Exploiting Users' Feedbacks

Towards a Task-based Evaluation of Application Ontologies Throughout Their Lifecycle

Perrine Pittet and Jérôme Barthélémy
Articque Software, 149 avenue Général de Gaulle, 37230 Fondettes, France

Keywords: Application Ontology, Task-based Ontology Evaluation, Ontology Revision, Semantic Annotation, Ontology Lifecycle, Crowdsourcing.

Abstract: This paper presents the basis of our approach for evaluation of application ontologies. Adapting an existing task-based evaluation, this approach explains how crowdsourcing, involving application users, can efficiently help in the improvement of an application ontology all along the ontology lifecycle. A real case experiment on an application ontology designed for the semantic annotation of geobusiness user data illustrates the proposal.

1 INTRODUCTION

Ontology development is becoming a common task, which is nowadays not just a matter for ontologists. In the literature many ontology development methodologies have been proposed to help non-experts build their own ontologies such as (Noy and McGuinness, 2001), (Sure et al., 2002), (Sure et al., 2009) and (Suarez-Figueroa et al., 2012). However, according to (Neuhaus et al., 2013), currently, there is no agreement on a methodology for development of ontologies, and there is no consensus on how ontologies should be evaluated.

As said by (Brank et al., 2005), ontology evaluation is the problem of assessing a given ontology from the point of view of a particular criterion of application. It could help ontology developers evaluating their results and possibly guiding the construction process and any refinement steps. According to (Vrandečić, 2009) this would make them feel more confident about their results, and thus encourage them to share their results with the community and reuse the work of others for their own purposes. Though, because of the lack of a consensus, evaluation techniques and tools are not widely utilized in the development of ontologies (Neuhaus, et al., 2013). This can lead to ontologies of poor quality and is an obstacle to the successful deployment of ontologies as a technology.

In this paper we focus on the evaluation of application ontologies. Application ontologies describe the domain of specific applications (Malone and Parkinson, 2010). They are built from scratch to make it stick to applications specific requirements. Consequently a pertinent evaluation consists in assessing their effectiveness against the different tasks they have to solve within the application for which they have been built (Porzel and Malaka, 2004). This evaluation step is crucial to guide their refinement. But in practice it is often skipped. It may be due to the difficulty of distinguishing which part of the application outputs really depends on the ontology itself and not on the application. Also few studies have addressed the evaluation of application ontologies through their specific uses within the application (Brewster et. al., 2004). Therefore we attempt to promote the systematic effectiveness evaluation of application ontologies by proposing a simple adaptation of the task-based approach of (Porzel and Malaka, 2004) using crowdsourcing all along their lifecycle.

The rest of the paper is articulated in 4 sections. The second section presents a state of the art on application ontologies and ontology evaluation and the related works. The third section presents our proposal and an application experience. The fourth section presents our conclusions.

Pittet, P. and Barthélémy, J..
Exploiting Users' Feedbacks - Towards a Task-based Evaluation of Application Ontologies Throughout Their Lifecycle.
In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 2: KEOD*, pages 263-268
ISBN: 978-989-758-158-8

2 BACKGROUND

This section introduces the related works about application ontologies, ontology evaluation and the deadlocks identified for the evaluation of application ontologies.

2.1 Application Ontologies

According to (Malone and Parkinson, 2010), an application ontology is an ontology engineered for a specific use or application focus and whose scope is specified through testable use cases. Application ontologies usually reuse, derive or reference recognized ontologies to construct ontological classes and relationships between classes (Shaw et al., 2008). According to (Guarino, 1998), this top-down approach promotes the reuse of ontologies. However, in practice, building reusable ontologies is a costly process. Consequently a frequent alternative consists in building application ontologies from scratch and then generalizing them to domain and task ontologies (bottom-up approach). For instance a bottom-up approach called Goal-Oriented Application Ontology Development Technique, presented in (Santos et al., 2013), has been designed to guide the development of application ontologies from the explicit specification of their application goals translated into rules and facts. Generally, both top-down and bottom-up approaches require well defining the tasks the application ontology has to solve through the application. These tasks can be multiple (conceptual similarity calculation, disambiguation, knowledge extraction, semantic annotation, etc.) but are generally closely related to the application processes.

2.2 Ontology Evaluation

According to (Gómez-Pérez, 2001), ontologies, such as any other resources used in software applications, should be evaluated before (re)using it in other ontologies or applications. Evaluation of the ontology content, i.e. its concepts definitions, its taxonomy and its axioms, as well as evaluation of the software environment are therefore critical before integrating them in final applications. Many ontology evaluation methodologies have been proposed since 1995, among those, the well-known works of (Gómez-Pérez, 1995) and (Guarino and Welty, 2002). In 2005, (Brank et al., 2005)'s survey identified the main ontology evaluation approaches types: those based on comparing the ontology to a *gold standard* (Maedche and Staab, 2002), those based on using the ontology in an application and evaluating the results (Porzel and Malaka, 2004), those involving comparisons with a source of data about the domain to be covered by the ontology (Brewster et al., 2004), and those where evaluation is done by humans (Lozano-Tello and Gómez-Pérez, 2004). In addition, authors grouped these approaches based on the level of evaluation: vocabulary level, taxonomy level, semantic relations level, context or application level, syntactic level, and structure, architecture and design level evaluation. They are all suited for the three first levels but only human-based evaluation can cover the three other levels. Several issues are still addressed today, such as the need to have a detailed methodology to allow performing evaluation throughout the entire ontology lifecycle (Staab and Studer, 2013). Also evaluation of application ontologies has been little addressed (Malone and Parkinson, 2010).

2.3 Application Ontology Evaluation

Authors of (Malone and Parkinson, 2010) state that application ontologies should be evaluated against a set of use cases and competency questions, which represent the scope and requirements of the particular application. This approach, called application-based evaluation, is founded on the fact that the outputs of the application or its performance on a given task might be better or worse depending partly on the ontology used in it. According to (Brank et al., 2005), one might argue that a good ontology is one, which helps the application in question produce good results on the given task. However authors identify some drawbacks concerning application-based evaluation approaches: (1) as an ontology is good or bad considering a particular task, it is difficult to generalize the approach, (2) if the ontology is only a small component of the application, its effect on the output may be relatively small and indirect, (3) comparison of different ontologies cannot be handled if they cannot all be plugged in the same application. In (Vrandečić, 2009) the author minimizes these drawbacks by stating that ontologies are often tightly interwoven with an application, and, that the user never accesses an ontology directly but always through this application. Therefore the application often needs to be evaluated with the ontology, regarding the ontology as merely another component of the used tool. He adds that such a situation has the advantage that well-known software evaluation methods can be applied.

2.4 The Task-based Approach

One of the major works addressing application-based ontology evaluation is the task-based approach of (Porzel and Malaka, 2004). This approach provides an evaluation scheme on three basic ontological levels: vocabulary (concepts), taxonomy and semantic relations. For these three levels, the authors define three shortcomings the evaluation results have to show: insertion, deletion and substitution errors. Insertion errors indicate superfluous concepts, isa- and semantic relations, deletion errors indicate missing ones and substitution errors indicate off-target ones. Then given appropriate tasks and maximally independent algorithms operating on the ontology in solving these tasks, and, given a *gold standard*, this approach allows calculating the error rates corresponding to specific ontological shortcomings at each ontology level. A *gold standard* is the set of "perfect" outputs the task is expected to provide on the corpus on which it is run. Therefore a *gold standard* must be defined for each task as such as the explicit definition of the tasks, the ontology, and the application. A task is required to be sufficiently complex to constitute a suitable benchmark for examining a given ontology. The performance results must substantially depend on the way relations are modelled within the ontology. One ontology is sufficient as it is evaluated in terms of its performance on a given task within the application. The application is defined as a specific algorithm that uses the ontology to perform the task. However the algorithm can have more or less influence in the application output and it is sometimes difficult to distinguish which of the algorithm or the ontology is in cause. Therefore to obtain meaningful performances results, the algorithm must be kept constant within the ontology evaluation/revision cycle. Once all these components are defined, the evaluation can be launched. Then the revision phase of the ontology simply consists in undertaking the changes corresponding to the to the identified errors before running another evaluation round.

In the next section, we propose an adaptation of this approach taking into account user contributions to improve the evaluation accuracy all along the ontology use.

3 PROPOSAL

This section presents the basis of our proposal and illustrates it through a real case experiment.

3.1 A Task-based Evaluation through Ontology Lifecycle

Our approach consists in adapting the task-based approach of (Porzel and Malaka, 2004) by delegating the evaluation job to the users, using the application as a crowdsourcing platform, instead of using a *gold standard*. In (Porzel and Malaka, 2004), the *gold standard* is actually produced by means of annotators (trained by humans) agreeing on mutually satisfactory solutions for the cases of disagreement. Therefore its accuracy clearly depends on the performance of these annotators. Besides, if the application domain changes and the ontology needs to evolve, a new *gold standard* has to be produced, which is not convenient to evaluate an ontology during its entire lifecycle. Instead, a crowdsourced evaluation, in which users participate, can be realized all along the application lifecycle. Crowdsourcing is defined in (Hosseini et al., 2014) as a business model, where tasks are accomplished by a general public, called the crowd. This model has recently been promoted for the domain of information systems analysis and design namely through the involvement of users in evaluating the software (Ali et al., 2012) (Pagano and Brügge, 2013). In our approach, we consider the application users as the crowd, the evaluation procedures as the crowdsourcing tasks and the application as the crowdsourcing platform.

Our task-based evaluation methodology comprises the 6 following steps:

1. Definition of the application used as a crowdsourcing platform: distinction between tasks driven by the ontology and other tasks,
2. Definition of the (consistent) ontology: domain and scope, role within the application, the level of contribution regarding each task it drives,
3. Definition of the tasks driven by the ontology.
4. For each task, choice of the ontology levels to evaluate: vocabulary, taxonomy, semantic relations.
5. For each ontological level, definition of the corresponding error types: deletion, addition, and substitution errors.
6. For each task to evaluate, definition of the crowdsourcing task: explicit process each user participating in the evaluation has to follow, guidelines for the qualification of the tasks outputs and their evaluation w.r.t. error types, indications for potential revision. Within this step users can contribute to the ontology revision by proposing a change (ex: addition of a missing

concept, etc.). Therefore if a majority of users agrees on a proposed revision, it can be translated into ontological change and applied on the ontology. Then a reasoner can assess the ontology consistency after the application of the change in order to decide to commit or rollback it. Inversely if a consensus cannot be reached, the different points of view can be taken into account by applying each suggestion of change one by one until the ontology is no more consistent.

We believe this approach has several advantages. First, the ontology effectiveness is tested in the real production environment by real users on real data. Second, evaluation can be done during the application use if the users are given the ability of revising the tasks outputs; then a decision algorithm can be implemented to assess the user suggestions and translate them into ontological changes (Klein, 2004). Third as users point of view on the application domain can change over time, they may revise application outputs they used to consider correct. In this case the ontology is able to evolve and stay up-to-date with the application community of users all along its lifecycle.

3.2 Experience of Semantic Annotation Task Evaluation

Here we describe the application of this methodology on a real case experiment.

1. The application considered is the SaaS geo-business decision software, CD7Online, within which users build maps by processing and representing statistical data on geographical basemaps stored in their workspaces.
2. The ontology considered describes the CD7Online application domain including descriptive statistics data files (tables of data), geographical data files (base maps) and maps projects (organization charts), but also the platform related business processes and uses. It is specified with the OWL DL language and contains about 10000 axioms.
3. The ontology has been developed to drive a task of semantic annotation, supporting a recommender system suggesting users relevant data and processes. This task allows extracting metadata such as geographical level, year, theme, statistical indicator, etc., from the user data workspaces and to represent their relations with user data within RDF annotations. Built on the top of a triplestore containing these annotations a visualization tool, available in the form of a graphical 2D interactive graph (cf. Fig. 1) allows users intuitively browsing their own workspaces, navigating through the annotations, starting from their "Home" (cf. “Mes cartes et mes données” on Fig.1 step 1). We chose to use this existing tool to establish and perform the evaluation procedure.
4. The evaluation Level: Like in (Porzel and Malaka, 2004) performance of the ontology can be evaluated at the semantic relation level as annotations are semantic relations between data and metadata instantiated from the ontology model.
5. Three semantic relation error types are defined. A correct annotation corresponds to a correct non-taxonomic relation between a data and a metadata. An annotation with one of the following errors is assessed incorrect: missing annotation (deletion), superfluous annotation (insertion), and wrong annotation (substitution).
6. The evaluation procedure given to the users is articulated in 5 steps: (1) choosing a data to search, (2) navigating in the graph according to the metadata they consider the most related to this data, (3) assessing the accuracy, (4) if not accurate, identifying the error type and eventually (5) proposing a revision. During the process, if the users manage to find it within the first direct path they take within the graph, the corresponding annotation of this data is considered accurate. If they need to use at least another path to find it, it means the annotation corresponding to the expected data is missing, this is a deletion error. If they find the data but if this data is only related to a wrong metadata, this is a substitution error. And if they find an unexpected metadata linked to the data in addition to accurate ones, it is an insertion error.

Fig.1 describes an application example of the process. The first step shows the default graph on which the user can see the metadata extracted from his workspace (cf. Fig.1 step 1). If the user considers his data is related to population, he selects the “th_population” category. Second, the graph morphs to show the statistical indicators related to this category (cf. Fig.1 step 2). Among these, the user chooses the “Femme” indicator (i.e. “Woman”), because the data he is looking for is related to a population of women. Third, the graph morphs to show the statistical data corresponding to this indicator (cf. Fig.1 step 3). The user finds the data “Femmes chômage” (i.e. “Unemployed women”), and validates the annotation.

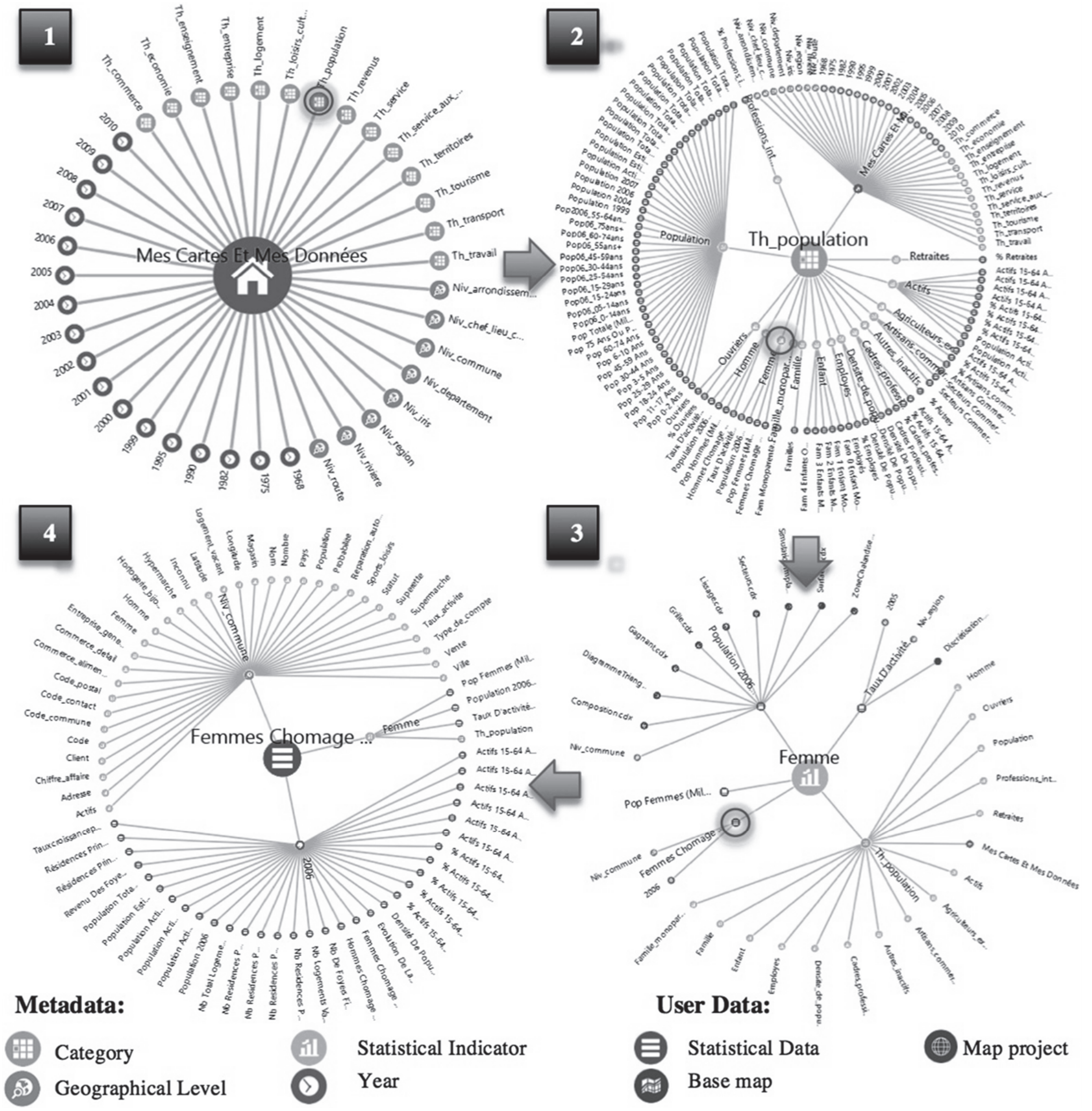

Figure 1: Data search test within the interactive semantic visualization of a working group.

3.3 Results

A first evaluation round, conducted on a sample of ten voluntary users, showed a percentage of 35,2% accurate annotations with statistical indicators obtained on a set of 571 real data series (against 38,4% deletions, 26,5% substitutions, 0% insertions). These results showed the ontology needed to be enriched and refined to obtain better outcomes. After the revision guided by these results, the percentage of accurate annotations resulted from a second evaluation conducted on a sample of ten other users, reached 82,8% of good annotations.

4 CONCLUSIONS

This paper presented a simple task-based ontology evaluation approach adapted from (Porzel and Malaka, 2004)'s methodology. Dedicated to application ontologies, it aims at facilitating evaluation and revision of application ontologies during their entire lifecycle, by delegating these tasks to voluntary users given an explicit procedure. Following (Staab and Studer, 2013)'s recommendation, evaluation can be done all along the ontology lifecycle and fit to the users' evolving points of view. Also first results of the experiment on a semantic annotation task, conducted on small samples of voluntary users, are promising. Today we are working on extending this evaluation experience to a true crowdsourcing one, by opening it to all users of CD7 in order to assess the relevance of the approach across the application lifecycle. The next step will consist in automating the inclusion of users revision suggestions by implementing a decision algorithm and translating the results to ontological changes. A future step would be to generalize this approach on different types of tasks in order to establish template procedures.

REFERENCES

Ali, R., Solis, C., Omoronyia, I., Salehie, M., & Nuseibeh, B. (2012). Social adaptation: when software gives users a voice.

Brank, J., Grobelnik, M., & Mladenic, D. (2005, October). *A survey of ontology evaluation techniques*. In *Proceedings of the conference on data mining and data warehouses (SiKDD 2005)* (pp. 166-170).

Brewster, C., Alani, H., Dasmahapatra, S., & Wilks, Y. (2004). Data driven ontology evaluation.

Gómez-Pérez, A. (1995, February). Some ideas and examples to evaluate ontologies. In *Artificial Intelligence for Applications, 1995. Proceedings, 11th Conference on* (pp. 299-305). IEEE.

Gómez□Pérez, A. (2001). Evaluation of ontologies. *International Journal of intelligent systems*, *16*(3), 391-409.

Guarino, N. (1998). *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy* (Vol. 46). IOS press.

Guarino, N., & Welty, C. (2002). Evaluating ontological decisions with OntoClean. *Communications of the ACM*, *45*(2), 61-65.

Hosseini, M., Phalp, K., Taylor, J., & Ali, R. (2014, May). The four pillars of crowdsourcing: A reference model. In *Research Challenges in Information Science (RCIS), 2014 IEEE Eighth International Conference on* (pp. 1-12). IEEE.

Klein, M. C. A. (2004). *Change management for distributed ontologies*.

Lozano-Tello, A., & Gómez-Pérez, A. (2004). Ontometric: A method to choose the appropriate ontology. *Journal of Database Management*, *2*(15), 1-18.

Maedche, A., & Staab, S. (2002). Measuring similarity between ontologies. In *Knowledge engineering and knowledge management: Ontologies and the semantic web* (pp. 251-263). Springer Berlin Heidelberg.

Malone, J., & Parkinson, H. (2010) *Reference and Application Ontologies*. *Ontogenesis*. http://ontogenesis.knowledgeblog.org/295.

Neuhaus, F., Vizedom, A., Baclawski, K., Bennett, M., Dean, M., Denny, M., & Yim, P. (2013). *Towards ontology evaluation across the life cycle. Applied Ontology*, *8*(3), 179-194.

Noy, N. F., & McGuinness, D. L. (2001). *Ontology development 101: A guide to creating your first ontology.*

Pagano, D., & Brügge, B. (2013, May). *User involvement in software evolution practice: a case study. In Proceedings of the 2013 international conference on Software engineering (pp. 953-962). IEEE Press.*

Porzel, R., & Malaka, R. (2004, August). *A task-based approach for ontology evaluation*. In *ECAI Workshop on Ontology Learning and Population, Valencia, Spain*.

Santos, L. E., Girardi, R., & Novais, P. (2013, April). A Case Study on the Construction of Application Ontologies. In *Information Technology: New Generations (ITNG), 2013 Tenth International Conference on* (pp. 619-624). IEEE.

Shaw, M., Detwiler, L. T., Brinkley, J. F., & Suciu, D. (2008). Generating application ontologies from reference ontologies. In *AMIA Annual Symposium Proceedings* (Vol. 2008, p. 672). American Medical Informatics Association.

Staab, S., & Studer, R. (Eds.). (2013). *Handbook on ontologies*. Springer Science & Business Media.

Suarez-Figueroa, M. C., Gomez-Perez, A., & Fernandez-Lopez, M. (2012). The NeOn methodology for ontology engineering. In *Ontology engineering in a networked world* (pp. 9-34). Springer Berlin Heidelberg.

Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., & Wenke, D. (2002). *OntoEdit: Collaborative ontology development for the semantic web* (pp. 221-235). Springer Berlin Heidelberg.

Sure, Y., Staab, S., & Studer, R. (2009). Ontology engineering methodology. In *Handbook on ontologies* (pp. 135-152). Springer Berlin Heidelberg.

Vrandečić, D. (2009). *Ontology evaluation* (pp. 293-313). Springer Berlin Heidelberg.