation language consist of the domain independent concepts. However, a domain model created by a knowledge engineer using such a language defines new concepts based on the concepts in the knowledge representation language. We frame the QR knowledge representation on these different types of ontologies (Figure 3).

OWL provides the *representational ontology* we use to define the general model ingredients that can be used in a QR model (i.e. the QR vocabulary). We call the formalisation of the QR vocabulary in OWL the *DynaLearn QR ontology*[5]. This ontology

---

[5] http://staff.science.uva.nl/~jliem/ontologies/QRvocabulary.owl

defines the *generic building blocks* (e.g. the concepts entity, configuration and different kinds of causal relations and inequalities) that can be used in a QR model. The DynaLearn QR Ontology functions as our *generic ontology* that extends the ontological commitments made by OWL.

When modellers create QR models, they extend the QR vocabulary by defining domain specific model ingredients, called *domain building blocks*, such as entities, configurations, and quantities. Creating such a domain specific vocabulary can be seen as refining some of the generic building blocks in the generic ontology to define a *domain ontology*. Note that the domain building blocks correspond to the *model ingredient definitions* (Section 2). The generic building blocks in the DynaLearn QR ontology, and the domain building blocks in the QR model ontology (which are all represented as classes) are instantiated in model fragments and scenarios to represent specific situations and processes. We refer to these ontologies as *QR model ontologies*. The QR model ontologies refer to concepts in the DynaLearn QR ontology. In the rest of this paper, when we use the word *QR model* to refer to QR model ontologies.
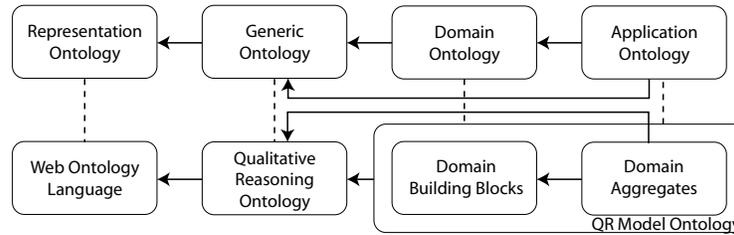
Fig. 3. Correspondences between the QR ontologies and ontology types based on the type of ontological commitments made.

4 Semantic Grounding

The text above details how QR models can be represented in terms of an ontological language. The next step is to link the unrestricted terminology utilized by users in the QR models into well defined external vocabularies. We refer to this process as *grounding*. Technically speaking, this is performed by an *anchoring* [1] process which connects model concepts to one or more equivalent concepts in a background knowledge ontology (or network of ontologies).

4.1 Grounding Process

From a user perspective, the grounding process follows a *semiautomatic* approach: for a given model term, a list of candidate ontology terms (representing the possible meanings of the model term) is automatically proposed to the user. Such a list is ranked according to the probability of being the right meaning. Then, the user can accept the

first proposed ontology term (the most probable one) or may choose another one in the list, and move to the next model term to ground it.

In order to save time and effort, a more automatic way of operating is allowed, called *whole model grounding* (see Figure 4). This way, the whole model is grounded at once, and only the most probable grounding of each term is shown to the user (separated by types of model ingredients: entities, quantities, etc.). If the user is not satisfied with some default grounding, he/she can ask for other proposals and the whole list of candidate senses is shown.
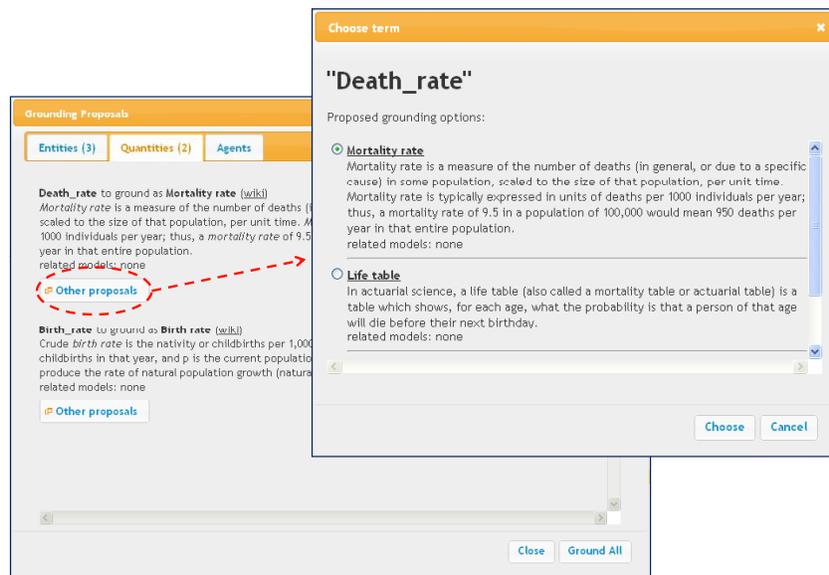
Fig. 4. Example of model grounding (left). When the user asks for alternative groundings for the term "death rate", the window on the right appears.

In case that the term to be grounded is not well covered by the proposed groundings (the user is not satisfied, or no sense was found), two actions are possible:

1. We obtain from WordNet [25] syntactic variations of the initial word, as well as approximate forms coming from Yahoo Spelling Suggestion Service[6]. These alternative terms are offered to the user for grounding, thus increasing the range of possibilities.
2. The user can insert the "ungroundable" term anyway, hence generating a new ontology term that is added into an ontology of *anchor terms*. This way, the information

---

[6] http://developer.yahoo.com/search/web/V1/spellingSuggestion.html

is not lost and can be proposed for future groundings jointly with the other background ontology terms. The anchor terms may be related afterwards to terms in other ontologies (by other domain experts).

Different algorithms can be applied for ranking the list of candidate senses, taking into account the context where the model term appears (surrounding terms in the model) to determine the probability of being the right sense [19]. In our approach, the system promotes the reuse of already utilized groundings, which are shown first. A list of synonyms is maintained in the system (fed by the information accessible in the background ontologies, e.g., `rdfs:label`). This is used for expanding the list of candidate senses (when searching for a term we can also search for their synonyms).

The system proposes by default the use of DBpedia [2] as the main knowledge source to support the grounding process, though it can be complemented by the use of other particular domain vocabularies. The choice of DBpedia as preferred source of knowledge in our system is supported by the results of experimenting with several sources of knowledge (see Section 6).

When the user confirms the grounding, we use the `owl:sameAs` construct for linking the model term with the background ontology term. The generated statement is stored jointly with the model. Finally, the grounded models (as well as the generated ontology of anchor terms) are stored in a *semantic repository*[7], where they remain accessible to the modelling tool for its later reuse (and to any other system interested in reusing the knowledge contained in the stored models)[8].

4.2   Benefits of Grounding in QR

By grounding a model, we are able to bridge the gap between the loosely and imprecise terminology used by a modeller and the well-defined semantics of an ontology. This facilitates interoperability among models or model fragments. Benefits following from this include:

1. In an educational context, a teacher might restrict the vocabulary used by the learner to the knowledge contained in a certain domain ontology, thus speeding up the training period required to learn that vocabulary.
2. New knowledge can be inferred using standard semantic reasoning techniques. For example, let us suppose that entities *whale* and *mammal* in a QR model are grounded to equivalent terms of the same background ontology. If this ontology asserts that *whale* is a subclass of *mammal*, then the same relationship can be inferred for the entities in the model. Other relations not explicitly declared in the model can be also inferred (such as *whale* is an *animal*).
3. Inconsistencies and contradictions between models can be more easily detected. Besides semantic inconsistencies (which can be discovered by applying a reasoner), other modelling issues can be detected. For example, suppose that a model asserts that the increasing *size* of a *population* increases the demand of *natural resources*

---

[7] Based on Jena semantic framework (http://jena.sourceforge.net/) in our current prototype.
[8] A set of web services has been developed to support the communication between repository and modelling tool.

of such a population, while another model establishes the opposite effect, that is, a growing *size* would decrease the demand of *natural resources*. If we are able to establish that both models are referring to the same concepts (*size, population, natural resources, etc.*), the contradiction between the shared concepts can be discovered and pointed out.

4. Additional knowledge and resources can be incorporated to the system. For example, DBpedia contains rich multilingual textual descriptions, links to pictures and web pages, etc. as part of a term description. This information can be imported if the term is grounded on that knowledge source, and shown to the user in the modelling tool.

Most of the previous features are exploited in our system for enabling knowledge-based feedback, as we will see in the following section.

5 Ontology Based Feedback

As aforementioned, the repository of semantically grounded models created in our system is intended to support feedback during the model creation process. For such a purpose, we devise the use of ontology matching techniques [16], semantic reasoning, and QR specific comparisons between models. Depending on the particular technique, our system provides different types of feedback (see Figure 5). Notice, however, that these types of feedback are not mutually exclusive and can be combined.
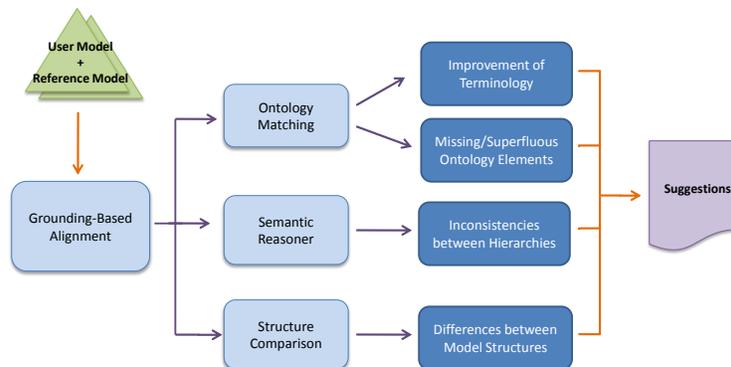
Fig. 5. Data flow diagram of the ontology-based feedback techniques.

The input to the feedback process is a pair of QR models: one corresponding to the user model (under construction) and other corresponding to a reference model made by an expert and already stored in the repository (we do not enter here in the particular technique utilized for selecting the reference model, i.e., if manually or if based on a dynamic evaluation of relevance).

The first step in the process is to derive mappings from the shared groundings. Since the concepts of both models are grounded to a common vocabulary, we can use

these relations to infer a preliminary set of mappings. For example, let us suppose that the user model has a concept labelled *Death* that is grounded to the DBpedia term *Mortality rate*[9], and the reference model has a concept labeled *Mortality* that is also grounded to the same DBpedia resource *Mortality rate*. In order to allow later inference, we determine that *Death* and *Mortality* are equivalent terms (expressed using `owl:EquivalentClass`). The next steps in the process depend on the particular technique:

Ontology Matching. The set of mappings inferred from the shared groundings are utilized, jointly with the user and reference models, as input of an ontology matching tool [18], to generate more pairs of equivalent terms. This enriched set of mapped terms are used to give two types of feedback:

Improvements of terminology. Two terms that have been deemed equivalent in the ontology matching process should share the same label and grounding. By comparing the user terms with their equivalent reference terms we are able to detect these differences and suggest a better option to the user. As an example of this, if a user has an entity labelled *Sustainable biomass* but the equivalent term in the reference model has the label *Carrying capacity*, the system suggests to the user the replacement of the current label by the one used in the reference model.

Missing or superfluous ontological elements. In this type of feedback, we use the set of mappings to find missing ontological elements in the user model, as well as elements that might be not necessary. The concepts of the reference model that have no equivalence in the user model are suggested to the user in order to enrich the model. On the other hand, those user terms with no equivalence in the reference model might be superfluous and hence proposed to be removed from the model.

Semantic Reasoning. We create a temporary ontology by mixing both the user and reference models with the set of previously found equivalences. Then, a semantic reasoner is applied[10] to detect *inconsistencies between hierarchies*. For example, let us suppose that the user model defines *whale* as subclass of *fish*. However, the reference model states that the equivalent term *whale* is subclass of *mammal*, and *mammal* and *fish* have been declared as disjoint classes. Therefore, these two statements are inconsistent. The system informs about this situation, so that the user can review the hierarchy and change it accordingly.

Structure Comparison. Besides ontology-based comparisons, we also exploit the particular semantics of the QR vocabulary to perform more QR-specific comparisons between the models. In fact, we can identify common model structures that are present in the reference model but not in the user model, thus revealing the *differences between model structures*. These missing structures can modify the final behaviour of the model. To detect them, we make a structural comparison between the models. First, patterns in the reference model are searched; then, by means of the set of mappings, we look for

---

[9] http://dbpedia.org/resource/Mortality_rate

[10] We use Jena built-in reasoner in our current prototype (http://jena.sourceforge.net/inference/), though any other reasoner can be used.

the same patterns in the user model. Once the mappings are established between the elements, the structure comparison process detects that some model structures are missing in the user model. Figure 6 exemplifies this. In the example, an inequality property in *Number of* quantity and the positive influence between the quantities *Birth* and *Number of* are pointed out to allow the user to make the corresponding changes.
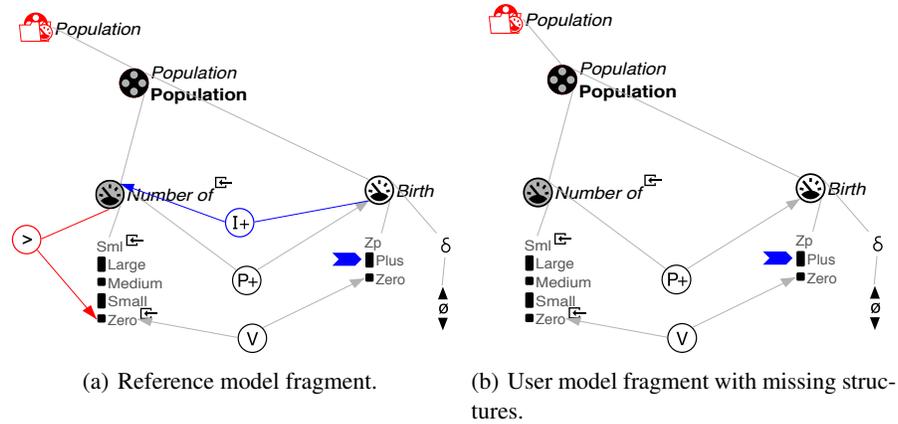

(a) Reference model fragment.   (b) User model fragment with missing structures.

Fig. 6. Example of feedback on missing model structures.

6 Experimental Evaluation

To adequately ground QR model terms in an external vocabulary and be able to explore similarities between models for quality feedback purposes, specific concerns need to be addressed:

- Q1: Are Semantic Web resources suitable for grounding the specific domain vocabularies that QR models typically contain?
- Q2: Are the state of the art ontology matching techniques suitable for mapping QR models?

In this section, we present the description of the experiments carried out to answer our motivating research (and use) questions.

6.1 Grounding Experiments

In order to answer our first question Q1, we performed an experiment to study the *coverage* of different ontologies and semantic resources in specific domains. We measure the coverage as the amount of terms that a resource is able to describe semantically, divided by the total examined terms.

In a realistic usage scenario, the QR models are constructed on the basis of specific domain vocabularies. Therefore, we have focused in our experiment on a set of domain glossaries in environmental science developed by several universities[11]. These vocabularies have been specifically created to be used in the context of QR modelling, so that they constitute very valuable material for our purposes. Each glossary consists of a set of English words which covers seven topics: *Earth systems and resources, the living world, human population, land and water use, energy resources and consumption, pollution*, and *global changes*. We merged these glossaries and removed duplicated terms, obtaining a dataset of 1686 different words.

This unified dataset was used to explore the coverage of knowledge sources of different type: lexical resources such as WordNet [25], common knowledge ontologies such as DBpedia [2] and OpenCyc[12], and the large amount of online ontologies accessible in Watson [13].

The first step of the experiment consisted in searching each word of the input dataset on each of the above external resources[13], obtaining (if available) a set of ontology terms from each resource that semantically describe the input word. Table 1 shows the different coverage degree obtained on each resource[14].

Knowledge Source	Coverage
DBpedia	72%
OpenCyc	69%
WordNet	45%
Watson	47%

Table 1. Coverage of knowledge sources.

The immediate conclusion from the given results is that DBpedia has a better coverage than the other resources, closely followed by OpenCyc, for the utilized domain specific vocabularies.

We have analysed the uncovered cases in the experiment, noticing that most of them corresponded to complex multiword terms (e.g., "cultural habit", "distributed water governance"). There was also a reduced amount of misspelling errors in the glossaries (e.g., "fiter feeding" for "filter feeding") and some terms that do exist in the resource however in other syntactic variation (e.g., "meandering" for "meander"). In such cases (misspelling errors and variations), the grounding could be assisted by services like Yahoo! Spelling Suggestion. In order to measure that effect, we repeated the experiment with DBpedia but searching for alternative suggested forms when the term was not

---

[11] University of Brasília (Brazil), Tel Aviv University (Israel), University of Hull (United Kingdom), Bulgarian Academy of Sciences (Bulgaria), and University of Natural Resources and Applied Life Sciences (Austria).

[12] http://www.opencyc.org

[13] The search mode was "exact match".

[14] The experimental data can be found in
http://delicias.dia.fi.upm.es/~jgracia/experiments/dynalearn/groundingcoverage.html

found in its initial form. The coverage of using DBpedia + Yahoo Spelling Suggestion service raised to a **78%**.

Though the reached 78% coverage indicates that DBpedia covers well the studied domain specific terminologies, other question arises, which is whether these proposed DBpedia groundings are acceptable according to human opinion or not. In order to answer that, we randomly selected 909 terms covered by DBpedia from the same glossaries used in the previous experiments. We asked 8 evaluators (experts in different fields of environmental science) to assess the correctness of the possible meanings given by DBpedia. Each evaluator assessed between 200 and 300 terms. The grounding of each term was double-evaluated.

We counted as positive groundings those terms for which there were at least one suitable meaning among the list of DBpedia results for such a term. We define *accuracy* as the amount of positive groundings divided by the number of evaluated groundings. The obtained average accuracy in the experiment was **83%**. The observed inter-evaluator agreement was 85%, and Cohen's kappa of inter-evaluator agreement [10] was 0.47, which can be considered as "moderate" and gives us an idea of the difficulty of the task (if another expert does not fully agree with me, why a computer should?).

Finally, although DBpedia exhibits a good coverage, we wonder whether it can be further improved with the addition of other resources. Table 2 shows the result of combining DBpedia with the other resources[15]. From the data we conclude that the combined use of OpenCyc and DBpedia increases the coverage significantly (while the further addition of other sources has only a minor effect).

Knowledge Source	Coverage
DBpedia and OpenCyc	87%
DBpedia and Watson	73%
DBpedia and WordNet	72%
DBpedia, OpenCyc, Watson, and WordNet	88%

Table 2. Combined coverage of knowledge sources.

Discussion The results of the grounding experiment show a high coverage degree of DBpedia (78%) when used to ground domain specific terminologies, as well as a high accuracy (85%) of the covered terms according to human opinion. These results support the use of DBpedia as preferred source of knowledge for the grounding of the vocabulary involved in QR modelling. Notice that although the coverage of OpenCyc and DBpedia are comparable (see Table 1), there are other good reasons for choosing DBpedia, such as its multilingual capabilities: DBpedia contains data in up to 92 languages while OpenCyc is monolingual in English, thus reducing its potential usage in a multilingual modelling environment. Nevertheless, the combined use of both ontologies further improves the coverage of English terms up to a 87%, as we have found out empirically.

---

[15] Notice that this time the experiment was run without applying Yahoo Spelling Suggestion service.

6.2 Ontology Matching Experiment

In our approach, we use ontology matching techniques for reconciliating models coming from different authors but modelling similar domains. Thus, the question that we posed above (Q2) emerges naturally. Our target is to study the applicability of already existent alignment approaches in the context of QR modelling.

We have tested the use of two state of the art ontology matching systems: Falcon-AO [21], and CIDER [18] (a complete evaluation of the large amount of alignment systems currently available is out of the scope of this paper). We chose these two systems owing to their good behaviour in previous Ontology Alignment Evaluation Initiative (OAEI) competitions[16] and to their complementary nature: Falcon is more focused on structure-based matching and incorporates efficient partition-based mechanisms, while CIDER relies on a context-based similarity measure and applies lightweight semantic reasoning. The evaluation was conducted as follows:

1. First, a golden standard[17] was defined by human experts[18]. They created specific QR models for this experiment and identified semantic equivalences between them. As result, eight QR models grouped in pairs were created, and a reference alignment file was produced for each pair, containing a total of 85 equivalences between the model terms[19].
2. Then, each ontology aligner system was run separately. Each one received the two QR models as input, and produced an alignment file as result, containing the found correspondences between the two models. Note that the models were not previously grounded, to allow a raw comparison.
3. Finally, each produced alignment was compared to the golden standard, and *precision* and *recall* computed.

In this study we focused on comparing certain types of QR model ingredients: entities, quantities, and configurations (those more closely identifiable as ontology elements). Table 3 shows the experimental result.

	Precision	Recall
CIDER	92%	95%
Falcon	67%	95%

Table 3. Averaged results for the QR Model matching experiment.

Notice that the lower precision given by Falcon is in part owing to the fact that Falcon also aligns the imported ontologies (thus, it aligns the QR vocabulary imported in the models). If the alignment is post-processed and these unnecessary alignments removed, precision also reaches 92%.

---

[16] http://oaei.ontologymatching.org/

[17] It can be found at http://delicias.dia.fi.upm.es/∼jgracia/experiments/dynalearn/omtechniques.html

[18] Three researchers in biological science from University of Brasília (Brazil).

[19] We expressed the alignments in the Alignment Format [15], to facilitate their later processing.

Discussion Although these results are not conclusive, they are indicative that the use of traditional ontology matching techniques perform well for giving the similarity between QR models. Only minor adaptation were required to reuse such techniques for the purposes described in this paper. Notice that the experiment is not intended to study the differences between the compared matching tools, but to determine whether these systems are suitable for the proposed task or not (a question that we have answered positively).

7 Related Work

To our best knowledge, the approach described in this paper has no counterpart in the field of qualitative modelling [5, 17, 28, 14]. Other modelling and simulation tools, such as Betty's brain [23] or Stella [11] neither ground terms to a common vocabulary, nor get quality feedback from other models.

With regard to other conceptual modelling techniques beyond QR, CmapTools [9] is a software for representing knowledge using such as *concept maps* [26]. Although CmapTools is also intended for collaborative use, it neither relies on Semantic Web standards to maximize its interoperability, nor uses common shared vocabularies to minimize the semantic gap between models. In [8] a method for suggesting concepts during concept map modelling based on Web mining techniques is proposed, though an effective grounding with external concepts is not performed.

Regarding the use of semantic techniques to enhance collaborative learning (one of the goals of DynaLearn), there have been some specific efforts, such as the work described in [22], where DEPTHS (Design Patterns Teaching Help System) is introduced. DEPTHS system establishes a common ontological foundation for the integration of different existing learning tools and systems in a common learning environment. Though our techniques differ, the motivation is along the same lines as our work. Nevertheless, DEPTHS focuses on the particular scenario of software engineering education, and supports recommendation more than quality feedback.

8 Conclusions and Future Work

In this paper we propose a method for the semantic enhancement of the modelling process in the field of Qualitative Reasoning. Our goal is to support the creation of semantically networked models as a means to share and reuse conceptual knowledge. In our approach, QR models are first exported into an ontological language and grounded to an external common vocabulary. Then, ontology matching techniques are used for getting quality feedback, by identifying pieces of common knowledge across models.

Our experiments show that the coverage of DBpedia, when used for grounding specific domain terminologies, is above other studied resources. We have also shown that 85% of the covered terms in DBpedia contain suitable meanings according to human opinion. Finally, our experiments indicated a good behaviour of the state of the art ontology matching systems when applied for the alignment of QR models.

As future work, we will especially focus on running usability studies on our ontology-based feedback functionalities. We will also enrich our ontology matching-based techniques with other advanced metrics that measure the agreement and disagreement degree between ontologies [12]. We devise also the application of collaborative filtering techniques for model recommendation based on the community of users of the system. We also plan to specifically use our system in the academic domain to support semantic-guided learning. In this real usage context, more human-based evaluations will be possible for the improvement of our approach.

Finally, although the semantic data that we generate is accessible on the Web by different means, a natural improvement of our system is to "open" this semantic data to the web of Linked Data, by adhering to the Linked Data principles. This will be favoured also by the preferred use of DBpedia (one of the most important nodes in the cloud of Linked Data [2]) in our system.

References

1. Z. Aleksovski, M. C. A. Klein, W. ten Kate, and F. van Harmelen. Matching unstructured vocabularies using a background ontology. In *EKAW*, volume 4248 of *LNSC*, pages 182–197, October 2006.
2. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, September 2009.
3. B. Bredeweg and K. Forbus. Qualitative modeling in education. *AI Magazine*, 24(4):35–46, 2004.
4. B. Bredeweg, J. Liem, F. Linnebank, R. Bühling, M. Wißner, J. Gracia, P. Salles, W. Beek, and A. Gómez-Pérez. Dynalearn: Architecture and approach for investigating conceptual system knowledge acquisition. In *Intelligent Tutoring Systems*, volume 6095, chapter 42, pages 272–274. Springer, Berlin, Heidelberg, 2010.
5. B. Bredeweg, F. Linnebank, A. Bouwer, and J. Liem. Garp3 - workbench for qualitative modelling and simulation. *Ecological Informatics*, 4(5-6):263–281, 2009.
6. B. Bredeweg and P. Salles. Qualitative models of ecological systems – editorial introduction. *Ecological Informatics*, 4(5-6):261 – 262, 2009. Special Issue: Qualitative models of ecological systems.
7. B. Bredeweg and P. Struss. Current topics in qualitative reasoning (editorial introduction). *AI Magazine*, 24(4):13–16, 2003.
8. A. Cañas, M. Carvalho, M. Arguedas, D. B. Leake, A. Maguitman, and T. Reichherzer. Mining the web to suggest concepts during concept mapping: Preliminary results. In *Proc. of the first international conference on concept mapping*, Pamplona, Spain, 2004.
9. A. J. Cañas, G. Hill, R. Carff, N. Suri, J. Lott, G. Gómez, T. C. Eskridge, M. Arroyo, and R. Carvajal. Cmaptools: A knowledge modeling and sharing environment. In *Proc. of the first international conference on concept mapping*, pages 125–133, Pamplona, Spain, 2004.

10. J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, April 1960.

11. R. Costanza and A. Voinov. Modeling ecological and economic systems with stella: Part iii. *Ecological Modelling*, 143(1-2):1–7, October 2001.

12. M. d'Aquin. Formally measuring agreement and disagreement in ontologies. In *Proc. of the 5th International Conference on Knowledge Capture (K-CAP 2009)*, pages 145–152, September 2009.

13. M. d'Aquin, C. Baldassarre, L. Gridinoc, S. Angeletou, M. Sabou, and E. Motta. Characterizing knowledge on the semantic web with Watson. In *5th International EON Workshop, at ISWC'07, Busan, Korea*, November 2007.

14. M. Dehghani and K. Forbus. QCM: A QP-based concept map system. In J. Žabkar and I. Bratko, editors, *the 23nd International Workshop on Qualitative Reasoning (QR09)*, pages 16–21, Ljubljana, Slovenia, June 2009.

15. J. Euzenat. An API for ontology alignment. In *3rd International Semantic Web Conference (ISWC'04), Hiroshima (Japan)*. Springer, November 2004.

16. J. Euzenat and P. Shvaiko. *Ontology matching*. Springer-Verlag, 2007.

17. K. D. Forbus, K. Carney, B. L. Sherin, and L. C. UreelII. Vmodel: A visual qualitative modeling environment for middle-school students. In *Proc. of the 16th Conference on Innovative Applications of Artificial Intelligence, San Jose, California, USA*, pages 820–827, July 2004.

18. J. Gracia and E. Mena. Ontology matching with CIDER: Evaluation report for the OAEI 2008. In *Proc. of 3rd Ontology Matching Workshop (OM'08), at ISWC'08, Karlsruhe, Germany*, volume 431, pages 140–146. CEUR-WS, October 2008.

19. J. Gracia, R. Trillo, M. Espinoza, and E. Mena. Querying the web: A multiontology disambiguation method. In *Sixth International Conference on Web Engineering (ICWE'06), Palo Alto, California, USA*, pages 241–248. ACM, July 2006.

20. B. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patelschneider, and U. Sattler. Owl 2: The next step for owl. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):309–322, November 2008.

21. W. Hu and Y. Qu. Falcon-ao: A practical ontology matching system. *Journal of Web Semantics*, 6(3):237–239, 2008.

22. Z. Jeremic, J. Jovanovic, and D. Gasevic. Semantic web technologies for the integration of learning tools and context-aware educational services. In *8th International Semantic Web Conference (ISWC2009)*, October 2009.

23. K. Leelawong and G. Biswas. Designing learning by teaching agents: The betty's brain system. *Int. J. Artif. Intell. Ed.*, 18(3):181–208, 2008.

24. J. Liem and B. Bredeweg. OWL and qualitative reasoning models. In C. Freksa, M. Kohlhase, and K. Schill, editors, *KI 2006: Advances in Artificial Intelligence. 29th Annual German Conference on AI*, number 4314 in Lecture Notes in Artificial Intelligence, pages 33–48, Bremen, Germany, 2007. Springer-Verlag Berlin Heidelberg.

25. G. A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, November 1995.

26. J. Novak and D. Gowin. *Learning How to Learn*. Cambridge University Press, 1984.

27. T. Nuttle, P. Salles, B. Bredeweg, and M. Neumann. Representing and managing uncertainty in qualitative ecological models. *Ecological informatics*, 4(5-6):358–366, November/December 2009.

28. E. Soloway, A. Z. Pryor, J. S. Krajcik, S. Jackson, S. J. Stratford, M. Wisnudel, and J. T. Klein. Scienceware's model-it: Technology to support authentic science inquiry. *Technological Horizons on Education*, 25(3):54–56, 1997.

29. G. van Heijst, S. Falasconi, A. Abu-Hanna, G. Schreiber, and M. Stefanelli. A case study in ontology library contruction. *Artificial Intelligence in Medicine*, 7(3):227–255, June 1995.