International Journal of Computer Applications (0975 – 8887)
Volume 128 – No.10, October 2015

A Study Investigating Knowledge-based Engineering Methodologies Analysis

Thabet Slimani
Computer Science Department
Taif University
Taif, Saudia Arabia

ABSTRACT
It is now commonly accepted that building ontology is a key step in the development of knowledge based systems. A quite number of methodologies for ontology development have been proposed in the literature. However, this field still lacks mature methodologies which provide a clear vision for how to build ontologies. Most methodologies provide insufficient details about their adopted techniques and activities. This paper gives a preliminary guide which analysis and discusses some selected methodologies for ontology building. The important criteria and guidelines resulting from current literature have been adapted to perform this discussion and analysis. The performed analysis showed as outcomes that the existing methodology needs and additional hard work to be mature.

Keywords
Ontology; Ontology Engineering; Knowledge based systems; Ontology Building;

1. INTRODUCTION
Ontologies are considered as a well-known topic in Computer Science where they are used as explicit conceptual knowledge models making domain knowledge available to information systems. In the vision of semantic web, ontologies are indispensable, given that they provide the semantic vocabulary useful to annotate websites in a meaningful manner for machine interpretation. Moreover, and in the context of information systems, ontologies have access to the fields of symbolic knowledge representation in Artificial Intelligence, from formal logic, first order logic, automated reasoning and from conceptual modeling, though also creation based on Web-enabling features.

With the Semantic Web and the knowledge based systems advances, the study of ontologies has become increasingly an active field of search. The most frequently accepted definition of an ontology, refers to Gruber (1993) [1] which consider an ontology as an explicit specification of a conceptualization. This definition was redefined by Guarino and Giaretta (1995) [2] and refined by Borst and Akkermans (1997)[3] stating that an ontology is a formal specification of a shared conceptualization. In the definition of Staab and Studer (2004), an ontology is a combination between the definitions presented above saying that an ontology was a formal, explicit specification of a shared conceptualization. Finally, merging the definitions above we can say that an ontology is a conceptualization that needs to be explicit, in such a way that should be precise in nature, a formal specification such that it is processable by machine that includes computational semantics, and that should be shared (it provides a commonly accepted understanding of a given community). The literature of Ontology Engineering methodologies highlights those of the most interesting.

Ontology engineering and software engineering have some similarities, since those ontologies have their own life cycle that starts from an initial idea and finish by a new ontology production. The building for such an ontology is based on the methodology that defines the activities that must be performed through the required ontology development life cycle. Several works have explored this topic and this paper will explore the history of Ontology Engineering Methodologies, highlighting those of most importance.

This paper presents and discusses some ontology engineering methodologies developed in the last two decades. The paper analyzes each methodology based on a set criterion and guidelines, and provides a deep and critical detail to methodologies presented in literature. The organization of the paper is as follows. Next section defines the basic concepts of ontologies. Then the related work of ontology engineering will be discussed, followed by discussion and analysis. Finally, the conclusions and future work are presented in the final section.

2. ONTOLOGIES: BASIC CONCEPTS AND DEFINITIONS
An ontology is a computational object encoding knowledge about a specific domain in machine-processable form making it available to information systems. In several application contexts ontologies have been investigated from different facets, and there exist several definitions of the meaning of ontology. According the community of Semantic Web, the prevailing definition of an ontology is the following definition, based on [4].

Definition 1: An ontology is a formal explicit specification of a shared conceptualization of a domain of interest.
Several aspects of an ontology are captured by this definition: the aspects of formality, explicitness, conceptuality, consensus, and domain specificity which explained in the following paragraphs [5]:

- **Formality:** ontology formality means that it is expressed in a knowledge representation language that is focused on the basis of formal semantics. This type of knowledge representation ensures that the specification of ontology domain knowledge is machine-processable and can be interpreted in a clear manner. Moreover, the techniques that adopt symbolic knowledge representation built on the principles of logic, can be used to realize this aspect.
- **Explicitness:** to prepare the knowledge for machines accessibility, it is primordial that the ontology states the knowledge clearly. Explicitness is a metric for ontology evaluation. For example, in the work of Kargl et al. (2006)[6], explicitness is defined in the context of metamodels as the

number of concepts in the modeling language that are first class concepts in the metamodel.

- **Consensus:** An ontology represents an agreement between people in a community that provides a domain conceptualization. It is difficult to an agreement to share a common conceptualization when the community is large. For that reason, the ontology construction is associated with a social process of consensus attainment.
- **Conceptuality:** The use of conceptual modeling practices for ontology engineering is eventually beneficial: the vast number of existing conceptual modeling methods (graphical notations, conceptual symbols and tools) can make ontologies more understandable, easier to adopt, construct, and visualize, etc. Additionally, an ontology illustrates a conceptualization in general terms and instead of making statements about a specific situation relating particular individuals, an ontology tries to cover as many situations as possible that can potentially occur [7].
- **Domain Specificity**: the knowledge about a particular domain of interest is the restrictive scope of the ontology specifications. The ontology engineer can more effectively focus on capturing the details in this domain rather than covering a broad range of related topics, if the scope of the domain for the ontology is narrower.

Briefly, an ontology can be considered as a conceptual executable model of a specific domain. It is made machine-interpretable by adopting knowledge representation techniques making it easier to be used by applications and to support decisions on reasoning about domain knowledge.

Knowledge system building requires the creation of a particular domain. The domain being modeled has an abstraction under format of model, specifically, useful as it abstracts from irrelevant details. The model of a domain building involves specifying the entities in the domain in a distinguishable manner, and the necessary relations relating these entities. Additionally, it requires specifying the types of entities, and the relations types existing between entities. Each domain requires to be split into concepts which constitute a conceptualization of the domain under consideration. The conceptualization defines the types of entities and the existing relations between them. It is important to specify that the created conceptualization is not, obviously, a defined process.

It is possible to not conceptualize some entities in the world, but other entities could be specified more or less abstractly. In brief, a conceptualization making is a process that comes with a considerable amount of autonomy. Figure 1 includes an example of Car ontology.Moreover, an ontology is a representation of the entities type, their relations, and their constraints [8]. It consists of a set of classes, relations, instances, functions and axioms ordered hierarchically. Formally, an ontology is a description of data that remains constant over various data/knowledge bases in a certain domain [9].

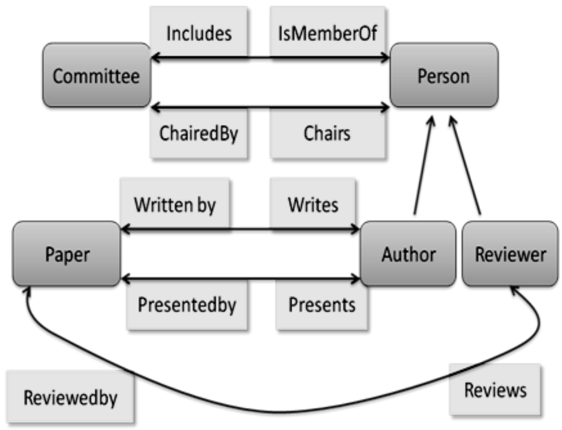

Fig1. Example of an ontology describing scientific publication.

In Figure 1, entities/concepts are shown as named rounded rectangle. The double arrow depicts an incoming and outgoing property (relation) to and from a concept. Each property appears as a blurred box. In Fig.1, the concepts are Committee, Reviewer, Person, Author and Paper. As an example Fig1 shows that 'an Author Writes a Paper and a Paper is Written by an Author'; 'Author Reviews a Paper and a Paper is Reviewed by a Reviewer', etc. The arrows connecting the entities Author and Reviewer to Person denote an IS-A relationship.

3. ONTOLOGY ENGINEERING

According Gruber (2009) [10], an ontology is defined as "a set of representational primitives with which to model a domain of knowledge or discourse". Three representational primitives can characterize an ontology: classes /concepts, relationships/properties, and attributes/ "datatype properties". The term semantic entity serves to generally refer any of the three types of the representational primitives.

An ontology covers only some aspects of the reality which are relevant to address its specific purpose, instead to represent the entire world of interest. Therefore, these essential aspects should be searched and selected for the ontology development process. Ontology engineering offers efficient and objective procedures for developing ontologies [11].

In the last few years, a great number of methodologies have been proposed for ontology engineering support. These methodologies have been compared and drawn in [12][13][14]. Each of these methodologies has advantages and disadvantages. Accordingly, methods and approaches of various ontology engineering methodologies are investigated in this paper.

Ontology development can be initiated by including three distinct motivating scenarios representing typical enterprise crowdsourcing activities which have already been presented in the work of Hetmank (2013)[15]:crowdsourcing system, crowdsourcing application and crowdsourcing platform.
All the procedure for building the crowdsourcing ontology is derived from the suggestion described by Uschold and King (1995) [16] providing four activities that must be performed: a) purpose and scope of ontology identification, b) ontology development, c) ontology evaluation, and (d) ontology documentation (see Figure 2). However, the activity of ontology development is more grouped into the following

steps: conceptualization, domain capture, integration, and implementation.

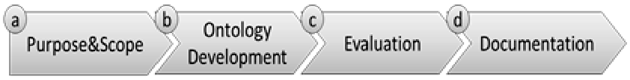

Fig2. Ontology engineering process of the enterprise crowdsourcing ontology.

The methodology described by Uschold and King (1995) presents some advantages, but the main weakness is the conceptualization missing, which starts with the development of a less formal domain model prior to the ontology implementation.

4. ONTOLOGY ENGINEERING METHODOLOGIES

Many works have explored the topic of ontology engineering and this paper explores the history of ontology engineering methodologies, highlighting those of most interest:

4.1 Cyc Methodology

The Cyc methodology [17] have been adopted in the Cyc project aiming to create a large knowledge base (KB) and which includes commonsense knowledge. The Cyc knowledge base development has adopted three main phases, the first is concentrated on manual extraction of commonsense knowledge from knowledge sources, the second adopts tools, guided by commonsense knowledge to help identifying more commonsense knowledge within the knowledge sources, and the third phase is completely automated with the use of the KB commonsense knowledge aiming to search more commonsense knowledge contained in the knowledge sources. A simple methodology can be extracted from these phases and each phase includes two common activities:

- Activity1: is devoted to "Develop a knowledge representation and top level ontology". This activity is charged to establish the knowledge representation terms (concept, attribute and attribute value), then using these terms for top level ontology codification which contains the most abstract concepts (Thing for example).
- Activity2: is devoted to "Represent the rest of the knowledge with the use of these primitives". The second activity uses the terms and the concepts created in the top level ontology codifying the knowledge of different domains.

4.2 Enterprise Ontology Methodology

Enterprise methodology is proposed within the Enterprise project proposed by Uschold and King in 1995 [16]. This proposed methodology is responsible to build ontologies based on their experience for building the Enterprise Ontology. This includes methodology four activities:

- Identification: This identification process begins with the purpose of the ontology identification aiming to clarify how it intends to be used and the relevant term identification in the domain.
- Building: Building is the second process of the ontology building, which itself includes three steps. The first activity is ontology capture which identifies the basic concepts in the ontology and the relationship between them, the second activity is the activity of coding which is responsible for knowledge representation, and finally the third activity is the activity of integration, which can be run in parallel with ontology capture and coding. The integration activity finds existing ontologies that could be integrated into the ontology being built and diagnosis their relevance and applicability.
- Evaluation: The evaluation process is an important activity that occurs after the process of ontology building is achieved. According Gómez-Pérez (1995) [18] definition, evaluation involves making a technical judgement of the ontologies, their associate software environment and documentation according to a frame of reference (competency questions, requirements specifications, and/or the real world).
- Documentation: Uschold and King [16] suggest some guidelines for activity of documentation taking into account the purpose and the type of the
- ontology that has been built.

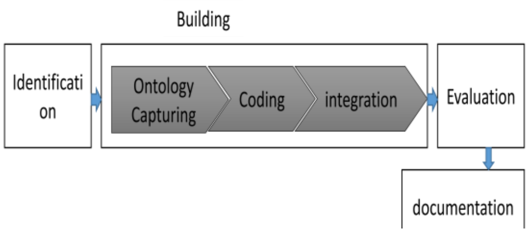

Fig3. Activities in the Uschold and King Methodology.

4.3 TOVE Methodology

Soon after, Grüninger and Fox (1995) [19] proposed several steps towards a methodology for ontology construction and evaluation. Those steps or scenarios are provided based on their experience acquired during the TOVE project ontology building:

- **Motivating scenario identification:** this step is devoted to specify a scenario that motivates the ontology and its applications. These scenarios should also give a set of possible solutions to the problems proclaimed in the scenario.
- **Informal competency question formulation:** this step is devoted to defining the requirements of the ontology as a set of competency questions that the ontology must be able to answer. This step involves the formulation of a set of natural language questions that the resulting ontology must be able to formally represent and answer once the anthology is completed.
- **Formal terminology specification:** this step aims to define the terminology of the ontology (its objects, attributes, relations, etc.). Having these informal competency questions in mind, from the previous step, the ontology engineer progresses to formally specify the terminology of the ontology within a formal language.
- **Formal competency question formulation:** Being given that the terminology of the formally defined, the ontology engineer can formally define

the competency questions identified earlier.

- **Axioms specification:** Certainly that the terms defined are inadequate except if they have a meaning associated with them, therefore the next step is to specify axioms in the ontology that offers a meaning to the terms and constraints over them. Finally, this step is devoted to specify the definitions and constraints on the terminology.
- **Completeness establishment:** In order to check the completeness of the conditions under which the competency questions are complete, it is vital to evaluate the ontology to test the competency questions against completeness theorems [19].

4.4 TOVE Methodology

KACTUS methodology [20] follows an engineering approach, highlighting the ontology development based on modular design, ontology redesign and reuse. The KAKTUS methodology is mainly focused on the ontologies building for particular applications and follows a bottom-up approach. The KAKTUS design process provides the possibility of reusing (refined and extended) ontologies even now developed for the use of different applications in the domain. As a first step, it is recommended to specify the application and the lists of relevant terms and tasks. As a second step, it is important to prepare a preliminary design based on the previous lists and specifications (that includes searching for already developed Ontologies). And finally, the ontology it should be refined towards the final design of the application. This methodology is not supported by any specific tools.

The ontology construction starts from a small-scale library ontology which requires the mapping between the various ontologies included in the new ontology development. A relevant ontology should be selected from a library and can be done by the use of ontology indexing schemas. Then, the interpretation context for the use of an ontology is characterized by three dimensions as follows:

- Task type;
- Problem solving method;
- Domain type.

4.5 CommonKADS methodology

CommonKADs is a methodology strongly based on ontology redesign and reuse, which provides an existing collection of ontologies for reuse by users. The CommonKADS methodology is mainly directed at the development of KBS [21]. This methodology offers useful features for ontology engineering and support the construction of a knowledge model. This knowledge model includes activities useful for ontology construction, and could identify different roles in knowledge-system development. The first step to do is to conduct an organizational analysis, which includes the feasibility study (studying the problems and opportunities, the organizational context and solutions) and improvement study. Being given that the requirements and needs are identified, the construction of the knowledge model starts with the following steps:

- **Knowledge identification** (Domain familiarization and identification of the list of potential model components for reuse)
- **Knowledge specification** (consists to choose the task template, construct initial domain conceptualization and specify the complete knowledge model).
- **Knowledge refinement** (which validate the knowledge model and refine the knowledge bases).

The CommonKADS method was adapted for legal knowledge-system development based on the following design steps: analysis, conceptual modelling, formal modelling, and implementation [22]. The method provided guidance for the legal knowledge system design, but not in the design of ontologies. However, this method is considered as ontology modelling methodological steps.

4.6 METHONTOLOGY methodology

METHONTOLOGY is an extensive methodology created in 1996 at the Universidad Politechnica de Madrid. METHONTOLOGY was the first methodology for ontology engineering that includes the rigors of the software engineering community in ontology engineering and has its roots in the software development process as presented by the IEEE in [23]. Activities in METHONTOLOGY are classified into several categories that include management activities (ensuring that ontology is built on time and ensuring a satisfactory quality level), a Development-Oriented activity, and support activities that run in parallel to the technical activities and that need to be performed to enable them. In addition, METHONTOLOGY take into account the evaluation, documentation and configuration management. Several tools were built to give technological support to this methodology like ODE and WebODE, In addition to protégé , OntoEdit and KAON.

The activities of management include scheduling and control, which consists to identify and monitor the tasks to be performed with the aim to guarantee their execution as they planned, and their quality assurance, where the quality of each output from the methodology is checked.

The activities of development are as follows:

- **Specification:** the purpose and scope of the ontology are realized informally or formally (why, what use, who are the end users).
- **Conceptualization:** is adopted to organize the knowledge acquired.
- **Formalization.**
- **Implementation.**
- **Maintenance.**

Development executes all the above activities either sequentially or iteratively. Support activities take into account knowledge acquisition, evaluation, integration, documentation and configuration management activities highlighting the activities in the management.

4.7 The Neon Methodoogy

As METHONTOLOGY has evolved into the NeOn [24] project. METHONTOLOGY defines its activities in a Software Development Process Model (SDPM) in a neutral manner that will be treated later by the ontology engineer to apply their desired process model to the activities defined in the methodology. This approach enables the use of the activities of an ontology engineering methodology directly into the life cycle of software development of an application together with activities related to database design and software engineering.

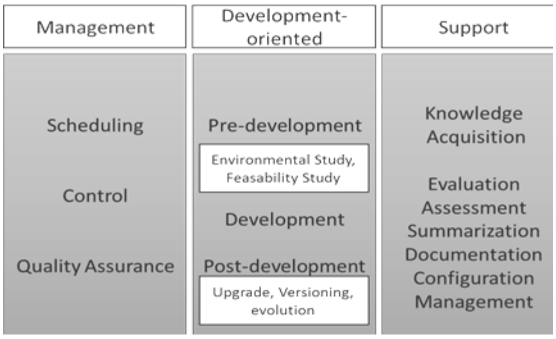

Fig4. Activities in the NeOn Methodology.

In the NeOn project, the activities number (in METHONTOLOGY) in each of category has grown, some activities have been refined and some moved to different categories (See Fig.4).

4.8 The ON-TO-KNOWLEDGE methodology

The On-To-Knowledge (OTK) Methodology [25] was inspired from the methodologies proposed by Uschold and others, CommonKADS and METHONTOLOGY. OTK is developed in 2001 during the project of the same name and follow an application-based driven development of ontologies. The On-To-Knowledge methodology applies many of the techniques inspired from methodologies like CommonKADS [21] to the ontology engineering domain. Two separate processes that constitute these knowledge management methodologies, the first process called the Knowledge Meta Process is adopted for the application establishment, while the second called the Knowledge Process is adopted to handle the maintenance of the established application. OTK methodology includes 5 steps (a number of activities are included in each step), as shown in Figure 5. We need to make a decision after each completed step based on the current state of the ontology. The steps are enumerated as follows:

- **Feasibility study:** this step identifies the problem area and the most promising solutions for this problem.
- **Kick-off step**: this step captures requirements based on the CommonKADS worksheets and the analysis of accompanying knowledge sources.
- **Refinement step:** this step extracts knowledge from the requirements and formalizes them into a semi-
- formal ontology description.
- **Evaluation step:** this step is performed, based on the technological, user and ontology perspectives.
- **Application and evolution:** this step applies the ontology in the given application and is evolved and maintained based on the needed changes. The refinement, evaluation, and application and evolution steps occur significantly in an iterative manner.

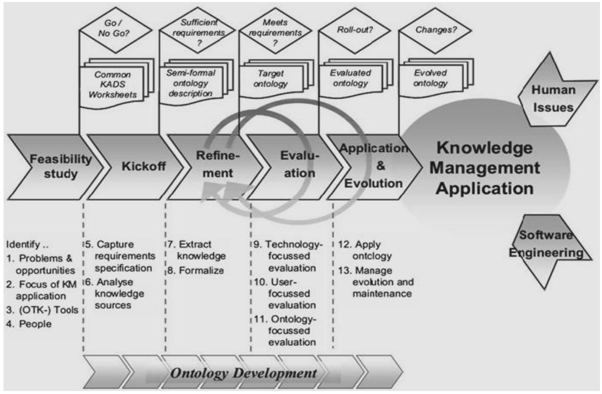

Fig.5. Steps of OTK methodology or knowledge meta-process (extracted from [26]).

4.9 The DILIGENT methodology

DILIGENT methodology was originated within the Semantic Web and Peer to Peer (SWAP) project aiming to address the problem of centralization in the area of ontology engineering. DILIGENT proposes an ontology life cycle model focused on prototypes evolving, where ontologies should be built in Distributed, loosely coupled, and evolving environment of ontology engineering [27], in addition to the building of an initial core ontology, which serves to its controls. This initial ontology should be complete. This core ontology can be taken by end users where he can perform local adaptation on it as they need for their given application, and can keep track the adjustment that they have made. Then, the changes made by each of the end user can be analyzed and revised by the central board to the core ontology. The end users in the local update activity can then adopt the new revision of the core ontology and this cycle can continue iteratively as changes are required. The DILIGENT methodology includes five steps presented by the iterative process as shown by Fig. 6.

Fig 6. Activities in the DILIGENT Methodology.

4.10 The TERMINAE methodology

The TERMINAE methodology [28][29] is a method and a platform for ontology engineering (from texts) that includes four steps as described below:

- **Corpus constitution**: this step is concerned by corpus selection and organization;
- **Linguistic study**: this step is concerned with the analysis of linguistic based on several natural language processing tools;
- **Normalization**: this step does the normalization according some structuring principles and criteria;
- **Formalization:** this step preoccupies by the formalization and validation using several languages of description logics.

A comprehensive knowledge of the domain should be provided by the ontology, to facilitate the distinction between terms (nouns, verbs, phrases or adjectives), domain terms and the concepts and the relations labeled with these domain terms.

4.11 The PLINUIS methodology

The PLINUIS project [30] aims to extract knowledge from natural-language texts (abstracts and title of bibliographic document descriptions extracted from the online version of Engineered Materials) in a semi-automatic manner. The general ontology development principles embody the design decisions taken during the ontology development that appears as domain-independent. Those principles are enumerated as follows:

- Discovery of conflicting assertions about the same entity, if the concepts are well defined.
- Take into account the pre-existing formal theory as they given and do not specify the semantics of logical constants for the domain ontology.
- Ensure the independence of an ontology from any particular knowledge representation language.
- Adopts the principle of conceptual construction kit stating that an ontology is a set of primitive concepts and construction rules allowing the definition of other concepts based on these primitives.
- Follow a bottom-up approach in-order that an ontology displays the required completeness for the intended tasks (principle 7).
- Adopts an engineering decisions approach for the ontology development.
- Evaluate the ontology in terms of its completeness with respect to the intended task. Coverage (every concept of interest is covered) and garrulity (every relevant distinction is made) are two sub-criteria of completeness.

4.12 The ONIONS methodology

ONIONS is an abbreviation for "ONtologic Integration Of Naïve Sources) and a methodology motivated by the knowledge integration problem (integrate heterogeneous sources of information in knowledge acquisition). The building of a formal domain ontology with the integration of existing repositories of knowledge can resolve this problem. ONIONS focus on problems in ontology acquisition, rather than focusing on the problem of final representation of an ontology. The modeling stopover and knowledge relevance are two examples of ontology acquisition [32].

4.13 Mikrokosmos methodology

Microkosmos is a project which provides thirty guidelines to assist the Micokosmos ontology development [33]. Mikrokosmos includes an ontology in addition to some lexicons in different languages that were developed for machine translation. The following section presents seven guidelines for the ontology acquisition process:

- Do not add instances as concepts (Instances do not have their own instances; or Concepts haven't fixed position in space/time);
- Do not decompose concepts further;
- Use close concepts;
- Do not add EVENTs with particular arguments;
- Do not add concepts with instance-specific aspects, temporal relations;
- Do not add concepts with specific language;
- Do not add ontological concepts for collections.

4.14 PHYSSYS Methodology

PHYSSYS is an approach which adopts a formal ontology based upon the system dynamics theory as accomplished in engineering modeling, design and simulation. Additionally, PHYSSYS is an ontology that expresses different conceptual viewpoints on a physical system. This method aims to facilitate the ontologies selection based on dynamic knowledge construction, rather than the simple knowledge component selection. The viewpoint formalization of PHYSSYS ontology is based on three engineering ontologies. Ontology projections are adopted to formalize the interdependencies between these ontologies. Furthermore, the view-points construction is focused on smaller abstract ontologies. The adopted whole set of ontologies includes ontologies with a variety of genericity and abstractness. The Identification of these separate ontologies makes it easier to understand the domain because classes and ontological commitments are added incrementally, in addition to the enhancement of the ability to share and reuse parts of PHYSSYS.

4.15 MENELAS Methodology

MENELAS ontology was adopted in the medical domain and designed as part of a natural language understanding system [34]. The methodology used in the development of MENELAS ontology follows conceptual graphs as its central formalism. Four useful principles in the development of taxonomic knowledge are included in MENELAS (similarity, opposition, specificity and unique semantic access). These principles greatly assist the ontology structuring and acquisition.

4.16 KOLON Methodology

The KOLON [35][36](an acronym for "Korean Lexicon mapped onto an ONtology") methodology is a project developed by Seoul National University Computational Linguistics Laboratory in 2007 and applied to the automatic mapping of lexemes from the Sejong Electronic Dictionary onto concepts extracted from the Mikrokosmos Ontology. Two main databases are included in KOLON Ontology: the ontological frame-system and the lexical frame-system. The heart of the KOLON System is the KOLON ontology and a set of programs developed in order to facilitate the use and visualization of the ontological data.

4.17 UPON Methodology

UPON is a methodology of ontology development inspired from the Unified Software Development Process [37]. UPON takes advantage of the Unified Process, in addition to the use of the Unified Modeling Language (UML). The use of these techniques facilitates the ontology development process. The use of UPON for ontology development includes cycles, phases, iterations and workflows, it pursuit the UP (Unified Process) paradigm. The iterative and incremental nature of UPON makes it different from other processes, especially for software and ontology engineering [37]. The limitation of UPON is its inability to provide comprehensive details and its omission of the collaborative construction aspect.

4.18 OntoClean Methodology

OntoClean [38] was the first attempt to formalize notions of the analysis of ontology for information systems. This methodology focused on formal-based ontology evaluation, which adopts the notions of essence, rigidity, identity, unity and dependency. Moreover, OntoClean presents a definition of metaproperties helping with the construction of ontology language, the descriptions of problem domains.

5. DISCUSSIONS AND ANALYSIS

5.1 Generic Ontology Engineering Steps

The ontology engineering methodologies include essentially three steps explained as follows [39][40][5]:

- **Requirements Analysis**: the process of ontology engineering starts, typically with a complete analysis of the requirements that occurs in the original application scenario. These requirements should be described by the ontology engineer or domain expert on the basis of the document of ontology requirements specification (useful as a basis for subsequent modeling activities and quality assurance). Consequently, the requirements description should include information about the ontology scope (competency questions), its intentional use, or its level of expressivity.
- **Conceptualization:** In this phase, the content of ontology takes into account a representation based on semantic vocabulary and statements about the concerned domain of interest, which involves the ontological entities choice and the axioms formulation. The ontology engineers and domain experts attempt to realize a common agreement upon the simple structure of the ontology based on the abovementioned requirements specification (based on the exchange of arguments for their respective design decisions supporting, for example) [41]. The output of this phase is a specification (whether it be informal or semiformal) of their shared conceptualization.
- **Implementation:** The implementation phase is the final step of the core ontology engineering methodology which provides an explicit formalization of this requirement in terms of a concrete ontology representation language. The appropriate ontology language selection depends mostly on the intentional use of the ontology and the required level of expressivity. The implementation phase can be realized by automatic approaches intended for ontology acquisition and reuse.

Moreover, it requires including additional ontology development activities such as ontology evaluation [42], versioning and documentation. Especially, it is habitually suitable to perform the ontology alignment with existing upper-level ontology. For example, an alignment can be obtained by the application of ontology design patterns.

5.2 Analysis and Recommendations

The steps mentioned in the previous section are part of more or less each ontology engineering methodology presented earlier. Some of the most well-known methodologies facilitating the effective and efficient construction of ontologies are Methontology, OntoClean, and DILIGENT. All these methodologies have been inspired by the first, general methodologies developed in the 1990s [16][19]. Currently, the state-of-the-art includes several methodologies for specific ontology building scenarios (e.g., collaborative ontology engineering [43]), distributed, and specific application domains, such as bioinformatics [44] or medicine [45]. Only a few number of the methodologies described earlier was integrated with major ontology development environments. Nevertheless, several plug-ins were provided for the application of the OntoClean application (especially is facilitated for Protege, OntoEdit and WebODE). These plugins serve for ontologies manual tagging facilities with the use of OntoClean meta-properties and offer means for ontology consistency checking according to the constraints includes in OntoClean. Despite the multiplicity of the methodologies for ontology engineering, the ontology construction remains time-consuming, labor-intensive, and error-prone if it is completely realized manually.

The TOVE methodology relates to the domain of business enterprises and adopt a stage based model. However, TOVE is actively referred in literature, it does not provide complete details about their used techniques, activities and if it provides support for collaborative construction and interoperability. METHONTOLOGY methodology adopts an evolving prototype model and the ontology development is application independent. Additionally, METHONTOLOGY take into account the notion of life cycle recommending. METHONTOLOGY provide sufficient details of the techniques and activities employed in it [46], differently to the most of the methodologies covered in this paper. Consequently, METHONTOLOGY has been adopted for the development of a quite number of ontologies when compared to other methodologies.

CommonKADs methodology provides an existing collection of ontologies for reuse to the users and is strongly centered on ontology redesign and reuse. The CommonKADs methodology is application dependent in nature. As stated before, as well stage based and developing prototype model there is the wide category called guidelines. Mikrokosmos, PLINIUS and MENELAS methodologies are included in the category providing guidelines. Guidelines are provided for recommending tips, rules and techniques aiming to obtain better design decisions making. Although, some of the provided guidelines are general in nature and can be applied to other domains. In other terms Mikrokosmos, ONIONS and MENELAS are classified as application dependent, whereas PLINIUS is classified as application semi-independent in nature. All of them provide a little detail about techniques and activities and require the collaborative construction and reusability support. Only ONIONS supports the notion of interoperability. In a similar manner to METHONTOLOGY, UPON recommends a life cycle. Additionally, METHONTOLOGY and UPON methodologies are application independent, follow an evolving prototype model and provide as a minimum some details about the adopted techniques and activities [47].

6. CONCLUSION

Ontology developing remains a very complicated and difficult task. The most of the methodologies described in literatures are based on the idea of one or few works, which is not sufficient to validate effectiveness. This paper clearly provides more or less detailed description of the existing methodologies. Consequently, the research found in this paper may act as a guidance and a state of art ontology for ontology engineering methodology, helping researchers interested in this domain. The discussion and analysis provided in previous sections, concludes with the following points:

- The requirement for fully mature methodologies for ontology engineering
- Most of the methodologies discussed in this paper offer insufficient details about the techniques and activities they use (exception for METHONTOLOGY).
- The notion of reusability and reengineering is supported only by few methodologies.

7. REFERENCES

[1] Gruber T. R. "A translation approach to portable ontologies", Knowledge Acquisition, vol. 5, no. 2, pp. 199-220, 1993.

[2] Sowa J. F.. Ontologies for Knowledge Sharing. In Manuscript of the invited talk at Terminology and Knowledge Engineering Congress (TKE ’96), Vienna, 1996.

[3] Borst P. and Akkermans H. An Ontology Approach to Product Disassembly, Knowledge Acquisition, Modeling and Management, pp. 33-48, 1997.

[4] Studer R., Benjamins V.R. and Fensel D.: Knowledge engineering: principles and methods. IEEE Trans. Knowl. Data Eng. 25(1–2), 161–197, 1998.

[5] Grimm, S., Abecker, A., Völker, J., & Studer, R. Ontologies and the Semantic Web. In J. Domingue, D. Fensel, & J. A. Hendler (Eds.), Handbook of Semantic Web Technologies (pp. 507–570). Berlin: Springer, 2011.

[6] Kargl, H., Strommer, M., Wimmer, M. Measuring the Explicitness of Modeling Concepts in Metamodels. In ACM/IEEE 9th International Conference on Model Driven Engineering Languages and systems (MoDELS/UML 2006), Workshop on model Size Metrics, Genova, Italy, October 2006.

[7] Guarino, N.. Formal ontology and information systems, preface. In: Guarino, N. (ed.) Proceedings of the First International Conference on Formal Ontologies in Information Systems (FOIS 1998), Trento, pp. 3–15. IOS Press, Amsterdam, 1998.

[8] Van Heijst, G., Schreiber, A.T., Wielinga, B.J.. Using explicit Ontologies in KBS development. International Journal of Human-Computer Studies, 1997, 46(2-3):183-292.

[9] Guarino, N., Giaretta, P. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In N. J. I. Mars (ed.), Towards Very Large Knowledge Bases. 1995, Amsterdam, The Netherlands, 25-32. IOS Press.

[10] Gruber, T. R. Ontology. In L. Liu & M. T. Özsu (Eds.), Encyclopedia of Database Systems (pp. 1963–1965). Berlin: Springer, 2009.

[11] Sure, Y., Staab, S., & Studer, R. Ontology engineering methodology. In S. Staab & R. Studer (Eds.), Handbook on Ontologies (2nd ed., pp. 135–152). Berlin, Heidelberg: Springer, 2009.

[12] Sicilia, M. A., Rodríguez, D., García Barriocanal, E., & Sánchez Alonso, S. . Empirical findings on ontology metrics. Expert Systems with Applications, 39(8), 6706–6711. doi:http://dx.doi.org/10.1016/j. eswa.2011.11.094, 2012.

[13] Gómez Pérez, A., Fernández López, M., & Corcho, O. Ontological engineering – With examples from the areas of knowledge management, e-commerce and the Semantic Web (4th ed.). London, Berlin, Heidelberg: Springer, 2005.

[14] Casellas, N. N. Methodologies, tools and languages for ontology design. In Legal Ontology Engineering (Vol. 3, pp. 57–107). Springer Netherlands. doi:10.1007/978-94-007-1497-7_3, 2011.

[15] Hetmank. Components and Functions of Crowdsourcing Systems – A Systematic Literature Review. In 11th International Conference on Wirtschaftsinformatik (pp. 55–69). Leipzig, 2013.

[16] Uschold, M., & King, M.. Towards a methodology for building ontologies. In Proceedings of the International Joint Conference on AI (IJCAI’95), Workshop on Basic Ontological Issues in Knowledge Sharing. Budapest, 1995.

[17] Lenat D., Guha R., Pittman K., Pratt D., and Shepherd M., "CYC: Toward Programs With Common Sense", ACM Communications, vol. 33, pp. 30-49, 1990.

[18] Gomez-Perez A., "Criteria to Verify Knowledge Sharing Technology", Knowledge Systems Laboratory, Stanford University, 1995.

[19] Grüninger M. and Fox M. "Methodology for the Design and Evaluation of Ontologies", in Proc. of the IJCAI Workshop on Basic Ontological Issues in Knoweldge Sharing, Montreal, Canada, 1995.

[20] Bernaras, A., I. Laresgoiti, and J. Corera. Building and reusing ontologies for electrical network applications. In Proceedings of the 12th European Conference on Artificial Intelligence (ECAI’96), ed. W.Wahlster, 298–302. Budapest, Hungary: Wiley, 1996.

[21] Schreiber, G., Akkermans H., Anjewierden A., Dehoog, R., Shadbolt N., Vandevelde W., and Wielinga B., "Knowledge Engineering and Management: The CommonKADS Methodology", December 1999.

[22] Visser, P. R. S., van Kralingen R. W., and Bench-Capon T. J. M. A method for the development of legal knowledge systems. In Proceedings of the International Conference on Artificial Intelligence and Law (ICAIL’97), ed. J. Zeleznikow, D. Hunter, and L. K. Branting, Melbourne, 151–160. New York: ACM Press, 1997.

[23] IEEE, "IEEE Std 1074-2006: IEEE Standard for Developing a Software Project Life Cycle Process", IEEE, 2006.

[24] NeoN, "The NeoN Project Web site". http://www.neon-pr oject.org/web-content/.

[25] Sure, Y., S. Staab, and R. Studer. On-To-Knowledge methodology (OTKM). In Handbook on ontologies, ed. S. Staab and R. Studer. International handbooks on information systems, 117–132. Berlin: Springer, 2004.

[26] Sure, Y., and R. Studer. September. On-To-Knowledge methodology – final version. Project Deliverable D. 18, EU IST-1999-10132 On-To-Knowledge (Institute AIFB, University of Karlsruhe), 2002.

[27] Pinto, S., Staab, S., Sure, Y., and Tempich C. OntoEdit empowering SWAP: a case study in supporting

DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies (DILIGENT). in Proc. of the 1st First European Semantic Web Symposium (ESWS 2004), S. Valencia (Eds.), 2004.

[28] Biebow, B. & Szulman, S. TERMINAE: A linguistics-based tool for the building of a domain ontology. In EKAW '99 - Proceedings of the 11th European Workshop on Knowledge. Acquisition, Modeling, and Management. Dagstuhl, Germany, LNCS, pages 49–66, Berlin. Springer, 1999.

[29] Aussenac-Gilles, N., Despres, S., Szulman, S. The TERMINAE Method and Platform for Ontology Engineering from texts. Bridging the Gap between Text and Knowledge - Selected Contributions to Ontology Learning and Population from Text. P. Buitelaar, P. Cimiano (Eds.), IOS Press, p. 199-223, 2008.

[30] Mars N.J.I., Ter Stal W.G, De Jong H., Van Der Vet P.E. and Speel, P.H. Semi-automatic Knowledge Acquisition in Plinuis: An Engineering Approach, in proceeding of 8th Banff Knowledge Acquisition for Knowledge-based Systems Workshop, Banff, January 30th-February 4th, 4. 1-4.15. 1994.

[31] Wielinga, B.J., Schreiber, A.TH., Jansweijer, W.H., Anjewierden, A. And Vam Harmelen, F. Framework and Formalism for Expressing Ontologies. KACTUS Project Deliverable DO1b.1, University of Amsterdam. 1994.

[32] Jones, D., Bench-Capon, T., & Visser, P. Methodologies for ontology development. In J. Cuena, editor, Proc. 15th IFIP World Computer Congress, pp. 62–75, London, UK. Chapman and Hall, 1998.

[33] MAHESH, K. "Ontology Development for Machine Translation: Ideology and Methodology", Technical Report MCCS 96-292, Computing Research Laboratory, New Mexico State University, Las Cruces, NM, 1996.

[34] Bouaud, J., Bachimont B., Charlet J. and Zweigenbaum P. 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, 1994.

[35] Jang, H.-Y. and H.-P. Shin. 2010. Language-Specific Sentiment Analysis in Morphologically Rich Languages. Proceedings of COLING 2010. To appear.

[36] Junho J.P, Jo J., Shin H. The KOLON System: Tools for Ontological Natural Language Processing in Korean. In the Proceeding of 24th PACLIC Conference. P. 425-432.

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

[38] Guarino, N., and Welty C. An overview of OntoClean. In Handbook on ontologies:International handbook on information systems, ed. S. Staab and R. Studer, 151–171. Berlin: Springer, 2004.

[39] Gomez-Perez A., Corcho-Garcia, O., Fernandez -Lopez, M.: Ontological Engineering. Springer, New York, 2003.

[40] Simperl, E., Tempich, C.: Exploring the economical aspects of ontological engineering. In:Staab, S., Studer, R. (eds.) Handbook on Ontologies. International Handbooks on Information Systems, 2nd edn, pp. 337–358. Springer, Heidelberg, 2009.

[41] Tempich, C., Simperl, E., Luczak, M., Studer, R., Pinto, H.S.: Argumentation-based ontology engineering. IEEE Intell. Syst. 22(6), 52–59, 2007.

[42] Vrandecic , D.: Ontology evaluation. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. International Handbooks on Information Systems, 2nd edn, pp. 293–313. Springer, Heidelberg, 2009.

[43] Tempich, C., Pinto, H.S., Sure, Y., Staab, S.: An argumentation ontology for DIstributed, Loosely-controlled and evolvInG Engineering processes of oNTologies (DILIGENT). In: Gomez-Perez, A., Euzenat, J. (eds.) Proceedings of the Second European Semantic Web Conference (ESWC 2005), Heraklion. Lecture Notes in Computer Science, vol. 3532, pp. 241–256. Springer, Berlin, 2005.

[44] Garcia Castro, A., Rocca-Serra, P., Stevens, R., Taylor, C., Nashar, K., Ragan, M.A., Sansone, S.-A.: The use of concept maps during knowledge elicitation in ontology development processes-the nutrigenomics use case. BMC Bioinform. 7, 267, 2006.

[45] Wennerberg, P., Zillner, S.,Moller,M., Buitelaar, P., Sintek, M.: KEMM: a knowledge engineering methodology in the medical domain. In: Eschenbach, C., Gruninger, M. (eds.) Proceedings of the Fifth International Conference on Formal Ontology in Information Systems (FOIS 2008), Saarbrucken. Frontiers in Artificial Intelligence and Applications, Saarbrucken, vol. 183,pp. 79–91. IOS Press, Amsterdam, 2008.

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

[47] Iqbal R., An Analysis of Ontology Engineering Methodologies: A Literature Review. Research Journal of Applied Sciences Engineering and Technology 6(16):2993– 3000, 2013.