

Fusing Interesting Topics in the Web

Fusión de temas importantes en la Web

A.D. Cuevas-Rasgado

Instituto Tecnológico de Oaxaca, México.

E-mail: almadeliacuevas@gmail.com

A. Guzman-Arenas

Centro de Investigación en Computación del IPN, México.

E-mail: a.guzman@acm.org

(Recibido: septiembre de 2007; aceptado: abril de 2008)

Abstract

A person builds his or her knowledge from different sources of information. In school he learns that Buonarroti was born in Caprese and at home they tell him that the name of the neighbor's dog is Fido. In order to know more, he combines information from many sources. But this multi-source information can contain repetitions, different level of details or precision, and contradictions. These problems are not easy to solve by computers. Nevertheless, the enormous masses of accumulated knowledge (in the Web there exist more than one billion different pages) demand computer efforts to combine them, since merging manually this information in a consistent way is outside human capabilities. In this paper, a method is explained to combine multi-source information in a manner that is automatic and robust; contradictions are detected and sometimes solved. Redundancy is expunged. The method combines two source ontologies into a third; through iteration, any number can be combined.

Keywords: Ontology fusion, knowledge representation, semantic processing, artificial intelligence, text processing, ontology.

Resumen

Una persona construye su conocimiento usando diversas fuentes de información. En la escuela aprende que Buonarroti nació en Caprese y en casa le dicen que Fido se llama el perro del vecino. Para saber más, él combina información de muchas fuentes. Pero esta multiplicidad de fuentes contiene repeticiones, distintos niveles de detalle o precisión, y contradicciones. Estos problemas no son nada fáciles para que una computadora los resuelva. Sin embargo, la enorme masa de conocimiento acumulado (en la Web existen más de mil millones de páginas) demanda esfuerzos computarizados para combinarlas, puesto que la fusión manual de esta información rebasa las capacidades humanas. En este artículo se explica un método para combinar información de varias fuentes en una manera que es automática y robusta, y donde las contradicciones se detectan y a veces se resuelven. La redundancia se elimina. El método combina dos ontologías fuentes en una tercera; por iteración, cualquier número de ellas puede ser combinada.

Descriptores: Fusión de ontologías, representación del conocimiento, procesamiento semántico, inteligencia artificial, procesamiento de texto, ontología.

1. The importance of knowledge fusion

Knowledge accumulation is important. A person accrues knowledge gradually, as he adds concepts to his previous knowledge. Initial knowledge is not zero, even for animals. How can a machine do the same?

Learning occurs by adding new concepts, associating them to the information already learnt. New information can contradict or confuse a human being, or be simply redundant (already known, said with more words) or less accurate (more vague). A person somehow solves these tasks, and keeps a consistent knowledge base.

This paper is centered in the fusion of ontologies (arising from different sources) between computers. During this fusion the same problems (redundancy, repetition, inconsistency...) arise; the difference is that the machines have no common sense (Lenat, *et al.*, 1989) and the challenge is to make them understand that beneficial is the same as generous, and that triangle represents:

- A three-sided polygon;
- A musical percussion instrument; or
- A social situation involving three parties.

The computer solution to fusion should be very close to people's solution.

Works exist (Dou *et al.*, 2002; McGuinness, *et al.*, 2000 and Noy, *et al.*, 2000) that perform the union of ontologies in a semiautomatic way (requiring user's assistance). Others (Kalfoglou, *et al.*, 2002 and Stumme, *et al.*, 2002) require ontologies to be organized in formal ways, and to be consistent with each other. In real life, ontologies coming from different sources are not likely to be similarly organized, nor they are expected to be mutually consistent. The automation of fusion needs to solve these problems.

This paper explains a process of union of ontologies in automatic and robust form. Automatic because the (unaided) computer detects and solves the problems appearing during the union, and robust because it performs the union in spite of different organization (taxonomies) and when the sources are jointly inconsistent.

The fusion is demonstrated by taking samples of real Web documents and converting them by hand to ontologies. These are then fed to the computer, which produces (without human intervention) a third ontology as result. This result is hand-compared with the result obtained by a person. Mistakes are low (table 1).

Table 1. Performance of OM in some real examples: C = A B. First column gives the execution times taken by the fusion in a laptop Pentium IV with 1.70 GHz. Error column gives the ratio of (number of wrong relations) / (total number of relations) or (number of wrong concepts) / (total number of concepts), respectively

Ontologies	Nodes that are relations	Error	Nodes which are concepts but not relations	Error
Turtles (4 sec)	A (8 relations), B (6 relations) and C (10 relations)	0	A (29 concepts), B (35 concepts) C (35 concepts)	0
Hammer (6 sec)	B (30), A (8) and C (35)	0	A (24), B (33) and C (51)	0
Poppy (14 sec)	A (21), B (20) and C (37)	0	A (34), B (35) and C (58)	0
100 Years o loneliness (10 min)	A (231), B (283) and C (420), manual method gave C (432) 12 out of 432 were not copied	0.027	A (90), B (126) and C (141), manual method gave C (149) 8 out of 149 were not copied	0.053
Oaxaca ⁶ (5 min)	A (61), B (43) and C(96)	0	A (234), B (117) and C (309), manual method gave C (310) 1 out of 310 were not copied	0.003

⁶ Oaxaca is located to the south of Mexico.

1.1 The problem to solve: To merge two data sources into a result containing its common knowledge, without inconsistencies or contradictions.

OM (Ontology Merging) is a program that automatically merges two ontologies into a third one containing the joint knowledge at the sources, without contradictions or redundancies. OM is based in

- The theory of confusion (2.1);
- The use of COM (2.3), to map a concept into the closest concept of another ontology;
- The use of the OM notation (2.3) to better represent ontologies.

These are briefly explained in section 2, whereas section 3 explains the OM Algorithm, and gives examples of its use.

1.2 The importance of automatic knowledge fusion

How can we profit from computers automatically fusing two ontologies?

- a. We could use crawlers or distributed crawlers (Olguin, 2007) to automatically find most Web pages and documents about a given topic (say, One Hundred Years of Solitude by Gabriel García-Márquez). After a good parser (3.4) converts these documents to their corresponding ontologies, OM can produce a large, well-organized, consistent and machine-processable ontology on a given topic, containing most of the knowledge about this theme.
- b. By repeating (a) on a large variety of topics, we could produce a single unified ontology containing most of the knowledge on whatever collection of topics¹ we wish to have. This ontology will contain not only common sense knowledge (Lenat & Guha, 1989), but specialized knowledge as well.
- c. Ontology (b) can be exploited by a question-answerer or deductive software (Botello, 2007), that answers complex questions (not just factual questions), thus avoiding the need to read and understand several works about One Hundred Years of Solitude to find out the full name of the father of the person who built small gold fish in Macondo, or to find out

why the text-processing company Verity was bought by rival Autonomy around 2005.

- d. Ontology (b) could be kept up to date by periodically running (a) and OM in new documents.

Commercial applications of automatic fusion appear in (Cuevas & Guzman, 2007).

2. Background and relevant work

This section reveals the work on which OM is based, as well as previous relevant work.

2.1 Hierarchy and confusion

A hierarchy (Levachkine *et al.*, 2007) is a tree where each node is a concept (a symbolic value) or, if it is a set, its descendants must form a partition of it. Example: see figure 1.

Hierarchies code a taxonomy of related terms, and are used to measure confusion, which OM uses for synonym detection and to solve inconsistencies.

Contradiction or inconsistency arises when a concept in ontology A has a relation that is incompatible, contradicts or negates other relation of the same concept in B. For instance, Isaac Newton in A may have the relation born in Italy; and in B Earth Isaac Newton may have the relation born in Lincolnshire, England. Contradiction arises from these two relations: in our example, the born in places are not the same, and they are inconsistent as born in can only have a single value. Since OM must copy concepts keeping the semantics of the sources in the result, and both semantics are incompatible, a contradiction is detected. It is not possible to keep both meanings in the result because they are inconsistent². OM uses confusion (Levachkine *et al.*, 2007) to solve this.

Function CONF(r, s), called the absolute confusion, computes the confusion that occurs when object r is used instead of object s, as follows:

$$\text{CONF}(r, r) = \text{CONF}(r, s) = 0, \text{ when } s \text{ is some ascendant of } r;$$

$$\text{CONF}(r, s) = 1 + \text{CONF}(\text{descendant of } (r), s) \text{ in other cases.}$$

¹ Or just by applying the parser in (a) to all articles of Wikipedia and then using OM to fuse the resulting ontologies.

² OM assumes A and B to be well-formed (each without contradictions and no duplicate nodes). Even then, an inconsistency can arise when considering their joint knowledge.

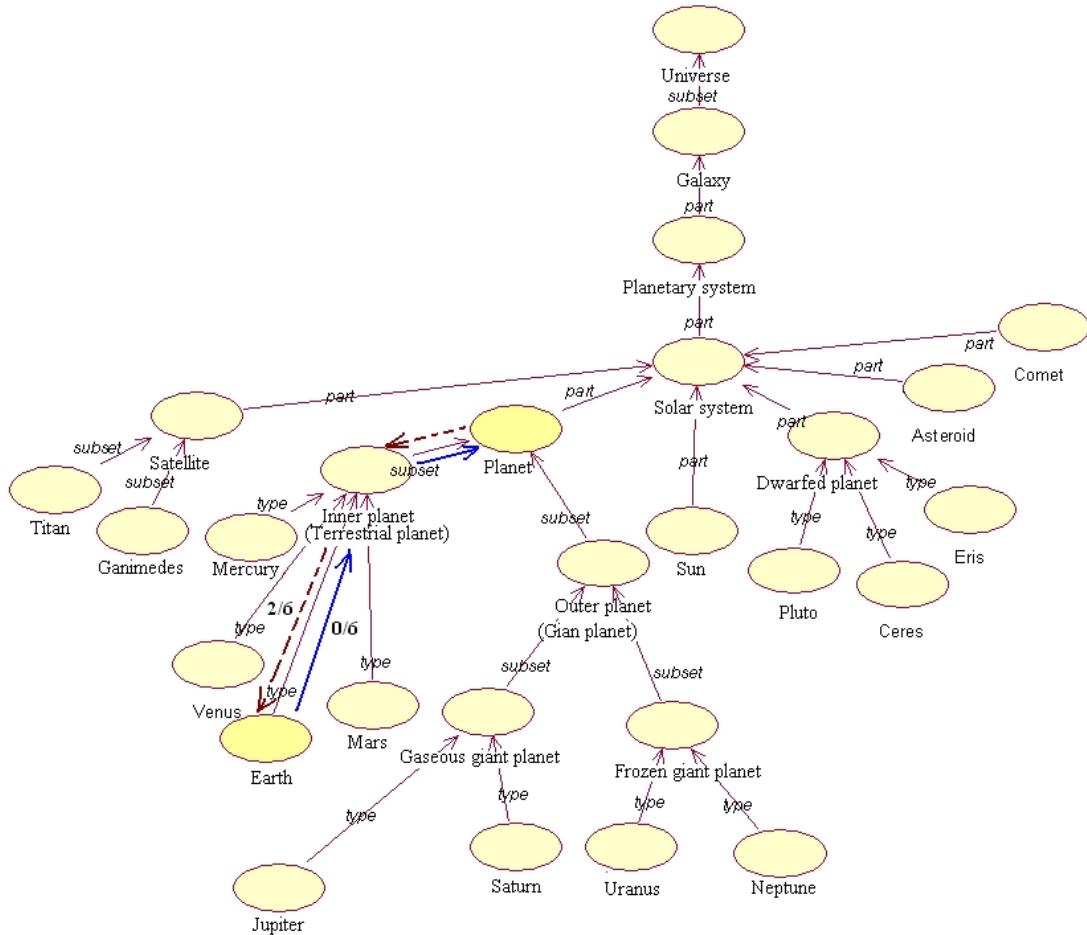


Figure 1. A hierarchy. $\text{conf}(\text{the Earth}, \text{Planet}) = 0$ but $\text{conf}(\text{Planet}, \text{Earth}) = 2/7$

CONF is the number of descending links when one travels from r (the used value) to s (the intended value), in the hierarchy to which r and s belong.

Absolute confusion CONF returns a number between 0 and h , where h is the height of the hierarchy. We normalize to a number between 0 and 1, thus:

Definition.

$\text{conf}(r, s)$, the confusion when using instead of s , is
 $\text{conf}(r, s) = \text{CONF}(r, s)/h$

conf returns a number between 0 and 1. Example: In figure 1, $\text{conf}(\text{Hydrology, river}) = 0.2$. OM uses conf, whereas (Levachkine *et al.*, 2007) describes CONF. The function conf is used by OM to detect apparent or real inconsistencies (3.1, example 1), and to solve some of them.

2.2 Ontology

Formally, an ontology is a hypergraph (C, R) where C is a set of concepts, some of which are relations; and R is a set of restrictions of the form $(r c_1 c_2 \dots c_k)$ among relation r and concepts c_1 through c_k . It is said that the arity of r is k .

Computationally, an ontology is a data structure where information is stored as nodes (representing concepts such as house, computer, desk) and relations (representing restrictions among nodes, such as shelters, rests in or weight, as in (shelters house computer), (rests on computer desk) (figure 2). Usually, the information stored in an ontology is “high level” and it is known as knowledge. Notice that relations are also concepts.

We have found current ontology languages restricted, so we have developed our own language, called OM notation (2.3).

An important task when dealing with several ontologies is to identify most similar concepts. We wrote COM (2.3) that finds this similarity across ontologies.

2.3 COM and OM notation

Given two ontologies B and C, COM (Guzman *et al.*, 2004) is an important algorithm that, given a concept $c_C \in C$, finds $cms = COM(c_C, B)$, the most similar concept (in B) to c_C . For instance, if B knows Falkland Islands, an archipelago in the Atlantic Ocean about 300 miles off the coast of Argentina, and C knows Islas Malvinas, a chain of islands situated in the South Atlantic Ocean about 480 km East of the coast of South America, COM may deduce that the most similar concept in C to Falkland Islands (in B) is Islas Malvinas. COM greatly facilitates the work of OM, which extensively uses an improved version (Cuevas, 2006) of it.

OM Notation (Cuevas, 2006) represents ontologies through an XML-like notation. The labels describe the concepts and their restrictions. In OM Notation:

- Relations are concepts;
- Relations are n-ary relations;
- A particular case of a relation is a partition.

2.4 Computer-aided ontology merging

Initially, merging was accomplished with the help of a user. Previous solutions to 1.1. (Kotis K. *et al.*, 2006), which applies WordNet and user intervention, focuses on a single aspect of the merging process. IF-Map (Kalfoglou *et al.*, 2002) and FCA-Merge (Stumme *et al.*, 2002), require consistent ontologies that are expressed in a formal notation employed in Formal Concept Analysis (Bemhard *et al.*, 2005), which limits their use. Prompt (Noy *et al.*, 2000), Chimaera (McGuinness *et al.*, 2000), OntoMerge (Dou *et al.*, 2002), are best considered as non automatic mergers, because many important problems are solved by the user. Also, [11] has a fusion method (applied in the ISI project) that requires human intervention.

Our solution to 1.1 is the OM algorithm (3), which performs the fusion in a:

- Robust (OM forges ahead and does not fall into loops),
- Consistent (without contradictions),

- Complete (the result contains all available knowledge from the sources, but it expunges redundancies and detects synonyms, among other tasks) and
- Automatic manner (without user intervention).

2.5 Knowledge support for OM

OM uses some built-in knowledge bases and knowledge resources, which help to detect contradictions, find synonyms, and the like. These are:

1. In the coding, stop words (in, the, for, this, those, it, and, or...) are expunged (ignored) form word phrases;
2. Words that change the meaning of a relation (without, except...) are considered;
3. Several hierarchies are built-in into OM, to facilitate the calculus of confusion;
4. In the near future (see Discussion at 3.4),
5. OM can rely on external language sources (WordNet, dictionaries, thesaurus..);
5. OM will use as base knowledge the results of previous merges!

3. Merging ontologies automatically: the OM algorithm

This algorithm fuses two ontologies (Cuevas, 2006) A and B into a third ontology $C = A \cup B$ ³ containing the information in A, plus the information in B not contained in A, without repetitions (redundancies) nor contradictions.

OM proceeds as follows:

1. C = A. Ontology A is copied into C. Thus, initially, C contains A.
2. Add to each concept $c_C \in C$ additional concepts from B, one layer at a time, contained in or belonging to the restrictions (relations) that c_C has already in C. At the beginning, concept c_C is the root of ontology C. Then, c_C will be each of the descendants of c_C , in turn, so that each node in C will become c_C ⁴. For each

³ Symbol \cup when it refers to ontology merging, it means not only set union, but “careful” merging of concepts, using their semantics.

⁴ The ontology C is searched *depth-first*: first, c_C is the root. Then, c_C is the first child of the root, then c_C is the first child of *this* child (a grand son of the root)... Thus, a branch of the tree is traveled only until the deepest descendant is

$c_C \in C$, COM (2.3) looks in B for the concept that best resembles c_C , such concept is called the most similar concept in B to c_C , or cms. Two cases exist:

A. If c_C has a most similar concept cms in B, then:

- i. Relations that are synonyms (3.1, example 2) are enriched.
- ii. New relations (including partitions) that cms has in B, are added to c_C . For each added relation, concepts related by that relation and not present in C are copied to C.
- iii. Inconsistencies (2.2) between the relations of c_C and those of cms are detected.

If it is possible, by using confusion, to resolve the inconsistency, the correct concepts are added to C.

When the inconsistency can not be solved, OM rejects the contradicting information in B, and c_C keeps its original relation from A.

- 3. c_C next descendant of c_C (Take the next descendant of c_C).
- 4. Go back to step 2 until all the nodes of C are visited (including the new nodes that are being added by OM as it works). (Cuevas, 2006) explains OM fully.

3.1 Examples of merges by OM

In this section, figures show only relevant parts of ontologies A, B and the resultant C, because they are too large to fit.

Example 1. Merging ontologies with inconsistent knowledge. Differences between A and B could be due to: different subjects, names of concepts or relations; repetitions; reference to the same facts but with different words; different level of details (precision, depth of description); different perspectives (people are partitioned in A into male and female, whereas in B they are young or old); and contradictions.

Let A (the information was obtained in [2]) contains: The Renaissance painter, sculptor, architect and poet Michelangelo di Lodovico Buonarroti Simoni was born in Caprese, Italy while B [7] contains: The painter Michelangelo Buonarroti was born in Caprese, Italy. Both ontologies duplicate some information (about Michelangelo's place of birth), different expressions (painter, sculptor, architect and poet versus painter), different level of details (Michelangelo di Lodovico Buonarroti Simoni versus Michelangelo Buonarroti),

and contradictions (Caprese vs. Capre). A person will have in her mind a consistent combination of information: Michelangelo Buonarroti and Michelangelo di Lodovico Buonarroti Simoni are not the same person, or perhaps they are the same, they are synonyms. If she knows them, she may deduce that Michelangelo di Lodovico Buonarroti Simoni is the complete name of Michelangelo Buonarroti. We solve these problems everyday, using previously acquired knowledge (2.5) and common sense knowledge (Lenat et al., 1989), which computers lack. Also, they did not have a way to gradually and automatically grow their ontology. OM measures the inconsistency (of two apparently contradicting facts) by asking conf to determine the size of the confusion in using Caprese in place of Capre and vice versa, or the confusion of using Michelangelo Buonarroti instead of Michelangelo di Lodovico Buonarroti Simoni. In the example Caprese is a write error, therefore in C the value of A is conserved (Caprese).

OM does not accept two different names for a birthplace (a person cannot be born at the same time in two places). If A said that Michelangelo Buonarroti was born in Caprese and B Michelangelo Buonarroti was born in Italy, OM chooses Caprese instead of Italy because it is more specific place whereas Italy that is more general (it deduces this from a hierarchy of Europe). Small inconsistencies cause C to retain the most specific value, while if it is large, OM keeps C unchanged (ignoring the contradicting fact from B). In case of inconsistency, A prevails⁵.

Example 2. Joining partitions, synonym identification, organization of subset to partition, identification of similar concepts, elimination of redundant relations and addition of new concepts. Figure 2 displays ontologies A, B and the fusion of these, C. Cases of OM exemplified in the figure are shown with underlined terms.

⁵ We can consider that an agent's previous knowledge is A, and that such agent is trying to learn ontology B. In case of inconsistency, it is natural for the agent to trust more its previous knowledge, and to disregard inconsistent knowledge in B as "not trustworthy" and therefore not acquired – the agent refuses to learn knowledge what it finds inconsistent, if the inconsistency (measured by conf) is too large.

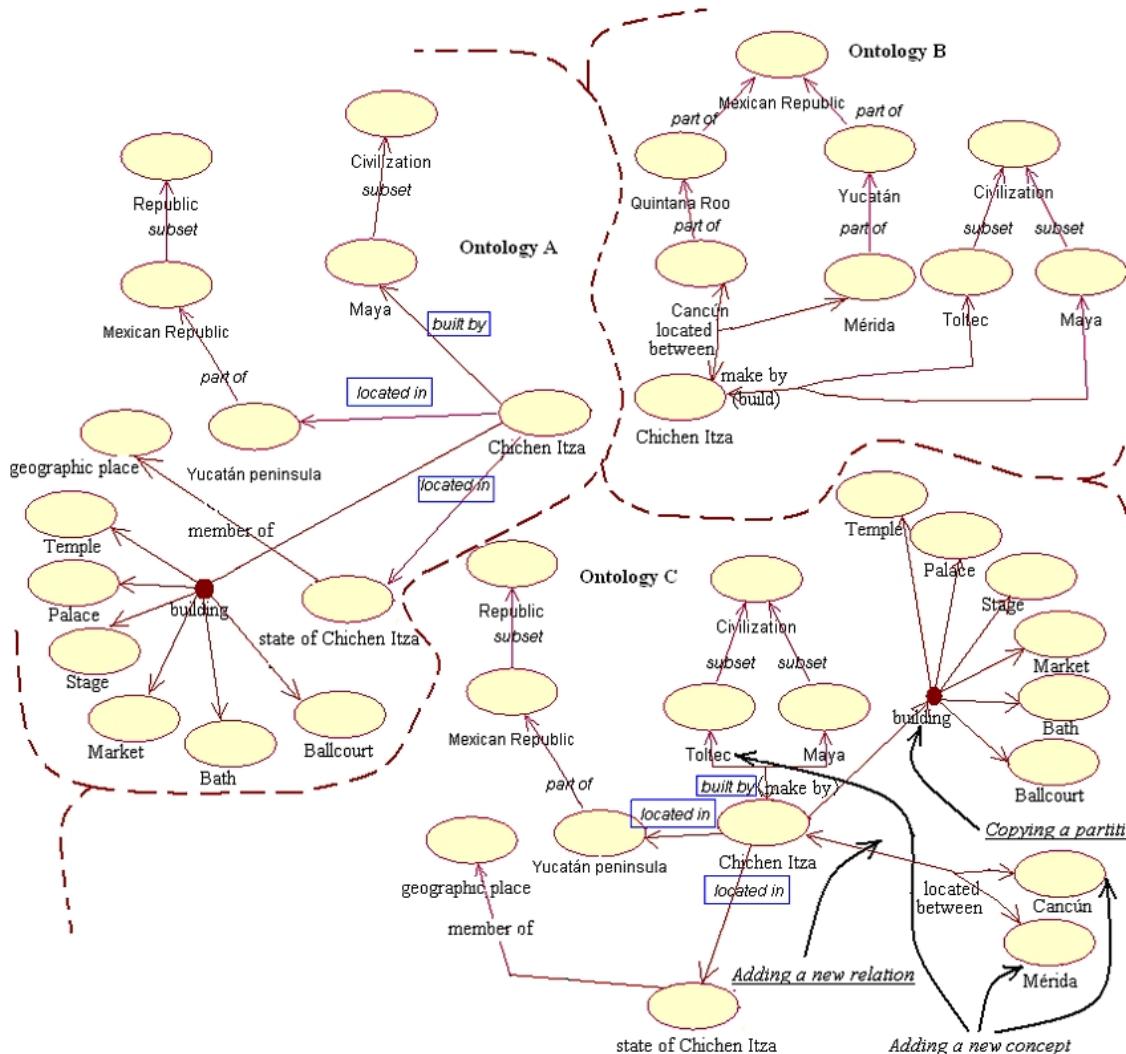


Figure 2. Two different ontologies A and B describing Chichen Itza, and its fusion C (only relevant parts of them are shown)

Cases of OM: the fusion is accomplished through seven cases:

1. Copying new partitions. building is a partition in A (indicated in the small circle) of Chichen Itza, therefore it is added to the resulting ontology C.
2. Copying new concepts. Concepts Toltec, Mérida and Cancún were not in A, but they appear in B. Therefore, they were copied by OM to C.
3. Reorganization of relations. Relation located in it appears twice but with different values, therefore they are added to C because it is possible for that relation to have several values. In case of single-valued relations, confusion is used, as in Example 1.

4. Synonym identification. Concept Chac Mool in A (figure 3) has Chac in it definition (the words that defines it, between parenthesis), and Chac in B is synonymous of chac Mool in
5. Identification of similar concepts. Concept sculpture of a jaguar in A and throne in the shape of jaguar in B they have the same properties (Color and its value) therefore, OM fuses them into a single concept. The same happens with El Castillo and Pyramid of Kukulkan since they have the same properties and children.
6. Removing redundant relations. In A, Chichen Itza is member of pre-Columbian archaeological site (figure 4), which is in turn a member of archaeological sites.

In B, Chichen Itza is member of archaeological site (which is parent of pre-Colombian archaeological site in B), therefore it is eliminated in C because it is a redundant relation. In C, pre-Colombian archaeological site is parent of Chichen Itza.

7. Organization of subset to partition. In the building partition in A there are six subsets (figure 4): Ballcourt, Palace, Stage, Market and Bath. OM identifies them in B, where they appear as subsets of Chichen Itza. OM thus copies them into C like a partition, not as simple subsets. OM prefers the partition because it means that the elements are mutually exclusive and collectively exhaustive.

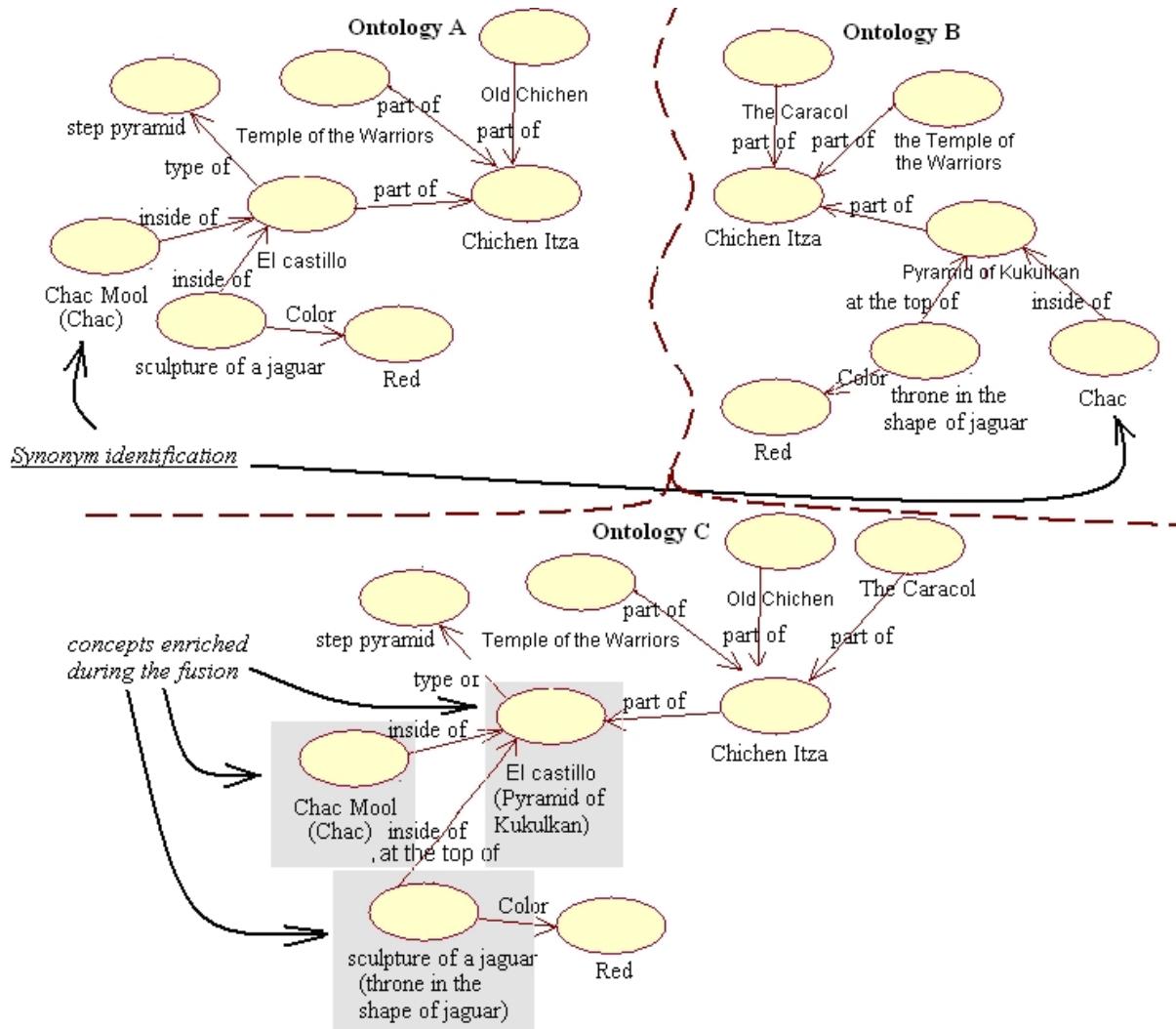


Figure 3. Chac Mool in ontology A and Chac in B are identified (case 4) as synonyms. A more interesting case is case 5, that identifies sculpture of a jaguar in A as a similar concept (a synonym) to throne in the shape of jaguar in B. Also El Castillo in A and Pyramid of Kukulkan in B are found to be the same

3.2 More applications of OM in real cases taken from the web

OM has merged ontologies derived from real documents. The ontologies were obtained manually from several documents (100 Years of Loneliness [8 and 10], Oaxaca [4 and 9], poppy [1 and 3] and turtles [5 and 6]) describing the same thing. The obtained ontologies were merged (automatically) by OM. Validation of results has been made manually, obtaining good results (table 1).

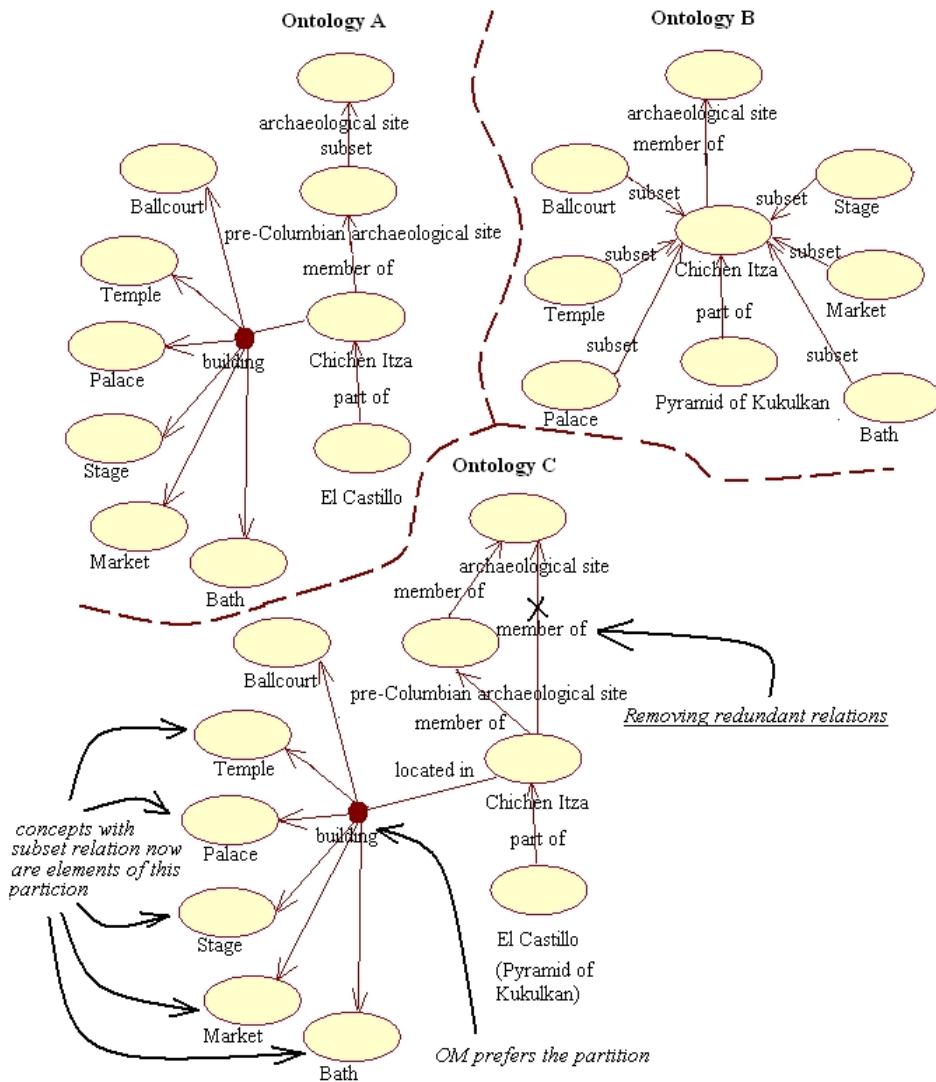


Figure 4. Case 6 removes redundant relations (marked with an X in the result C).
Case 7 (see text) upgrades a set of subsets into a partition

3.3 Conclusions

The paper presents an automatic, robust algorithm that fuses two ontologies into a third one, which preserves the knowledge obtained from the sources. It solves some inconsistencies and avoids adding redundancies to the result. Thus, it is a noticeable improvement to the computer-aided merging editors currently available (2.4).

The examples shown, as well as others in (Cuevas, 2006), provide evidence that OM does a good job, in spite of joining very general or very specific ontologies. This is because the algorithm takes into account not

only the words in the definition of each concept, but its semantics [context, synonyms, resemblance (through conf) to other concepts...] too. In addition, its base knowledge (2.5) helps.

3.4 Discussion

Is it possible to keep fusing several ontologies about the same subject, in order to have a larger and larger ontology that faithfully represents and merges the knowledge in each of the formant ontologies? OM seems to say "yes, it is possible." What are the main roadblocks? As we perceive them, they are:

- a. A good parser. Documents are now transformed into ontologies by hand, thus fusing of these hand-produced ontologies, although fully automated, it is hardly practical. It has been found difficult to build a parser that reliably transforms a natural language document into a suitable ontology, due to the ambiguity of natural language and to the difficulty of representing relations (verbs, actions, processes) in a transparent fashion (see next point).
- b. Exploitation of hypergraphs. Although we define ontologies as hypergraphs (2.2), the restrictions ($r\ c1\ c2\ \dots\ ck$), where r is a relation, are lists, and consequently, order matters. For instance, it is not the same (kills; Cain; Abel; jaw of donkey) that (kills; Abel; Cain; jaw of donkey). Moreover, the role of each argument (such as jaw of donkey) matters and must be explained –in the example it is the instrument used in the killing. Restrictions have different number of arguments, each with different roles: consider (born; Abraham Lincoln; Kentucky; 1809; log cabin). Many arguments may be missing in a given piece of text.

The role of each argument must be explained or described in a transparent (not opaque) fashion⁷, so that OM can understand such explanations, manipulate them and create new ones. For instance, from a given argument, it should be able to take two different explanations (coming from ontologies A and B, respectively) and fuse them into a third explanation about such argument, to go into C. Ways to do all of this should be devised.

- c. A query answerer that queries a large ontology and makes deductions. It should be able to provide answers to complex questions, so that “reasonable intelligence” is exhibited. (Botello, 2007) works on this for databases, not over a large ontology. He has obtained no results for real data, yet.
- d. Additional language-dependent knowledge sources could further enhance OM. For instance, WordNet, WordMenu, automatic discovery of ontologies by analyzing titles of conferences, university departments (Makagonov, P).

In this regard, probably the best way to proceed is (1) carefully building by hand a base ontology, and then (2) fusing to it (by OM) ontologies hand-translated from

⁷ Ideally, in OM notation.

carefully chosen documents, while (3) building the parser (a). This parser could very well use as built-in knowledge the very ontology that (2) produces. Also, OM can use as its built-in knowledge (2.5) the ontology (2). In parallel, (4) the language-dependent knowledge sources of (d) can also be somehow parsed by (a) into ontologies in OM notation (2.3), thus “including” them or absorbing them inside OM’s built-in knowledge. All of this while (5) the question-answerer (c) is finished and tested, first on federated or independent databases, then (6) on ontologies. An alternative to (6) is (7) to build the question-answerer or deductive machinery based on Robinson’s resolution principle, helped by the theory of confusion (2.3). We see four parallel paths of work: $[1 \rightarrow 2] ; [3] ; [4] ; [5 \rightarrow (6 \parallel 7)]$.

Acknowledgments

Work herein reported was partially supported by CONACYT Grant 43377, EDI-IPN and EDD-COFAA scholarships. A.G. has support from SNI. A-D. C. had an Excellence student grant from CONACYT.

References

- Bemhard G., Stumme G., Wille R. *Formal Concept Analysis: Foundations and Applications*. LNCS 3626. Springer 2005. ISBN 3-540-27891-5.
- Botello A. Query Resolution in Heterogeneous Data Bases by Partial Integration. Thesis (Ph. D. in progress). Mexico. Centro de Investigación en Computación (CIC), Instituto Politécnico Nacional (IPN). 2007.
- Cuevas A. Merging of Ontologies Using Semantic Properties. Thesis (Ph. D) CIC-IPN [on line], 2006. In spanish. Available on: <http://148.204.20.100:8080/bibliodigital>ShowObject.jsp?idobject=34274&idrepositorio=2&tpe=recipiente>
- Cuevas A. Guzman A. A Language and Algorithm for Automatic Merging of Ontologies. At: Chapter of the Book Handbook of Ontologies for Business Interaction. Peter Rittgen, ed. Idea Group Inc. 2007. In press.
- Dou D., McDermott D., Qi. P. *Ontology Translation by Ontology Merging and Automated Reasoning*. Proc. EKAW Workshop on Ontologies for Multi-Agent Systems. 2002.
- Guzman A., Olivares J. *Finding the Most Similar Concepts in two Different Ontologies*. LNAI 2972. Springer-Verlag. 2004. Pp. 129-138.
- Kalfoglou Y., Schorlemmer M. Information-Flow-based Ontology Mapping. At: Proceedings of the International Conference on Ontologies (1st, 2002, Irvine, CA, USA). Databases and Application of Semantics, 2002, (ODBASE’02).

Kotis K., Vouros G., Stergiou, K. Towards Automatic of Domain Ontologies: The HCONE Merge Approach. *Journal of Web Semantics (JWS)* [on line], Elsevier, 4(1): 60-79. 2006. Available on: ScienceDirect:<http://authors.elsevier.com/sd/article/S1570826805000259>.

Lenat D., Guha V. *Building Large Knowledge-Based Systems*. Addison-Wesley. 1989.

Levachkine S., Guzman A. Hierarchy as a New Data Type for Qualitative Values. *Journal Expert Systems with Applications*, 32(3). June 2007.

Makagonov P. Automatic Formation of Ontologies by Analysis of Titles of Conferences, Sessions and Articles. Work in Preparation.

McGuinness D., Fikes R., Rice J., Wilder S. The Chimaera Ontology Environment Knowledge. At: Proceedings of the International Conference on Conceptual Structures Logical, Linguistic, and Computational Issues. (Eighth, 2000, Darmstadt, Germany).

Noy N., Musen M. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In Proc. of the National Conference on Artificial Intelligence, 2000.

Olguin Luis-Antonio. Distributed Crawlers. Effective Work Assignment to Avoid Duplication in Space and Time. Thesis (M. Sc.). Mexico. CIC-IPN. In Spanish. 2007.

Stumme G., Maedche A. Ontology Merging for Federated Ontologies on the Semantic Web. At: E. Franconi K. Barker D. Calvanese (Eds.). Proc. Intl. Workshop on Foundations of Models for Information Integration, Viterbo, Italy, 2001. LNAI, Springer 2002 (in press).

URLs:

1. es.wikipedia.org/wiki/Amapola
2. es.wikipedia.org/wiki/Miguel_%C3%81ngel
3. www.buscajalisco.com/bj/salud/_herbolaria.php?id=1
4. www.elbalero.gob.mx/explora/html/oaxaca/geografia.html
5. www.damisela.com/zoo/rep/tortugas/index.htm
6. www.foyle.com/cartillas/37/tortugas_-_accesorios_para_acuarios_i.html
7. www.historiadelartemgm.com.ar/biografiamichelangelobuonarroti.htm
8. www.monografias.com/trabajos10/ciso/ciso.shtml
9. www.oaxaca-mio.com/atrac_turisticos/infooaxaca.htm
10. www.rincondelvago.com/cien-anos-de-soledad_gabriel-garcia-marquez_22.html
11. <http://plainmoor.open.ac.uk/ocml/domains/aktive-portal-ontology/techs.html>

Semblanza de los autores

Alma Delia Cuevas-Rasgado. Recently obtained her masters and Ph degrees in Computer Science at CIC-IPN (Centro de Investigación en Computación, Instituto Politécnico Nacional) in Mexico. Her lines of research: Software engineering -specifically about Quality of software and representation knowledge- specifically about representation and fusion of ontologies. University professor at the Technological Institute of Oaxaca since October 1992, teaching courses in the areas of: Data base and Information systems, Programming and Technology for the Web and Low level Programming.

Adolfo Guzman-Arenas. Is a computer science professor at Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico City, of which he was Founding Director. He holds a B. Sc. in Electronics from ESIME-IPN, and a Ph.D. degree from MIT; he is an ACM Fellow, an IEEE Life Senior Member, a Member of the Academia de Ingeniería and the Academia Nacional de Ciencias (Mexico). He has received (1996) the National Prize in Science and Technology (Mexico) and (2006) the Premio Nacional a la Excelencia "Jaime Torres Bodet." His work is in semantic information processing and AI techniques, often mixed with distributed information systems.