Research Journal of Applied Sciences, Engineering and Technology 6(16): 2993-3000, 2013
DOI:10.19026/rjaset.6.3684
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: January 08, 2013 Accepted: January 31, 2013 Published: September 10, 2013

Research Article

An Analysis of Ontology Engineering Methodologies: A Literature Review

Rizwan Iqbal, Masrah Azrifah Azmi Murad, Aida Mustapha and Nurfadhlina Mohd Sharef
Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

Abstract: It is now widely accepted that ontologies play a critical role in achieving the goal of machine understandable web, also known as semantic web. In order to develop ontologies, several methodologies have been proposed during the last two decades. Despite the fact, that quite a number of ontology engineering methodologies have been proposed, still the field lacks widely accepted and mature methodologies. Most methodologies lack sufficient details of techniques and activities employed in them. However, some methodologies provide sufficient details including METHONTOLOGY. This article discusses and reports a critical analysis and comparison of these methodologies. The analysis is performed based on a criterion, derived from related literature, trends and needs which evolved over the years. The results of the analysis showed that there is no completely mature methodology. Therefore, this research may act as a preliminary guide to come with a state of art ontology engineering methodology, bridging up the existing gaps and shortfalls.

Keywords: Knowledge engineering, ontology engineering, ontologies, semantic web

INTRODUCTION

The future of machine understandable web, the semantic web has its root deep down in to ontologies. Ontologies explicitly define the concepts in a domain and the relationships between those concepts. They are critical for many knowledge based applications since they are the formal model and detailed machine understandable description of a domain, which can be utilized in many ways. There are several definitions available to define ontology in different fields. In the area of information science, according to Smith (2003) definition of ontology is 'an ontology is a dictionary of terms formulated in a canonical syntax and with commonly accepted definitions designed to yield a lexical or taxonomical framework for knowledge representation which can be shared by different information systems communities'.

In the last two decades, ontologies have gain considerable attention and focus in the world of research. In the present day ontologies are extensively used in different domains like knowledge engineering, artificial intelligence, natural language processing, e-commerce, intelligent information integration, information retrieval, database design and integration, bio-informatics and etc. In order to support the development of ontologies several methodologies have been proposed to date, facilitating the process of ontology development or ontology engineering.

Ontology engineering is the discipline that investigates the principles, methods and tools for creating and maintaining ontologies. An ontology engineering methodology caters the methodological aspect of ontology development. It gives a set of guidelines and activities to develop ontologies. In order to assist ontology engineers and domain experts in building ontologies several ontology engineering methodologies have been proposed to date. Some methodologies were initially proposed, while some emerged as a result of experience and insights attained during development of ontologies for different projects.

This study will discuss the ontology engineering methodologies developed over the years. The study will not only discuss these methodologies, but will analyse each methodology based on a set criterion, providing a deeper and critical insight to methodologies developed to date. The organization of the paper is as follows. Next section is the related work. Then the criteria for analysing ontologies will be discussed, followed by discussion and analysis. Finally, the study is concluded by the conclusions and future work section.

LITERATURE REVIEW

In the field of ontology engineering, many ontology engineering methodologies have been developed to date. The available methodologies have either been proposed initially or emerged from experiences and insights attained during ontology

Corresponding Author: Rizwan Iqbal, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

development for different projects. Despite the fact, that quite a number of ontology development methodologies have been proposed over the last two decades, still the field lacks widely accepted methodologies (López, 1999). There might be many reasons for this but one of the main reasons is that mostly methodologies were applied for developing ontology for a project, which does not unveil much insight to encourage others to adopt it. Below is a brief discussion on ontology development methodologies which evolved over the passage of time.

Methodology developed by Ushold and King was based on the experience for developing the Enterprise Ontology (Uschold and King, 1995). The Enterprise Ontology is a collection of terms and definitions relevant to business enterprises. Ushold and king were the first who felt the need to propose a methodology for the purpose of developing ontologies (Fernández-López and Gómez-Pérez, 2003). Previously, only guidelines were considered sufficient for this purpose. Though it was the first methodology proposed for ontology creation, it does not precisely describe the techniques and activities (López, 1999).

Similarly, Grüninger and Fox proposed a methodology related to the domain of business (Gruninger and Fox, 1995). It was proposed on the experience of creating the TOVE project ontology. The methodology first focuses to capture the ontology requirements by means of informal description. Later, informal description is transformed to formal language, which is a computable model expressed in first order logic. Motivation scenarios are used to capture the informal intended semantics which should be introduced in the ontology (Gruninger and Fox, 1995). From these motivation scenarios evolve the competency questions, they are considered as expressive requirements which the ontology should answer. Like the methodology proposed by Uschold and King (1995), activities and techniques used in this methodology also lack sufficient details and remain abstract (López, 1999).

METHONTOLOGY methodology was introduced for ontology engineering to build domain ontologies from scratch (Fernández-López *et al*., 1997). Unlike methodologies discussed by Uschold and King (1995) and Gruninger and Fox (1995), METHONTOLOGY covers the employed activities and techniques in detail. It supports the creation of ontologies at knowledge level and follows a life cycle based on evolving prototypes. It includes development oriented activities, namely specification, conceptualization, formalization, integration and implementation. In parallel to development oriented activities some support activities are also part of it. Support activities include knowledge acquisition, evaluation, integration and documentation. In addition, it has been tested by developing ontologies for different domains like chemicals (Fernández, 1996; Gómez-Pérez, 1996; López *et al*., 1999), environmental pollutants (Rojas, 1998; Gómez-Pérez and Rojas, 1999), monatomic ions , Silicate ontology, to name a few (Fernández-López and Gómez-Pérez, 2003; Arpírez and Gómez-Pérez, 2000). Furthermore, it supports the notion of reusability.

Like METHONTOLOGY, IDEF5 is another methodology that follows an evolving prototype model and is application independent in nature (KBSI, 1994). This methodology supports gradual refinement through the use of two representation languages. Initially the ontology is defined using a schematic language, based on graphical notation to express the most common ontological information. The schematic language is used as a means of communication between the domain expert and the ontology developer. The initial representation is later analyzed and transformed in to a structured language based on KIF (KBSI, 1994). On the one hand a useful addition in IDEF5 is the library maintained for commonly used relations. This library contains definitions and characterizations of classification relations, meronymic relations, temporal relations, spatial relations, influence relations, dependency relations and case relations (KBSI, 1994; Jones *et al*., 1998). On the other hand, it does not recommend a life cycle and provide limited details of the methodology.

The importance of reusability in the field could not be denied as ontology development is a time consuming and tedious job. To overcome this bottleneck Ontolingua server was introduced (Farquhar *et al*., 1995). The Ontolingua server provides a library of previously defined ontologies to its users. The users can reuse and redesign existing ontologies and further extend the library by adding their new ontologies to the server. The guides for using Ontolingua comprises of advices on browsing, developing, maintaining and sharing of ontologies stored at the server end. Another distinguishing feature of Ontolingua is its provision for collaborative ontology construction (Farquhar *et al*., 1995). However, Ontolingua seem to ignore details about mapping functions that convert from one ontology to another (Jones *et al*., 1998).

On the same footsteps as Ontolingua, CYC (Lenat and Guha, 1990) and SENSUS (Swartout *et al*., 1996) supports the notion of reusability and redesign and both of them relate to the domain of natural language processing. SENSUS ontology comprises of more than 50,000 concepts in a hierarchy consisting of terms belonging to different levels of abstraction but it does not specifically cover a particular domain. It was

developed using various sources of knowledge including PENMAN Upper Model, ONTOS, WordNet and some electronic dictionaries (English, Spanish and Japanese) (Swartout *et al*., 1996). In the case of developing domain ontology, some terms are selected as seeds relevant to a particular domain. These seed terms are manually linked to SENSUS. The final ontology includes only the seed terms extended to the root of SENSUS, all irrelevant terms are pruned (Swartout *et al*., 1996; López, 1999). However, no particular techniques and a lot of details are mentioned (López, 1999).

The CYC methodology (Lenat and Guha, 1990) emerged from the experience of developing the CYC knowledge base. The ontology is the core component of the CYC knowledge base. This ontology has been developed to represent enough common sense and encyclopedic knowledge, sufficient to provide natural language capabilities to natural language understanding systems, expert systems and machine learning systems (Lenat and Guha, 1990). CYC methodology is based on three phases. First phase requires manual coding, second phase proposes knowledge codification aided by tools and third phase relies majorly on the tools for work requiring little human intervention (Fernández-López and Gómez-Pérez, 2003). Both CYC and SENSUS do not recommend a life cycle and lack the details for pre and post development processes.

Mikrokosmos is another project in the domain of natural language processing (Mahesh, 1996). It was based on developing ontology particularly for the purpose of machine translation. The project has discussed about the methodology applied during the ontology development process. Among the set of guidelines proposed, some guidelines were general in nature and can be applied to other domains. Although the methodology contains some general development guidelines and useful heuristics for making design decisions, another more comprehensive methodology would be more appropriate for other domains.

The Plinius project (Mars *et al*., 1994) is somewhat similar to Mikrokosmos project (Mahesh, 1996). The project developed ontology to assist translation of natural language text into expressions in a knowledge representation language. The approach is based on engineering decisions and recommends adoption of cost benefit analysis for inclusion of concepts. Like Mikrokosmos, it included some general guidelines applicable to other domains but these guidelines might not always be feasible or comprehensive enough to cater other domains (Jones *et al*., 1998).

Methodologies supporting the introduction of knowledge management solutions into enterprises are important. Common KADs is a widely used methodology for developing knowledge based systems in which ontologies play a vital role (Schreiber *et al*., 1995; Wielinga *et al*., 1994). The KACTUS project was a follow up project which focused to investigate the following issues: feasibility of knowledge reuse in complex technical systems and the role of ontologies to support it (Schreiber *et al*., 1995). The approach is conditioned by applications development. So, whenever an application is built, an ontology encapsulating the knowledge required for creating the application is also constructed. KACTUS follows an engineering approach, with emphasis on modular design, ontology redesign and reuse. Although it supports reusing ontologies, the work on mapping functions is quite limited (Jones *et al*., 1998).

Besides CommonKADS, another methodology specifically handling enterprise solutions is On-To-Knowledge methodology (Sure *et al*., 2003). This methodology presented an approach which intends to bring a balance between human problem solving and automated IT solutions. The methodology focuses on knowledge meta process and knowledge process. A sample case study was also conducted, illustrating the instantiation of knowledge meta process in all the phases of ontology development.

Methodologies including ONIONS (ONtologic Integration Of Naive Sources (Gangemi *et al*., 1996; Steve and Gangemi, 1996) and MENELAS (Bouaud *et al*., 1994) have been applied for the medical domain. ONIONS addressed the issue of integrating heterogeneous sources of information in knowledge acquisition. The proposed solution does not focus on the final representation of the ontology but it focuses on problems related to ontology acquisition. It particularly focuses on ontology acquisition issues including modeling stopover (how to control over refining an ontology?) and knowledge relevance (stating what is conceptually relevant?).

MENELAS ontology was designed as part of a natural language understanding system in the medical domain (Bouaud *et al*., 1994). The methodology used in developing MENELAS ontology used conceptual graphs as its core formalism. It includes four principles useful in the development of taxonomic knowledge namely similarity, specificity, opposition and unique semantic access. These principles greatly assist in controlling the acquisition and structuring of the ontology. The first two relations relate to Aristotle's definition principles by genus and differentiae, while the remaining caters with relations between the siblings. However, this methodology presumes an idealized view of taxonomies which makes it incompatible to many domains (Jones *et al*., 1998).

101 method is another guide for developing domain ontologies (Noy and McGuinness, 2001). It is iterative in nature and based on some fundamental rules which assist in making design decisions during

ontology development. The guide sequentially covers all the phases of ontology development, including complex issues related to defining class hierarchies and properties of classes and instances. For explanation and elaboration purposes the authors have extensively used a wine ontology throughout the guide (Noy and McGuinness, 2001). Though the methodology covers some critical design issues, life cycle recommendation seems to be absent.

So far the methodologies discussed above refer to well-known and widely used standards from the areas like software engineering and knowledge representation. UPON is another ontology development methodology derived from the Unified Software Development Process (Nicola *et al*., 2005). The methodology takes advantage of the Unified Process (UP) and adopts the Unified Modeling Language (UML) as well. Adoption of these techniques makes the ontology development process handier, for both the domain experts and knowledge engineers. Ontology development using UPON consists of cycles, phases, iterations and workflows, it follows the UP (Unified Process) paradigm. The use-case driven, iterative and incremental nature of UPON makes it unique from other processes, respectively for software and ontology engineering (Nicola *et al*., 2005). However, it does not provide comprehensive details and neglects the collaborative construction aspect.

DISCUSSION

The previous section briefly talks about different ontology engineering methodologies, which emerged over the years. It was revealed during literature review that different methodologies lay stress and focus on distinct aspects of ontology development. For example, some methodologies focus a lot on domain analysis and scope identification, but at the same time lack due attention on the design phase, which is as equally important as the prior phases. In the same way, some methodologies talk about covering distinct phases of ontology development, but their documentation does not report about particular techniques which should be employed during these phases.

During the literature review pros and cons of different methodologies were identified and a criterion was needed to analyze and compare the methodologies. Therefore, a criterion is established for analyzing and comparing different ontology engineering methodologies. The criterion is established after reviewing the related literature and observing the trends and needs which evolved over the years in the field of ontology engineering. The criteria cover eight different aspects of any ontology engineering methodology. The defined criteria will allow readers to develop a quick understanding of different methodologies. This will

Table 1: Comparison of methodologies based on the established criterion

Methodologies	Type of development	Collaborative construction	Reusability support	Degree of application dependency	Life cycle recommendation	Strategies for identifying concepts	Methodology details	Interoperability support
TOVE	Stage based	No	Yes	Application semi independent	No	Middle out strategy	Some details	No
Enterprise model approach	Stage based	No	Yes	Application independent	No	Middle out strategy	Some details	No
METHONTOLOGY	Evolving prototype	No	Yes	Application independent	Yes	Middle out strategy	Sufficient details	No
KBSI IDEF5	Evolving prototype	No	Yes	Application independent	No	Not clear	Some details	No
Ontolingua	Modular development	Yes	Yes	Application independent	No	Not clear	Some details	Yes
Common KADS and KACTUS	Modular development	No	Yes	Application dependent	No	Top down strategy	Insufficient details	No
PLINIUS	Guidelines	No	No	Application independent	No	Bottom up strategy	Some details	No
ONIONS	Modular development / Guidelines	No	No	Application dependent	No	Not clear	Insufficient details	Yes
Mikrokosmos	Guidelines	No	No	Application dependent	No	Rule based strategy	Some details	No
MENELAS	Guidelines	No	No	Application dependent	No	Concepts Graphs (CG)	Insufficient details	No
SENSUS	does not mention any preference	Yes	Yes	Application semi independent	No	Bottom up	Some details	Yes
Cyc methodology	Evolving prototype	No	Yes	Application independent	No	Not clear	Some details	No
UPON	Evolving prototype	No	Yes	Application independent	Yes	Middle out strategy	Some details	No
101 method	Evolving prototype	No	Yes	Application independent	No	Developer's consent	Some details	No
On-To-Knowledge	Evolving prototype	No	No	Application dependent	Yes	Middle out strategy	Some details	No

also facilitate the readers to choose the right methodology for their projects, depending on the project needs and preferences/priorities.

Criteria for analysis: The detail of criteria for analyzing and comparing ontology engineering methodologies is defined below. The first four aspects of the criteria namely, type of development, support for collaborative construction, support for reusability and support for interoperability reflect high level details of a methodology. They do not discuss about specific and technical details. The last four aspects namely, degree of application dependency, life cycle recommendation, strategies for identifying concepts and details of methodology cover the technical side of a methodology. They help the reader to quickly grasp the technical insights of a methodology.

Table 1 manifests a detailed and comprehensive comparison of methodologies based on the established criterion. Furthermore, Table 1 allows the readers to explore insights of all methodologies covered in this study.

Criteria 1: Type of development: Literature reveals that methodologies can be can be divided in to three broad categories namely, stage based model, evolving prototype model and guidelines, depending on the type of development model they follow. Different approaches have their respective pros and cons. Stage based methodologies may be suitable for scenarios where the purpose and requirements are clear. On the contrary, evolving prototype may be the best choice when requirements are initially not clear and need refinement over time. Guidelines mainly focus on recommending useful tips, rules and techniques, for making better design decisions rather than focusing on the overall development model.

Criteria 2: Support for collaborative construction: Ontologies can be constructed in isolation as well as in collaboration. Collaborative construction support allows different members of the ontology development team to work on a single ontology, at the same time. The team members are not restricted to a geographical location, they can contribute being at any location, without effecting the project efficiency. All the discussed methodologies will be analyzed for this aspect.

Criteria 3: Support for reusability: Ontology development is a time consuming and tedious task. In order to save time and efforts, the notion of ontology reusability gained popularity over the years. Methodologies supporting reusability allow ontology engineers to make use of existing ontologies, reducing the overall ontology development time and efforts. The time saving facilitate ontology engineers to focus on other advance issues, like ontology quality. Therefore, it's important to analyze that whether a methodology supports the notion of ontology reusability or not.

Criteria 4: Support for interoperability: Interoperability is an important aspect for today's ontology engineering. Some approaches support interoperability between systems. Domain ontologies developed using these methodologies share the same skeleton or high level concepts. Therefore, systems adopting such ontologies will have a similar knowledge skeleton and it would be easier for them to communicate and share knowledge with each other.

Criteria 5: Degree of application dependency: Different methodologies adopt distinct approaches for application dependency, during ontology development. A methodology can opt for one of the three scenarios namely, application dependent (ontology is developed on the basis of an application knowledge base in mind), application semi-independent (possible scenarios of ontology use are kept in mind during the specification stage) and application independent (no assumption is made regarding the uses to which the ontology will be put in knowledge-based systems, agents, etc).

Criteria 6: Life cycle recommendation: An ontology life cycle identifies the set of stages through which the ontology moves during its life. Many of the methodologies do not clearly recommend a life cycle. It will be analyzed whether a methodology proposes a life cycle or not.

Criteria 7: Strategies for identifying concepts: Identification of candidate concepts for inclusion in the ontology design is undoubtedly a crucial process. There are some techniques available for identifying concepts; some commonly used techniques include the bottom-up approach, top-down approach and middle-out approach. Researchers have their respective point of views for preferring one technique over another, depending on their experiences and nature of projects.

Criteria 8: Details of methodology: Every methodology comprises of some activities and techniques to support ontology development. Interestingly, the literature revealed that a number of methodologies do not provide sufficient details of their employed techniques and activities. For analysis purpose, this research will classify the methodologies to have three degrees of details namely, sufficient details,

some details and insufficient details. Those methodologies that give no or very vague details of their employed techniques are classified as having insufficient details. Methodologies which do not cover complete details but at least provide some details of their employed techniques are classified to have some details. Methodologies classified to have sufficient details cover the employed techniques with reasonable level of details, allowing the reader to clearly understand the technique and its application in the ontology development process.

Summary of analysis of methodologies: The methodologies employed by Enterprise Ontology and TOVE project relate to the domain of business enterprises. Both the methodologies follow a stage based model, compared to an evolving prototype model. Though both of them are actively referred in literature, none of them provide complete details about the techniques and activities involved in them. Furthermore, they do not offer any support for collaborative construction and interoperability. IDEF methodology follows an evolving prototype model and the ontology development is application independent, like the well-known METHONTOLOGY methodology. Similar to Enterprise Ontology and TOVE project, IDEF neither provides complete details of the techniques nor it recommends a life cycle. It also neglects the aspects for supporting collaborative construction and interoperability.

METHONTOLOGY considered the notion of recommending a life cycle as well as kept reusability perspective in focus. Unlike most of the methodologies covered in this research, METHONTOLOGY provide sufficient details of the techniques and activities employed in it (Fernández-López *et al*., 1997). Therefore, it is one of the reasons that it has been adopted for building quite a number of domain ontologies when compared to other methodologies. The authors also improved and revised it over the years, which made it even more adoptable (López, 1999). The aspect of collaborative construction and interoperability still remained untouched.

Collaborative construction, reusability and interoperability are crucial aspects of ontology development. The Ontolingua server focused on the aspects above and also provided advices on browsing, developing, maintaining and sharing of ontologies at the server end. The vision of distributed collaborative construction was initially explored by Ontolingua server (Farquhar *et al*., 1995; Swartout *et al*., 1996). Ontolingua strongly supports reusability, although it does not seem to cover mapping functions that convert from one ontology to another (Jones *et al*., 1998). Furthermore, it particularly lack details for engineering ontologies and doesn't recommend a life cycle.

SENSUS methodology resembles Ontolingua for supporting collaborative construction, reusability and interoperability. Unlike Ontolingua which uses an existing collection of ontologies for reuse, SENSUS already consists of more than 50,000 concepts in a hierarchy. For creating domain ontologies seed terms are manually linked to SENSUS. The final ontology includes only the seed terms extended to the root of SENSUS, all irrelevant terms are pruned (Swartout *et al*., 1996; López, 1999).

Similar to Ontolingua and SENSUS, CommonKADs methodology strongly focuses on ontology redesign and reuse, providing an existing collection of ontologies for reuse to the users (Schreiber *et al*., 1995; Wielinga *et al*., 1994). The degree of application dependency is a point of distinction between these approaches. With respect to degree of application dependency Ontolingua is application independent and SENSUS is application semi-independent, whereas CommonKADs is application dependent in nature.

As mentioned earlier, besides stage based and evolving prototype model there is a third broad category called guidelines. PLINIUS, Mikrokosmos and MENELAS all belong to the category of guidelines. Guidelines mainly stress on recommending tips, rules and techniques for making better design decisions. Mostly, methodologies belonging to this category are project specific. However, some of the guidelines they provide are general in nature and can be applied to other domains. Still there are many concerns left unanswered by these approaches (Jones *et al*., 1998). It was identified that ONIONS, Mikrokosmos and MENELAS are application dependent, whereas PLINIUS is application semi-independent in nature. All of them lack support for collaborative construction and reusability. Moreover, they all provide little details about techniques and activities. It was found that amongst this category only ONIONS supports the notion of interoperability.

UPON methodology exploits the Unified Process (UP) and Unified Modeling Language (UML) for ontology development. Like METHONTOLOGY, it recommends a life cycle and its iterative nature makes it somewhat closer to the 101 method. A closer insight shows that METHONTOLOGY, UPON and 101 method all follow an evolving prototype model, their natures are application independent and provide at least some details about the techniques and activities they employ.

CONCLUSION AND RECOMMENDATIONS

It is quite apparent that developing methodologies remains a complicated and tedious task. The existing literature depicts that most of the methodologies are based on experience of one or few projects, which is not enough to fully validate effectiveness. The analysis clearly highlights the shortcomings of the existing methodologies. Therefore, this research may act as a preliminary guide to come with a state of art ontology engineering methodology, bridging up the existing gaps and shortfalls. Based on the discussion and analysis carried out in previous sections, hence the following points converge to conclude this study.

- None of the methodologies are fully mature, if they are analyzed and compared on the basis of the established criteria.
- Most of the methodologies discussed in the study do not provide sufficient details about the techniques and activities employed in them. Some exceptions are there, most prominently METHONTOLOGY.
- Some methodologies support the notion of reusability and reengineering, but only few amongst them provide detailed recommendations for reusing and reengineering ontologies.
- Collaborative construction is an important aspect of ontology engineering but still little attention has been paid to this aspect. Ontolingua platform is worth mentioning when discussing collaborative construction.
- Most methodologies opt for conventional strategies for identifying ontology concepts. More new methods and techniques should be explored to make this process more efficient and handier for the ontology engineers, as it plays a critical role in the ontology designing phase.

REFERENCES

Arpírez, J.C. and A. Gómez-Pérez, 2000. Reference ontology and (ONTO) 2 agent: The ontology yellow pages. Knowl. Inform. Syst., 2(4): 387-412.

Bouaud, J., B. Bachimont, J. Charlet and P. Zweigenbaum, 1994. Acquisition and structuring of an ontology within conceptual graphs. Proceedings of Workshop on Knowledge Acquisition using Conceptual Graph Theory, University of Maryland, College Park, MD, 94: 1-25.

Lenat, D.B. and R.V. Guha, 1990. Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project. Addison-Wesley, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts.

Farquhar, A., F. Richard, P. Wanda and R. James, 1995. Collaborative Ontology Construction for Information Integration, Retrieved from: http://citeseerx. ist. psu. edu/viewdoc/summary?doi = 10.1.1.50.5303.

Fernández, M., 1996. CHEMICALS: Ontology of chemical elements. Final-Year Project. Faculty of Informatics at the University of Madrid.

Fernández-López, M., A. Gómez-Pérez and N. Juristo, 1997. Methontology: From ontological art towards ontological engineering. Proceeding of the Spring Symposium on Ontological Engineering (AAAI), pp: 33-40.

Fernández-López, M. and A. Gómez-Pérez, 2003. Overview and analysis of methodologies for building ontologies. Knowl. Eng. Rev., 17(02): 129-156.

Gangemi, A., G. Steve and F. Giacomelli, 1996. ONIONS: An ontological methodology for taxonomic knowledge integration. Proceeding of the Workshop on Ontological Engineering, ECAI-96, Budapest, pp: 95.

Gruninger, M. and M.S. Fox, 1995. Methodology for the design and evaluation of ontologies. Proceeding of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI.

Gómez-Pérez, A., 1996. Towards a framework to verify knowledge sharing technology. Exp. Syst. Appl., 11(4): 519-529.

Gómez-Pérez, A. and M.D. Rojas, 1999. Ontological reengineering for reuse. Proceeding of the 11th European Workshop Knowledge Acquisition, Modeling and Management (EKAW), Dagstuhl Castle, Germany, pp: 139-156.

Jones, D., T. Bench-Capon and P. Visser, 1998. Methodologies for ontology development. Proceeding of the 15th IFIP World Conference, pp: 20-35.

KBSI, 1994. The IDEF5 ontology description capture method overview. KBSI Report, Texas

López, F., 1999. Overview of Methodologies For Building Ontologies, Retrieved from: http://oa. upm.es/5480/1/Overview_Of_Methodologies.pdf.

López, M.F., A. Gomez-Perez, J.P. Sierra and A.P. Sierra, 1999. Building a chemical ontology using methontology and the ontology design environment. IEEE Intell. Syst. Appl., 14(1): 37-46.

Mahesh, K., 1996. Ontology Development for Machine Translation: Ideology and Methodology, Retrieved from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi = 10.1.1.47.3449.

Mars, N.J.I., W.G. Ter-Stal, H. De-Jong, P.E. Van-Der-Vet and P.H. Speel, 1994. Semi-automatic knowledge acquisition in plinius: An engineering approach. Proceeding of 8th Banff Knowledge Acquisition for Knowledge-based Systems Workshop, Banff, January 30th-Febraury 4th, pp: 4.1-4.15.

Nicola, A.D., M. Missikoff and R. Navigli, 2005. A proposal for a unified process for ontology building: UPON. Proceeding of the Database and Expert Systems Applications, pp: 655-664.

Noy, N. and D. McGuinness, 2001. Ontology development 101: A guide to creating your first ontology. Stanford Knowledge Systems Laboratory Technical Report.

Rojas, M.D., 1998. Ontologies Monatomic Ion in Physical Environmental Variables. Final-Year Project, Faculty of Informatics at the University of Madrid, (In Spanish).

Schreiber, G., B. Wielinga and W. Jansweijer, 1995. The KACTUS view on the'O'word. Proceeding of the IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing, pp: 159-168.

Smith, B., 2003. Blackwell guide to the philosophy of computing and information: Chapter ontology. Blackwell, 39: 61-64.

Steve, G. and A. Gangemi, 1996. ONIONS methodology and the ontological commitment of medical ontology ON8. 5. Proceeding of the Knowledge Acquisition Workshop.

Sure, Y., S. Staab and R. Studer, 2003. On-To-Knowledge Methodology. In: Staab, S. and R. Studer (Eds.), Handbook on Ontologies. Springer, Berlin, pp: 811, ISBN: 3540926739.

Swartout, B., R. Patil, K. Knight and T. Russ, 1996. Toward distributed use of large-scale ontologies. Proceeding of the 20th Workshop on Knowledge Acquisition for Knowledge-Based Systems, pp: 138-148.

Uschold, M. and M. King, 1995. Towards a methodology for building ontologies. Proceeding of the Workshop on Basic Ontological Issues in Knowledge Sharing, pp: 74.

Wielinga, B., A.T. Schreiber, W. Jansweijer, A. Anjewierden and F. Van Harmelen, 1994. Framework and Formalism for Expressing Ontologies. ESPRIT Project, 8145 KACTUS, Free University of Amsterdam deliverable DO1b.1.