

# NATURAL LANGUAGE SUPPORT FOR COMPETENCY EVALUATION OF WEB-ONTOLOGIES

Muthukkaruppan Annamalai<sup>1</sup>, Zulhizam Sanip<sup>2</sup>

<sup>1</sup> Faculty of Computer and Mathematical Sciences  
Universiti Teknologi MARA  
40450 Shah Alam, Selangor, Malaysia  
mk@tmsk.uitm.edu.my

<sup>2</sup> TM Research & Development  
TM Innovation Centre  
63000 Cyberjaya, Selangor, Malaysia.  
zulhizam@tmrnd.com.my

**Abstract** - Ontology is a key element of the Semantic web that facilitates a common understanding of the structure of knowledge in a domain, and they can back applications such as semantic annotation, consistency checking, search and retrieval. The increasing use of the Semantic web has escalated the demand for competent ontologies. One way to check on the competency of an ontology is by use of competency questions to ascertain whether the ontological commitments are adequate to support its purpose of design. Thus, it is useful to have a competency evaluation supporting tool to assist an ontology engineer to perform this analytic task during the development of an ontology. At present, such support is noticeably weak. It led us to construct a semi-automated competency evaluation supporting tool for the popular Protégé ontology editor. The tool, supported by a natural language program interface to the editor, is currently equipped with simple functions, but has demonstrated its potential as a dependable tool for performing the competency evaluation of web ontologies more effectively, in less time and with reduced effort.

**Keywords:** Web-ontologies, Ontology engineering, Competency evaluation, Natural language program interface.

## 1. WEB-ONTOLOGIES

An ontology is a formal specification of conceptualisation of a representation vocabulary specialised in a knowledge domain (Chandrasekaran & Josephson, 1999; Gruber, 1995). The role of the model is to make the underlying assumptions about the domain concepts and the relationships among them explicit. The idea is to help to elucidate the meaning of expressions in the shared knowledge resources. Consequently, ontology has emerged as a key element of the Semantic web that is founded on meaning, and is used to support applications as semantic annotation, consistency checking, searching and retrieval of the shared knowledge resources distributed on the web.

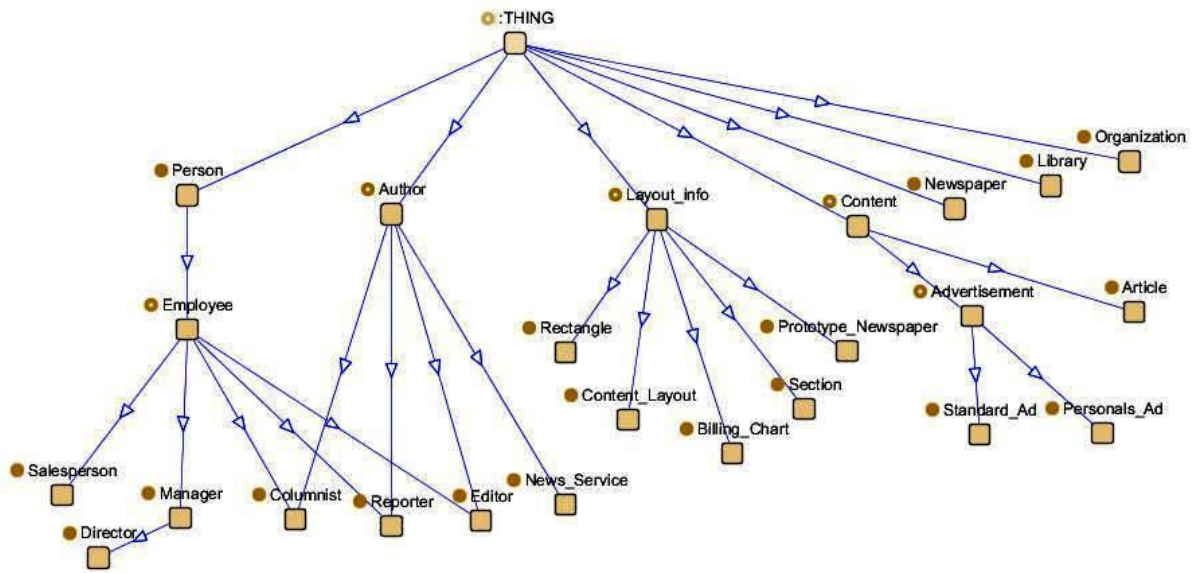
Ontologies can be designed with increasing levels of formality, from simple glossaries and thesauri to rigorously formal logical theories; the higher the degree of formality, the less ambiguous and stronger the power of automated reasoning (Uschold & Gruninger, 2004). On this formality scale, a web-ontology is regarded as a light weight ontology that characterises the concepts of a problem domain with somewhat lower degree of formality.

There are two ways of organising the concepts defined in a web-ontology, i.e., hierarchically according to the subsumption association, and crosswise according to various many domain-specific association between concepts. For purpose of illustration, graphical representations of hierarchically organised concepts in a Newspaper ontology and network of concepts in a Family ontology are shown in Figure 1(a) and (b), respectively. The Newspaper ontology, which is sourced from the Protégé ontology library of the Protégé Community (n.d.), highlights the subsumption associations between some pairs of concepts defined in the ontology. In the smaller Family ontology, which we have developed, we extend the subsumption with other domain-specific associations between the related concepts.

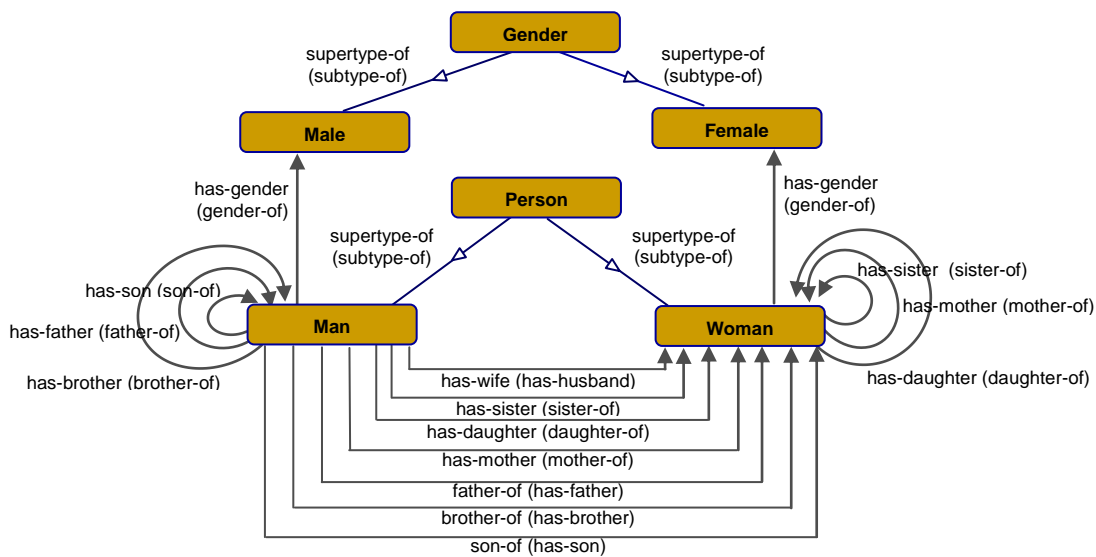
The concepts in a web-ontology are defined as classes of domain objects whose attributes are captured as properties and relations. The properties are data type attributes, while the relations are object type attributes.

For example, *name* and *date-of-birth* are properties of the *Person* concept in the Family ontology. The properties of *Person* are passed down to the concepts *Man* and *Woman* via the subsumption operator.

There are five types of relationships to describe the association between concepts in a web-ontology. We identify them as: normal, inverse, parallel, self-referencing and indirect relations. A normal relation is a primary relation defined for a concept that directly relates it to an immediate concept. For example *supertype-of*, *has-gender* and *has-wife* are examples of normal relations defined for *Man* in the ontology. An inverse relation is the opposite of a normal relation. The inverse relation name appears in a bracket next to the normal relation name. For example, *son-of* is defined as the inverse relation of *has-son*, and vice versa in this ontology. Similarly, *subtype-of* is the inverse of *supertype-of*. The *subtype-of* relation is often referred to as an *is\_a* relation. Parallel relations are said to exist when there are two or more distinct relations connecting a same pair of concepts. For example *has-wife* and *has-sister* are examples of parallel relations from *Man* to *Woman*. Self-referencing relation refers to a relation that relates a concept to itself. For example, *has-son* and *has-father* are examples of self-referencing relations defined for *Man*. Indirect relations refers to the sequence of relationships that connect two concepts through one or more intermediate concepts. For example, *has-mother* and *has-gender* are indirect relations that relate the concepts *Man* to *Female* through an intermediate concept *Woman*.



a) A snapshot of Newspaper ontology



b) A snapshot of Family ontology

Figure 1. The graphically reproduced segments of example web-ontologies

While it is possible to describe the domain-specific associations using all the five types of relations mentioned in the previous paragraph, the subsumption associations can only be represented as normal (*supertype-of*), inverse (*subtype-of*) or indirect relations. Subsumption is a special kind of association that characterises a parent-child relationship between a pair of immediate concepts or an ancestor-descendant relationship between a pair of distant concepts in a hierarchy. Therefore, it is represented in a distinct fashion in the ontology.

First, the association between an ancestor and its descendant concepts must be acyclic. A cycle is said to exist when a set of concepts are related through a particular relation, in such way that a traversal of the ontology over the relation forms a loop. This means, we cannot allow a concept to relate to itself through the *supertype-of* or *subtype-of* relations, neither directly (self-referencing) nor indirectly. On the other hand, some domain-specific associations admit cycles.

Second, because the relation between a parent and its child concepts are unique, it is not conceptually right to represent the association between the ancestral concepts in more than one way. Further, it is bound to introduce unwanted cycles in the concept hierarchy. For example, it may appear sensible to represent the *has-son*, *has-brother* and *has-father* relationships between *Person* and *Man*, or *has-daughter*, *has-sister* and *has-mother* relationships between *Person* and *Woman*, instead of the way they are currently described in the Family ontology. However, the additional parallel relations would taint the distinct associations between *Person* and *Man*, and *Person* and *Woman* concepts. Therefore, we have described the above mentioned relations crosswise between the children of *Person* as shown in Figure 1(b). As a result, some unavoidable redundancies are apparent in the ontology.

For sake of reasoning efficiency, domain-specific associations are also generally free of cycles with exception of the self-referencing (circularity at distance zero) and the inverse (circularity at distance one) relations. The ontologies described in the paper are based on this assumption. Nevertheless, we regard this as a restriction in the ongoing research, and plan to overcome this limitation by providing for admissible cycles involving indirect (circularity at distance  $n$ ) relations in the future.

Web-ontologies are formally represented using Descriptive logic (Wang et al., 2006), a restricted logical formalism that is capable of expressing only binary relationships between concepts. The concepts in the ontology are defined as classes using Frames. The ontologies are usually developed with the aid of editors such as Protégé (Noy, et al., 2000), OntoEdit (Sure, et al., 2003) and WebODE (Arpirez, et al., 2001), and are typically encoded in ontology languages like OWL (W3C Recommendation, 2004), DAML (DARPA Agent Markup Language Program, 2000) and RDF (McBride, 2004). DAML was a precursor to OWL, and OWL is a layer on top of RDF. OWL has become the defacto web-ontology language and Protégé is a freely available, popular web-ontology editor.

## 2. ONTOLOGY EVALUATION

In general, an ontology is evaluated on the syntax, structure and semantics of its conceptual definitions (Annamalai, 2005). The syntax of an ontology is dependent on the formalism used to represent the ontology. Syntax analysers incorporated within ontology editors, parsers and validators help to check the syntactic correctness of the definitions and also highlight the violations of user defined constraints in the ontology.

The structure of an ontology is determined by the organisation of its concepts. Structural evaluation concerns the topographical properties of an ontology. Metrics are often used to calculate the measure of structural complexity of an ontology (Tartir, et al., 2005), and this technique has been incorporated within some web-ontology editors (Corcho, et al., 2004).

The semantic evaluation validates and verifies of the semantic features of an ontology. The semantic evaluation is the focus of this paper. The evaluation analyses the vocabulary used to represent the concept, relation and property terms, and how well these terms relate to their corresponding things in the problem domain.

The semantic evaluation has been approached from the perspectives of software engineering, knowledge engineering and philosophy. The data-driven ontology evaluation proposed by Brewster, et al. (2004) and the evaluation based on the unified software process proposed by Nicola, et al. (2009) are examples of methods that are evolved from the software engineering approach. The former analyses the lexicon constructed from a text corpus to identify suitable terms of an ontology, while the latter exploits a software engineering standard to propose a method for building ontologies, which includes a test workflow that verifies the semantic and pragmatic quality of an ontology.

Conventionally, the semantic evaluation is based on the knowledge engineering approach. It makes use of a set of criteria of desirable qualities that has been evolved from the best practices in knowledge systems development such as consistency, completeness, conciseness, and extensibility to evaluate an ontology (Fox &

Gruninger, 1998; Gómez-Pérez, 2004; Gruber, 1995; Uschold, 1996). These criteria often emphasise on the aspects of generality to enable ontology reuse, and the analysis is by and large performed manually by an ontology engineer. Lately, a method of evaluation that relates these criteria to applicable measures have been proposed (Yu, et al., 2009). The approach seeks to quantify the measurement of the desirable qualities that an ontology must possess.

From the philosophical perspective, an ontology captures some part of reality, and so ought to reflect the entities in the real world. The task is to organise the concepts in the ontology in correspondence to the natural categorisation of the real world entities. Following this, Guarino and Welty (2002) proposed OntoClean, a method to verify the inclusive dependencies in concept taxonomy using ontological meta properties such as rigidity, identity, dependence and unity. Semantic inconsistencies with respect to the real world are detected by analysing the meta properties assigned to each concept using philosophical principles. Fernandez-Lopez and Gomez-Perez (2002) have developed a tool support for OntoClean in WebODE.

Nevertheless, we are hesitant to apply the philosophical approach to the semantic evaluation of ontology because it is still unclear as to how to assign ontological properties to terminological concepts corresponding to abstract entities and entities that originated from empirical knowledge that cannot be grounded in the real world. Moreover, the evaluation of semantic consistency is restricted to subsumption associations, and cannot be extended to domain-specific associations in the ontology. Therefore, we are inclined to follow the criteria based knowledge engineering approach to perform the semantic evaluation.

We concur that the list of general criteria suggested by the knowledge engineering researchers are useful for semantic evaluation. Apart from logical consistency, the conceptual definitions in the ontology ought to be semantically consistent with the area of knowledge to be modelled, which is determined by the need and use of the ontology. Therefore, Annamalai and Sterling (2003) argued that when developing a new ontology or when adapting an existing one, an ontology engineer should consider ontology's ultimate purpose of use. Yu, et al. (2009) agree with our argument and also highlighted in their paper that an ontology should be evaluated in the context of its roles in its applications.

Further, Annamalai (2005) stresses that competency is a consequential quality of a usable ontology that must be present. A competent ontology espouses adequate ontological commitments to provide for the ontology's purposive mechanisms. In other words, a competent ontology is not only conceptually consistent with respect to its frame of reference, but is also fit for its purpose of use, functionally complete in the sense that it can adequately represent the terminological knowledge required to support the functional tasks for which it is designed. This line of reasoning reverberates in the Nicola, et al. (2009)'s proposal that calls for the evaluation of the pragmatic quality of an ontology, which concerns its usability and usefulness for users.

One way to evaluate the competency of an ontology is by means of a premeditated set of competency questions. The notion of competency question was originally conceived by Gruninger and Fox (1995) in the context of knowledge base systems development. The competency questions can be regarded as a form of ontology modelling requirements. They are typically identified by the ontology engineers before the ontology is developed.

There are various approaches for obtaining the competency questions. One approach is through end-users brainstorming. Gruninger and Fox (1995) suggest the use of motivating scenarios to help end-users to articulate the expected queries for a knowledge base system. Annamalai and Sterling (2003) recommend that competency questions are identified in the context of the planned applications to support a domain task; the questions are solicited by interviewing practitioners and experts in the application domain. Gangemi (2005) advocates a top-down approach to construct competency questions by refactoring generic, recurrent questions that arise in the domain knowledge modelling practice.

### 3. COMPETENCY EVALUATION

At present the competency evaluation is performed summatively, i.e., after the related ontology has been fully developed and implemented. This approach is used to corroborate the reliability of the knowledge base constructed based on an ontology, i.e., the possibility to answer the competency questions by using the ontology content. It is seen as an informal method to profile the type of queries that users want to submit to a knowledge base specified by an ontology. The use of competency questions in this manner is espoused in some existing web-ontology development methodologies (Noy & McGuinness, 2001; Sure, et al., 2004; Nicola, et al., 2009).

Even the contemporary web-ontology editors including Protégé provide limited support for competency evaluation based on the idea of summative evaluation. As such, the query is only allowed over the instances of the defined concepts. Instances are extensions of the concepts used in the description of content in the knowledge base, which unfortunately cannot be plausibly created until the ontology has been fully developed.

### 3.1 Competency evaluation as a formative method

Annamalai (2005) argues that competency evaluation should also be performed as part of formative evaluation, i.e., when the ontology is being developed. Formative evaluation helps to ensure that the right ontology is built from the start through progressive validation of the conceptualisation. From this standpoint, the formative evaluation of competency can better guide the development of usable ontologies.

The problem is the iterative development of an ontology undergoes successions of additions, deletions and modifications. Therefore, the attempt to extend the evolving concepts during the development of an ontology will give rise to inconsistencies in the knowledge base; a useless effort and a waste of time. So, the competency evaluation currently practised, which may be suitable for assessing the utility of an implemented ontology, is impracticable for the formative evaluation of ontologies.

Accordingly, Annamalai and Teo (2007) proposed an alternative method that backs the use of competency questions to formatively evaluate an ontology. Their approach checks the competency questions directly against the ontology, and not against the improperly extended instances of its evolving concepts. The fundamental assumption is that the premeditated competency questions assert the requirements that an underlying ontology must meet in terms of its planned applications. Therefore, a usable ontology should specify the conceptualisation required in order to express all of the competency questions and their corresponding answer statements. The competency evaluation corroborates the applicability of the conceptual definitions by checking and ensuring that the ontology represents all the necessary concept, property and relation terms in a computable manner.

### 3.2 Role of competency questions

As a case in point, using an envisaged newspaper company's information system as the context, the first author solicited sixteen competency questions from his Knowledge Management class by means of brainstorming. We give three example competency questions for a Newspaper ontology: “Who authored this article and who edited it?”, “In which news sections are advertisements allowed?” and “Who is responsible for reporting the domestic crimes?”

The evaluation checks the ability of the ontology to express the competency questions and their corresponding answer statements, and the concomitant search for more information that can help to build up the ontology. We illustrate the value of the competency questions in enhancing the Newspaper ontology shown in Figure 1(a) with the aid of the three competency questions mentioned in the previous paragraph.

In order to express these questions, we first try to match the underlined key words in the questions with certain concept, property or relation terms defined in the ontology. The ensuing semantic analysis checks if the requisite concepts and the relationships between them are properly defined in the ontology, and to deliberate on the improvements that are needed.

The first competency question: “Who authored this article and who edited it?” implies that some persons are involved in the authoring a news article and some other persons edit it. The inspection of the relationships between *Article* and *Author* shows that these two concepts are currently related via a normal relation *author*, i.e., “*Article* *author* *Author*”, but they are not related in the inverse; meaning it is not possible to know the articles authored by a person. So, it is a good idea to include the definition of the inverse relation for *author*. Currently, the concepts *Editor* and *Article* are not directly related. A normal relation, say *edit* that relates *Editor* to *Article*, and its inverse relation are needed.

In addition, the inspection of the relationships between *Author* and *Editor* reveals that *Editor* is a subtype of *Author* in the ontology, and so is *News\_Service*. First, the conceptualisation assumes that editors are authors, which may not always be the case. Second, it does not make sense to place an inanimate entity like *News\_Service* as a subtype of a being, *Author*. So, both these conceptually inconsistent subsumption relationships need to be restructured.

Interestingly, *Author* is also not defined as a subtype of *Person* as expected of it. The former is indirectly related to the latter via *News\_Service*, i.e., “*Author* *supertype-of* *News\_Service* *contact-person* *Person*”. A corrective alternative is to replace *Author* with a more general concept, say *News\_Creator*, and subsequently relegate *Author* as the subtype of *News\_Creator*. It should also be made the subtype of *Employee*. Since *Employee* is the subtype of *Person*, the reorganization will make *Author* a subtype of *Person*, and the sibling concept of *Editor* and *News\_Service*. The concepts *Columnist* and *Reporter* can remain as subtype of *Author*.

The second competency question: “In which news sections are advertisements allowed?” entails association between *Newspaper*, *Section* and *Advertisement*. Currently there is an indirect relation between *Newspaper* and *Section* via *Content*, i.e., “*Newspaper* *contents* *Content* *containing-section* *Section*”, as well as via *Prototype\_Newspaper*, i.e., “*Newspaper* *prototype* *Prototype\_Newspaper* *sections* *Section*” (further illustrated in Section 4). However, the indirect relations that currently exist between *Advertisement* and *Section* (via *Person* – *Employee* – *Editor*), i.e., “*Advertisement* *purchaser* *Person* *supertype-of* *Employee* *supertype-of* *Editor* *sections*”

*Section*” fails to express the desired association between *Advertisement* and *Section*. A remedial alternative is to create two concepts, say *Section\_with\_Ad* and *Section\_without\_Ad* as subtypes of *Section*, and to introduce a normal relation, say *advertisement* to relate *Section\_with\_Ad* to *Advertisement*. With these changes, the normal relation *advertisements* that currently relates *Library* and *Advertisement* needs to be reviewed to avoid ambiguity.

The third competency question: “*Who is responsible for reporting the domestic crimes?*” indicates that there are specific persons who are tasked with the responsibility of reporting about certain types of crime-related news. The ontology lacks the necessary concepts and relations to express this question. There are no definitions that typify news or crimes in the ontology. The necessary concept categories need to be introduced. Moreover, the competency question warrants a relationship between *Reporter* and the newly defined news category extended via a relation, say *responsible\_for*. At present, there are no domain-specific associations between *Reporter* and any other concepts in the ontology. Further, the normal relation *responsible\_for* that currently relates *Editor* to *Employee* needs to be reviewed. The above illustrations shed some light on the role of competency questions to instate the requisite ontological commitments in an ontology. The evaluation helps an ontology engineer to validate the ontology’s relevance (fit for purpose of use) by checking the adequacy (functional completeness), coverage (domain space) and depth (level of detail) of the terminology, and exposes the gaps that exist in the ontology; thereby drawing the attention to the need for additional terms and of any required modifications to the existing terminological definitions.

### 3.3 Competency evaluation steps

The formative evaluation of the competency of an ontology is a demanding task undertaken by an ontology engineer. The competency evaluation steps are outlined in Figure 2. First, a competency question or its answer statement is considered (*Step 1*). Second, the key words are extracted from the input sentence (*Step 2*). Next, we try to match the extracted key words with certain concept, property or relation terms defined in the ontology (*Step 3*). If we cannot find a match for a key word, we need to ask whether the unmatched key words are indispensable; if yes, we need to consider defining new terms that could be associated with these key words. The ensuing semantic analysis includes selecting the relevant matched concepts, properties or relations to examine (*Step 4*), followed by the retrieval of the background information about those selected terms (*Step 5*), and the checking if the requisite concepts and the relationships between them are properly defined in the ontology, before allowing for necessary changes to correct the improperly defined terms (*Step 6*).

### 3.4 Competency evaluation issue and solution

The semantic analysis required to check the ability of the ontology to express a competency question, as well as its solution statements is subject to understandability and intellectual judgment, which only humans can perform. Even the preceding steps, which occasionally requires the tracing of the indirect relationships between a pair of concepts can be challenging, especially when dealing with large, rich and dense ontologies.

It may be possible to perform the competency evaluation by examining the concepts and relations in a graphically captured ontology. Sometimes, an ontology engineer has to undertake additional preliminary work such as re-engineering the ontology and producing an intermediate representation before the evaluation can be done. Some web-ontology editors are equipped with visualisation plug-ins that can help to render the graphical representation of the ontology being developed. Nevertheless, it is still cumbersome to trace the intricate links in dense graphs of rich ontologies, and miniaturised graphs of large ontologies. Besides, the established ontology visualisation support for Protégé such as the TGVizTab (Alani, 2003) and Jambalaya developed by the CHISEL Software Engineering Group (2001) lack the simplicity and expressiveness we desire.

Hence, the existing support for competency evaluation of web-ontologies during their formative stage is noticeably weak. Therefore, we have taken the initiative to develop a back-end competency evaluation supporting tool to the popular Protégé ontology editor, which can assist an ontology engineer to perform the competency evaluation of an ontology under development.

The competency evaluation steps that can be automated are tagged with computer icons in Figure 2, and those which cannot be automated are tagged with human icons. The former consists of *Step 2* and *Step 5*, and part of *Step 3*. On the other hand, *Step 1* is a manual process, while *Step 4* and *Step 6* leverage much on analytical reasoning to support the inquiry and the analysis of the semantic structure of the concepts in the ontology. Clearly, no automatic method will ever suffice the task that comprises the latter.

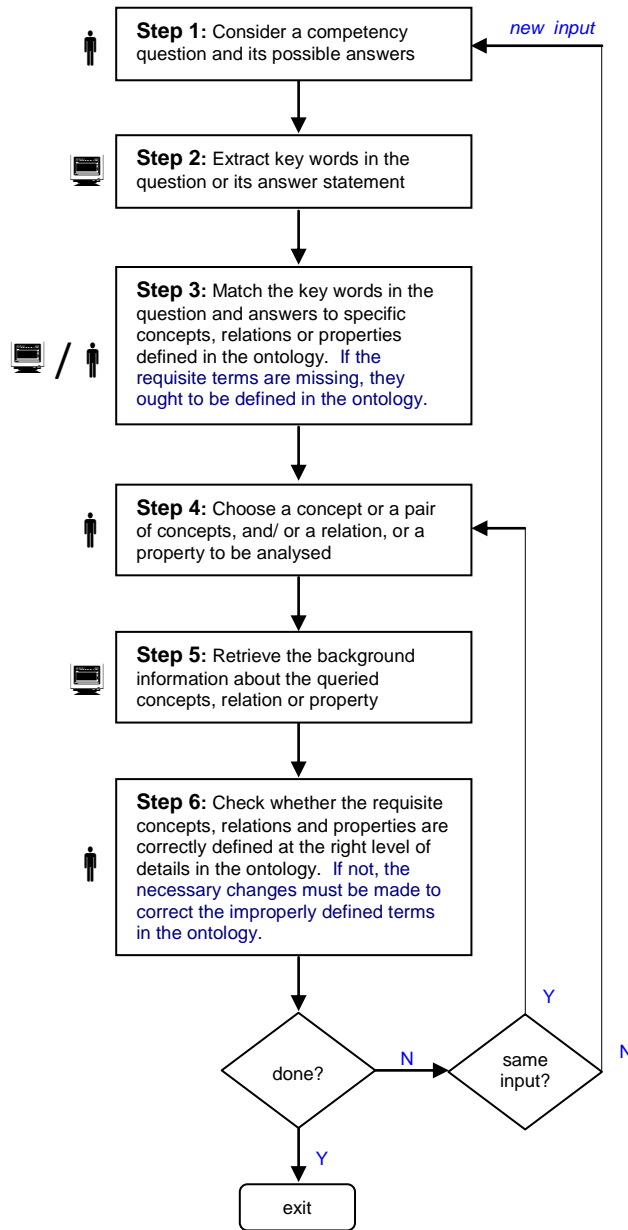


Figure 2. The six steps involved in the competency evaluation of a web-ontology.

For that reason, we plan to semi-automate the competency evaluation, in particular *Step 2*, *Step 5*, and to a lesser extent of *Step 3*. The intention is to dispense with human intervention at these critical points. The tool is especially designed to circumvent the additional preliminary work that is currently performed by an ontology engineer as part of the competency evaluation. The automation of these evaluation steps can help to overcome mistakes due to human oversights, and makes the competency evaluation less subjective and more complete.

The following sections discuss about the competency evaluation supporting tool design and development.

#### 4. TOOL DESIGN

The competency evaluation supporting tool has been designed to closely support the automatable competency evaluation steps described in Section 3.3. The interface window is shown in Figure 3.

First, the user chooses an ontology to evaluate by opening its project file. Next, a sentence representing a competency question or its answer statement is typed in the input box. When the *Find Terms* button is clicked, the tool extracts the key words in the input sentence, and the corresponding concepts, relations and properties that can be matched with these words are identified and listed in the *Concepts*, *Relations* and *Properties* list boxes, respectively. After that, the user can request for the background information about the terms defined in the ontology to be analysed by selecting the relevant concepts, relation, or property in the list boxes and place them in the appropriate text boxes beneath the list boxes.

We have identified six query options by which a user can request for background information about the terms in the ontology, i.e., by providing the name(s) of:

- I. a particular concept,
- II. a particular relation,
- III. a particular property,
- IV. a particular relation directed *from* a certain concept,
- V. a particular relation directed *to* a certain concept, or
- VI. a pair of concepts.

For option I, the tool lists the properties associated with the named concept, and all relations that originate from it or directed to it. For option II, the tool lists all the concept pairs that are directly related through the named relation. For option III, the tool lists all the concepts that are defined by the named property. For option IV, all the concepts immediately succeeding the named predecessor concept through the named relation are listed. For option V, all the concepts immediately preceding the named successor concept through the named relation are listed. Options IV and V are actually subset of option I, but focused on specific relation and its orientation. Finally, for option VI, the tool lists all the direct as well as the indirect relationships between the named source and the target concepts.

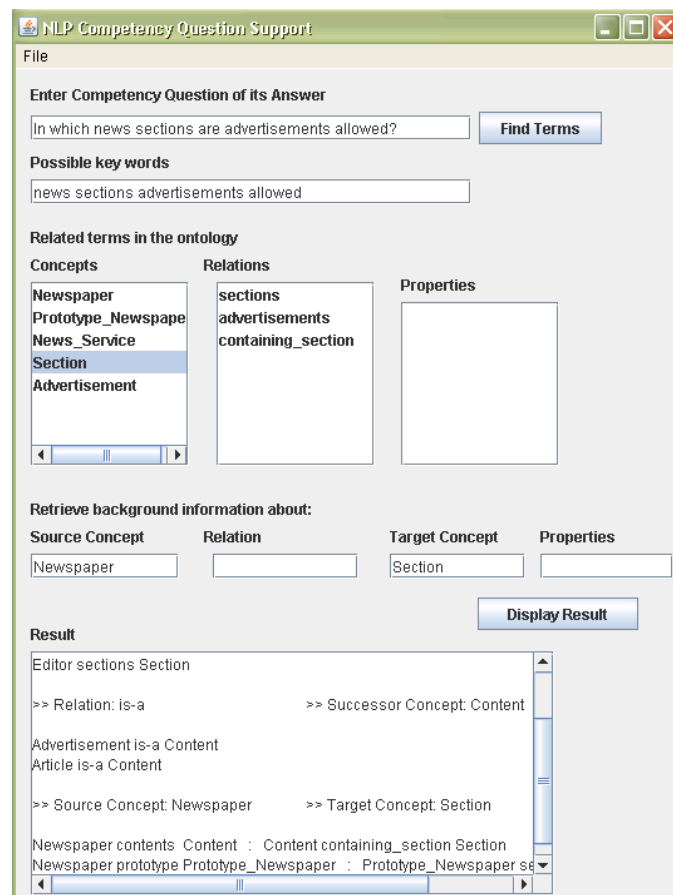


Figure 3. The interface window of the competency evaluation supporting tool



For example, the existing terms in a Newspaper ontology that can be matched with the key words in the competency question: “*In which news sections are advertisements allowed?*” are concepts as *Newspaper*, *Section* and *Advertisement*, and relations as *containing\_section* and *advertisements*. If for instance, we want to find the existing relationships between the concept *Newspaper* and the concept *Section*, we choose option VI, and copy *Newspaper* to the *Source Concept* text box and *Section* to the *Target Concept* text box, or vice versa.

The background information about the queried concepts, relation or property is displayed in the *Result* window at the bottom. In the above example, the tool generates and outputs all the possible relationships between *Newspaper* and *Section*. One possible relationship between these two concepts is through the intermediary concept *Content*, i.e., “*Newspaper contains Content containing\_section Section*”, as conveyed by the output statement on the second last line in the *Result* window in Figure 3. Subsequently, the user can refer to the structured result to make informed competency evaluation as illustrated in Section 3.2.

The user can continue to do the analysis by finding the background information about the other terms related to the same input that are listed in the *Concepts*, *Relations* and *Properties* list boxes, or repeat the process for a new input sentence.

## 5. TOOL DEVELOPMENT

The tool is developed using the Java programming language and is casted as a backend application equipped with a natural language interface to the Protégé ontology editor. We developed the tool in two parts. We begin by automating the retrieval of the background information related to the queried terms (*Step 5* in Figure 2).

At this stage, the user has to manually identify the key words in an input sentence, and try to match these words with specific concept, property or relation terms in the ontology; a daunting task when evaluating large ontologies. It will be better if the user can enter the competency question or its answer statement verbatim as input, and let the tool help to find fitting terms in the ontology for the ensuing analysis. So, we extended the first part with a natural language program interface that facilitates the automatic extraction of the key words from the input sentence (*Step 2* in Figure 2), and then try to match these words with specific terms defined in the ontology (*Step 3* in Figure 2). The following subsections describe the tool development in detail.

### 5.1 Retrieval of background information about the terms

The names of the concept, property and relation terms defined in an ontology are extracted from the Protégé project file, and stored in three distinct vectors. In addition, the conceptual relationships in the ontology are captured separately as a directed graph and represented in an adjacency matrix that facilitates easy access and manipulation. Consequently, the first five options mentioned in Section 4, by which a user can request the background information about the terms in the ontology can be implemented quite straightforwardly. Nevertheless, it requires systematic searching of the data in the term vectors and the adjacency matrix to retrieve the relevant background information.

On the other hand, the implementation of the sixth and last option is rather challenging because we need to find not only the direct relationships between two concepts, but also all the indirect relationships that exist between them. For this, the algorithm of Warshall (1962) is adapted to compute the transitive closure of the binary relationships in the ontology. The original Warshall’s algorithm simply manipulates boolean values, and so can only indicate the presence or the absence of path connecting two vertices in a graph. Therefore, the algorithm has to be modified to store and manipulate textual information in the adjacency matrix. A marking scheme is also employed to represent the tracking information constructed during the computation of the transitive closure. The tracking information is updated to the adjacency matrix and is later utilised during the generation of the output. The work on this part of the tool development is reported in a complementary paper by Annamalai and Teo (2007).

### 5.2 Natural language program interface

In the context of ontology evaluation, the key words are descriptive parts of a competency question or its answer statement, whose meaning, if important, should be defined in the ontology. Therefore, we started out by studying the morphology of concepts, relations and properties defined in web-ontologies. For this, we selected five prototype ontologies of disparate domains in the Protégé ontology library of the Protégé Community (n.d.); each developed by different groups of people. The ontologies are for the domains of Newspaper, Time, Wine, Clinical and Petri-net. The following are our general observation about the syntax and the morphological forms of the terms defined in these ontologies.

- a. A single term is denoted singly by a word, while a phrasal term is denoted by a combination of words.

- b. The words in a phrasal term are joined using underscores or hyphens (e.g. *containing\_section*, *number-of-pages*)
- c. Concepts are commonly described using singular nouns.
- d. Properties are described using singular or plural nouns.
- e. Relations are often described using verbs; and at times using nouns. The verbs can be in present or past tense, or participles.
- f. Prepositions (e.g. *the*, *of*, *on*) are not used to denote a single term; however, their appearance in a phrasal term is always followed by a verb or a noun.
- g. Adjectives are sometimes used together with a noun to describe a phrasal term.
- h. Adverbs are seldom used in the description of a phrasal term.
- i. Pronouns are also not used to denote a single term.
- j. Interrogatives (e.g. *who*, *what*, *why*) are not used to denote a single term.

The above observation provides useful information for developing the natural language interface functions. We established that either a noun or a verb is always used to constitute a term name, and words that do not carry independent meaning are not used to denote a single term. If at all, they appear in a phrasal term together with a noun or a verb. Such words whose semantic value is dependent on their lexical context are generally referred to as stopwords. A list of English stopwords is given in (Snowball, 2001a). Since stopwords are not used to denote a single term, we can safely ignore them when processing an input sentence. The idea is to limit the number of words in the input that could be associated with the terms in the ontology. Consequently, we identified the key words in an input by removing the formatting symbols and the stopwords with the aid of a look-up list of exceptions.

We face two main challenges when trying to match the key words with the terms in the ontology. First, since the competency questions and their corresponding answers are conceived independent of the terms defined in the ontology, the morphology of the key words in the input sentence and the related terms in the ontology can vary. Direct matching will not be able to match a key words with an allied term although they are inflected forms of a particular root word (e.g. trying to match the key word *writing* with the relation term *wrote*). Second, a key word can appear as part of a term (e.g. the key words *print* in the concept term *Printing\_Service*).

In order to overcome these challenges, we resorted to Porter (1997) and Irregular word stemming, as a means to reconcile the morphological variance between the key words and the terms before checking whether they can be matched. The Porter stemming attempts to conflate the singular and plural nouns (e.g. *editorial*, *editors*), and the inflective variants of the verbs (e.g. *edits*, *editing*) to their basic forms, or stems by algorithmically removing the suffixes.

A shortcoming of Porter stemming is that it cannot handle irregular flexion forms (e.g. *wrote*, *men*). However, such irregularities in English are small, and so can be controlled with the aid of a list of exceptional stems, similar to the way we dealt with the stopwords. In this work, we find the stem of a word by first checking if the word is in the exception list; if not, we apply Porter stemming to strip the suffixes and output the resulting stem. Though not complete, the combined stemming approach conflates a much wider variety of morphological forms. We describe the steps for stemming the key words and the terms below.

#### **A. Stemming key words and single terms**

1. Apply irregular word stemming by looking up the list of exceptional stems.
2. If the word is not stemmed in *Step A1*, apply Porter stemming.
3. Maintain a relation that connects the original term to its stem (It is not necessary to do this for the stemmed key words).

#### **B. Stemming phrasal terms**

1. Remove the formatting symbols.
2. Remove the stopwords.
3. Apply irregular word stemming on the remaining part, i.e., each of the leftover words.
4. Apply Porter stemming on each of the unstemmed words in *Step B3*.
5. Maintain a relation that connects the original term to each of its stems.

We adapted the natural language processing software utilities developed by the researchers in the Computational Semantics Laboratory (n.d.) at Stanford University to implement the stopword removal and word stemming.

Lastly, to check whether a key word can be matched with a term, substring comparison is performed between a key word stem and a term stem. For example, *edit* is the stem of the key word *editing*, which is related to the term *editorial* that is stemmed to *editori*. Therefore, only through substring comparison of *edit* to *editori*, it is possible to match *editing* with *editorial*. Since *editing* can be the term and *editorial* be the key word, the

substring comparison must be done both ways. It is also necessary to maintain a relationship between the original term and its stems because when a substring comparison succeeds, we need to output the related term in the ontology, not its stem.

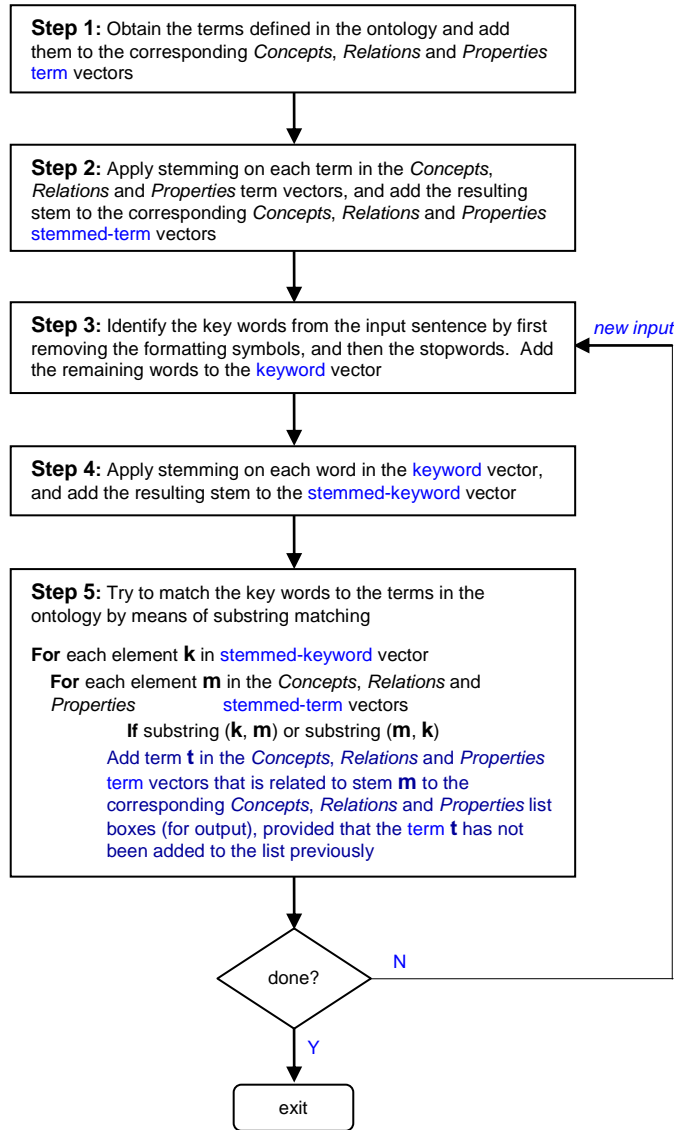


Figure 4. The natural language program interface functioning steps

The key words stemming and the matching processes will be repeated for each input sentence. The program interface functioning steps are outlined in Figure 4.

## 6. TOOL PERFORMANCE ASSESSMENT

The assessment of the developed tool’s ability to meet its purpose of design is organised into two integral parts, namely the assessment of the tool’s utility and its usability. The utility assessment verifies the ability of the tool to perform the prescribed part of the competency evaluation task, while the usability testing is an alpha test conducted in the lab involving potential users to observe how well the tool assists the users in achieving the evaluation task.

## 6.1 Utility assessment

The utility assessment of the developed tool mainly checks on the functionalities of its two key components, namely the natural language program interface and the ensuing background information retrieval functions.

### 6.1.1 Natural language program interface functions

We assessed the key word extraction, stemming and the key word matching functions individually and collectively.

We verified the key word extraction function using many possible competency questions that can be posed to our prototype ontologies. We duly updated the exceptional list of the stopwords to ensure all words that do not fit to be a term name are separated.

We checked the stemming function using the terms in our prototype ontologies mentioned in Section 5.2, and the words in the sample English vocabulary (Snowball, 2001b). The result of the irregular word stemming is checked against sample words that have been arranged into conflation groups of irregular plurals maintained by University of Victoria's English Language Centre (n.d.) and in the Dictionary of irregular verbs (English Page Online, n.d.). We updated the exceptional list of irregular words accordingly to minimise errors due to under-stemming. In the case of Porter stemming, we cross-checked the result of the functions against the output of the online English stemming checker (Snowball, 2001c) that is accessible from the official Porter stemming page (Tartarus, n.d.). In the process, we became aware of the limitations of the Porter's stemming function.

Next, we assessed the interface functions collectively, that includes the key word matching using a set of sixteen competency questions for the Newspaper ontology. The competency questions were solicited from the first author's Knowledge Management class by means of brainstorming. Note that we are less concerned with the validity of the competency questions at this stage than with the questions being conceived independent of the terms defined in the ontology. In this sample evaluation, the interface functions achieved an average of 76% precision and 100% recall. The result of the analysis is shown in Table 1.

The Precision and Recall measurements follow the widely used primary metrics in information retrieval (Baeza-Yates & Ribeiro-Neto, 1999). The Precision measures the ability of the functions to correctly identify the relevant terms in the ontology that are matched with the key words in the questions, while the Recall measures the ability of the functions to identify all the relevant terms in the ontology that should have been identified for a given question. Therefore, based on the number of *actual* terms to be identified, the number of *returned* terms and the number *correct* terms that have been returned, Precision and Recall scores are calculated according to the formula:  $Precision = Correct / Returned$ , and  $Recall = Correct / Actual$ .

Overall, the functions performed satisfactorily and produced the expected total recall of the relevant terms in the Newspaper ontology. The lower precision, however, is due to over-stemming (e.g. *related* and *relatively* results the same stem *relat*), and substring matching (e.g. *man* is matched with *manager* and *secret* is matched with *secretary*) is not an issue because what concerns us most right now is the high recall rate. The tool cannot afford to miss any terms that can somehow be matched with the key words. Since the semantic analysis is performed by an ontology engineer, the redundant terms returned by the interface functions will be ignored.

Table 1. Precision and Recall scores for the sixteen competency questions of the Newspaper ontology

Question	Key words	Terms			Precision	Recall
		Actual	Returned	Correct		
1	4	6	10	6	0.60	1.00
2	4	3	3	3	1.00	1.00
3	4	0	0	0	1.00	1.00
4	3	3	6	3	0.50	1.00
5	2	4	4	4	1.00	1.00
6	3	2	2	2	1.00	1.00
7	2	2	2	2	1.00	1.00
8	7	4	7	4	0.57	1.00
9	6	1	6	1	0.17	1.00
10	7	6	12	6	0.50	1.00
11	4	2	2	2	1.00	1.00
12	3	5	6	5	0.83	1.00
13	5	3	8	3	0.38	1.00
14	3	7	7	7	1.00	1.00
15	5	6	11	6	0.55	1.00
16	4	5	5	5	1.00	1.00
<b>Average</b>					<b>0.76</b>	<b>1.00</b>

### 6.1.2 Background information retrieval functions

We verified the functioning of the background information retrieval functions by checking whether they meet the design requirements. In particular, we assessed the functions ability to properly process the six types of user requests mentioned in Section 4, and accurately retrieve the background information about the terms defined in the ontology. The retrieval process includes the correct generation of indirect relationships that exist between the concepts. We performed this assessment using the Family, Newspaper and Wine prototype ontologies. The Newspaper and Wine ontologies are obtained from the Protégé ontology library. The Wine ontology is large with richly defined relationships. It has 108 concepts and 109 relations. The mid-size Newspaper ontology has 24 concepts and 42 relations. The smaller Family ontology has 6 concepts and 34 relations. In all our test cases, the background information retrieval functions have demonstrated their capability to respond correctly to the six types of retrieval requests by producing the expected result as output.

### 6.2 Usability testing

Finally, we also conducted a simple usability test to observe the ease of using the tool by involving three potential users who are familiar with the development of web-ontologies. The users were given a set of sixteen competency questions to evaluate the competency of the Newspaper ontology. Their task is to determine whether the concepts, relations and properties defined in the ontology are adequate to express the competency questions.

Using eight questions, the competency evaluation was first performed directly on the Protégé ontology editor windows with the aid of paper and pencil. Then, the competency evaluation supporting tool was employed in the evaluation of the remaining eight questions. From our observation, and as acknowledged by the users, the tool was easy to use, and has helped to reduce the competency evaluation time by half with apparent reduction in the effort required to perform the evaluation task. The Newspaper ontology has 24 concepts and 42 relations. Clearly, the saving on time and effort will be significantly higher when evaluating larger ontologies.

The only usage problem we observed was with the tracing long output statements displayed in the *Result* window (see Figure 3). We suppose a graphical representation of the concepts and relations in the structured output (a sub-ontology) will be more appealing to the users.

## 6. CONCLUSION AND FUTURE WORK

A competent ontology is characterised by its ability to express competency questions, as well as to represent their solutions. Competency evaluation is a semantic analysis that checks if the requisite concepts and the relationships between them are properly defined in the ontology, and to deliberate on the improvements that are needed. Currently the competency evaluation that is performed as part of formative evaluation of an ontology being developed is by and large carried out manually. Consequently, we have taken the initiative to develop a semi-automated competency evaluation supporting tool.

The tool is developed as a backend application equipped with a natural language interface (in English) to Protégé, a popular web-ontology editor. The tool dispenses with much of the human effort in the steps preceding the semantic analysis.

The natural language processes mainly uses stopword elimination, stemming and substring matching functions to enact *Step 3* in Figure 2. We have decided to keep it simple for two reasons: a) English is considered to have a simple morphology, the use of basic stemming approach is regarded sufficiently effective for information retrieval; b) the index (number of terms in the ontology) is not too large to affect the precision of the retrieved result. Nevertheless, algorithmic stemmers such as the Porter stemmer that we have employed, conflate different word variants using a rule based scheme, which ignore word meanings. As a result, the algorithmic stemming does suffer from loss of precision in the key word matching (see Section 6.1.1) due to over- or under- stemming.

We think, it is possible to improve the pertinent matching between the key words and terms by first determining the lemma for the words. Using the base form of a word and the context, we can find more information about the word such as its synonyms, hypernyms and hyponyms. We can then try to apply lexical restrictions to increase the precision through semantic word matching. With the help of a synonym matcher, we can also retrieve the terms with the same meaning as a key word. Hence, it has the potential of increasing both precision and recall. We intend to incorporate these natural language functions to further enhance the matching process as part of our future work. The competency evaluation supporting tool described in this paper is a working prototype of this idea.

Although this prototype tool is currently equipped with simple features, it has demonstrated its potential as a useful and reliable support for an ontology engineer to make informed evaluation (see Section 6.2). From our observation, and as acknowledged by the users, there is an apparent gain in transparency and analysability; and, as a consequence, it makes the competency evaluation easier, less subjective, more complete and faster.

## REFERENCES

- Alani, H., 2003. TGVizTab: An ontology visualisation extension for Protégé. *International Conference on Knowledge Capture*. Florida, USA.
- Annamalai, M & Sterling, L., 2003. Guidelines for constructing reusable domain ontologies, *International Conference in Autonomous Agents and Multi-agent Systems Workshop on Ontologies in Agent Systems*. Melbourne, Australia.
- Annamalai, M., 2005. Formative evaluation of ontologies for information agents. *Conference in Computer Science, Technology and Networking*. Shah Alam, Malaysia.
- Annamalai, M. & Teo, N. I., 2007. *Tool support for competency evaluation of web-ontologies*, *International Conference on Informatics*. Petaling Jaya, Malaysia.
- Arpirez, J. C., Corcho, O., Fernandez-Lopez, M. & Gomez-Perez, A., 2001. WebODE: A scalable workbench for ontological engineering. *International Conference on Knowledge Capture*. Victoria, Canada.
- Baeza-Yates, R. & Ribeiro-Neto, B., 1999. *Modern Information Retrieval*. Addison Wesley.
- Brewster, C., Alani, H., Dasmahapatra, S. & Wilks, Y., 2004. Data Driven ontology evaluation. *International Conference on Language Resources and Evaluation*. Lisbon, Portugal.
- Chandrasekaran, B. & Josephson, J., 1999. What are ontologies, and why do we need them? *IEEE Intelligent Systems*, January/February issue, pp. 20-26.
- CHISEL Software Engineering Group, 2001. *Jambalaya – Information Browser for Protégé*. Retrieved from <http://protege.stanford.edu/plugins/jambalaya/jambalaya-simple-backup.htm> [Accessed on 5 December 2009].
- Computational Semantics Laboratory, (n.d.). *API Resources*. Retrieved from <http://godel.stanford.edu/public/doc-versions/util/doc/api/> [Accessed on 5 December 2009].
- Corcho, O., Gomez-Perez, A., Gonzalez-Cabero, R. & Suarez-Figueroa, M.C., 2004. ODEval: A tool for evaluating RDFS, DAML+Oil and OWL concept taxonomies. *International Conference on Artificial Intelligence Applications and Innovations*. Toulouse, France.
- DARPA Agent Markup Language Program, 2000. *DAML DARPA Agent Markup Language*. Retrieved from <http://www.daml.org/> [Accessed on 5 December 2009].
- English Language Centre, (n.d.). *Irregular Plurals of Nouns*. Retrieved from <http://web2.uvcs.uvic.ca/elc/studyzone/330/grammar/irrplu.htm> [Accessed on 5 December 2009].
- English Page Online, (n.d.). *Irregular Verb Dictionary*. Retrieved from <http://www.englishpage.com/irregularverbs/irregularverbs.html>. [Accessed on 5 December 2009].
- Fernandez-Lopez, M. & Gomez-Perez, A., 2002. The integration of OntoClean in WebODE. *International Conference on Knowledge Engineering and Knowledge Management Workshop on Evaluation of Ontology-based Tools*. Sigüenza, Spain.
- Fox, M. & Gruninger, M., 1998. Enterprise modelling, *AI Magazine*, 19(3), pp. 109-121.
- Gangemi, A., 2005. Ontology design pattern for semantic web content. *International Semantic Web Conference*. Galway, Ireland.
- Gómez-Pérez, A., 2004. Ontology evaluation. In: S. Staab & R. Studer, eds. *Handbook on ontologies*, Springer, pp. 251-274.

- Gruber, T. R., 1995. Toward principles for the design of ontologies used for knowledge sharing. *Journal of Human-Computer Studies*, 43(5), pp. 907-928.
- Gruninger, M. & Fox, M., 1995. Methodology for the design and evaluation of ontologies. *International Joint-Conference in Artificial Intelligence Workshop on Basic Ontological Issues in Knowledge Sharing*. Montreal, Canada.
- Guarino, N. & Welty, C., 2002. Evaluating ontological decisions with OntoClean, *Communications of the ACM*, 45(2), pp. 61-65.
- McBride, B., 2004. The Resource Description Framework (RDF) and its vocabulary description language RDFS. In: S. Staab & R. Studer, eds. *Handbook on ontologies*, Springer, pp. 51-66.
- Nicola, A. D., Missikoff, R. & Navigh, R., 2009. A software engineering approach to ontology building. *Information Systems*, 34, pp. 258-275.
- Noy, N. F., Ferguson, R. W. & Musen, M. A., 2000. The knowledge model of Protégé-2000: Combining interoperability and flexibility. *International Conference on Knowledge Engineering and Knowledge Management*. Juan-les-Pins, France.
- Noy, N. & McGuinness, D. L., 2001. *Ontology development 101: A guide to creating your first ontology*. Technical Report SMI-2001-0880. Stanford Medical Informatics: Stanford University, USA.
- Porter, M. F., 1997. An algorithm for suffix stripping. In: K. Jones & P. Willet eds. *Readings in Information Retrieval*, Morgan Kaufmann, pp. 313-316.
- Protégé Community, (n.d.). *Protégé Ontologies Library*. Retrieved from <http://protege.cim3.net/cgi-bin/wiki.pl?ProtegeOntologiesLibrary> [Accessed on 5 December 2009].
- Snowball, 2001a. *An English Stopword List*. Retrieved from <http://snowball.tartarus.org/algorithms/english/stop.txt> [Accessed on 5 December 2009].
- Snowball, 2001b. *Sample English Vocabulary and their Stemmed Equivalent*. Retrieved from <http://snowball.tartarus.org/algorithms/english/diffs.txt> [Accessed on 5 December 2009].
- Snowball, 2001c. *Snowball - Demo*. Retrieved from <http://snowball.tartarus.org/demo.php> [Accessed on 5 December 2009].
- Sure, Y., Jurgen, A. & Staab, S., 2003. OntoEdit: Multifaceted inferencing for ontology engineering. *Journal on Data Semantics*, 1(1), pp. 128-152.
- Sure, Y., Staab, S. & Studer, R., 2004. On-To-Knowledge methodology. In: S. Staab & R. Studer, eds. *Handbook on Ontologies*, Springer, pp. 117-132.
- Tartarus, (n.d.). *Porter Stemming Algorithm*. Retrieved from <http://tartarus.org/~martin/PorterStemmer/index.html> [Accessed on 5 December 2009].
- Tartir, S. et al., 2005. OntoQA: Metric-based ontology quality analysis, *International Conference in Data Mining Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*. Texas, USA.
- Uschold, M., 1996. Building ontologies: Towards a unified methodology. *Conference of the British Computer Society Specialist Group on Expert Systems*. Cambridge, UK.
- Uschold, M. & Gruninger M., 2004. Ontologies and semantics for seamless connectivity, *SIGMOD Record*, 33(4), pp. 58-64.
- Wang, H. et al., 2006. Frame and OWL side-by-side. *International Protégé Conference*. Stanford, USA.
- Warshall, S., 1962. A theorem on boolean matrices. *Journal of the ACM*, 9(1), pp. 11-12.
- W3C Recommendation, 2004. *OWL Web Ontology Language*. Retrieved from <http://www.w3.org/TR/owl-ref/> [Accessed on 5 December 2009].
- Yu, J., Thom, J.A. & Tam, A., 2009. Requirements-oriented methodology for evaluating ontologies. *Information Systems Databases: Their Creation, Management and Utilization*, 34(8), pp. 686-711.