INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, France

A METHODOLOGY OF ENGINEERING ONTOLOGY DEVELOPMENT FOR INFORMATION RETRIEVAL

Zhanjun Li[1, 2], Victor Raskin[1, 3] and Karthik Ramani[1, 2, 4]
[1]Purdue Research and Education Center for Information Systems in Engineering (PRECISE)
[2]School of Mechanical Engineering
[3]Department of English and Linguistics
[4]School of Electrical and Computer Engineering (by courtesy)
Purdue University, West Lafayette IN, USA

ABSTRACT

When engineering content is created and applied during the product lifecycle, it is often stored and forgotten. Since search remains text-based, engineers do not have the means to harness and reuse past designs, experiences, and mistakes. On the other hand, current information retrieval approaches based on statistical methods and keyword matching are not directly applicable to the engineering domain. The long term goal of this research is to develop an engineering ontology based computational framework in order to (1) structure unstructured engineering documents; and (2) achieve more effective information retrieval. This paper focuses on the method and process that acquire and evaluate the engineering ontology. We propose a principled, systematic, and semi-automatic ontology development methodology that is based on ontological semantics and is integrated with Protégé, one of the most widely used ontology engineering tools. The methodology is applied in acquiring the established engineering knowledge from various resources. A preliminary test based on engineering catalogs, CAD drawings, and project reports has been conducted. The results validate the effectiveness of the engineering ontology as well as the methodology.

Keywords: Engineering ontology, Knowledge acquisition, Engineering information retrieval, Reuse

1 INTRODUCTION

Engineers are dependent on accessing documents in order to fulfill various design and engineering tasks. In fact, today's engineers simply do not make an effort to find engineering content beyond mere keyword searches [1]. However, current information retrieval (IR) approaches either retrieve too much or irrelevant results for engineering. In industry sectors, it was reported that design engineers spent 20% to 30% of their time retrieving and communicating information [2]. "Delivering the right information to the right people at the right time" plays an important role in supporting engineers' memory extension, knowledge sharing, design concept exploration, design reuse, and the learning process particularly of novice engineers [3, 4]. However, current engineering practices ignore reuse of previous knowledge because appropriate engineering information retrieval tools have not been developed. As a result, a large amount of time is spent reinventing what is already known in the company or is available in outside resources [5, 6]. It is, therefore, imperative to minimize such overhead by developing the science base for contextual retrieval and then using this knowledge to create effective computer-aided tools.

Statistics-based methods and keyword-based input have been prevalent in IR research such as vector space model [7], latent semantic analysis [8], language modeling [9], and probabilistic model [10]. They can be viewed as sophisticated stochastic techniques for matching terms from queries with terms in documents under the assumption of term independence. They try to derive the meaning of the text from the observable syntactic and statistical behavior of its units without any attempt to represent the meaning directly. However, words alone cannot capture the semantics or meanings of the document and query intent. To put it differently, the search results should satisfy the users, who are looking for something that matches their understanding of pertinent text—an understanding that includes, among other things, the relations among the terms and the ability to disambiguate and to infer. This is where the statistical keyword-based techniques fail the users and defeat their purposes.

In the engineering domain, there has been very limited research aimed at analyzing unstructured engineering documents for retrieval purposes. Most of it has been based on IR approaches. Dong and Agogino [11] proposed to use vector space model and belief networks to represent design manuals. Ahmed et al. [12] developed taxonomies in order to index corporate documents. The vector space model was also used to classify the documents against the terms in the taxonomies. Yang et al. [13] attempted to automate the population of a thesaurus from notebooks by using the latent semantic analysis. McMahon et al. [4] employed predefined taxonomy to classify documents by rule-based matching.

Current approaches (1) do not attempt to provide the semantics-based representation of engineering documents or provide for engineers' information needs; (2) do not reflect and utilize engineering knowledge in the organization of the search; and (3) are unable to handle complex queries that have qualitative as well as quantitative engineering specifications.

The long term goal of this research is to develop a content-oriented, knowledge and meaning based computational framework to form the ontological basis of the search, browsing, and learning tasks in the engineering domain. This paper focuses on investigating the method and process required to develop such ontological basis. The proposed methodology

1. Specializes the ontological semantics methodology proposed by Nirenburg and Raskin [15] for machine translation and natural language understanding;
2. Represents a structured process in developing the engineering ontology (EO) and its associated engineering lexicon (EL);
3. Formalizes the cumulative domain knowledge such as the classification of mechanical elements, their function, design, and manufacturing knowledge and formulates in a single standard format;
4. Integrates with an ontology engineering tool;
5. Incorporates semi-automatic tools into the practical acquisition process; and
6. Evaluates the acquired EO by using principled and empirical methods.

Section 2 first provides definitions of an ontology and its distinct features compared to other representation schemas. Current ontology acquisition methods are summarized. An overview of our knowledge-based computational framework is described in section 3. Section 4 discusses in detail the proposed engineering ontology development methodology and the acquired EO. The evaluation method and experiment are introduced in section 5. Section 6 concludes the paper.

2 RELATED WORK

2.1 Ontology Definition

An ontology is a constructed model of reality, a theory of a domain. In more practical terms, it is a highly structured system of concepts covering the processes, objects, and attributes of a domain as well as all their pertinent complex relations. The grain sizes of the concepts are determined by considerations such as the need for an application or for computational complexity.

From one aspect, an ontology can be viewed as a decomposition of a domain: it is a tangled hierarchy of conceptual nodes, each of which can be represented as:

property-slot (CONCEPT-NAME, PROPERTY-VALUE/FILLER-CONCEPT-NAME) +

Every concept but the root of the ontology has the property-slot is-a, and the value of this property is the parent of this concept. A concept may have multiple parents and multiple inheritances.

From the other aspect, an ontology reflects the correlations among concepts across sub-domains: the PROPERTY-VALUE of a concept refers to its filler concept, i.e., these two concepts are connected by the specific property-slots, i.e., (binary) relationships.

Ontologies share the inheritance feature with the object-oriented (OO) programming languages, which are indeed suitable for implementing ontological procedures. However, in OO programming, the focus is on designing the operational properties, i.e., the methods of a class, whereas ontology development is based on the structural properties, i.e., relationships of a class. More importantly, the OO approach lacks the conceptual content of ontologies, and it is not sufficient for addressing the rich knowledge modeling needs discussed here. The distinction between form and content is crucial for understanding the proposed ontology model. It is the content of ontologies that makes them useful for this application, independent of the choice of form, i.e., format or language. Currently, there is also confusion between taxonomy and ontology based applications. One of the major differences between taxonomies and ontologies is that an ontology represents much richer domain contexts than a taxonomy or a list of taxonomies. A taxonomy is a hierarchical classification of concepts in a sub-domain. These concepts are connected only by domain-independent (or taxonomic) relationships such as is-a. An ontology, however, consists of several taxonomies, along with multiple domain-specific (or non-taxonomic) relationships to connect concepts across taxonomies. See [14] for comparisons between ontologies and database schema, as well as those between ontologies and knowledge representation; [15] for an extended view of what a full-fledged ontology must be and how to bring that about; and [16] for an extensive survey on existing ontological systems from manufacturing and knowledge sharing perspectives.

2.2 Methods for Ontology Development

While ontologies have found many applications in the fields where semantics-based communications among people and systems are crucial [14], only a few methods for developing ontologies have been reported.

The method used to build the Cyc [17] ontology consists of general steps and codification of articles and pieces of knowledge. Manual process is used to extract the common sense knowledge that is implicit in different sources. Latterly proposed methods all start from the identification of the scope and the need for the ontology: The work by Gruber [18] represents the first attempt to consolidate experience gained in developing ontologies. It can be summarized as five ontology design criteria: clarity, coherence, extensibility, minimal ontological commitment, and minimal encoding bias. Uschold and King [19] developed Enterprise Ontology for enterprise modeling processes. Their development method includes four activities: 1) identification purpose, 2) build ontology, 3) evaluation, and 4) documentation. They also proposed three strategies for identifying the concepts in the ontology: top-down, bottom-up, and middle-out. Grüninger and Fox [20] proposed an ontology design and evaluation method while developing the TOVE (Toronto Virtual Enterprise) project ontology for business processes and activities modeling. It uses a set of natural language questions, called competency questions to determine the scope of the ontology and to extract the main concepts of the ontology as well. However, the major focus is on in building the first order logical model representation of the ontology. A similar method was introduced by Noy and McGuinness [21] with an example of wine ontology acquisition using Frame-based representation. Fernandez et al. [22] presented a more structured method and life cycle definition for developing ontologies from scratch, called METHONTOLOGY. However, the evaluation is purely subjective. Notice that the implementation step (manually editing and coding ontologies in a specific language) specified in most of these methods is not necessary any more because of the maturity of the ontology engineering tools nowadays [23].

Among the recently proposed ontology acquisition methods in engineering, Eris et al. [24] presented an initial ontology framework in modeling product development projects in small teams. Nanda et al. [25] applied the formal concept analysis to form the product family ontology of one-time-use cameras. Ahmed and Wallace [12] intended to design an ontology development process which can be customized for a particular manufacturing company. However, their acquisitions did not explicitly explore the domain-specific relationships among concepts and therefore, the acquisition result is a list of independent taxonomies, not an ontology.

In summary, very little effort has been made to systemize the established knowledge in the engineering domain by using formal ontology representation for the purpose of more effective information retrieval. Most of the current ontology development methods still require tremendous effort and subjective judgments from the ontology developers to acquire and maintain the ontology. To our limited knowledge, no attempt has been made to evaluate the resulting ontologies both from the principle perspective and the application perspective. It is critical to investigate a principled, systematic, and more structured acquisition method combined with an evaluation process for developing such an engineering ontology in order to support knowledge and meaning based information retrieval. Our method also is different because it uses EL to formalize the lexicon knowledge in order to bridge the concept-based representation of the ontology and the word-based representation of documents and user queries.

3 OVERVIEW OF EO-SEARCH

Figure 1 shows the overall architecture of interactions between the ontological basis, i.e., the EO and EL, with other functional modules applied to the knowledge-based engineering information retrieval framework (EO-Search). The framework comprises six portions: pre-processing, ontology basis, ontology acquisition and maintenance, concept tagging, concept indexing, and query processing. Note that concept tagging also refers to the empirical process of evaluating the EO and EL in section 5.

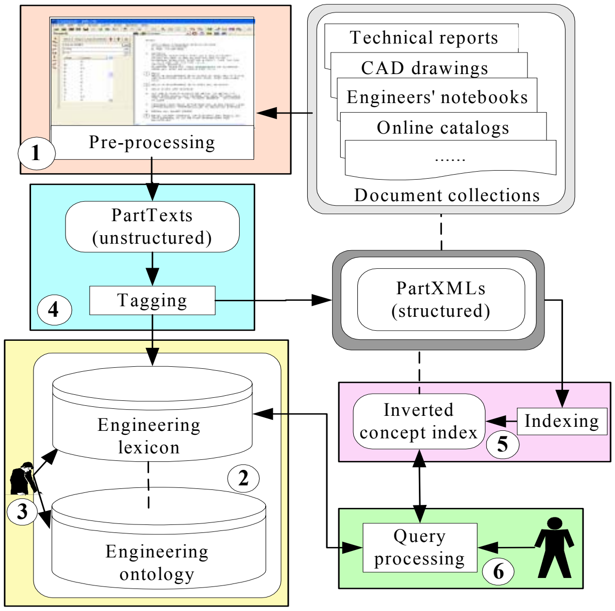


Figure 1. System architecture and functional modules

1. Pre-processing: Consolidating heterogeneous documents
2. Ontology basis: Engineering Ontology & Engineering Lexicon
3. Ontology acquisition and maintenance
4. Concept tagging
5. Concept indexing
6. Query processing

1. Pre-processing: The task of pre-processing is to convert engineering documents into a unified format such as .txt files, which can then be processed by the system. The inputs may include catalog descriptions, CAD drawings, technical reports, and engineers' notebooks.
2. Ontology basis: This consists of domain knowledge and lexical knowledge, i.e., the EO and its associated EL, respectively. They are used to assist in recognizing technical terms (in documents and queries) at the concept level.
3. Ontology acquisition and maintenance: Protégé 3.1 (http://protege.stanford.edu) is used to build and update the EO and EL. The output scripts from Protégé record the content of the EO and EL. These Frame-based XML scripts are then read into the system to generate the EO and EL in the memory.
4. Concept tagging: The documents in the unified format are tagged by using the concepts in the EO and are then transformed into an XML-based representation. Using EO and EL makes the tagging process less dependent on NLP techniques in understanding the texts. Metadata, such as names of the original documents, are also stored.
5. Concept indexing: An inverted index is generated to index the XML documents. The filenames and the locations where the concept (tag) appears are listed along with the concept. This index is accessed when the system ranks the documents in query processing.
6. Query processing: EO plays an important role in interpreting the user's queries accurately, and therefore improves retrieval performance. Queries with qualitative or quantitative property-value pairs are also handled. Ontology-based query processing algorithms are developed to fulfill these tasks.

   Please refer to [28, 29] for more details of concept indexing and ontology-based query processing.

4 DEVELOPING EO AND EL

The process of developing the EO and its associated EL includes six steps. These are 1) Specification: determining the scope and granularity of the EO; 2) Conceptualization: acquiring the EO (according to the scope and granularity) and EL from various knowledge resources; 3) Formalization: the acquired knowledge is put into structured formats; 4) Population: the formalized knowledge is converted into Protégé's Frame-based representation, manually or automatically; 5) Evaluation: using principled and empirical methods to validate the quality of the EO and EL; and 6) Maintenance: revising the EO and EL. Figure 2 illustrates the development process and the supporting activities in each step. Note that the de facto development of EO and EL is an iterative process. Indeed, the specifications of an ontology may change throughout its life cycle as the definitions are initialized, modified, and deleted.

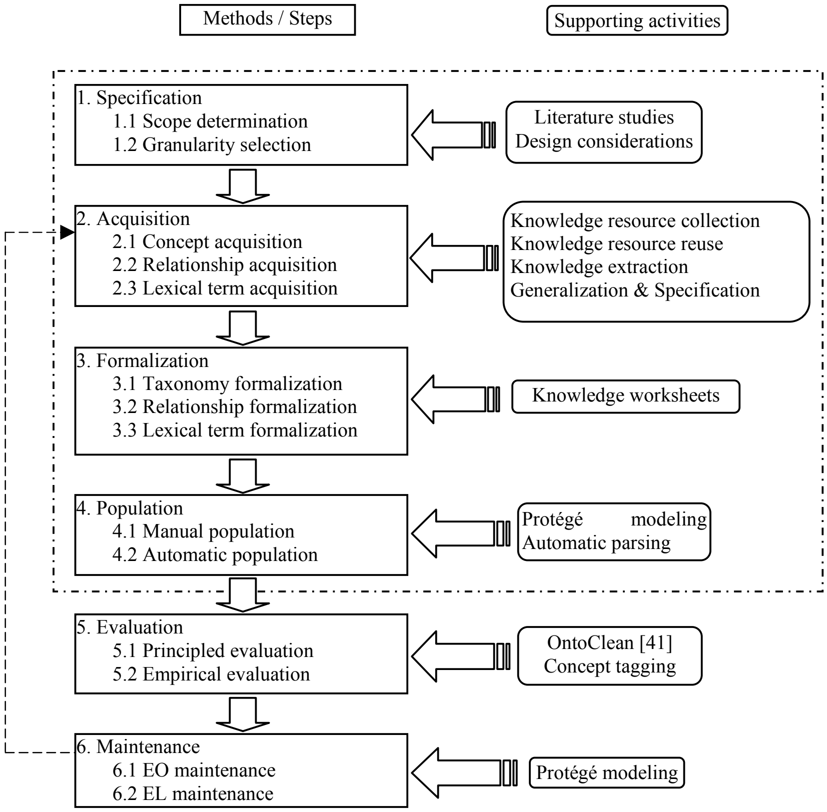


Figure 2. Methodology and process of developing the EO and EL

4.1 Specification

The first step is to identify the scope or themes of the EO for information retrieval purpose. These themes are determined based on the discoveries by cognitive studies in the engineering domain, such as [12, 30-33]. The prior studies have categorized the domain-specific issues being documented during the product development process as well as the information needs of engineers. Currently, in designing the themes of the EO, this research considers issues such as the products or components being designed (i.e., devices), their functions, properties, material selections, shape features, various processes (e.g., manufacturing) in product development, the environmental objects with which the product or component interact, and the standards that certain design or manufacturing entities comply with. The overall schema of the EO is shown in Figure 3. Each taxonomy represents an issue or a sub-domain of the EO. Recall that a taxonomy consists of concepts organized in a hierarchy. However, the EO is differentiated from simply a set of loosely connected taxonomies (at their root level) by having other domain-specific inter-relationships among concepts across these taxonomies. Therefore, what types of inter-relationships exist among concepts and should be acquired must also be determined.

Now the question becomes what level of granularity of knowledge should be taken into account in the EO. Since the goal is to build a search mechanism that is more effective than keyword-based search while less dependent on using NLP techniques to understand documents or queries, the EO must include more specific concepts (lower-level concepts), such as spur gears, as well as more general concept categories (upper-level concepts), such as mechanical components. This is because specific concepts are usually used in documentation while both general and specific concepts may be the interest of users' queries. Note that 1) different brand names of the product or components are not treated as separate concepts; and 2) the instances of the concepts will appear only in the documents by concept tagging rather than as part of the EO.

4.2 Acquisition

Most of the ontology development methods conduct the ontology acquisition in a subjective manner. They generate concepts either by brainstorming (i.e., randomly enumerating a list of terms and then figuring out how they are related to each other), or by interviewing with experts. The first approach may be effective in creating ontologies for simple domains with shallow knowledge.

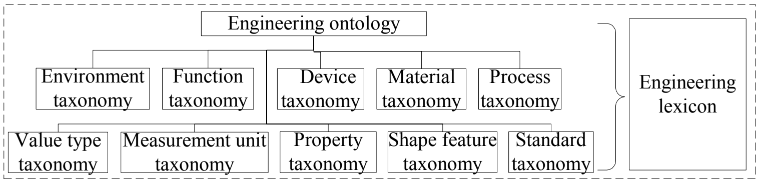


Figure 3. The schema of the ontology basis

However, it is not feasible in developing the EO, which includes broader as well as complex domain knowledge. The second approach may be appropriate if the ontology is built based upon the knowledge in a small domain, such as a company. However, the content of the ontology may be skewed and limited.

The knowledge acquisition task is conducted mainly by utilizing the established engineering knowledge resource (EKR) and analyzing the content of these resources based on our domain knowledge background. Examples of the EKR are engineering handbooks, textbooks, online catalogs, literature, and bill of materials (BOMS). The last one is analyzed in order to acquire the desired knowledge from a specific company. The ontology acquisition consists of three tasks: concept acquisition or taxonomy acquisition, relationship acquisition, and lexicon acquisition. In practice, the first two tasks are done simultaneously. First, the EKRs corresponding to a specific taxonomy, or part of the taxonomy, are collected, for example, material selection handbooks for the material taxonomy. Second, the sentences and phrases which describe the concepts of this taxonomy as well as their relationships with other concepts are extracted, and then documented in free texts. Certain EKRs, such as Function Basis from [34] and motor and pump ontologies from [35], where concepts are already organized, are reused directly. The references from the investigated EKR are recorded. The generated descriptions/documentations are informal representations of the domain knowledge. The lexicon acquisition is integrated with the lexical term formalization in the next step.

4.3 Formalization

When most of the knowledge has been acquired, it is unstructured and needs to be organized and structured by using representations that both computers and humans can understand. Such representations are named "knowledge worksheets." They are formatted templates and independent of the ontology engineering tools or implementation languages used. The worksheets 1) are used as formal documentations of the EO and EL development; 2) direct the acquisition of the EO and EL; and 3) improve the efficiency of the ontology development process by enabling automatic upload of the acquired knowledge into Protégé. They have been used extensively by the undergraduate students who fulfill the acquisition and formalization tasks in our group. In the process of full deployment, the ontological semantic toolbox [15, 36] will be used.

Basically, there are two types of worksheets: taxonomy worksheets and relationship worksheets. Each taxonomy corresponds to a taxonomy worksheet while each concept, in general, has a relationship worksheet. The taxonomy worksheet is used in organizing the unstructured results from the concept acquisition into a hierarchical structure. In our experience, this is the most challenging step of the overall development process. For example, different EKRs may classify the same taxonomy or concept from different perspectives and therefore have to be merged carefully. For

Table 1. The EO contents and acquisition resources

Taxonomies		Num. of concepts	Examples of concepts	Acquisition resources	Examples of acquisition resources
Device	Engineering component	451	D-LOCK-WASHER, D-LINEAR-SLIDE	Engineering texts, Handbooks, Online catalogs	[35, 37], www.globalspec.com
	Proprietary product	190	D-BASE-COVER	BOMs (or Product dissection)	BOMs of the base cover assembly
Function		246	F-SUPPORT, F-LOCK	Existing taxonomies	[34, 38]
Material		1017	M-STAINLESS-STEEL, M-2008-T4 AL	Engineering texts, Handbooks, Online catalogs	[39], www.matweb.com
Process		252	MF-CASTING, MF-WELDING	Engineering texts, Handbooks	[39, 40]
Property		378	P-SHAFT-DIAMETER, P-DUCTILITY	Same as Device taxonomy	Same as Device taxonomy
Measurement unit		64	MU-INCH, MU-FT-LB/SECOND	Online resources	www.ex.ac.uk/cimt/dictunit/dictunit.htm
Shape feature		47	SF-LINEAR-SLOT, SF-TOOTH	Existing taxonomies	STEP AP224, vocabularies of major CAD packages
Environment object		135	E-HEAT, E-AXIAL-LOAD	Engineering texts, linguistic resources	[32], WordNet2.1
Standard		31	S-MIL-STD-130	Standard libraries	www.nssn.org
Value-type		8	V-FLOAT (Numerical), V-HIGH (Symbolic)	Engineering common sense; Online catalogs	N/A

instance, the manufacturing process can be classified from either the functional aspect or the material removal/addition aspect. And some EKRs, especially the online catalogs, may have contradictory classifications (e.g. a child concept becomes an ancestor of its parent). In this case, re-classification is needed. Note that concept naming conventions are applied in order to 1) make the EO more readable; and 2) make each concept unambiguous even at its (term) representation. Otherwise for example, 'cylinder' can refer to both a device concept and a shape feature concept and therefore, can cause ambiguities in the EO, which is not allowed. The naming conventions require that each concept consist of a prefix representing the taxonomy the concept belongs to, in upper case, and that its terms be connected by "-." Therefore, the two concepts in the previous example are written as D-CYLINDER and SF-CYLINDER, respectively. Table 1 lists more details of the EO concepts and the acquisition resources.

In general, concepts in the EO are connected with their relevant concepts through relationships. For instance, a property concept (e.g., P-PITCH-DIAMETER) is related with some measurement unit concepts (e.g., MU-MILLIMETER) and value type concepts (e.g. V-FLOAT). Exceptions include value type concepts and standard concepts, which are self-contained. Note that these relationships are one-way connections. Definitions of the relationships are given in Table 2. The relationship worksheet describes the relevant knowledge (concepts + relationships) of a concept. Table 2 illustrates that the most significant relationship worksheet is the one for device concepts because a device concept has correlations with most of the other types of concept (see Figure 4). Note that some relationship descriptions may be empty either because such knowledge has not been acquired or because of the characteristics of the concept being described. For example, D-LOCK-WASH does not have has-part relationships with a device concept since it is a part type of component.

Note that the device taxonomy includes classifications of engineering catalog components and proprietary products. The latter one needs to be customized for each specific company including product line classifications, subassembly classifications, and part inventory classifications. The properties of the device concepts are conceptualized in the property taxonomy and connected with the device concepts through the has-property relationship. This is also true for the properties of the material concepts and shape feature concepts.

The lexical terms are the natural language phrases of the corresponding concept. They are used to match with word(s) in documents or queries. Therefore, morphology forms, abbreviations, acronyms,

Table 2 Definitions of the relationships

Relationship	(Concept*,	Filler concept)	Definitions of the relationship	Examples
is-a	Child	Parent	Describes the generalization from a child concept to its parent concepts or the specification from a parent concept to its child concepts	is-a (D-ELECTRICAL-MOTOR, D-MOTOR)
has-part	DC	DC	Represents the part-whole between an DC and the other DC	has-part (D-LINEAR-SLIDE, D-BALL-BEARING)
has-function	DC	FC	Refers to the connection between a DC and one of its FCs	has-function (D-LOCK-WASHER, F-LOCK)
interface-with & Interact-with	DC	DC EC	Complements the has-function relationship when there is an 'object' in the function description of 'subject + verb [+ objects]'. Together, they represent the interactions between an DC and the other DC or EC	interface-with (D-LOCK-WASHER, D-FASTENER); interact-with (D-LOCK-WASHER, E-FRICTION)
has-material	DC	MC	Describes the type of materials used in making the DC	has-material (D-WASHER, M-METAL)
has-process	DC	MFC	Describes the type of manufacturing process used to make/fabricate the DC	has-process (D-GEAR, MF-HOBBING)
use-material	MFC	MC	Describes the type of possible raw materials that certain manufacturing processes act on	user-material (MF-COATING, M-NONFERROUS-METAL)
has-property	DC/MC/SFC	PC	Each DC has several PCs characterizing its attributes such as various physical attributes and geometry attributes; each MC may also have several PCs specifying its characteristics such as physical and mechanical attributes	has-property (D-PLAIN-WASHER, P-INSIDE-DIAMETER); has-property (M-METAL, P-HARDNESS)
has-measurement	PC	MUC	Most of the PCs have one or several MUCs	has-measurement (P-LENGTH, MU-METER)
has-value	PC/MUC	VC	Each PC may have numerical VC or symbolic VC while MUC only has numerical VC	has-value (P-DIAMETER, V-NUMERICAL)
has-feature	DC	SFC	Describes the significant shape features a device may have	has-feature (D-SCREW, SF-THREAD)
has-standard	DC/MC/MFC	SC	Specifies the standard a DC/MC/MFC may comply with	has-standard (D-WASHER, S-ASME B18.13)

and synonyms of the word/phrase are also lexical terms and share the same concept with the original lexical term. For example, *move* and *moving* are lexical terms of the functional concept F-MOVE.

Currently, there are 10 taxonomies, 2,819 concepts and 13 types of relationships in the EO, and more than 10,000 lexical terms in the EL. The EO represents the general domain knowledge as well as the company-specific or proprietary product knowledge. We have investigated the design of a commercialized surgery robot as an example of the proprietary products. The general domain knowledge refers to the knowledge about the more-standardized catalog components, such as gears, and more-customized catalog components, such as linear slides.

D-LOCK-WASHER

Definition
A washer designed to prevent undesired loosening of a nut after it has been tightened

Lexical terms
lock washer, lock washers

Sub-part

Function descriptions
F-LOCK D-FASTENER, F-DISTRIBUTE E-FORCE

Properties
P-INSIDE-DIAMETER, P-OUTSIDE-DIAMETER, P-THICKNESS

Material
M-FERROUS-METAL, M-THERMOPLASTICS

Manufacturing process
MF-COATING

Shape feature
SF-HOLE, SF-TOOTH

Figure 4. Relationship worksheet for 'lock washer'

4.4 Population

The population step refers to modeling the EO and EL by using the ontology engineering tool, i.e., Protégé as well as the generated knowledge worksheets. Two options are provided: manual population and automatic population. The modular structure of the EO and EL lend themselves easily to an expansion, such as the addition of a new relationship or new concept. In Protégé, concepts are modeled as classes while relationships are slots. An attribute (unary relationship) slot named lexical-terms is assigned to each class. This attribute contains all the lexical terms of the pertinent concept. In automatic population, a Protégé plug-in is developed by using Protégé APIs that can read in the knowledge worksheets and generate the EO and EL model. The manual approach is more appropriate for maintenance purposes where limited number of concepts, relationships, or lexical terms need to be changed sometimes. It is more efficient to use the automatic population especially when building the EO and EL from scratch. In automatic population, taxonomy worksheets are loaded prior to the relationship worksheets. However, it is possible that certain concepts which are part of the descriptions in the relationship worksheets may not be defined in the EO yet. Therefore, some human interventions are expected.

5 EVALUATION

The resultant EO is organized in a directed graph or lattice: each node represents a concept and each arc represents a relationship. A portion of the EO is shown in Figure 5.

Now the questions are: How is the EO to be validated? How much does this ontology cover? And how accurate are the concept definitions? The OntoClean [41] is applied in order to validate the choice of the concepts. This method is based on general ontological notions drawn from philosophy, such as essence, rigidity, identity, and unity, which are used to characterize relevant aspects of the intended meaning of the ontology concepts and relationships. Regarding the completeness and accuracy of the EO, because lower-level concepts are taken from various EKRs and more upper-level concepts are added in the EO, we believe the EO covers the domain within the scope of the components and proprietary products modeled.

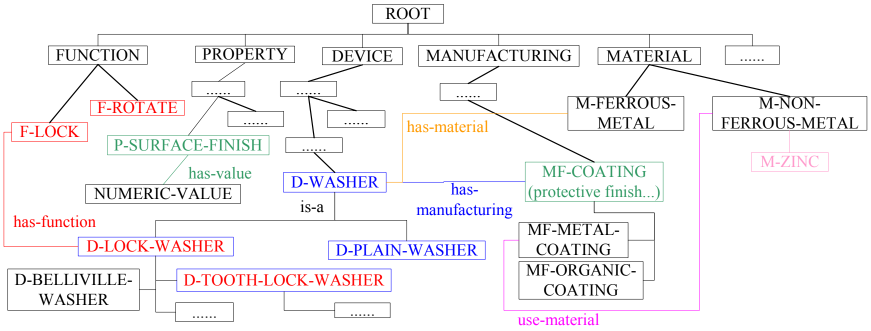


Figure 5. A Portion of the EO

DC: device concept; FC: function concept; EC: environment concept; MC: material concept;
MFC: manufacturing process concept; SFC: shape feature concept; SC: standard concept;
PC: property concept; MUC: measurement unit concept; VC: value type concept

In order to estimate the coverage and accuracy, an experiment is established by using the concept tagging. Recall that concept tagging is also part of the framework for EO-based information retrieval. However, in this experiment, our goal is to select concepts from the tagged document, which includes 1,000 catalog descriptions downloaded from 62 manufacturers and 205 CAD drawings and 11 technical reports from a design company. It is observed that 92.7% of the test documents are associated with the concept of the EO, while 7.3% of the documents failed to associate with any concept of the EO due to incompleteness of the EO or EL. However, after the EO and EL get updated accordingly, tagging errors can be neglected. This observation indicates that the completeness and maintenance issues will be the life cycle issues in using EO for information retrieval.

In order to tag engineering documents by using concepts in the EO, documents from various resources are first converted into .txt files, i.e., *PartTexts*, during pre-processing as in Figure 1. XPDF (www.foolabs.com/xpdf/) is incorporated into our framework in order to convert the PDF documents into the unstructured PartTexts. Certain symbols such as Ø are replaced by their textual descriptions (e.g., diameter). Texts in CAD drawings are extracted using I-migrate, a software program that employs various CAD application program interfaces (APIs) and was donated by our industry partner.

Our method makes use of the EO and EL to recognize concepts contained in the documents. By doing so, the PartTexts are converted into an XML and concept based representation, i.e., *PartXMLs*, where each recognized word/phrase is tagged by the corresponding concept. Figure 6 shows the modules and process of concept tagging.

1. Tokenization: The input character streams are parsed into tokens and punctuation marks.
2. Segmentation: Sentences are formed by using punctuation marks and symbols such as "\n."

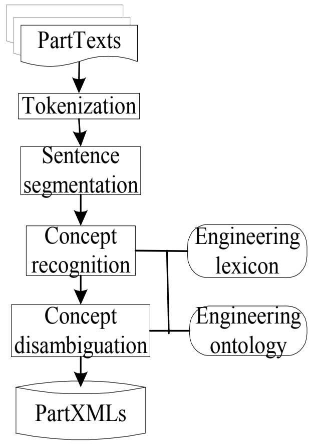


Figure 6. Modules and process of concept tagging

3. Concept recognition:

3.1 Cardinal number recognition: Cardinal numbers such as 3.2, 1:20, and 200 are identified.

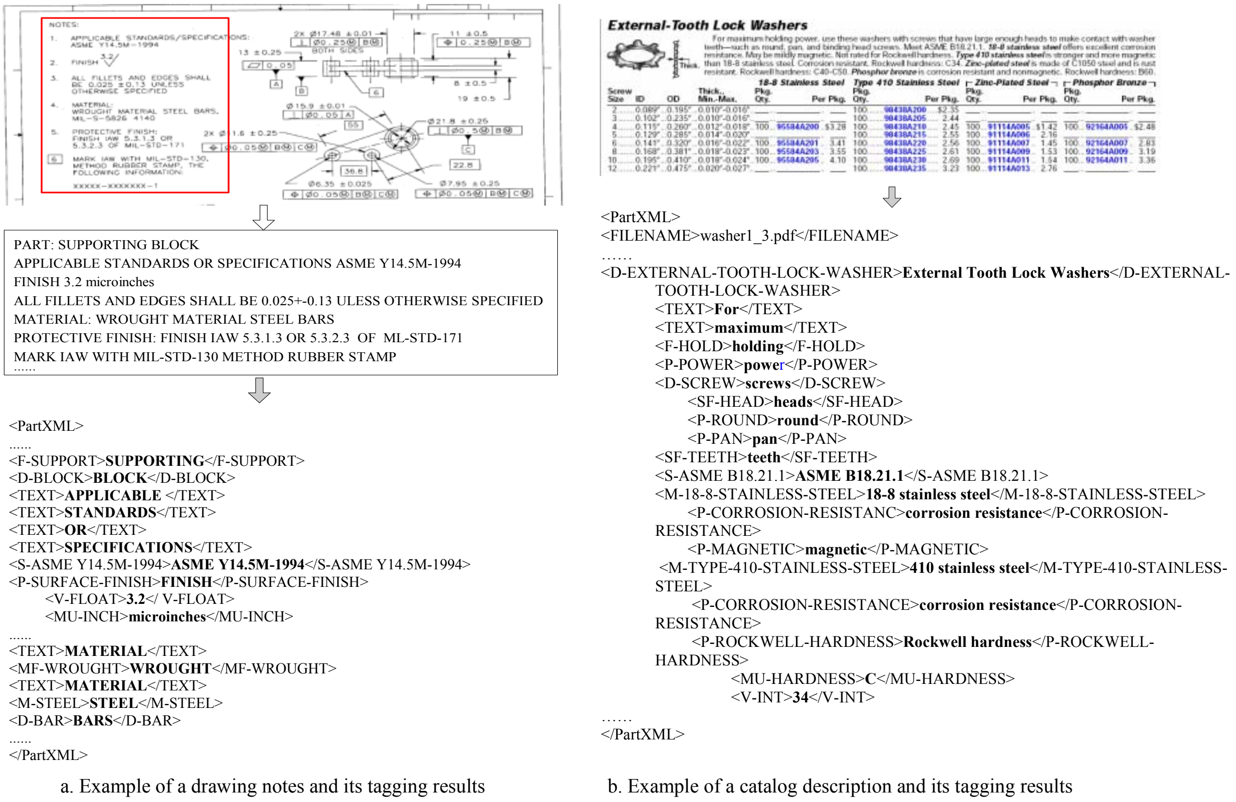
3.2 Concept matching: Assigning each word/phrase the concepts it refers to. This process takes two iterations. The first iteration is *full matching*, where lexical terms are retrieved in an orderly manner and matched against words in each sentence sequentially. Word(s) that fully match with a lexical term will be assigned the pertinent ontology concept. Note that multiple concepts may be assigned to a single word or a series of words (i.e., a phrase) because different concepts may have the same lexical term. The next iteration is *partial matching*, where each unrecognized word is matched against lexical terms sequentially. The concept will be assigned if the word matches part of its lexical term.

3.3 Numerical value recognition: The system recognizes numerical values such as 3.2 inch, HRC 55, and 32°F - 212°F. First, it recognizes a single numerical value by converting the cardinal number recognized in step 3.1 to a single numerical value if the number is adjacent to a measurement unit concept such as 'mm' (MU-millimeter), a property concept such as 'diameter' (P-DIAMETER), or certain symbols, such as '+/-.' Next, range values are recognized. Currently, the system recognizes five types of numerical values: integer, float (e.g., 3.2 and 1/2), percentage (e.g., 20%), ratio (e.g., 1:4), and tolerance (e.g., +/- 0.001).

4. Concept disambiguation: A word or phrase which matches multiple concepts causes ambiguities. There are two major types of ambiguities:

- Polysemy: for example, the word *cylinder* may refer to a shape feature concept, SF-CYLINDER, or a device concept, D-CYLINDER, because both concepts have the same lexical term *cylinder*.
- Ellipsis: for instance, the word *finish* may (partially) match the lexical term *surface finish*, which is associated with the property concept P-SURFACE-FINISH, and *protective finish*, which is associated with the manufacturing process concept, MF-COATING.

Ambiguities are resolved by referring to the contexts of the word/phrase that is ambiguous. The contexts of a word refer to the concepts its adjacent words/phrases are tagged. For example, if the untagged word *finish* is followed by a phrase tagged material concept, e.g., M-ZINC, then the word finish must be tagged MF-COATING. If the word is followed by a numerical value concept such as +/-0.001, it must be tagged P-SURFACE-FINISH, because this property concept is related to



NOTES:

1. APPLICABLE STANDARDS/SPECIFICATIONS: ASME Y14.5M-1994
2. FINISH 3.2
3. ALL FILLETS AND EDGES SHALL BE 0.025 ±0.13 UNLESS OTHERWISE SPECIFIED
4. MATERIAL: WROUGHT MATERIAL STEEL BARS, MIL-S-5826 4140
5. PROTECTIVE FINISH: FINISH IAW 5.3.1.3 OR 5.3.2.3 OF MIL-STD-171

6 MARK IAW WITH MIL-STD-130, METHOD RUBBER STAMP, THE FOLLOWING INFORMATION:

XXXXX-XXXXXXX-1

PART: SUPPORTING BLOCK
APPLICABLE STANDARDS OR SPECIFICATIONS ASME Y14.5M-1994
FINISH 3.2 microinches
ALL FILLETS AND EDGES SHALL BE 0.025+-0.13 ULESS OTHERWISE SPECIFIED
MATERIAL: WROUGHT MATERIAL STEEL BARS
PROTECTIVE FINISH: FINISH IAW 5.3.1.3 OR 5.3.2.3 OF ML-STD-171
MARK IAW WITH MIL-STD-130 METHOD RUBBER STAMP
......

<PartXML>
......
<F-SUPPORT>**SUPPORTING**</F-SUPPORT>
<D-BLOCK>**BLOCK**</D-BLOCK>
<TEXT>**APPLICABLE** </TEXT>
<TEXT>**STANDARDS**</TEXT>
<TEXT>**OR**</TEXT>
<TEXT>**SPECIFICATIONS**</TEXT>
<S-ASME Y14.5M-1994>**ASME Y14.5M-1994**</S-ASME Y14.5M-1994>
<P-SURFACE-FINISH>**FINISH**</P-SURFACE-FINISH>
<V-FLOAT>**3.2**</ V-FLOAT>
<MU-INCH>**microinches**</MU-INCH>
......
<TEXT>**MATERIAL**</TEXT>
<MF-WROUGHT>**WROUGHT**</MF-WROUGHT>
<TEXT>**MATERIAL**</TEXT>
<M-STEEL>**STEEL**</M-STEEL>
<D-BAR>**BARS**</D-BAR>
......
</PartXML>

External-Tooth Lock Washers

Thick.

For maximum holding power, use these washers with screws that have large enough heads to make contact with washer teeth—such as round, pan, and binding head screws. Meet ASME B18.21.1. ***18-8 stainless steel*** offers excellent corrosion resistance. May be mildly magnetic. Not rated for Rockwell hardness. ***Type 410 stainless steel*** is stronger and more magnetic than 18-8 stainless steel. Corrosion resistant. Rockwell hardness: C34. ***Zinc-plated steel*** is made of C1050 steel and is rust resistant. Rockwell hardness: C40-C50. ***Phosphor bronze*** is corrosion resistant and nonmagnetic. Rockwell hardness: B60.

				18-8 Stainless Steel			Type 410 Stainless Steel			Zinc-Plated Steel			Phosphor Bronze		
Screw Size	ID	OD	Thick., Min.-Max.	Pkg. Qty.		Per Pkg.	Pkg. Qty.		Per Pkg.	Pkg. Qty.		Per Pkg.	Pkg. Qty.		Per Pkg.
2	0.089"	0.195"	0.010"-0.016"	—	—	—	100	98438A200	$2.35	—	—	—	—	—	—
3	0.102"	0.235"	0.010"-0.016"	—	—	—	100	98438A205	2.44	—	—	—	—	—	—
4	0.115"	0.260"	0.012"-0.018"	100	95584A200	$3.28	100	98438A210	2.45	100	91114A005	$1.42	100	92164A005	$2.48
5	0.129"	0.285"	0.014"-0.020"	—	—	—	100	98438A215	2.55	100	91114A006	2.16	—	—	—
6	0.141"	0.320"	0.016"-0.022"	100	95584A201	3.41	100	98438A220	2.56	100	91114A007	1.45	100	92164A007	2.83
8	0.168"	0.381"	0.018"-0.023"	100	95584A203	3.55	100	98438A225	2.61	100	91114A009	1.53	100	92164A009	3.19
10	0.195"	0.410"	0.018"-0.024"	100	95584A205	4.10	100	98438A230	2.69	100	91114A011	1.54	100	92164A011	3.36
12	0.221"	0.475"	0.020"-0.027"	—	—	—	100	98438A235	3.23	100	91114A013	2.76	—	—	—

<PartXML>
<FILENAME>washer1_3.pdf</FILENAME>
……
<D-EXTERNAL-TOOTH-LOCK-WASHER>**External Tooth Lock Washers**</D-EXTERNAL-TOOTH-LOCK-WASHER>
<TEXT>**For**</TEXT>
<TEXT>**maximum**</TEXT>
<F-HOLD>**holding**</F-HOLD>
<P-POWER>**power**</P-POWER>
<D-SCREW>**screws**</D-SCREW>
<SF-HEAD>**heads**</SF-HEAD>
<P-ROUND>**round**</P-ROUND>
<P-PAN>**pan**</P-PAN>
<SF-TEETH>**teeth**</SF-TEETH>
<S-ASME B18.21.1>**ASME B18.21.1**</S-ASME B18.21.1>
<M-18-8-STAINLESS-STEEL>**18-8 stainless steel**</M-18-8-STAINLESS-STEEL>
<P-CORROSION-RESISTANC>**corrosion resistance**</P-CORROSION-RESISTANCE>
<P-MAGNETIC>**magnetic**</P-MAGNETIC>
<M-TYPE-410-STAINLESS-STEEL>**410 stainless steel**</M-TYPE-410-STAINLESS-STEEL>
<P-CORROSION-RESISTANCE>**corrosion resistance**</P-CORROSION-RESISTANCE>
<P-ROCKWELL-HARDNESS>**Rockwell hardness**</P-ROCKWELL-HARDNESS>
<MU-HARDNESS>**C**</MU-HARDNESS>
<V-INT>**34**</V-INT>
……
</PartXML>

a. Example of a drawing notes and its tagging results

b. Example of a catalog description and its tagging results

Figure 7. Examples of the document tagging results
Note that letters in bold are the words from the original document or PartText.
For the sake of clarity, 1) the title block in the drawing is not shown; 2) only parts of the tagged documents are illustrated; and 3) the PartText is ignored in b.

numerical value concepts as defined in the EO. See [29] for details about the concept disambiguation method.

5. PartXML generation: The processed partText is converted to PartXML, where each word/phrase is enclosed with its concept as tags. Figure 7 presents examples of a 2D drawing (notes) and component catalog descriptions before and after the tagging process. Note that the tag <TEXT> serves as a containment of words not semantically tagged. These tags are used for a repeated updating of the EO and EL because the words can easily be pulled out and analyzed.

6 CONCLUSION

A systematic, principled, and semi-automatic ontology development methodology has been described. It has several distinctive characteristics: 1) The acquisition method is based upon the ontological semantics theory which has been justified in various semantics-based applications; 2) The elicitations of the EO take into account both general EKRs independent of a particular company, EKRs specific to a company, and the information needs of engineers; 3) The knowledge worksheets further structure the acquisition process and enable an automatic ontology population; 4) The evaluations of the EO using the principled method and the empirical experiment; and 5) The overall development process is integrated with Protégé in order to utilize its wide range of plug-ins (e.g. ontology visualization and reasoning) and language formats such as XML and OWL (Web Ontology Language). In addition, though the EO is developed for information retrieval purpose in particular, it has a broader array of potential applications, such as semantics-based system interoperability by extending the EL, and knowledge reuse.

ACKNOWLEDGEMENTS

Our special thanks go to Professor Bill Peine, School of Mechanical Engineering, Purdue University, for his help in knowledge acquisition about the surgery robot design, and to Professor David C. Anderson, School of Mechanical Engineering, Purdue University, for his insightful suggestions. We would also like to thank Imaginestics LLC for their help in preparing the test data and for donating I-migrate. Chris Bence, In Chul Chang, and Kaushik Mantri helped in acquiring the engineering ontology. This research is supported by the 21st Century R&T funds, the National Science Foundation Partnership for Innovation Award, and University Faculty Scholar Award for Professor Ramani. We are grateful to the Purdue Center for Education and Research in Information Assurance and Security (CERIAS) for making its ontological semantics resources available to us and to its internal pilot grants, funded initially by an Eli Lilly Foundation grant, as well as to its external NSH-ITR and NSF CyberTrust grants on the extension of the resources to new domains.

REFERENCES

[1] McMahon C.A., Lowe, A., Culley, S.J., Corderoy, M., Crossland, R., Shah, T., and Stewart, D. Waypoint: An Integrated Search and Retrieval System for Engineering Documents. *ASME J. Computer and Information Science in Engineering (JCISE)*, 2004, 4(4), 329-338.

[2] Court, A.W., Ullman, D.G., and Culley, S.J. A Comparison between the Provision of Information to Engineering Designers in the UK and the USA. *Int. J. Information Management*, 1998, 18(6), 409-425.

[3] Ullman, D.G. *The Mechanical Design Process.* 2001 (New York: McGraw-Hill).

[4] Ahmed, S. and Wallace K.M. Identifying and Supporting the Knowledge Needs of Novice Designers within the Aerospace Industry. J. of Engineering Design, 2004, 15(5), 475-492.

[5] Sivaloganathan, S. *Engineering Design Conference 98: Design Reuse*, 1998 (ASME 98).

[6] Hertzum, M. and Pejtersen, A.M. The Information-seeking Practices of Engineers: Searching for Document as well as for People. *J. Information Processing and Management*, 2000, 36(5), 761-778.

[7] Salton, G. *Automatic Text Processing.* 1989 (Wokingham: Addison Wesley).

[8] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. Indexing by Latent Semantic Analysis. *J. American Society for Information Science*, 1990, 41(6), 391– 407.

[9] Ponte, J. and Croft, W. A Language Modeling Approach to Information Retrieval. *ACM SIGIR*, 1998.

[10] Robertson, S.E., Walker, S., and Hancock-Beaulieu, M. Okapi at TREC-7: Automatic Ad hoc Filtering, VLC and Interactive. *TREC-7*, 1999.

[11] Dong, A. and Agogino, A.M. Text Analysis for Constructing Design Representations. *J. Artificial Intelligence in Engineering*, 1996, 11, 65-75.

[12] Ahmed, S., Kim, S., and Wallace, K.M. A Methodology for Creating Ontologies for Engineering Design. *Proc. ASME / IDET&CIE Conf.*, Long Beach, CA, 2005.

[13] Yang, M.C., Wood, W.H. and Cutkosky, M.R. Design Information Retrieval: A Thesauri-based Approach for Reuse of Informal Design Information. *J. Engineering with Computers*, 2005, 21(2), 177-192.

[14] Uschold, M. and Grüninger, M. Ontologies and Semantics for Seamless Connectivity. *SIGMOD Record*, 2004, 33(4), 58-64.

[15] Nirenburg, S. and Raskin, V. *Ontological Semantics.* 2004 (Cambridge, MA: MIT Press).

[16] Schlenoff, C. Denno, E., Ivester, R., Libes, D., and Szykman, S, An Analysis and Approach to Using Existing Ontological Systems for Applications in Manufacturing. *J. Artificial Intelligence for Engineering Design, Analysis and Manufacturing* (*AIEDAM)*, 2000, 14, 257-270.

[17] Lenat, D.B. and Guha, R.V., *Building Large Knowledge-based Systems: Representation and Inference in the Cyc Project.* 1990 (Addison-Wesley, Boston).

[18] Gruber, T. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *Intl. J. of Human-Computer Studies*, 1995, 43(5/6), pp. 907-928.

[19] Uschold, M. and King, M. Towards a Methodology for Building Ontologies. *IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing*, 1996, Montreal.

[20] Grüninger, M. and Fox, M.S. Methodology for the Design and Evaluation of Ontologies. *Proc. Int'l Joint Conf. AI Workshop on Basic Ontological Issues in Knowledge Sharing,* 1995.

[21] Noy, N.F. and McGuinness, D.L. Ontology Development 101: A Guide to Creating Your First Ontology. *Technical Report, KSL-01-05 and SMI-2001-0800,* 2001, Stanford, Knowledge Systems Laboratory and Stanford Medical Informatics.

[22] Fernández-López, M., Gómez-Pérez, A., and Sierra, J.P. Building a Chemical Ontology Using Methonology and the Ontology Design Environment. *IEEE Intelligent Systems*, 1999, 14(1).

[23] Gómez-Pérez, A., Angele, J., Fernández-López, M., Christophides, V. Sure, Y. (ed.) *A Survey on Ontology Tools.* OntoWeb Deliverable, Universidad Politecnia de Mardrid, 2002.

[24] Eris, O., Hansen, P.H.K., Mabogunge, A., and Leifer, L. Toward a Pragmatic Ontology for Product Development Projects in Small Teams. *In Proc. Int'l Conf. on Engineering Design (ICED'99)*, 1999, Munich.

[25] Nanda, J. Simpson, T.W., Kumara, S.R.T., and Shooter, S.B. A Methodology for Product Family Ontology Development Using Formal Concept Analysis and Web Ontology Language. *JCISE*, 2006, 6, 1-11.

[26] Hwang, C. H. Incompletely and Imprecisely Speaking: Using Dynamic Ontologies for Representing and Retrieving Information. *Technical Report*, 1999, Austin: Microelectronics and Computer Technology Corp., TX.

[27] Shamsfard, M. and Barforoush, A.A. Learning Ontologies from Natural Language Texts. *Int. J.Human-Computer Studies*, 2004, 60, 17-63.

[28] Li, Z., Raskin, V., and Ramani, K. Developing Ontologies for Engineering Information Retrieval. *Proc. ASME 2007 IDETC/CIE Conf.*, Las Vegas, NE, September, 4-7 (accepted).

[29] Li, Z. and Ramani, K. Ontology-based design information extraction and retrieval. *AIEDAM*, 21(2), 2007.

[30] Kuffner, T.A. and Ullman, D.G. The Information Request of Mechanical Design Engineers. *Design Studies*, 1991, 12, 42-50.

[31] Baya, V, Gevins, J, Baudin, C, Mabogunje, A, Leifer, L., and Toye, G. An Experimental Study of Design Information Reuse. *Proc. 4th ASME/DTM Conf.*, Scottsdale, AZ, 1992, 42, 141-147.

[32] Pugh, S. *Total Design: Integrated Methods for Successful Product Engineering.* 1997 (Wokingham: Addison-Wesley).

[33] Lowe, A. McMahon, C. Shah, T., and Culley, S. An Analysis of the Content of Technical Information Used by Engineering Designers. *Proc. of ASME/DET Conf.*, 2000, Baltimore.

[34] Hirtz, J., Stone, R.B., McAdams, D.A., Szykman, S., and Wood, K.L. A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts. *Research in Engineering Design*, 2002, 13(2), 65–82.

[35] Kim, J., Will, P., Ling, S.R., & Neches, B. Knowledge-rich Catalog Services for Engineering Design. *AIEDAM*, 2003, 17 (4), 349-366.

[36] Triezenberg, K.E. *Ontology of Emotion.* PhD thesis, 2006, Purdue University.

[37] Rothbart, H.A. *Mechanical Design Handbook*, 1996 (McGraw-Hill, NY).

[38] Collins, J.A., Hagan, B.T., and Bratt, H.M. The Failure-Experience Matrix – A Useful Design Tool. *J. of Engineering for Industry*, 1976, August, 1074-1079.

[39] Kutz, M. *Handbook of Materials Selection.* 2002 (John Wiley & Sons, NY).

[40] Kutz, M. *Mechanical Engineers' Handbook, Manufacturing and Management.* 2005 (John Wiley & Sons, NY).

[41] Guarino, N. and Welty, C. Towards a Methodology for Ontology-based Model Engineering. I*n Proc. of the ECOOP-2000 Workshop on Model Engineering*, 2000.

Contact info:

Zhanjun Li
Purdue University
School of Mechanical Engineering
West Lafayette, IN 47907, USA
Phone: 01-765-4945653
Fax: 01-765-4940539
liz@purdue.edu
https://engineering.purdue.edu/precise

Professor Karthik Ramani
Purdue University,
School of Mechanical Engineering
West Lafayette, IN 47907, USA
Phone: 01-765-4945725
Fax: 01-765-4940539
ramani@purdue.edu
http://widget.ecn.purdue.edu/~ramani