Automatic Ontology Generation: State of the Art

Ivan Bedini
Orange Labs
ivan.bedini@orange-ftgroup.com

Benjamin Nguyen
PRiSM Laboratory
University of Versailles – France
benjamin.nguyen@prism.uvsq.fr

ABSTRACT

In recent years the growth of internet applications has highlighted the limit of traditional data model representation like UML. Ontologies are means of knowledge sharing and reuse and promise a more suitable knowledge representation for building more "intelligent" applications. In this paper we define the requirements that an ontology must meet in order to fit these new use cases and we provide a meticulous survey with a comparative analysis of experiences and software for automatic ontology generation, investigating in detail which aspects of ontology development can be done automatically and which ones require further research. The main contributions of this paper are the presentation of a new framework for evaluating the automation of ontology generation and an exhaustive comparative analysis of existing software geared towards automatic ontology generation.

Keywords

Automatic Ontology Generation, Semantic Web.

1. Introduction

Nowadays more and more use cases need of a more dynamic machine interpretation of input and output data of applications. With the development of open applications like Web Services and Web 2.0 widgets or also enterprise application integration, existing knowledge representation show its limit [3]. Currently, applications mostly exchange information on the basis of passing parameters or data, formatted according to pre-defined strict syntaxes. We define this approach as the *exactness method*. This method has the advantage of allowing total error management, except application bugs of course, but leaves no space for data interpretation. In consequence, reasoning on data of this type is virtually impossible because of the limits of its definition. Ontologies provide a richer knowledge representation that improves machine interpretation of data. For this they become to be widely used in information systems and applications, and ontology construction has been addressed in several research activities.

A rich quantity of papers describing ontology development and management are available in scientific papers, but the bottleneck of knowledge acquisition remains one of the major problems to be solved. In fact classical development of domain ontology is typically entirely based on strong human participation. It does not adequately fit new applications requirements, because they need a more dynamic ontology and the possibility to manage a considerable quantity of concepts that human can not achieve alone.

For this reason we have investigated most, if not all existing solutions to automatically construct an ontology in a given domain and we have asked ourselves the following questions:

- Is there already an existing system that can do this?
- If an exhaustive system does not exist, how can we use parts of existing systems in order to propose a methodology to achieve this goal?
- Are there any extra parts that need to be developed?

In order to give factual answers to these questions, we provide in this article the following contributions: a new definition for the ontology life-cycle that we adopted for evaluating the ontology generation and a state of the art in automatic ontology generation software with their comparative analysis.

Since, as this paper will show, there are shortcomings in existing solutions, we are currently in the process of developing a new methodology. However, its description is out of the scope of the paper. We will start, in Section 2, with the description of the important aspects of an ontology, with regards to the application integration and the automatic ontology generation process as framework for evaluating research works provided in the a state of the art (Section 4). Section 5 will provide the comparative analysis. Section 6 is a conclusion.

2. Ontology Requirements

In this section we give some ideas on how we can evaluate good or bad ontologies with regards to their structure.

2.1 Definition

In existing literature there are many definitions for ontology [1], [2], [3], [4], which range from antiquity, with Aristotle, to current practices, which fit the computer science domain better. Rather than giving yet another new definition, we align ourselves with the definition that seems to gather the widest consensus:

*An **ontology** is an explicit representation of concepts of some domain of interest, with their characteristics and their relationships.*

We do not focus on the definition, our main interest here is the use that can be made of ontologies in order to develop applications that can share and improve information integration.

2.2 Ontology Requirements

What we are looking for is a knowledge representation that is able to maintain all relevant information for the domain. Thus otology must be able to grow dynamically without bustling existing applications. At the same time computational time for discovering the best matches between several ontologies is expensive, therefore the technique must maintain previous discovered alignments and common usages in order to quickly recognize similarities between concepts and to compute only new information. We decode these characteristics with the following attributes for the ontology: memory, dynamism, polysemy and automation.

2.2.1 Memory

An ontology is designed not only to provide a complete view of domain concepts but also to identify quickly and accurately similarities between concepts, even if not identical, and to conduct consistent alignments. For example a concept like *Address* can be called *Postal Address* or *Delivery Location* depending on application behaviour, but it always represents the same information, a concept of the ontology.

An ontology is not only a classification, or taxonomy of general concepts, it is also a model that includes and maintains the most common properties of concepts, their relationships existing alignments and known semantics.

2.2.2 Dynamism

Identifying new concepts or new semantics, structural and syntactic forms and knowing how to include them in the ontology is another important feature, for two reasons: one is that the similarity search and alignment between concepts is very costly, which heavily penalizes performance in real time; the second is that it is possible to benefit from consecutive alignments, for example, the matching of two concepts is facilitated if we use an intermediary concept.

From this viewpoint ontology an ontology is a dynamic characteristic of the domain, thus evolution should not be a classical versioning system, but more a learning system. We call this feature the *dynamism* of an ontology.

2.2.3 Polysemy

A third characteristic an ontology must have is the ability to provide the polysemeous forms that a term associated to a concept can have. Indeed a term can have several different uses depending on the context. For example, in English the term *Individual* can be used to define *Person* and in another context it can be synonymous with *Alone*. This difference can be detected by making an grammatical analysis of the text to see whether it appears an adjective or a noun, but if the corpus source is not a text, but as in our use case an XML Schema, its meaning must be drawn from its properties only. Thus the concepts must maintain the various groups of common properties and their type.

2.2.4 Automation

A fourth characteristic that we seek is to be able to generate and enrich the ontology automatically. Indeed, even in a specific field, the concepts handled by the applications can be numerous and the quantity of information which we wish to maintain for each concept is vast. Solely relying on human management could quickly become impossible: consider a corpus source made up of a thousand files and the concepts themselves are thousands.

3. Automatic Ontology Generation Process and Evaluation Criteria

The aim of this document is to provide an exhaustive view of the automation aspect of the ontology generation, thus in the rest of this document we focus only on the this requirement defined above and we will provide some elements that we consider crucial in order to achieve this result.

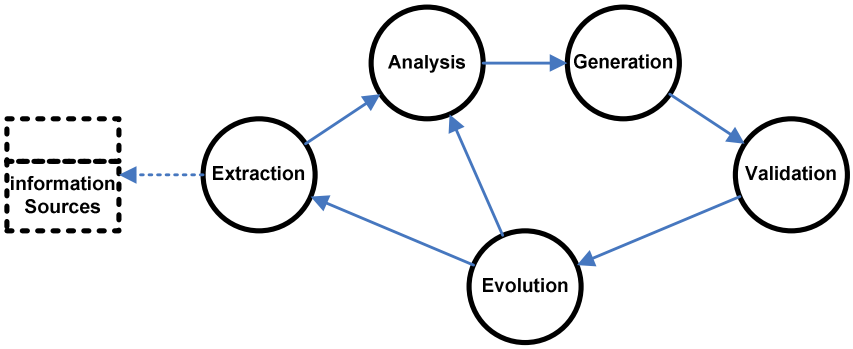

Figure 1 – Automatic ontology generation process

Several methodologies for building ontologies exist, such as OTK [5], METHONTOLOGY [6] or DILIGENT [7], but they target ontology engineers and not machines. We do not develop here a new methodology yet, but we define the **automatic ontology generation life cycle** as a process composed of 5 main steps that we consider necessary to achieve our goal. These steps represent the main tasks of

the process for building ontologies starting from an existing corpus source. In this document we do not focus on what techniques are available for each task, but mainly describe what we expect from a task. The process is depicted in Figure 1, and in detail the 5 steps are:

- **Extraction**: this step provides the acquisition of information needed to generate the ontology (concepts, attributes, relationships and axioms) starting from an existing corpus source. Input resources can be of many types: structured, semi-structured or unstructured. Techniques for information retrieval and extraction can be of different types, such as: NLP (Natural Language Process) techniques, clustering, machine learning, semantics, morphological or lexical and more often a combination of them.
- **Analysis**: this step focuses the matching of retrieved information and/or alignment of two or more existing ontologies, depending on the use case. This step requires: techniques already used in the first stage, as morphological and lexical analysis of labels; a semantic analysis to detect synonyms, homonyms and other relations of this type; an analysis of concepts' structure to find hierarchical relationships and identify common attributes; techniques based on reasoners to detect inconsistencies and induced relations.
- **Generation**: this stage deals with the ontology merging, if appropriate, and the formalization of the meta-model used by the tool in a more general formalism which is interpretable by other applications, such as OWL and RDF/S.
- **Validation**: all previous steps may introduce wrong concepts and relationships, thus an automated validation phase of the result is needed. Conversely, a validation task can be introduced at the end of each previous step. This step is often done by hand, but in some cases validation can be automated.
- **Evolution**: an ontology is not a static description of a domain, but with the evolution of applications, in quality and number, the ontology may also require some changes. The number of concepts as well as properties relationships and other parameters can be added or modified. This operation is considered as an addition of new requirements and as such it could be followed by a new step of information extraction, if new resources are not yet in ontology format, or directly by the analysis step in order to provide new matches and alignments. Anyway, this criterion evaluates the ability of tools to solve and take care of this problem.

Existing literature about ontology generation is rich because of the broad research domains that it involves and it is often difficult to clearly understand who makes what and why, mainly because defined frameworks for evaluation and analysis tend to organize methods according to adopted technologies. Our approach wants to facilitate the understanding of what a method does within the ontology generation life cycle. For this reason this process will also constitute our base framework for evaluating software and experiences of paragraph 5 and will provide us elements to evaluate which part of the process can be automated and how, as well as what techniques are more appropriate, and which part still requires human intervention, thus further research.

4. State of the art of Automatic Ontology Generation

A brief glance at current solutions of automatic ontology building system is enough to understand that we are asking a lot because a few problems still need solving. The purpose of this study is not only to identify existing tools, but also to understand which parts of the generation can be done automatically following our requirements, classify them and define a methodology for this task.

Several *states of the art* are currently available about ontology generation, also referred to as Ontology Learning, but papers focusing on automation for the whole process as defined above are rare. Shamsfard Mehrnoush and Barforoush Abdollahzadeh [8], present a complete framework that classifies software and techniques for building ontologies in six main categories (called *dimensions*). It is a detailed and interesting classification, but it focuses only on the learning method. In [9] the authors

provide a comprehensive tutorial on learning ontology from text, which is really useful, but the considered corpus source does not fit our use case. Euzenat et al. in [10] provide a detailed analysis of technical alignment for ontologies and a state of the art on existing tools, probably the best known matching and alignment software, but they concentrate on the one task of aligning two ontologies already existing at the time, without investigating other steps in the generation process, such as information extraction and the problem of multiple merging. Castano et al. [11] provide a comprehensive and easily understandable classification of techniques and different views of existing tools for ontology matching and coordination, but also limited to the case of two existing ontologies.

In this paper we simply include the complement to the documents cited above, and we overlap on those tools that are closer to our interests.

4.1 Ontology Generation Classification

It appears that ontology generation is mainly hand-made by domain experts, but this approach is of no interest to us. In this paper we have grouped experiences and software in four main categories as follow:

- **Conversion or translation** for those applications that make the hypothesis that an ontology is already well defined by someone or somewhere. What is interesting here is that they prove that the ontology format representation is wider than other common knowledge representation, such as XML or UML, and they also build software that produces this transformation. Experiences show that this approach presents a high degree of automation, but mainly because it does not address the whole problem of the ontology generation, merely a specific task. However it still remains an interesting result to know that if we are in confronted with two different representation formats, the solution is not always complex.
- **Mining based** for those applications implementing some mining techniques in order to retrieve enough information to generate an ontology. Most experiences are focused on unstructured sources, like text documents or web pages and implement Natural Language Processing (NLP) techniques. These experiences tell us that recovering structured concepts from unstructured documents still requires human assistance and that mining techniques from natural text can be used only in complement with other existing structured knowledge representations or techniques.
- **External knowledge based** for those applications that build or enrich a domain ontology by using an external resource. This category may sometime overlaps the mining based because techniques applied to retrieve information can be the same, nevertheless we classify here experiences with an approach closer to the integration of external dictionaries, existing ontology or from a more general knowledge resource, like WordNet [12] or the WWW.
- **Frameworks** for those works providing an approach based on different modules to achieve the goal.

As always when creating a classification the border line is not well defined and in our case applications can present more aspects matching our classification, therefore we classify works with respect to their automation approach rather than with regards to the techniques they implement. In fact we support the thesis that there is not a single technique to develop, but that only an appropriate mix of techniques can bring us to our goal.

In the following paragraphs, we describe the software and experiences, using our classification.

4.2 Conversion / Translation

4.2.1 Mapping XML to OWL Ontologies

Sören Auer of the University of Leipzig (Germany) and Hannes Bohring have developed a tool that converts given XML files to OWL format [13]. It is based on the idea that items specified in the XSD file can be converted to ontology's classes, attributes and so on. Table 1 shows in detail the mapping between these two formalisms. Technically they have developed four XSLT[1] instances to transform XML files to OWL[2], without any other intervention on semantics and structures during the transformation. Finally the OWL file (read ontology) is automatically generated, but under the assumption that concepts were already correctly represented in the source file. This method has been also applied to the Ontowiki platform [14].

Table 1 - XSD to OWL correspondences

XSD	OWL
xsd:elements, containing other elements or having at least one attribute	owl:Class, coupled with owl:ObjectProperties
xsd:elements, with neither sub-elements nor attributes	owl:DatatypeProperties
named xsd:complexType	owl:Class
named xsd:SimpleType	owl:DatatypeProperties
xsd:minOccurs, xsd:maxOccurs	owl:minCardinality, owl:maxCardinality
xsd:sequence, xsd:all	owl:intersectionOf
xsd:choice	combination of owl:intersectionOf, owl:unionOf, owl:complementOf

4.2.2 UML to OWL

Dragan Gasevic et al. [15] advocated the use of UML profiles to extend the possibilities of representation of UML. In this way they get a larger UML representation that overcomes its limitations and that can be *translated* into OWL, again through a system of XSLT instances. As before the hypothesis is that the source of the transformation is complete and well-defined by an expert at an early stage to represent the ontology, the subsequent ontology generation is performed automatically.

4.2.3 Generating an ontology from an annotated business model

The L3I laboratory of the University of Rochelle has developed a semi-automatic ontology generation process [16]. This process starts from a UML class diagram representation of the ontology domain, made by an expert that annotates the elements to be introduced into the ontology. This UML model is then transformed into ODM format[3] as pivot model before automatically generating the ontology in RDFS format. As in the previous case some degree of human intervention is needed at an early stage.

4.2.4 Semi-automatic Ontology Building from DTDs

Within the PICSEL project, a collaboration between INRIA Future and France Telecom, Gloria Giraldo and Chantal Reynaud [17] have developed a semi-automatic ontology generation software for the tourism industry domain extracting information contained in DTD files. This experience is interesting because it goes further, in respect to the XML to OWL transformation seen previously, and shows that *tags and structure of XML files have sufficient information to produce an ontology*. What makes their solution semi-automatic is the fact that the detection of abbreviations or false positives[4] is

[1] Extended Style Sheet Transformations - http://www.w3.org/TR/xslt

[2] OWL - Web Ontology Language Overview. http://www.w3.org/TR/owl-features/

[3] Ontology Definition Metamodel – http://www.omg.org/ontology/

[4] A false positive is a misjudgement detection of a program.

left to an expert during the ontology validation task. This experience is really close to our use case but is limited to the sole domain of tourism, which is defined in advance with great precision, and therefore the detection of relevant concepts does not produce conflicts between different representations.

4.3 Mining based

4.3.1 TERMINAE

Brigitte Biebow et Sylvie Szulman [18] of the University of Paris Nord present the TERMINAE method and tool for building ontological models from text. Text analysis is supported by several NLP tools (such as LEXTER [19]). The method is divided into 4 stages, corpus selection and organisation; linguistic analysis with the help of several NLP tools; normalization according to some structuring principles and criteria; formalization and validation. An expert is called to select the most important notions (concepts) for the targeted ontology from the list of *candidate terms extracted by the tool* and to provide a definition of the meaning of each term in natural language. The new terminological concept finally may or may not be inserted into the ontology, depending on the validity of the insertion.

4.3.2 A method to build formal ontologies from text

Originating from the same University, Jerome Nobécourt has developed a method [20] based on TERMINAE that allows an automation of the insertion of concepts into the ontology by the adoption of ***successive refinements*** of the selected concepts: while the classic TERMINAE approach requires the hypothesis that the ontology is a static property of the domain, the latter introduces a more dynamic environment for domain ontology.

4.3.3 SALT

D. Lonsdaleet al. of Brigham Young University, England, propose a process to generate domain ontologies from text documents [21]. Their methodology requires the use of three types of knowledge sources: one is a more general and well defined ontology for the domain, a dictionary or any external resource to discover lexical and structural relationships between terms and a consistent set of training text documents. With these elements they are able to automate the creation of a new sub-ontology of the more general ontology. User intervention is required at the end of the process because it can generate more concepts then required. This behavior is acceptable because the withdrawal of false positives is easier than adding missing concepts. The authors state that with a large set of training documents their solution can achieve really good results. However the hypothesis of having an upper ontology well defined beforehand proves that the **NLP approach can be used in complement** of the automatic ontology generation process.

4.3.4 Learning OWL ontologies from free texts

He Hu and Da-You Liu from Renmin and Jilin University, China, have developed an automatic generation [22] based on an analysis of a set of texts followed by the use of WordNet. The analysis of the corpus retrieves words as concepts. These words are then searched in WordNet to find the concepts associated with these words. The ontology generation seems to be one of the most automated, but no details of how the terms are extracted from the body is available as well as any qualitative assessment of the work are provided. Nonethelss, it remains an interesting experience to the extent it demonstrates once again that automation is easier if a *more general reference knowledge* already exists, which the authors argue can be represented by WordNet.

4.3.5 Ontology Construction for Information Selection

Latifur Khan and Luo Feng of the University of Texas demonstrate a method to automatically construct an ontology from a set of text documents [30]. Their overall mechanism is as follow: 1) terms are extracted from documents with text mining techniques; 2) documents are grouped hierarchically

according to their similarity using a modified version of SOTA algorithm[5] and then; 3) assign concepts to the tree nodes starting from leaf nodes with a method based on the Rocchio algorithm[6]. Concept assignment is based on WordNet hyponyms[7]. This experience introduces a new *bottom-up* approach for ontology generation that seems to produce good results without any human intervention. The bad news is that it also needs a more general ontology to define concepts for the targeted ontology, but as we can see, this is generally the case of all mining base methods.

4.4 External Knowledge based

4.4.1 Design of the Automatic Ontology Building System about the Specific Domain Knowledge

Hyunjang Kong, Myunggwon Hwang and Pankoo Kim of the University Chosun, Korea, have developed a method [23] based on WordNet. In this method, WordNet is used as a *general ontology* from which they *extract a subset of "concepts"* to build a domain ontology. For example, consider a user trying to generate an ontology on wine. The software will query WordNet using this term and create classes of concepts based on the results of the query. After this initial pass, the user can extend the ontology by entering new concepts to be included. The ontology is then exported to OWL format. Depending on the quality of the starting knowledge resource, this approach will be more or less satisfactory. It is also dependant on the targeted area.

4.4.2 Domain-Specific Knowledge Acquisition and Classification Using WordNet

Dan Moldovan and Roxana Girju from the University of Dallas expose a method for generating ontologies [24] based on WordNet. The approach is almost the same as the previous [23], a user defines some "seed", i.e. concepts of the domain, but with the difference that if a word is not found in WordNet then a *supplementary module will look for it over the Internet*. Then linguistic and mining techniques extract new "concepts" to be added to the ontology. This method automatically enriches its corpus retrieving sentences about the seeds of the ontology that were not found in WordNet. User intervention is necessary here to avoid incongruous concepts.

4.4.3 Enriching Very Large Ontologies Using the WWW

E. Agirre, O. Ansa, E. Hovy and D. Martinez have developed a strategy to enrich existing ontologies using the *WWW to acquire new information* [25]. They applied their approach to WordNet, which is often accused of two flaws: the lack of certain links between concepts, and the proliferation of senses for the same concept. The method takes as input a word which one wants to “improve” the knowledge of. WordNet is questioned about this word, and the different meanings of words are used to generate queries for the web. For each query, that constitutes a “group”, different search engines are queried and the first 100 documents are recovered. Terms frequencies are then calculated and compared with each group, and of course the winning group, (i.e. sense), for the concept is the one with the highest frequencies. In addition a *statistical analysis* is performed on the result, in order to estimate the *most common meaning* of the concept. This method alone can not be adopted to build ontologies, but it has the merit to be able to iterate with an external knowledge base to provide further information that may be used for the validation task of an ontology in absence of human intervention.

---

[5] Joaquin Dopazo and Jose Maria Carazo. Phylogenetic reconstruction using an unsupervised growing neural network that adopts the topology of a phylogenetic tree. Journal of Molecular Evolution, Volume 44(2) :226/233, 02 1997.

[6] Thorsten Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In Douglas H. Fisher, editor, Proceedings of ICML-97, 14th International Conference on Machine Learning, pages 143/151, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

[7] A word that denotes a subcategory of a more general class. Opposite of hypernym.

4.4.4 A new Method for Ontology Merging based on Concept using WordNet

Miyoung Cho, Hanil Kim and Pankoo Kim from Chosun and Cheju Universities, Korea, present the problem of proximity between two ontologies as a choice between alignment and merging [26]. The first case is limited to establishing links between ontologies while the second creates a single, new ontology. With their experience they directly merge two ontologies based on WordNet. For this they use two approaches in their method that they call the horizontal approach and the vertical approach. The horizontal approach first checks fall the relationships between concepts of the "same level" in the two ontologies and merges or ties them as defined by WordNet, while the vertical approach completes the merging operation for concepts with "different levels", but belonging to the same branch of the tree. In this case they fill the resulting ontology with concepts from both ontologies and do not make a choice. A similarity measure is calculated in order to define the hierarchy between these concepts in the resulting tree.

This method, while not providing an adequate solution to automation, does provide a *purely semantic approach* to the merging solution.

4.4.5 A Method for Semi-Automatic Ontology Acquisition from a Corporate Intranet

Similar to [21], Joerg-Uwe Kietz, Alexander Maedche and Raphael Volz [27] describe a generic approach for the creation of an ontology for a domain based on a source with multiple entries which are: a generic ontology to generate the main structure; a dictionary containing generic terms close to the domain; and a textual corpus specific to the area to clean the ontology from wrong concepts.

This approach combines several input sources, allowing great generality and a better reliability of the result. The user must manually check the ontology at the end of the generation process.

4.5 Frameworks

4.5.1 Symontox: a web-ontology tool for e-business domains

SymOntoX [28] is an OMS (Ontology Management System[8]) specialised in the e-business domain, which provides an editor, a mediator and a versioning management system. With SymOntoX the creation of the ontology is mainly done by an expert using the editor. But the framework contains a first step towards an easier generation: it contains high-level *predefined concepts* (such as Business Process, Business Object, Business Actor, etc.), as well as *different modules* used for ontology mapping and alignment to simplify the work of the expert. Here, ontology generation is merely assisted.

4.5.2 Protégé

Protégé [31] is a free, open source, platform to design ontologies. Developed by the Stanford Medical Informatics group (SMI) at the University of Stanford, it is supported by a strong community and experience shows that Protégé is one of the most widely used platforms for ontology development and training. This software has an *extensible architecture* which makes it possible to integrate plug-ins[9]. Some of these modules are interesting and relevant in our case, like those from the PROMPT Suite [32]. They automate, or at least assist, in the mapping, merging and managing of versions and changes. Also the related project Protégé-OWL offers a library of Java methods (API-Application-Programming Interface) to manage the open-source ontologies formats OWL (Web Ontology Language) and RDF (Resource Description Language).

The glue between these pieces of software still remains human, yet program modules and libraries provide a fundamental basis for developing the automation of ontology generation.

---

[8] Ontologie Managment System. http://sw-portal.deri.at/papers/deliverables/d17_v01.pdf.

[9] A hardware or software module that adds a specific feature or service to a larger system.

4.5.3 Ontology Learning Framework

Alexander Maedche and Steffen Staab at the University of Karlsruhe, Germany, are contributors of several interesting initiatives within the ontology design field as well as the automation of this process, like the MAFRA Framework [34], Text-To-Onto [35] and KAON [36]. In this paper we focus on their framework for ontology learning [33].

They propose an ontology learning process that includes five steps: import, extraction, pruning, refinement, and evaluation. This approach offers their framework a flexible architecture that consists of many extensible parts, such as: a component to manage different input resources, capable of providing information extraction from a large variety of formats (UML, XML, database schema, documents text and web); a library of algorithms for acquiring and analyzing ontology concepts; a graphical interface that allows users to modify the generated ontology, but also to choose which algorithms to apply and treatments to perform.

They bring together many algorithms and methods for ontology learning. Despite their framework not allowing a completely automatic generation process, they are the only people to propose a *learning process* close to a methodology of automatic ontology generation.

4.5.4 LOGS

A group of researcher from Kansas State University has developed LOGS (Lightweight universal Ontology Generation and operating architectureS) [29]. They state that generating ontology automatically from text documents is still an open question. Therefore they developed LOGS with a modular architecture that integrates the core functionality that can be expected by automatic ontology building software. It consists of the following modules: document source parser, NLP engine, analyser, ontology engine, interface, integrator, ontological database and dictionary. It also contains other modules able to crawl an intranet, to refine the process of ontology design and a module implementing trial and error iterative analysis of related texts to find known patterns. Although no qualitative analysis is provided, the authors argue that they obtained significant results.

Unfortunately this software seems to have not met great consensus within the community.'

5. Comparative Analysis and Discussion

Works presented above are only a part of all studied experiences; nevertheless they represent a significant sample covering the essential steps in the generation of ontologies. We now provide a comparative analysis of methods following the 5 steps composing the automatic ontology generation process defined in section 3. Our exercise consists to focus over experiences that have implemented steps of this process and analysing results in order to understand what are strengths and weaknesses of each approach.

Firstly we can note that modules implementing a step have a different degree of automation, which can not be measured exactly. It should also be noted that qualitative results were not always available and when conducting this assessment only 3 tools presented in this paper were both freely available and able to process XML Schema files (as required by our use case), and therefore specifically tested by us. These are Protégé, XML2OWL and MAFRA. Despite this lack of availability, the purpose of this study is mainly theoretical, thus information obtained by public material was enough to perform a qualitative evaluation. Values are assigned to each step according to the following criteria:

- A (marked with'-') – when step is not developed ;
- B – for solutions using a semi-automatic approach ;
- C – for solutions where human intervention is optional;
- D – for solutions that are, a priori, completely automatic.

The analysis of Table 2 below draws several remarks about ontology generation automation.

Information extraction can reach good results. The most studied input corpora are for text documents, a lot of information can be reached from this type of corpus source. Methods based on this type of resource have the advantage to have a lot of resources, that can be found over Internet or an Intranet, and that several tools for NLP and mining are available. Nevertheless they require a most important human validation task and are preferred for defining a high level definition of concepts. Structures, like classes, attributes and relationships, are mostly provided by other external resources. Thus mining directly structured documents can reach better results with less validation, but not so much methods deepening study this approach.

To this end WordNet surely deserves some special attention because we can observe that it is an essential resource for the automation process. In fact it plays different roles. The first is that of an electronic dictionary and thesaurus, which is fundamental. The other is that of a reference ontology, mainly by using its sibling or hierarchical terms discovery, with relationships like hyponym, meronym, holonym and hyperonym. But for this WordNet has the drawback of being too generic and not adapted to specific domain ontology development. Even so, it remains an important module to further be developed.

Matching and alignment modules are the most challenging tasks but, as told in [37], they are growing and methods and techniques in the future should achieve valuable results. For this because the complexity of the development of such modules the best should be to have these modules available as shared libraries.

Merging, which is strictly related to alignment, is currently implemented with two input ontologies, thus multi ontology alignment and merging seems to be an open question yet to be investigated in detail. This point could be resolved with consecutives mergings, but it appears that the final ontology can be different depending on the sequence in which the ontologies are merged.

Validation is almost always human and only automatic consistency checking has been implemented. The only solution to improve it, is to limit its range, thus: adopting a bottom-up approach, which has shown better results; to use successive refinements and reasoners, in order to guarantee consistency in the resulting ontology and; by querying external resources like Watson [38], rather than the WWW directly, that provides the great advantage of returning structured information, which is more suitable for machine interpretation. It could even be left to be managed by applications, by improving the exception management due to bad alignment for example.

Evolution management is still rare. Some methods manage versions and other go further and provide automatic detection of changes. But in reality what we are really looking for is an ontology able to grow and not a static adaptation of some knowledge representation.

One important aspect is that most successful solutions integrate different resources for retrieving information and also as reference knowledge for detecting wrong alignments. Thus building reference ontologies, or others reference knowledge representations seems to gather the most important point to be further developed.

As final consideration we can say that most methods offer automations of only some steps of the generation process. Modular solutions, rather then monolithic applications should offer a better architecture for covering the larger part of the ontology life cycle, although integration of steps is mostly manual.

Now, in order to be able to fulfil our goal, there still remains a lot of work that we could divide into three main actions: i) one is the automatic construction of a dynamic reference ontology; ii) the second is to build applications able to integrate this new approach (that we could call *semantic method* in opposition to the *exactness method* seen in section 1) and further investigating the new types of exceptions that it could involve; iii) and more in the semantic web area to further develop a new methodology for automatic ontology generation and to provide the definition of modules for steps defined in paragraph 3.

Table 2 - Comparative analysis of methods

	Extraction	Analysis	Generation	Validation	Evolution
Generating an ontology from an annotated business model	- Human	-	C – No merging. Direct transformation using XSLT files.	- Human, upstream to the generation	-
XML2OWL	B – Static table of correspondences	-	C – No merging. Direct transformation using XSLT files.	- Human, upstream to the generation	-
UML2OWL	B	-	C – No merging. Direct transformation using XSLT files.	- Human, upstream to the generation	-
Semi-automatic Ontology Building from DTDs	C – automatic extraction from DTD Sources	B – structure analysis without alignment	C – No standard ontology representation	- Human	-
Learning OWL ontologies from free texts	C – Text sources. NLP techniques. WordNet as resource dictionary/ontology	-	C – OWL format	-	-
Ontology Construction for Information Selection	C -	-	C	-	-
TERMINAE	C – Text sources. NLP techniques	B – Concept relationships analysis	C – No standard ontology representation	- Human	-
SALT	D – Text sources. NLP techniques. Multi entries.	C – Similarity analysis of concepts	B – No standard ontology representation	B –Limited human intervention	-
A new Method for Ontology Merging based on Concept using WordNet	-	B	C – Automatic merging. No standard ontology representation.	-	-
Design of the Automatic Ontology Building System about the Specific Domain Knowledge	B – Main concept defined by a domain expert.	-	C	-	-
Enriching Very Large Ontologies Using the WWW	C – Enrich existing ontology	-	C	-	-
Domain-Specific Knowledge Acquisition and Classification Using WordNet	C – Main concept defined by a domain expert.	B – Grammatical analysis of text	C	- Human	-
A Method for Semi-Automatic Ontology Acquisition from a Corporate Intranet	C – NLP techniques. Multi entries source.	B – Meaning analysis of concepts	B	B – User required for undecidabe cases	B – Cyclic approach can manage evolutions
SymOntoX	-	C – Matching analysis	B - Provide some predefined concepts.	- Human	B – Manage versions, but still human.
Protégé (Mainly from plug-in)	B – extraction from Relational DB and some XML format	D – Matching and Alignment analysis.	B – Assisted merging. Export in several ontology formats.	- Human	C – Ontology evolution detection
LOGS	C – Text source analysis. NLP engine. Morphological and semantic analysis. Machine learning approach for rules.	C – Similarity based on concepts and relationships analysis.	C – Different format. Internal ontology structure based on a lattice.	B – Validation at the end of each module	-
Ontology Learning	D – Extraction from several formats (XML, UML, OWL, RDF, text…). NLP, Semantic and lexical analysis. Multi entries source.	C – Libraries for clustering, formal concept analysis and associations rules	C - OWL and RDF/S	B - Assisted	-

6. Conclusion

An ontology in most use cases is not a static behaviour of the domain, but should be able to guarantee the natural evolution of its domain. Neglecting automation would fail the adoption of ontologies for several use cases, since we would only be developing yet another static knowledge representation, *yet another standard.*

Let us answer to our opening questions:

• Is there already an existing system that can do this? Not yet. We have tried to develop ontology with a corpus of XML files using available software, but we have not be able to do this automatically.

• How can we use parts of existing systems in order to propose a methodology to achieve this goal? Different kinds of approaches can be considered to achieve our goal, such as multi-entries information extraction, bottom-up development, modular architecture, and looking for existing modules for format transformations, matching and alignment seen in this paper.

• Are there any extra parts that need to be developed? Yes, indeed. We think that further work is needed in the semantic web area and semantic applications development. Specific work should be done for: 1) XML Mining techniques for information extraction/retrieval; 2) the development complex matching approaches for alignments, maybe with simpler use cases and starting to collect common concepts; 3) working with more than 2 input ontologies because ontology merging may be uncomfortable; 4) limiting the validation task as much as possible introducing different kinds of input resources and; 5) the development of specialised modules and libraries for each step of the generation process.

7. REFERENCES

[1] Jean Charlet, Bruno Bachimont and Raphaël Troncy. Ontologies pour le Web sémantique. In Revue I3, numéro Hors Série «Web sémantique», 2004.

[2] C. Welty. Ontology Research. AI Magazine, 24(3), 2003

[3] Fensel, D. Ontologies: Silver bullet for knowledge management and electronic commerce. Springer-Verlag, Berlin, 2001

[4] Asuncion Gomez Perez and V. Richard Benjamins. Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods. IJCAI-1999, Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends

[5] York Sure, Rudi Studer. On-To-Knowledge Methodology Final Version Project Deliverable D18, 2002.

[6] Lopez, M.F., "Overview of the methodologies for building ontologies". Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden, August 1999.

[7] Denny Vrandecic, H. Sofia Pinto, York Sure, Christoph Tempich. The DILIGENT Knowledge Processes. Journal of Knowledge Management 9 (5): 85-96. October 2005. ISSN: 1367-3270.

[8] Shamsfard Mehrnoush, Barforoush Abdollahzadeh. The State of the Art in Ontology Learning: A Framework for Comparison. The Knowledge Engineering Review, Volume 18, Issue 4. December 2003

[9] Paul Buitelaar, Philipp Cimiano, Marko Grobelnik, Michael Sintek. Ontology Learning from Text Tutorial. ECML/PKDD 2005 Porto, Portugal; 3rd October - 7th October, 2005. In conjunction with the Workshop on Knowledge Discovery and Ontologies (KDO-2005)

[10] Euzenat J., Le Bach T., Barrasa J., Bouquet P., De Bo J., Dieng R., Ehrig M., Hauswirth M., Jarrar M., Lara R., Maynard D., Napoli A., Stamou G., Stuckenschmidt H., Shvaiko P., Tessaris S., Van Acker S. & Zaihrayeu I., State of the Art on Ontology Alignment. Knowledge Web Deliverable #D2.2.3, INRIA, Saint Ismier, 2004.

[11] S. Castano, A. Ferrar, S. Montanelli, G. N. Hess, S. Bruno. State of the Art on Ontology Coorination and Matching. Deliverable 4.4 Version 1.0 Final, March 2007. BOEMIE Project

[12] Miller, G.A. (1995). WORDNET: A lexical database for English. Communications of ACM (11), 39-41.

[13] Bohring H, Auer S. Mapping XML to OWL Ontologies, Leipziger Informatik-Tage. 2005: 147-156.

[14] Auer, S.; Dietzold, S.; Riechert, T. OntoWiki – A Tool for Social, Semantic Collaboration. 5th International Semantic Web Conference, Nov 5th-9th, Athens, GA, USA. In I. Cruz et al. (Eds.): ISWC 2006, LNCS 4273, pp. 736–749, 2006.

[15] D. Gasevic, D. Djuric, V. Devedzic, and V. Damjanovic. From UML to ready-to-use OWL ontologies. In Intelligent Systems, 2004. Proceedings. 2nd International IEEE Conference, volume 2, pages 485-490, jun 2004.

[16] Frédéric Bertrand and Cyril Faucher and Marie-Christine Lafaye and Jean-Yves Lafaye and Alain Bouju. Génération d'une ontologie à partir d'un modèle métier annoté. In proceedings IDM 06, Jun 2006

[17] Gloria Giraldo, Chantal Reynaud, Construction semi-automatique d'ontologies à partir de DTDs relatives à un même domaine, 13èmes journées francophones d'Ingénierie des Connaissances, Rouen, 28-30 Mai 2002

[18] Biebow, B., Szulman, S.: TERMINAE: A linguistics-based tool for the building of a domain ontology. In Proc. of EKAW'99, (1999)

[19] D. Bourigault. Lexter, a natural laguage processing tool for terminology extraction. In Procedings of the 7th EURALEX InternationalCongress, Goteborg, 1996.

[20] Nobécourt J. A method to build formal ontologies from texts. InWorkshop on ontologies and text, Juan-Les-Pins, France, 2000.

[21] D. Lonsdale and Y. Ding and D. Embley and A. Melby. Peppering knowledge sources with SALT : Boosting conceptual content for ontology generation, 2002.

[22] He Hu and Da-You Liu. Learning OWL ontologies from free texts. In Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on, volume 2, pages 1233_1237, 2004.

[23] Hyunjang Kong, Myunggwon Hwang, and Pankoo Kim. Design of the automatic ontology building system about the specific domain knowledge. In Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference, volume 2, feb 2006.

[24] Dan I. Moldovan and Roxana Girju. Domain-specific knowledge acquisition and classification using wordnet. In Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference, pages 224/228. AAAI Press, 2000.

[25] E. Agirre and O. Ansa and E. Hovy and D. Martinez. Enriching Very Large Ontologies Using the WWW. in Proc. of the Ontology Learning Workshop, ECAI, Berlin, Germany, 2000.

[26] Miyoung Cho, Hanil Kim, and Pankoo Kim. A new method for ontology merging based on concept using wordnet. In Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference, volume 3, pages 1573/1576, February 2006.

[27] Kietz, J.; Maedche, A. and Volz, R. A Method for Semi-Automatic Ontology Acquisition from a Corporate Intranet. In Proceedings of EKAW-2000 Workshop Ontologies and Text, Juan-Les-Pins, France, October 2000.

[28] M. Missikoff and F. Taglino. Symontox : a web-ontology tool for ebusiness domains. In Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on, pages 343/346, 2003.

[29] Raghunathan, P., “Fast semi-automatic generation of ontologies and their exploitation”, Department of Computer Science, Technical Report, Kansas State University, 2003

[30] Latifur Khan and Feng Luo. Ontology Construction for Information Selection. In ICTAI '02: Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'02), page 122, Washington, DC, USA, 2002. IEEE Computer Society.

[31] N. F. Noy, R. W. Fergerson, & M. A. Musen. The knowledge model of Protege-2000: Combining interoperability and flexibility. 2th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000), Juan-les-Pins, France, 2000.

[32] N. F. Noy, M. A. Musen. The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping. International Journal of Human-Computer Studies, 2003.

[33] Alexander Maedche and Steffen Staab. Ontology Learning for the Semantic Web. IEEE Intelligent Systems, 16(2): 72/79, 2001.

[34] Maedche, A., Motik, B., Silva, N., and Volz, R.. MAFRA - Mapping Distributed Ontologies in the Semantic Web. Proc. 13th European Conf. Knowledge Eng. and Management (EKAW 2002), Springer- Verlag, 2002, pp. 235–250.

[35] Maedche, A. and Staab, S.: The Text-To-Onto Ontology Learning Environment. Software Demonstration at ICCS-2000 - Eight International Conference on Conceptual Structures. August, 14-18, 2000, Darmstadt, Germany.

[36] E. Bozsak, M. Ehrig, S. Handschuh, A. Hotho, A. Mädche, B. Motik, D. Oberle, C. Schmitz, S. Staab, L. Stojanovic, N. Stojanovic, R. Studer, G. Stumme, Y. Sure, J. Tane, R. Volz, V. Zacharias. KAON - Towards a large scale Semantic Web. In: Proceedings of EC-Web 2002. Aix-en-Provence, France, September 2-6, 2002. LNCS, Springer, 2002.

[37] J. Euzenat, M. Mochol, P. Shvaiko, H. Stuckenschmidt, O. Svab, V. Svatek, W. van Hage, and M. Yatskevich. Results of the ontology alignment evaluation initiative 2006. In Proceedings of the ISWC workshop on Ontology Matching, pages 73–95, 2006.

[38] Mathieu d'Aquin, Claudio Baldassarre, Laurian Gridinoc, Sofia Angeletou, Marta Sabou, and Enrico Motta. Watson: A Gateway for Next Generation Semantic Web Applications. Poster session of the International Semantic Web Conference, ISWC 2007.

[39] Niles, I & Pease A. (2001) “Towards A Standard Upper Ontology.” In Proceedings of FOIS 2001, October 17-19, Ogunquit, Maine, USA.