2011 10th Mexican International Conference on Artificial Intelligence

CreaDO – A Methodology to Create Domain Ontologies using Parameter-based Ontology Merging Techniques

Sabino Pariente Juárez, Hugo Estrada Esquivel, Alicia Martínez Rebollar
Departamento de Ciencias Computacionales, Centro Nacional de Investigación y Desarrollo Tecnológico (cenidet)
Cuernavaca, Morelos, México
e-mail: spariente, hestrada, amartinez@cenidet.edu.mx

Mari Carmen Suárez-Figueroa
Ontology Engineering Group (OEG)
Facultad de Informática, Universidad Politécnica de Madrid
Campus de Montegancedo, Boadilla del Monte, Madrid, España
e-mail: mcsuarez@fi.upm.es

Abstract: **Nowadays, ontologies have become a key mechanism to represent the knowledge of a specific domain. Domain ontologies can be used for different purposes; one of them is the development of semantic search engines that obtain precise results by considering the meaning of the Web content. The construction of these ontologies usually requires a large amount of effort and time to be completed. One way of reducing such effort and time is using the reuse approach. When ontologies overlap, alignment techniques to merge such ontologies can be applied. However, the result of the ontology merging activity can be a very large ontology, difficult to understand and use. To overcome this problem, this paper describes a methodology, called CreaDO, to semi-automatically create domain ontologies. CreaDO focuses on performing a parameter-based ontology merge that allows the creation of a domain ontology only with relevant information for the ontology purpose.**

Keywords- Domain Ontology, Ontology Merging, Creation of Domain Ontology, Semantic Web.

I. INTRODUCTION

Nowadays, ontologies have become a key mechanism in different contexts where the knowledge representation is needed. In the Semantic Web, ontologies allow us to represent the knowledge included in Web sites and its corresponding resources. In this context, it is important to point out that the ontologies for a specific Web site usually only represent a fragment of the knowledge of the domain. This is because, a Web site does not cover all the knowledge of a complete domain, but it usually covers a specific topic. Therefore, we can argue that by means of joining all Web sites of a domain (such as health, tourism, government, etc) we can obtain an almost complete knowledge of the domain that can be represented using a domain ontology.

Domain ontology can be used as knowledge sources by semantic search engines to perform more precise searches when a user asks for information concerning a particular domain. This is a reason for which the construction of domain ontologies is an important issue in current Semantic Web approaches.

Building domain ontologies is a process that normally requires large amount of effort and time. This is because both domain experts and ontology engineers are needed in the building process: The first ones in order to provide their knowledge about the domain and the second ones to build the ontology using the knowledge provided by the domain experts. To reduce the effort and time, a reuse approach (e.g., based on merging techniques) [1], [2], [3] and/or a semi-automatic approach [4], [5] can be used.

d'Aquin [6] stated that "*the amount of knowledge published on the Semantic Web (i.e., the number of ontologies and semantic documents available on-line) is rapidly increasing, having reached the critical mass required to enable the vision of a truly large scale, distributed and heterogeneous web of knowledge*". In this context, we consider ontology reuse techniques as a solution to reach the goal of producing a heterogeneous corpus of knowledge by joining fragments of lightweight ontologies included in Web sites. In this reuse context when two or more ontologies overlap, ontology merging techniques can be used in order to build a domain ontology. This is performed through the join of the knowledge represented in lightweight ontologies. However, the result of ontology merging can be very large, which makes the merged ontology difficult to understand and use by ontology practitioners. To overcome this problem, this paper describes a methodology, called CreaDO, to semi-automatically create domain ontologies. CreaDO Methodology grounds on a parameter-based ontology merging technique that allows the creation of a domain ontology. Such a domain ontology only contains the relevant information for a specific purpose. To do this, the CreaDO Methodology uses a merge parameter that represents a concept of the domain that will be represented in the new ontology. This parameter is used to drive the knowledge selection in the domain ontology creation process. The parameter will help to ignore the irrelevant knowledge and preserve the relevant knowledge for the ontology purpose.

The rest of this paper is organized as follows. Section 2 presents an overview of the previous work related to our proposal. Section 3 describes the CreaDO Methodology used to create domain ontologies using a parameter-based ontology merging technique. Finally, section 4 closes our paper with the conclusions of our contribution and directions for further work.

978-0-7695-4605-6/11 $26.00 © 2011 IEEE
DOI 10.1109/MICAI.2011.39

II. Related works

This research work involves three main topics: ontology building methodologies, semi-automatic building of domain ontologies, and ontology merging techniques. Related research works in such topics are presented in this section.

A. Ontology building methodologies

This section presents the most well-known methodologies for building ontologies.

The NeOn Methodology [7] for building ontology networks is a scenario-based methodology that supports the collaborative aspects of ontology development and reuse, as well as the dynamic evolution of ontology networks in distributed environments. The key assets of the NeOn Methodology are (a) a set of nine scenarios for building ontologies and ontology networks, emphasizing the reuse of ontological and non-ontological resources, the reengineering and merging, and taking into account collaboration and dynamism; (b) the NeOn Glossary of Processes and Activities, which identifies and defines the processes and activities carried out when ontology networks are collaboratively built by teams; and (c) a set of methodological guidelines for different processes and activities of the ontology network development process

METHONTOLOGY [8] enables the construction of ontologies at the knowledge level. It includes (a) the identification of the ontology development process; (b) a life cycle based on evolving prototypes; and (c) some techniques to carry out management, development-oriented, and support activities.

The On-To-Knowledge methodology [9] proposes to build ontologies taking into account how these are going to be used in knowledge management applications. The processes proposed by this methodology are the following: feasibility study, kickoff, refinement, evaluation, and maintenance.

The DILIGENT methodology [10] is intended to support domain experts in a distributed setting in order to engineer and evolve ontologies. This methodology is focused on collaborative and distributed ontology engineering. Its ontology development process includes the following five activities: building, local adaptation, analysis, revision, and local update.

B. Semi-automatic bulding of domain ontologies

The main goal of the research works in this section is to propose mechanisms to semi-automatic build domain ontologies.

Cristani and Cuel [11] propose a framework that defines a meta-methodology for the creation of a domain ontology without reference to a prior ontology. This method defines five phases to achieve its goal: plan phase, introspective phase, bottom up phase, provision of basic axioms phase, and validation phase.

Dahab and collegues [4] propose a tool called TextOntoEx. This tool is used for the construction of domain ontologies from text in natural language following a pattern-based approach. TextOntoEx extract non-taxonomic relationship in a specific domain from free technical text. TextOntoEx does not discover new relationships, but only instantiates well-known relationships based on the domain and the theme chosen by the user.

Sánchez [5] proposes a method that creates domain ontologies from knowledge defined in the web. The method defines two tasks to achieve its goal: (a) extraction and selection of domain related terms, organizing them in a taxonomical way; and (b) discovery and label of no-taxonomical relationships among concepts. . In addition, the method includes guides for improving the final structure.

C. Ontology merging techniques

The main goal of the research works in this section is to propose techniques for the creation of ontologies using ontology merging techniques.

Ganter and Stumme [1] propose a method to create and merge top level ontologies. This method is suited especially for creating and merging the top level of the ontologies, where high accuracy is required, and for supporting the merger of two (or more) ontologies on that level.

Stumme and Maedche [2] propose the method FCA-Merge for merging ontologies following a bottom-up approach which offers a structural description of the merging process. FCA-Merge takes as input data of two ontologies and a set D of natural language documents and generates a set of instances from information contained at the documents set.

Raunich and Rahm [3] propose an approach of target–driven merging of taxonomies. This approach merges a source taxonomy into a target taxonomy. They also discuss how to extend the merge algorithm by providing auxiliary information, such as additional relationships among source and target concepts, in order to semantically improve the final result.

III. CreaDO - Methodology to Create Domain Ontology Using Parameter-Based Ontology Merging Techniques

CreaDO is a novel methodology for building domain ontologies which only contain the relevant knowledge for a specific purpose. The methodology receives as input (a) a set of ontologies called source ontologies (they are obtained from the analysis of single documents), and (b) a merge parameter which is a concept concerning the domain of the source ontologies. As output the methodology provides a domain ontology which represents all the knowledge from the source ontologies related to the merge parameter. Currently, the methodology only works with knowledge represented in English.

The methodology uses techniques related to ontology reuse, such as ontology merging and ontology modularization techniques.

- Ontology merging techniques are used to join consensual knowledge represented in the source ontologies. However, since the ontologies used have been probably built by different people and/or diverse software systems, several problematic issues can be found, for example, using different named patterns, different taxonomies, or even ontologies poorly designed and structured. To solve this specific issue, the methodology includes a method to review ontologies. This method analyzes the source ontologies with the aim of identifying structural and functional errors and extending ontology elements with lexical information. It is important to mention that ontology merging needs to find coincidences among ontologies. To perform this search, a set of mappings rules is required. In the methodology, we define a set of mapping rules and an algorithm for identifying coincidences among entities of multiples ontologies (currently only those coincidences related to equivalent knowledge). The set of ontology mapping rules are defined using the proposal of Ehrig [12]. However, since not all the entities involved in the mappings could be interesting for the ontology purpose, then not all the mappings should be taken into account. For this reason, we define a method to extract a subset of mappings using the merge parameter with the aim of obtaining only entities that are relevant to the ontology purpose.

- Ontology modularization techniques are used in this research to reduce the complexity of the merged ontology with the objective of representing only interesting information for the ontology purpose. To do this, the merge parameter is used to extract ontology modules from each source ontology. The method used to the ontology modularization is an extension of the Doran algorithm presented in [13].

The domain ontology is created by joining the ontology modules extracted by ontology modularization by means of the mappings subset selected on the filtering mappings. Once the domain ontology has been created, a method is applied to evaluate the inconsistencies of the domain ontology and its features.

The CreaDO methodology is composed by six methods, which were briefly mentioned in the aforementioned strategy. These six methods are presented in detail in the following sections. Figure 1 shows the overview of the CreaDO Methodology.

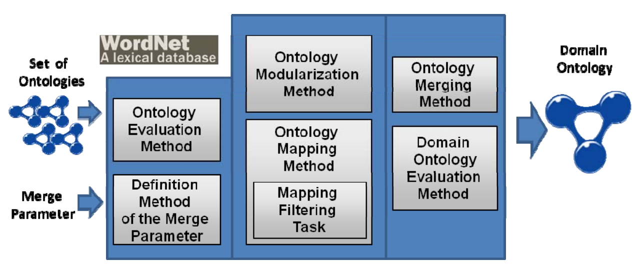

Figure 1. Overview of CreaDO Methodology

In addition, a small case study is presented for each method in order to validate the proposed approach. This case study is conformed of two ontologies about tourism: (a) the OTN ontology[1] and (b) the ETP-Tourism[2]. Figure 2 presents fragments of these ontologies that show its key concepts. The merge parameter used for the case study is the concept "Tourism".

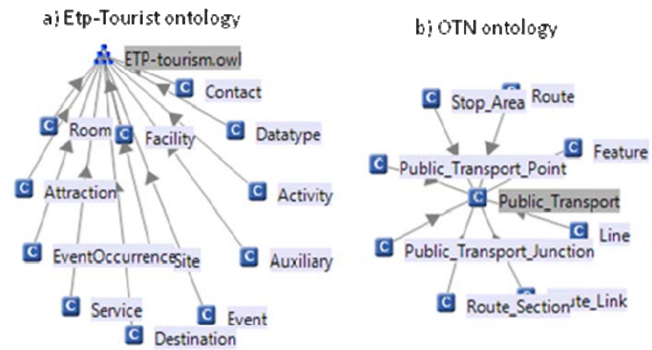

Figure 2. Fragments of the tourism ontologies

A. *Ontology evaluation method*

The ontology evaluation method has the objective of performing an evaluation of the source ontologies in order to identify possible design errors at structural and functional levels. In addition, this method gets lexical information to each class represented into the source ontology. This lexical information provides more knowledge to classes that can facilitate the finding of more correspondences between two classes in two different ontologies. The lexical information we consider in this research is the one related to synonyms; and this information is recovered from WordNet. This lexical information is used to determine the similarity between two concepts and all the synonyms associated to these concepts. With this approach we consider more possibilities than only the one that makes the comparison between two concepts as were originally defined. For example, in our case study the *ETP ontology* contains the classes *Recreation* and *Safari*. The synonym of the class *Recreation* is *Diversion*; and the synonyms of the *Safari* are *Camping*, *Hunting*, and *Expedition*.

The evaluation method proposed here evaluates the ontologies according with some common pitfalls presented in [14]. These common pitfalls may occur in the time of building the ontology or extending it. These common pitfalls refer to the structural, functional and usability-profiling dimensions of the ontology. The method needs to look into each element of each ontology to try to identify the common pitfalls. Therefore, the CreaDO Methodology defines an algorithm to carry out this evaluation. The algorithm is defined based on the ontological element where the error is located. Therefore, this algorithm is composed by three main tasks: (1) a task to identify errors in the classes, (2) a task to identify errors in the properties (both datatype and object properties), and finally, (3) a task to identify errors in the ontology structure. Not all the common pitfalls presented in [14] can be evaluated in our

[1] http://rewerse.net/A1/otn/OTN.owl
[2] http://www.info.uqam.ca/Members/valtchevp/mbox/ETP-tourism.owl

method, because some pitfalls are related to the evaluation of knowledge representation issues, which are not considered in our research. Then, only common pitfalls related to general design of the ontology are taken into account in our method. The common pitfalls that we evaluate were classified in three groups according to the aforementioned tasks, as shown in Table 1.

TABLE I. COMMON PITFALLS CLASSIFICATION ACCORDING TO TASKS OF THE EVALUATION METHOD

Tasks	Ontological elements	Common pitfalls
Evaluation of classes	Simples and complex classes, individuals	• Misusing primitive and defined classes • Using different naming criteria in the ontology
Evaluation of properties	Properties, properties restriction, properties characteristics	• Misusing primitive and defined classes • Using different naming criteria in the ontology • Creating unconnected ontology elements • Missing domain or range in properties • Using incorrectly ontology elements • Using recursive definition
Evaluation of architecture	Taxonomy of ontology	• Including cycles in the hierarchy

Here, we include some examples taken from our case study that show how the evaluation was performed.

- **Evaluation of classes:** for the classes *Cinema and CasualDining* of *ETP ontology* we found that the element is defined using the primitives of OWL to define ontological class.
- **Evaluation of properties:** in the *ETP ontology* the property *hasGPSCoordinate* has as domain the class *Site* and as range the class *GPS_Coordinate.* In this case, the primitives of OWL are correctly used because *hasGPSCoordinate* is defined as an object property, and *Site* and *GPS_Coordinate* as classes. The property contains both domain and range, thus, it does not contain unconnected elements or missing domain and range. However, the property *hasLanguage* has as range the class *Language* but has not domain defined.
- **Evaluation of architecture:** Cycles in the hierarchy was not found in this case study.

B. Definition method of the merge parameter

The CreaDO Methodology considers a merge parameter, that is, a concept related to the domain that will be represented in the domain ontology. As commented before, this parameter is used to filter the knowledge obtained from the source ontologies. The merge parameter is provided by the user.

In the same manner than classes in the ontology have been extended with lexical information, particularly, synonym information; the merge parameter needs to be also extended with lexical information. This enrichment is performed in this method using WordNet. In our case study the merge parameter is the concept *Tourism*. Therefore, in the method this term is extended with its synonyms (*Touristy* and *Touring*).

C. Ontology modularization method

The ontology modularization method has the objective of obtaining an ontology module for each one of the source ontologies. The approach in this method is to use the merge parameter as a filter of knowledge that allows us to obtain a concrete ontology module based on the merge parameter. This method is an extension of the algorithm to extract of ontology modules presented by Doran in [13]. The method uses as input: a) an ontology, b) a set of allowed properties, c) a set of not allowed properties, d) a set of classes to visit, and finally e) a set of visited classes. A relevant element in our method is the *set of classes to visit* because the merge parameter and its synonyms are defined in this input set.

Given an ontology O and a concept c, the algorithm for extracting on ontology modules gets the ontology module of the ontology O related to concept c. The ontology module is composed by all the elements related with the merge parameter by the allowed properties. The properties allowed by the method are the following: *subClassOf*, *subProperty*, *instanceOf*, *disjointOf*, *all no taxonomic properties*, and restriction properties (*transitiveProperty*, *simmetricProperty*). With this input, the methodology tries to obtain an ontology module that is autonomous, centered to the class and consistent in order to add the modules to the domain ontology without have relationships with the source ontologies. It is possible to find ontologies for which the method does not generate an ontology module; this situation is treated in the ontology merging method.

In our case study, the ontology modularization performed on the ETP ontology using the merge parameter "*Tourism*" and its synonyms "*Touristy* and *Touring*" produce an empty ontology *module*, which does not contain any elements. On the other hand, the ontology modularization performed on the OTN ontology using the merge parameter "*Tourism*" produces an ontology module that contains

- Nine class (*Tourism*, *Service*, *Historical_Monument*, *Tourist_Atraction*, *Tourist_Office*, *Vantage_Point*, *Location_Reference*, *Feature*, *Node*),
- Two object properties (*locationReference* (*Feature*, *Location_Reference*), *isDisplayedAt* (*Service*, *Node*)), and.
- Four datatype properties (*alternativeName* (*Feature*, *String*), houseNumber (*Service*, *String*), *externalLink* (*Feature*, *URL*), *externalLink* (*Feature*, *String*)).

D. Ontology mapping method

The ontology mapping method has the objective of identify mappings among the source ontologies. To do this, a mapping is defined between these two elements elements when there is a relation between two ontological elements that belong to two different ontologies. This method takes into account only the equivalence mapping. These mappings can be identified using a pattern language and a set of mapping rules.

On the one hand, the pattern language defines a correspondence between two ontological elements. The pattern language is defined using the following three correspondence ontology patterns[3], defined in the Ontology Design Pattern (ODP): a) *class to class pattern*, b) *object property to object property pattern*, c) *datatype property to datatype property pattern*. In addition to these existing patterns, we have also defined a new pattern: an equivalence relationship between an object property and a datatype property that we call *object property to datatype property pattern*.

On the other hand, the set of mapping rules is responsible for calculating the ontological similarity between two elements that belongs to two ontologies. To do this, we have used the approach proposed by Ehrig [12] based on a model to measure the ontological similarity of two ontological entities. This model proposes three layers: (a) a data layer that compares entities taking into account non-ontological aspects, namely the labels are consider as human identifiers; (b) an ontology layer that compares each entity, taking into account aspects of semantic networks, Description Logics, restrictions and rules (e. g. ontological elements associated to the entities compared); and (c) a context layer that compares entities in the context for which they were created. In this paper, we only focus on the data layer and the ontology layer. Ehrig and Sure [15] propose a set of rules of mappings to calculate the ontological similarity. These rules are classified on the two layers as shown in Table 2.

TABLE II. RULES OF MAPPINGS [15]

Layer	Rule of mappings
Data	R1. If two labels are the same, the entities probably also the same R2. If two entities have the same identifier they are identical
Ontology	R3. If the properties of two concepts are same, the concepts are also same R4. If domain and range of two properties are same, the properties are also the same R5. If super-concepts are the same, the actual concepts are similar to each other R6. If sub-concepts are the same, the compared concepts are similar R7. If concepts have similar siblings, they are also similar R8. If super-properties are the same, the actual properties are similar R9. If sub-properties are the same, the compared properties are similar R10. Concepts that have the same instances are the same R11. Instances that have the same mother concept are similar R12. If concepts have a similar fraction of instances, the concepts are similar R13. If two instances are linked to another instance via the same property, the two original instances are similar R14. If two properties connect the same two instances, the properties can be similar R15. They explicitly state that two entities are the same.

In our method, the mapping rules are applied to each pair of elements to calculate the similarity between two ontological elements, and the patterns are used to define mappings between ontological elements of the same type (e. g. class to class, property to property, etc) of the two ontologies. This method considers first the evaluation of the rules in the data layer, and second the ontology layer. This process needs to be executed for all the source ontologies with the objective of finding mappings among all the ontologies.

Here, we include some examples taken from our case study that show some mappings identified with the ontology mapping method.

- *equivalentClass(Ferry, Ferry)*: the *Ferry* element of the *ETP ontology* corresponding with the *Ferry* element of the *OTN ontology*.
- *equivalentClass(Swimming, Swimming_Pool)*: the *Swimming* element of the ETP ontology corresponding with the *Swimming_Pool* element of the OTN ontology.
- *equivalentClass(Service, Service):* the *Service* element of the *ETP ontology* corresponding with the *Service* element of the *OTN ontology*. Also is important to point out that the mapping between the *Service* classes contains the *Service* element of the *OTN module* identified in the modularization method. Nevertheless, has no reference to the *Service* element of ETP ontology due to this has not ontology module.

Other relevant task in the ontology merging method is the mapping filtering task, which identifies the mappings that have correspondence with the merge parameter.

Mapping filtering task

This task has the objective of filtering the mappings identified in the ontology mapping method. This filtering is performed due to not all the mappings can be taken in account into the new ontology. The only mappings that will be used are those that contain in their domain or range elements belonging to the ontology modules. If either domain or range belongs to ontology modules, then the mapping is selected to be included in the domain ontology.

In our case study, the mapping filtering is executed over the three *equivalentClass* mappings identified by the ontology mapping method. In this example, the mapping obtained by the mapping filtering task is the *equivalentClass* property of *Service* elements.

E. Ontology merging method

The ontology merging method has the objective of joining all the ontology modules obtained by the ontology modularization method in order to create the new ontology. This new ontology is our domain ontology. This method uses the subset of mappings obtained by the mappings filtering task to define the union among ontology modules into the domain ontology.

However, when a mapping is defined in the domain ontology, it is possible that the element (a class) of the domain or range is not defined in any of the modules included in the domain ontology. To solve this situation, these non-defined classes are aggregated to the domain ontology as a new element, which will have the knowledge described in the source ontology where the class is defined. The knowledge taken into account is the class and its datatype properties.

[3] http://ontologydesignpatterns.org/wiki/Category:AlignmentOP (Accessed on 22-Sep-2011

In our case of study, we have taken the ontology module of OTN ontology generated by the ontology modularization method and the mapping of *equivalentClass*(*Service*, *Service*) obtained by the mapping filtering task, and we have obtained an ontology that contains

- Ten classes (*OTN:Tourism*, *OTN:Service*, *ETP:Service*, *OTN:Historical_Monument*, *OTN:Tourist_Atraction*, *OTN:Tourist_Office, OTN:Vantage_Point, OTN:Location_Reference, OTN:Feature, OTN:Node*),
- Two object properties (*OTN:locationReference (Feature, Location_Reference), OTN:isDisplayedAt (Service, Node)*), and
- Four datatype properties (*alternativeName (Feature, String), houseNumber (Service, String), externalLink (Feature, URL), externalLink (Feature, String)*).

F. Domain ontology evaluation method

Finally, the domain ontology evaluation method has the objective of evaluating the domain ontology in order to identify inconsistencies at the structure level. In this method the following inconsistencies are detected: (a) a class with two super classes and (b) two classes with an equivalent property and a disjoint property. In addition, this method uses the pitfalls found in the ontology evaluation method in order to present this information to the ontology practitioner.

In our case of study, the generated domain ontology does not present inconsistencies in the structure.

IV. CONCLUSIONS AND FUTURE WORKS

In this paper, we have presented, in a general novel, the CreaDO Methodology. This methodology aims to create domain ontologies focuses on filtering only the relevant knowledge needed by the ontology practitioner. This methodology receives as input a set of lightweight ontologies and a concept that plays the role of a merge parameter. This parameter is used to filter the relevant knowledge to create a new specific domain ontology.

The methodology is grounded on techniques related to ontology reuse, such as ontology merging and ontology modularization. The methodology is formed by 6 methods: (a) ontology evaluation method, (b) definition method of the merge parameter, (c) ontology modularization method, (d) ontology mappings method, (e) ontology merging method, and (f) domaing ontology evaluation method.

The CreaDO Methodology will help ontology practitioners to build more understandable and usable domain ontologies to be applied in specific purposes.

Some important issues remain as directions for further work. The CreaDo Methodology only works with ontologies described in the same natural language (English). We have planned to update the methods to allow having ontologies in the same domain but in different natural languages.

In addition, in our ontology mapping method we calculate the semantic similarity between elements that only cover entity and ontology levels. Thus, another future work is to improve this mapping method in a way that it is able to take into account also the context level.

ACKNOWLEDGEMENTS

This work has been partially supported by the PROMEP project LeMoNe-Pai /103-5/10/4641.

REFERENCES

[1] B. Ganter and G. Stumme, “Creation and Merging of Ontology Top-Levels,” *Proc. of the 11th International Conference on Conceptual Structures, ICCS*, vol. 2746, pp. 131-145, 2003.

[2] G. Stumme and A. Maedche, “FCA-Merge: Bottom-up merging of ontologies,” in *International Joint Conference on Artificial Intelligence*, 2001, vol. 17, pp. 225–234.

[3] S. Raunich and E. Rahm, “Target-driven merging of Taxonomies,” *CoRR informal publication*, 2010.

[4] M. Dahab, H. Hassan, and A. Rafea, “TextOntoEx: Automatic ontology construction from natural English text,” *Expert Systems with Applications*, vol. 34, pp. 1474-1480, 2008.

[5] D. Sánchez, “Domain Ontology Learning from the Web,” *The Knowledge Engineering Review*, vol. 24, p. 413, 2009.

[6] M. d’ Aquin, C. Baldassarre, L. Gridinoc, S. Angeletou, M. Sabou, and E. Motta, “Characterizing knowledge on the semantic web with Watson,” *EON CEUR Workshop Proceedings*, vol. 329, pp. 1-10, 2007.

[7] M. C. Suárez-Figueroa, “NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse,” Universidad Politécnica de Madrid, Madrid, Spain, 2010.

[8] M. Fernandez, A. Gomez-Perez, and N. Juristo, “Methontology: from ontological art towards ontological engineering,” in *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering*, 1997, pp. 33–40.

[9] D. Fensel et al., “On-to-knowledge: Ontology-based tools for knowledge management,” in *Proceedings of the eBusiness and eWork*, 2000, pp. 18–20.

[10] H. S. Pinto, S. Staab, and C. Tempich, “DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies,” *in Pproceedings of the 16th european conference on artificial intelligence (ECAI)*, pp. 393-397, 2004.

[11] M. Cristani and R. Cuel, “A Comprehensive Guideline for Building a Domain Ontology from Scratch,” *Proceedings of International Conference on Knowledge Management*, 2004.

[12] M. Ehrig, *Ontology Alignment: Bridging the Semantic Gap*, vol. 4. New York: Springer, 2007.

[13] P. Doran, V. Tamma, and L. Iannone, “Ontology module extraction for ontology reuse: an ontology engineering perspective,” in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007, pp. 61–70.

[14] M. Poveda-Villalón, M. C. Suárez-Figueroa, and A. Gómez-Pérez, “A Double Classification of Common Pitfalls in Ontologies,” *Workshop on Ontology Quality Co-located with EKAW 2010*, 2010.

[15] M. Ehrig and Y. Sure, “Ontology mapping–an integrated approach,” *The Semantic Web: Research and Applications*, pp. 76–91, 2004.