Zhanjun Li
Graduate Research Assistant
Purdue Research and Education Center for Information Systems in Engineering (PRECISE), Purdue University, West Lafayette, IN 47907-2024;
School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907-2024
e-mail: liz@purdue.edu

Victor Raskin
Professor
Purdue Research and Education Center for Information Systems in Engineering (PRECISE), Purdue University, West Lafayette, IN 47907-2024;
Department of English and Linguistics, Purdue University, West Lafayette, IN 47907-2024
e-mail: vraskin@purdue.edu

Karthik Ramani[1]
Professor
Mem. ASME
Purdue Research and Education Center for Information Systems in Engineering (PRECISE), Purdue University, West Lafayette, IN 47907-2024;
School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907-2024;
School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-2024
e-mail: ramani@purdue.edu

Developing Engineering Ontology for Information Retrieval

When engineering content is created and applied during the product life cycle, it is often stored and forgotten. Since search remains word based, engineers do not have the effective means to harness and reuse past designs and experiences. Current information retrieval approaches based on statistical methods and keyword matching do not satisfy users' needs in the engineering domain. Therefore, we propose a new computational framework that includes an ontological basis and algorithms to retrieve unstructured engineering documents while handling complex queries. The results from the preliminary test demonstrate that our method outperforms the traditional keyword-based search with respect to the standard information retrieval measurement. [DOI: 10.1115/1.2830851]

1 Introduction

Engineering design is a decision making process in which the basic sciences, mathematics, and engineering sciences are applied to convert resources optimally in order to develop a product [1]. During this process, a large amount of knowledge is generated in order to describe the product and the process: Some of this knowledge is captured in the form of documents such as reports, notebooks, memos, emails, sketches, and 2D/3D computer-aided design (CAD) drawings, while other knowledge is retained in the memory of the engineers. Some important roles of the documentation include legal issues, patent applications, international standard certifications, internal practices, and sales catalogs. These engineering documents can be classified into internal resources and external resources. Internal resources include documents from product specifications and memos to final project reports and CAD drawings. External resources include online catalogs of suppliers and patents. A major portion of the engineering documents are (1) textual descriptions and (2) CAD drawings that have embedded drawing notes, bill of materials (BOMs), and texts, which describe shape and assembly information [2–4]. The number of digital documents being generated for product development has exploded. For example, there are approximately 40,000 documents produced in the design of a single engine in an aerospace company [2]. In Boeing, digital documents have accumulated up to the scale of petabytes, a number which is expected to double over the next two years [5].

Engineers are dependent on retrieving and using these documents in order to fulfill various engineering design tasks, such as the following.

(1) Acting as "memory extension" for individual engineers and enabling information sharing among them [6].
(2) Exploring design concept alternatives during the early stage of the development. This helps engineers avoid the tendency to take their first idea and start to refine it into a final design [6].
(3) Learning from the original design process and understanding the rationale behind the decisions made. This is especially important for novice engineers since they are not always aware of what they need to know during the design process [7].
(4) Searching for past designs when working on a similar product or problem in order to gain insight from past design scenarios and experiences. This is known as design reuse [8].

In fact, today's engineers simply do not make an effort to find engineering content beyond doing mere keyword searches [9]. However, current information retrieval approaches either retrieve

[1]Corresponding author.
Contributed by the Computer Engineering Informatics (EIX) Committee of ASME for publication in the JOURNAL OF COMPUTING INFORMATION SCIENCE AND ENGINEERING. Manuscript received December 5, 2006; final manuscript received August 13, 2007; published online February 14, 2008.

too much or irrelevant results for engineering or are not in a form that users can navigate and explore. It was reported that design engineers spent 20–30% of their time retrieving and communicating information [10]. 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 [11]. The redundant effort per employee is increasing and causing enormous cost as the complexity of enterprises and products increases. 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.

Most engineering documents are unstructured, in contrast to structured data resources such as database tables. Their formalities also vary. Formal documents, such as project proposals and reports, are written to comply with grammars or specific regulations; informal documents, such as engineers' notebooks and some textual descriptions in the drawings, on the other hand, are fragmentary descriptions. In contrast to general text documents, engineering documents are different because of the syntax variations and semantic complexities of their contents [3]. The syntax variations refer to the usage of abbreviations, e.g., SLA for stereolithography, acronyms, e.g., AL ALY 6061, which stands for a type of aluminum alloy, and synonyms, e.g., hardened way slides versus rectangular way slides, of the regular terms. They reflect the company-specific or domain-specific naming conventions, the diverse background of the authors, and the compositional nature of the designs. The semantics complexities denote the wide range of domain-specific issues and the relationship among these issues that must be considered and documented during the life cycle of product development. Examples of these issues are customer requirements, specifications, functions, performances, structure design, material selections, and manufacturing process selections. Therefore, it is necessary to consolidate and contextualize heterogeneous engineering documents in order to reconstruct the prior knowledge in a more explicit and structured manner.

This research addresses the task of retrieving unstructured engineering documents with textual descriptions or having texts associated in CAD drawings using an engineering ontology based approach. In general, an ontology can be used as a sophisticated indexing mechanism in order to structure an information repository such as unstructured documents, and specifically to achieve high precision and recall in text retrieval systems [12]. It entails adding semantic annotations to the documents themselves. Past research has been done to carry out comparisons between queries and documents via concept distance measures [13,14], and to enable query expansion with semantically related terms [15,16].

2 Related Work

2.1 Document Analysis. The analysis of general unstructured documents has been studied mainly by researchers from information retrieval (IR) and information extraction (IE). There is also research and development in the engineering domain though limited. Below, we explain these three areas.

Statistics-based methods and keyword-based input have been prevalent in IR research such as vector space (VS) model [17], latent semantic analysis [18], language modeling [19], probabilistic model [20], and many variants. They can be viewed as sophisticated stochastic techniques for matching terms from queries with terms in documents. These approaches have the advantage that minimal effort is required to adjust it into different domains and behaves reasonably effective, hence their wide adoptions in web search engines and other general IR applications. However, a common limitation of many retrieval models is that similarity scores are solely based on exact word matching. Words alone cannot capture the semantics or meanings of the document and query intent due to their ineffectiveness in understanding the context of domain-specific content. Following are the examples of queries through which users want to investigate the designs that

- lock a car door with a curvilinear slot sliding along a cylindrical pin in the assembly, and
- have a dc motor with an output speed of 100–1000 rpm.

The first query is for a desired function of a mechanism and its components. In the second example, the main concern is the exact value of a property attributed to a specific component. It is impossible to represent these semantic descriptions accurately by using a few keywords. Google's Brin and Page [21] pointed out that approaches using the VS model work well only with small and homogeneous collections such as literature or news releases under a common topic. However, recall that engineering documents usually describe various complex design processes and specifications, and are rich in specific technical terms and abbreviations, which end users are usually not familiar. Another issue in applying the current IR in the engineering domain is the ambiguities. The same term may represent different meanings in different contexts, or multiple terms may be used to mean the same thing [22]. To put it differently, the search results should satisfy the users, who are looking for something that matches their understanding of a 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.

IE approaches bring together natural language processing (NLP) tools with domain knowledge to extract meaningful sentence constituents from unstructured texts for retrieval or for knowledge mining purposes [23]. The domain knowledge can either be formalized as expression patterns by experts [24] or learned from a large training corpus [25]. However, the research in IE usually deals with short texts such as news of terrorism reports or extracts very specific information such as name entity recognition [23]. Engineering documents are more diversified. Therefore, it is very labor intensive to form expression patterns manually. It is also unfeasible to use the training approach because both writing style and terminology usage change over time, tasks, and departments/companies.

In the engineering domain, there has been very limited research aimed at analyzing unstructured engineering documents for retrieval purposes. Most of them have been based on IR or IE approaches. Farley [26] extracted the equipment and the repair action on them from aircraft maintenance logbooks for case-based retrieval. This method was based on the IE approach and used existent domain vocabularies. Dong and Agogino [27] proposed to use VS model and belief networks to represent design manuals. Ahmed et al. [22] developed taxonomies in order to index corporate documents. The VS model was also used to classify the documents against the terms in the taxonomies. Yang et al. [28] attempted to automate the population of a thesaurus from notebooks by using the latent semantic analysis. The same method was also applied by Song et al. [29] to improve the search performance of a digital library of engineering education resources. McMahon et al. [9] employed a predefined taxonomy to classify documents by rule-based matching. Their method supports keyword searching and browsing.

Commercial development in engineering such as product data management (PDM)/product lifecycle management (PLM) systems relies on manual processes to upload the metadata (e.g., file names and date of creation) as well as the content (e.g., part names, material used, and property-value pairs) of the documents into databases. In recent commercial software such as ENOVIA MATRIXONE[2] and Autonomy Co. [30] statistical IR approaches for document retrieval have been used. However, there are often too few or too many results and these approaches are not used fre-

[2]www.matrixone.com/.

quently in practice [9,31].

In summary, current approaches (1) do not attempt to provide a semantics-based representation of engineering documents or provide for engineers' information needs; (2) do not provide a feasible and scalable computational framework in order to retrieve heterogeneous engineering documents; (3) do not deal with the relationships among various engineering taxonomy classes and the syntactic and semantic ambiguities, which are prevalent in engineering documents and user queries; and (4) do not utilize engineering knowledge in the organization of the search results in order to accelerate the information seeking process.

2.2 Structured Representation. Different from analyzing and retrieving unstructured engineering documents to assist the design process, the structured and semantics-based representation of designs has been studied in fields such as product modeling and ontology modeling. In product modeling, such as Refs. [32,33], design and development information are recorded by engineers through complying with formalized templates and rules. Ontology modeling, e.g., Refs. [34,35], systemizes semantic relations between elements of specific designs such as pumps and motors, as well as represents the functions and behaviors of design decompositions. Research in product modeling and ontology modeling has made significant progress in establishing complex models as well as in standardizing terminologies to describe the details of the design. In many cases, however, establishing the knowledge sharing agreements or mapping out the design decomposition is potentially less feasible [12]. Therefore, in our opinion, it is equally important to develop a strategy that is comprehensive and effective at retrieving valuable content about the design and design process from unstructured documents. Meanwhile, this strategy should cause less cognitive burden on engineers in generating and maintaining the model that understands the engineering context.

We propose a new, content-oriented knowledge and meaning based computational framework to form the ontological basis of the search, browsing, and learning tasks in the engineering domain. The cumulative domain knowledge is formalized. This knowledge must be brought to bear in developing an industry-wide, constantly upgradeable ontology for engineers, and is formulated in a single standard format. The framework intends to

- develop an engineering ontology (EO) and its associated engineering lexicon (EL) in design and manufacturing and use them to index the documents and interpret users' queries at the concept level,
- extract a structured and semantics-based representation from the unstructured documents based on the knowledge conceptualized in the EO,
- overcome the difficulties of regular search engines in not understanding the engineering context of a query, and
- design a novel user interface, which reflects the semantics-based representation of the documents and the EO interpretation with respect to users' queries. The purpose is to further improve the information seeking effectiveness. This is especially important for novice engineers since they require more support in identifying *what they need to know*, instead of just what they want to know [7].

3 Overview of the Approach

Figure 1 shows the overall architecture of interactions between the ontological basis, i.e., the domain knowledge source, and other functional modules applied to engineering information retrieval. It comprises six portions: preprocessing, knowledge source, knowledge source acquisition and maintenance, concept tagging, concept indexing, and document retrieval, i.e., query processing. The centerpiece of the architecture is the EO and EL models.

1. Preprocessing: The task of preprocessing 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, drawings, technical reports, and notebooks.
2. Knowledge source: It provides domain knowledge and lexical knowledge, i.e., the EO and its associated EL, respectively. They are used to assist in recognizing and indexing technical terms at the concept level and to understand user queries.
3. Knowledge source acquisition and maintenance: PROTÉGÉ 3.1[3] is used to build and update the knowledge source. 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 unified format are transformed into an XML and concept-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 index file is generated to index the XML documents. The file names 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 query accurately, and therefore improves retrieval performance. Ontology-based query processing algorithms are developed to fulfill this task.

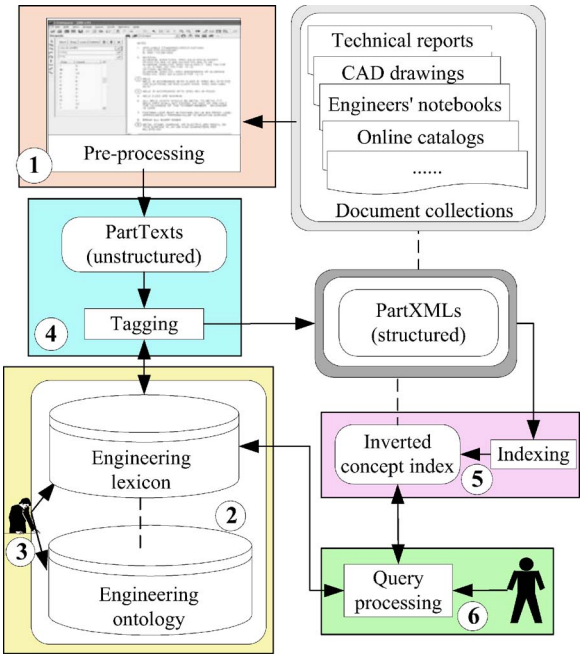


Fig. 1 System architecture and functional modules

4 Proposed Approach

4.1 Ontology Definition. An ontology is a constructed model of reality. 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 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

[3]http://protege.stanford.edu.

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 subdomains: 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*. This is similar to the predicate rule representation used in artificial intelligence research.

Ontologies share the inheritance feature with the object-oriented programming languages, which are indeed suitable for implementing ontological procedures. However, the object-oriented 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 in this application, independent of the choice of format. Currently, there is also confusion between taxonomy-based 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 subdomain. These concepts are connected only by domain-independent, i.e., taxonomic, relationships such as is-a. An ontology, however, consists of several taxonomies, along with multiple domain-specific, i.e., non-taxonomic, relationships to connect concepts across taxonomies. See Ref. [12] for comparisons between ontologies and database schema, as well as those between ontologies and knowledge representation; See Ref. [36] for an extended view of what a full-fledged ontology must be and how to bring it about.

The recently proposed ontology development in engineering can be categorized based on its intended usages, such as knowledge sharing [34,37,38], CAD interoperability (examples are STEP API 224 and [39,40]), design analysis and simulation [41–43], and product design and configuration [44,45]. Ahmed et al. [22] intended to design an ontology development process, which can be customized for a particular manufacturing company. However, their process does not explicitly explore the domain-specific relationships between concepts. Therefore, their acquisition result is a list of independent taxonomies, not an ontology.

Although significant progress has been made in ontology development in engineering, very little effort has been made to systemize the established knowledge in design and manufacturing by developing the correspondent ontological representation. No attempt has been made to formalize the associated lexical knowledge in order to bridge the concept-based representation of the ontology and the word-based representation of documents and queries. The proposed EO and its associated EL formalize engineering domain knowledge as well as general lexical knowledge in order to achieve more effective engineering IR. Examples of the domain knowledge are the classification of mechanical elements and their function, design, and manufacturing knowledge. The EO employs aforementioned concept definition to represent each class within a classification/taxonomy, formulates the hierarchy between classes in the same taxonomy by using an is-a relationship, and constructs other domain-specific correlations among concepts either in the same taxonomy or across taxonomies through relationships such as has-part (D-MOTOR, D-ROTOR) and has-function (D-MOTOR, F-ROTATE). The proposed EO development also distinguishes itself by incorporating semiautomatic tools into the practical acquisition process.

4.2 Developing Engineering Ontology and Engineering Lexicon. To build the EO, we specialize the ontological semantics method proposed by Nirenburg and Raskin [36] in developing large scale ontologies for machine translation. This method is similar to the ontology acquisition methods such as methontology [46] in that it uses a handcrafted acquisition process guided by ontological considerations. However, our method leverages the manual acquisition process by employing computer-assisted tools.

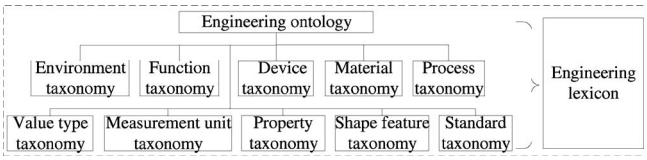


Fig. 2 The schema of the knowledge source

The first step is to identify the scope or themes of the EO. These themes are determined based on the discoveries by cognitive studies in the engineering domain, such as Refs. [3,22,47–49]. Prior studies investigated what types of information are requested by engineers as well as what domain-specific issues are documented during the product design process. The results of these studies are categorized and used to determine the themes of the EO. They include designed devices (products and components), functionalities and properties of the devices, common geometry and assembly information used in modeling the devices, the material selections and (design and manufacturing) processes applied in designing and making the devices, environmental objects which may interact with the devices in their working status, and the standards or specifications that certain design or manufacturing comply with. Measurement unit and value types are, in general, related to how device properties are described in the document. The overall schema of the EO is shown in Fig. 2.

In the second step, taxonomies under these themes are constructed. Concepts of each taxonomy are acquired from various engineering knowledge resources. These concepts are used in tagging (Sec. 4.3) and query processing (Sec. 4.4).

Then, (inter) relationships are formed between concepts across taxonomies. For example, has-material (D-PLAIN-WASHER, M-STAINLESS-STEEL): where D-PLAIN-WASHER represents a device concept (in device taxonomy), M-STAINLESS-STEEL is a material concept (in material taxonomy), and has-material is a relationship which associates a device concept with a material concept. The prefix in each concept represents the taxonomy which this concept belongs to. Table 1 lists more details of the EO and the acquisition resources. Definitions of the relationships are given in Table 2. The acquisition of relationships between concepts is important for query processing, as discussed in Sec. 4.4.

Note that the device taxonomy includes classifications of engineering catalog components and proprietary products. The latter needs to be customized for each specific company including product line classifications, subassembly classifications, and part inventory classifications, usually by referring to the BOMs or by product dissection [44]. The properties of the device concepts are conceptualized in the property taxonomy and connected with the device concepts through the has-property relationship.

The resultant EO is organized in a directed graph: Each node represents a concept; each arc represents a relationship. A portion of the EO is shown in Fig. 3.

The last step is to acquire the EL, which is a list of lexical terms in descendant order. They are used to match with word in documents or queries. Each lexical term is the actual word/phrase representation of the corresponding concept in the EO. Morphology forms, abbreviations, acronyms, and synonyms of the word/phrase are also lexical terms and share the same concept as the original lexical term. For example, move and moving are lexical terms of the functional concept *F-MOVE*.

The EO and EL lend themselves easily to an expansion, such as the addition of a new relationship or new concept. Then, the

Table 1 The EO concepts and acquisition resources

Taxonomies		No. 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	[51][a]
	Proprietary product	N/A	N/A	BOMs, product dissection	N/A
Function		246	F-SUPPORT, F-LOCK	Existing taxonomies	[52,53]
Material		1017	M-STAINLESS-STEEL, M-2008-T4 AL	Engineering texts, handbooks, online catalogs	[54][b]
Process		252	R-CASTING, R-DESIGN-REVISION	Engineering texts, handbooks, company regulations	[54,55]
Property		378	P-SHAFT-DIAMETER, P-DUCTILITY	Same as device taxonomy	Same as Device taxonomy
Measurement unit		64	MU-MILLIMETER, MU-FT-LB/SECOND	Online resources	[c]
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	[49][d]
Standard		31	S-MIL-STD-130	Standard libraries	[e]
Value type		8	V-FLOAT (numerical), V-HIGH (symbolic)	Engineering common sense; Online catalogs	N/A

[a]www.globalspec.com.
[b]www.matweb.com.
[c]www.ex.ac.uk/cimt/dictunit/dictunit.htm.
[d]WordNet2.1.
[e]www.nssn.org.

whole system can be updated automatically. 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.

Currently, there are 10 taxonomies, 2629 concepts and 13 types of relationships in the EO, and more than 10,000 lexical terms in the EL. We developed formatted worksheets as templates to (1) direct the acquisition of the EO and EL, and (2) improve the efficiency of the acquisition process (in the process of full deployment, the ontological semantic toolbox [36] will be utilized). These worksheets enable automatic uploading of the acquired data into the PROTÉGÉ editor. They have been used extensively by the undergraduate students who have design and manufacturing expe-

Table 2 Definitions of the relationships

Relationship	Concept[a]	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 a 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 and interact-with	DC	DC EC	Complement the has-function relationship when there is an "object" in the function description of "subject+verb [+objects]." Together, they represent the interactions between a 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	RC	Describes the type of process used in designing and manufacturing the DC	has-process (D-GEAR, R-HOBBING)
use-material	RC	MC	Describes the type of possible raw materials that certain manufacturing processes act on	user-material (R-COATING, M-NONFERROUS-METAL)
has-property	DC/MC	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-DIAMETER, MU-MM)
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/RC	SC	Specifies the standard a DC/MC/RC may comply with	has-standard (D-WASHER, S-ASME B18.13)

[a]DC: device concept; FC: function concept; EC: environment concept; MC: material concept; RC: process concept; SFC: shape feature concept; SC: standard concept; PC: property concept, MUC: measurement unit concept; VC: value type concept.

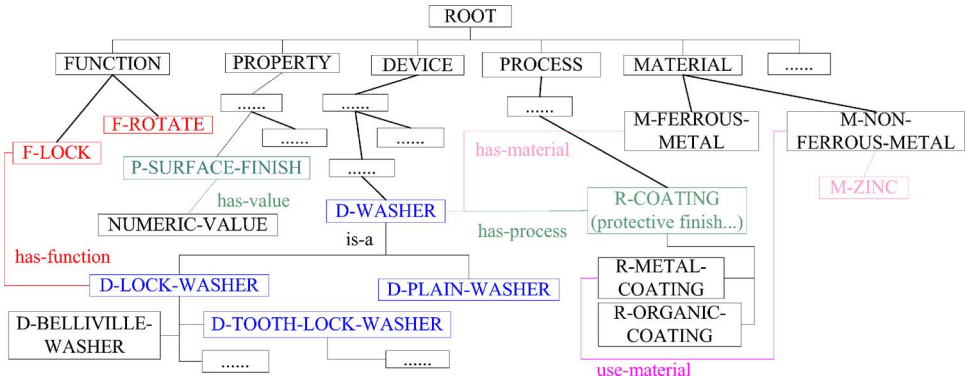


Fig. 3 A portion of the EO

rience and conduct the acquisition task in our group. The total amount of time spent for acquisition is about 500 h.

Now, the question is how complete are the EO and EL? Since both the lower-level concepts and more upper-level concepts and their lexical terms are manually acquired from a wide range of engineering resources, we believe that the EO and EL cover the selected scope reasonably well. In addition, we conducted experiments in order to estimate the coverage [50]. In these experiments, we selected concepts from the tagged documents, which include suppliers' catalogs, CAD drawings, and technical reports. We observed that more than 90% of the documents were associated with the concepts of the EO, while less than 10% of the documents failed to associate with any concepts of the EO due to its incompleteness.

4.3 Concept Tagging and Indexing. In order to represent unstructured engineering documents by using concepts in the EO, we first convert documents from various resources into unstructured .txt files, i.e., *PartTexts*, during preprocessing in Fig. 1. Note that .txt format is the only requirement for the input document to be processed by our prototype. We use XPDF,[4] which converts the PDF documents (e.g., catalogs) into a congruent stream of plain text while maintaining certain layouts of the documents. Engineering symbols such as ″ and Ø are replaced by their textual descriptions. Texts in CAD drawings are extracted by using I-PRAWLER,[5] a software program that uses various CAD application program interfaces (APIs), such as the APIs of SOLIDWORKS and AUTOCAD. It converts the textual descriptions such as drawing notes and title blocks (in 2D drawings) as well as shape features and mating relations (in 3D drawings) into .txt files.

Our method makes use of the EO and EL to recognize concepts contained in the documents. By doing so, the PartTexts are converted into a concept-based and XML representation, i.e., *PartXMLs*, where each recognized word/phrase is tagged by the corresponding concept in the EO. Then, the concepts in all PartXMLs are indexed.

In contrast to IE approaches and more recent ontology-based IE approaches, e.g., Refs. [56,57], our method is less dependent on NLP techniques. It has no syntax analysis and no phrase chunking and therefore is more robust in analyzing both formal and informal documents.

Figure 4 shows the modules and process of concept tagging and indexing.

1. Tokenization: The input character streams of a PartText are parsed into tokens and punctuation marks.
2. Sentence segmentation: Sentences are formed by using punctuation marks and symbols such as "\n."
3. Concept recognition:
   (a) Cardinal number recognition: The cardinal numbers such as 3.2, 1:20, and 200 are identified.
   (b) 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 with part of its lexical term.
   (c) Numerical value recognition: The system recognizes the numerical values such as 3.2 mm, HRC 55, and 32–212 F. First, it recognizes a single numerical value by converting the cardinal number recognized in Step (a) to 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

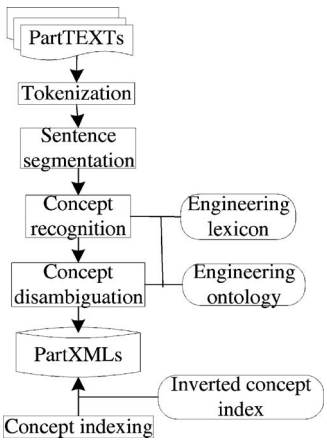
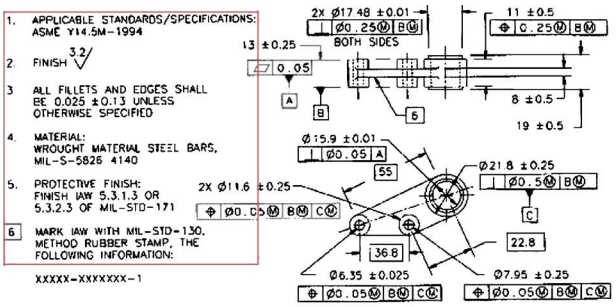
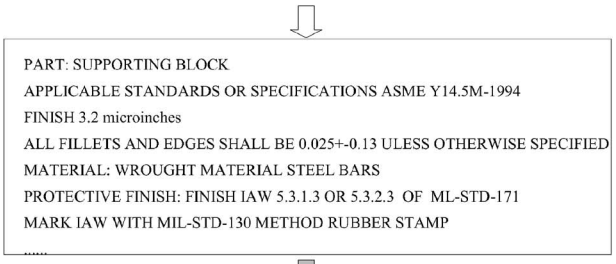


Fig. 4 Modules and process of concept tagging and indexing

[4]www.foolabs.com/xpdf/.
[5]www.imaginestics.com.



(*a*)



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

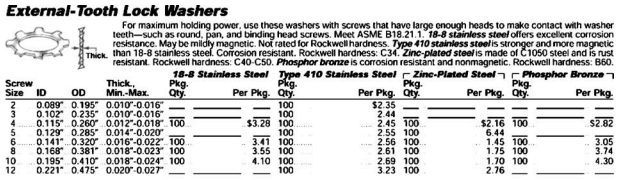
PROTECTIVE FINISH: FINISH IAW 5.3.1.3 OR 5.3.2.3 OF ML-STD-171

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>
```

(*b*)



External-Tooth Lock Washers

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.

Screw Size	ID	OD	Thick., Min.–Max.	18-8 Stainless Steel Pkg. Qty.	Per Pkg.	Type 410 Stainless Steel Pkg. Qty.	Per Pkg.	Zinc-Plated Steel Pkg. Qty.	Per Pkg.	Phosphor Bronze Pkg. Qty.	Per Pkg.
2	0.089"	0.195"	0.010"-0.016"	—	—	100	$2.35	—	—	—	—
3	0.102"	0.235"	0.010"-0.016"	—	—	100	2.44	—	—	—	—
4	0.115"	0.260"	0.012"-0.018"	100	$3.28	100	2.45	100	$2.16	100	$2.82
5	0.129"	0.285"	0.014"-0.020"	—	—	100	2.55	100	6.44	—	—
6	0.141"	0.320"	0.016"-0.022"	100	3.41	100	2.56	100	1.45	100	3.05
8	0.168"	0.381"	0.018"-0.023"	100	3.55	100	2.61	100	1.75	100	3.74
10	0.195"	0.410"	0.018"-0.024"	100	4.10	100	2.69	100	1.70	100	4.30
12	0.221"	0.475"	0.020"-0.027"	—	—	100	3.23	100	2.76	—	—

(*c*)

```
<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>
```

(*d*)

Fig. 5 Examples of the document tagging results: (*a*),(*b*) example of drawing notes; (*c*),(*d*) example of catalog descriptions. *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*).

with 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 with the lexical term *surface finish*, which is associated with the property concept P-SURFACE-FINISH, and *protective finish*, being associated with the manufacturing process concept, R-COATING.

Ambiguities are resolved by referring to the contexts of the word/phrase that is ambiguous. The context of a word refers to the concepts to which its adjacent words/phrases are tagged. For example, if the untagged word finish is followed by a phrase tagged as material concept, e.g., M-ZINC, then the word finish must be tagged as R-COATING. If the word is followed by a numerical value concept such as +/−0.002 (in drawing notes), it must be tagged as P-SURFACE-FINISH because this property concept is related to numerical value concepts as defined in the EO. More details about the concept disambiguation method are given in the next section.

5. PartXML generation: The processed partText is converted to PartXML, where each word/phrase is enclosed with its concept as tags. Figure 5 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 be easily pulled out and analyzed.
6. Concept-based indexing: In order to rank the relevancy of documents in query processing, we propose a data structure called *inverted concept index* (ICI) to index the concept (tag) and the PartXMLs. ICI is a variation of the inverted index. It lists each concept as well as the PartXML file name and the locations where the concepts are present.

4.4 Engineering Ontology Based Query Processing. Users' queries are assumed to be a list of words that may include property-value expressions, such as "linear slide surface finish <0.76 mm." Tokens, i.e., keywords, are generated from a query after tokenization and removal of stop words. Relational operators are recognized, and correspondent routines will be called when

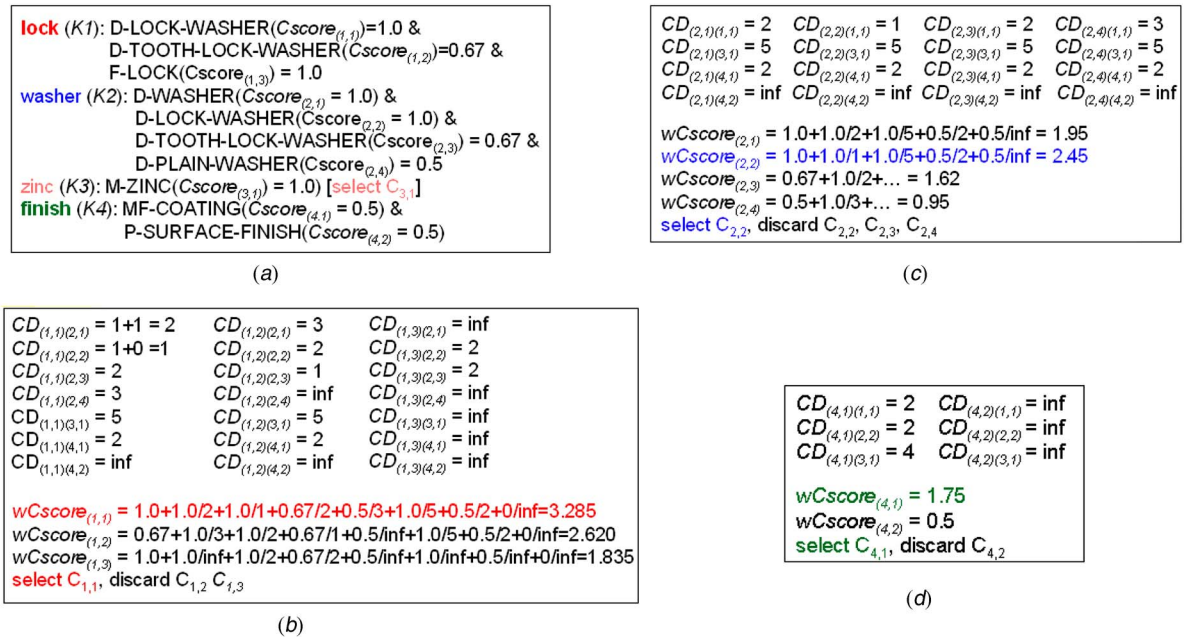


Fig. 6 Concept disambiguation process for the query "lock washer with zinc finish:" (*a*) Cscores of the matched concepts for the query keywords, (*b*) CDs and wCscores for the ambiguous concepts of keyword "lock," (*c*) CDs and wCscores for the ambiguous concepts of keyword "washer," and (*d*) CDs and wCscores for the ambiguous concepts of keyword "finish"

the system ranks the documents. The rest of the keywords are matched with the concepts in the EO through the EL.

4.4.1 Concept Disambiguation. Lexical ambiguity arises when a single keyword matches with multiple concepts. This includes the types of ambiguities described in the previous section, i.e., polysemy and ellipsis. Given a query "lock washer with zinc finish," the keywords "lock," "washer," and "finish" cause lexical ambiguities. For example, lock matches with the device concepts of D-LOCK-WASHER and D-TOOTH-LOCK-WASHER, and the function concept of F-LOCK; washer matches with D-WASHER, D-LOCK-WASHER, D-TOOTH-LOCK-WASHER, and D-PLAIN-WASHER; and finish matches with P-SURFACE-FINISH and R-COATING. We adapt the concept disambiguation metric proposed in Ref. [13]. It calculates the correlations between the matched concepts of all keywords in order to determine which of the ambiguous concepts should be retained. For a set of concepts, a concept is highly correlated with others if it is (1) less far away from them in the EO, i.e., semantically closer, and (2) has more words matching with the particular query keyword, i.e., lexically closer. The disambiguation metric is

Matched concepts of query keyword K_i:$C_{i,1}C_{i,2,\ldots},C_{i,h}$

Matched concepts of other keywords

$C_{1,1}C_{1,2,\ldots},C_{i-1,1}C_{i+1,1}C_{i+1,2,\ldots},C_{s,t}$

Lexical terms of $C_{i,h}$:$T_{i,h,1}T_{i,h,2,\ldots},T_{i,h,j}$

$$\text{Tscore}_{i,h,j} = \frac{\text{Number of keywords in query } T_{i,h,j} \text{ matches with}}{\text{Number of words in } T_{i,h,j}} \tag{1}$$

$$\text{Cscore}_{i,h} = \max(\text{Tscore}_{i,h,j}) \tag{2}$$

$$\text{CD}_{(i,h)(s,t)} = 1 + \min(\#\text{ of arcs}(C_{i,h}, C_{s,t})) \tag{3}$$

$$\text{wCscore}_{i,h} = \text{Cscore}_{i,h} + \sum_{s,t=1,1} \frac{\text{Cscore}_{s,t}}{\text{CD}_{(i,h)(s,t)}} \tag{4}$$

The calculations in disambiguating the given query example are described in Fig. 6. Figure 3 also illustrates the matched concepts. The formal definitions of the metric are the following.

Equation (1). Term score (Tscore): The score of a lexical term that matches with a keyword is the total number of keywords in the query the lexical term matches divided by the total number of words in the lexical term.

Equation (2). Concept score (Cscore): This is the maximum Tscore among its lexical terms. For the given example, P-SURFACE-FINISH has a Cscore of 0.5 because its lexical term is "surface finish." It consists of two words and matches with finish in the query. Similarly, R-COATING has a Cscore of 0.5, D-LOCK-WASHER and F-LOCK have 1.0, D-TOOTH-LOCK-WASHER has 0.67, and D-PLAIN-WASHER has 0.5.

Equation (3). Concept distance (CD): The shortest distance or least number of arcs between two matched concepts in the EO which belong to distinct keywords. For instance, CD (D-LOCK-WASHER, D-WASHER) =2 in the given query. CD is 1 if the matched concepts are coincident. It is infinite if there is no directed path between the two concepts. Note that CD is symmetric, i.e., $\text{CD}_{i,j}=\text{CD}_{j,i}$.

Equation (4). Weighted concept score (wCscore): This relates to the Cscores of all its correlated concepts but inversely relates to the CDs with them.

Finally, wCscores are compared among the matched concepts of a keyword: Only the concept with the maximum wCscore is selected. This metric is applied to all the matched and ambiguous concepts, keyword by keyword. Finally, all of the selected concepts are added into a list in descending order with respect to their wCscores. The selected concepts are further expanded to add their relevant concepts such as child concepts to the list. The newly added concepts share the same wCscore with their parent concepts. This is called query expansion, which might increase the number of relevant documents to be retrieved.

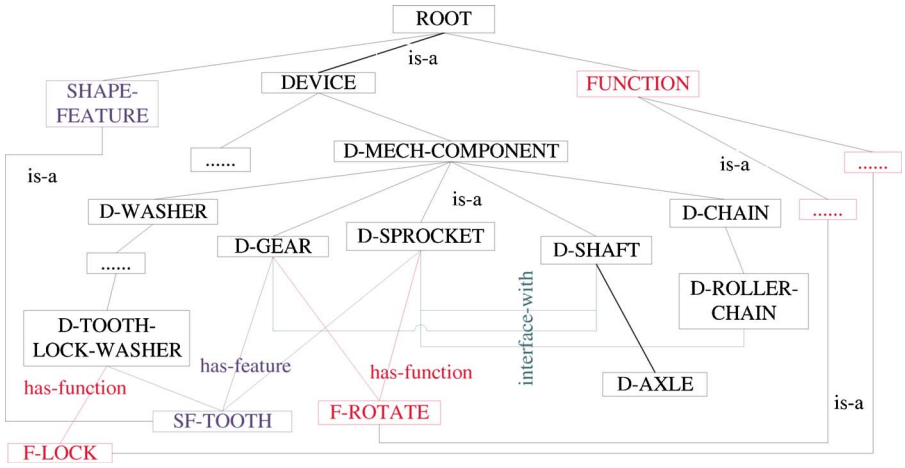


Fig. 7 A portion of the EO in concept abstraction

One issue of this metric is the computational cost related to the ontology traversal when computing the CDs. To ameliorate the problem, we propose a *pairwise concept distance lookup table* (PCDT) that stores any pair of concepts and their CD. PCDT is precomputed (i.e., when the EO is uploaded into the memory) by implementing the Floyd–Warshall algorithm. Therefore, the online computing cost is transferred to offline processing.

4.4.2 Concept Abstraction. Semantic ambiguity occurs in the queries when users may not know the exact words of the design or the related issues they want to find though they may have some contextual clues, such as the functionality of the design and other interacting parts/components of the querying product/component. One example is when users want to find "a component which rotates, has teeth, and connects to a chain and shaft." The system should return a sprocket even though the word "sprocket" does not appear in the query. However, in traditional IR methods, because the query is treated as a list of independent words when searching for relevant documents, the meaning of the query is lost. Therefore, the retrieval performance is hurt. In the EO-based method, the engineering contexts of the query are recovered at the system representation by using the proposed concept abstraction metric. The metric takes advantage of the structure and content of the EO to ferret out the true meaning, i.e., the target concept(s), of the query.

Given a user's query "rotates teeth chain shaft," the query keywords are first processed by Eqs. (1) and (2) in concept disambiguation, i.e., only the Cscores of the matched concepts are calculated. Next, all of the matched concepts are added to a list. Query expansion is executed on the matched concepts. The newly added concepts share the same Cscore with their parents. All these concepts are called Probe concepts. For the given example, assuming the Probe concepts (and their scores) are SF-TOOTH (1.0), D-SHAFT (1.0), D-AXLE (1.0), D-CHAIN (1.0), D-ROLLER-CHAIN (0.5), and F-ROTATE (1.0), the system traverses the EO by breadth first search (BFS) to collect the concepts (Cconcept). Each of these Cconcepts must have at least one of the Probe concepts as the filler concepts in one of its relationships, except for is-a. The wCscore of this Cconcept is then obtained by adding up the Cscores of all its filler concepts, which are also Probe concepts. Figure 7 is part of the EO which illustrates the concept abstraction process. For example, the wCscores are 1.0 for D-TOOTH-LOCK-WASHER, 3.0 for D-GEAR, and 4.0 for D-SPROCKET. The average wCscore is then set as threshold and only the Cconcepts with above average wCscores are selected. They are added into the selected concept list in descendent order. In this example, the average wCscore is 2.67. Therefore, D-SPROCKET and D-GEAR are selected.

4.4.3 Ranking and Property-Value Search. In general, the relevance of documents with respect to a query is ranked by the similarity score between the selected query concepts and the document. The score is calculated by using Boolean model [58] and the ICI.

Different from regular users' requests, engineers often need to search for designs that satisfy some specifications or property-value pairs, including quantitative/numerical values such as washer with inside diameter (I.D.)=2.0 cm and qualitative/symbolic values, e.g., motors with *high* load capacity.

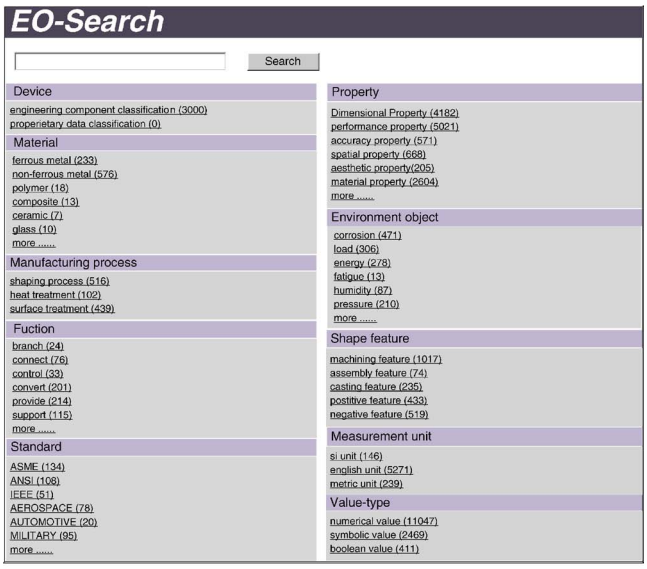
In order to enable query by numerical property-value pairs, the system first understands the meaning of relational operators such as "> (greater than)," "< (less than)," and "= (equal to)." Second, we identify the numerical value in the query such as 2.0, which is assumed to be associated with the selected property concept such as P-INSIDE-DIAMETER and the measurement unit concept, e.g., MU-MILLIMETER. Last, recall that ICI represents the distance among concepts in the PartXML by recording their locations; therefore, in order to rank documents with respect to queries having numerical values, the difference between the numerical value in the query and the correspondent numerical values in the PartXMLs is taken into account in the final ranking. Corresponding routines of the relational operators are developed to (1) find the PartXMLs that have the numerical value concepts, which are comparable to the one in the query and are adjacent to the same type of property concept (and the measurement value concept), and (2) rerank these PartXMLs after the Boolean ranking.

Note that both numerical and symbolic values are recognized as a value-type concept. The value-type taxonomy includes numerical value concepts such as V-INT and V-FLOAT and symbolic value concepts such as V-HIGH, V-MEDIUM, and V-LOW. To quantify the symbolic property values, the system first finds the PartXMLs that have numerical value concepts adjacent to the same type of property concept as in the query, then orders the PartXMLs into three sets according to the values of these numerical value concepts from lowest to highest.

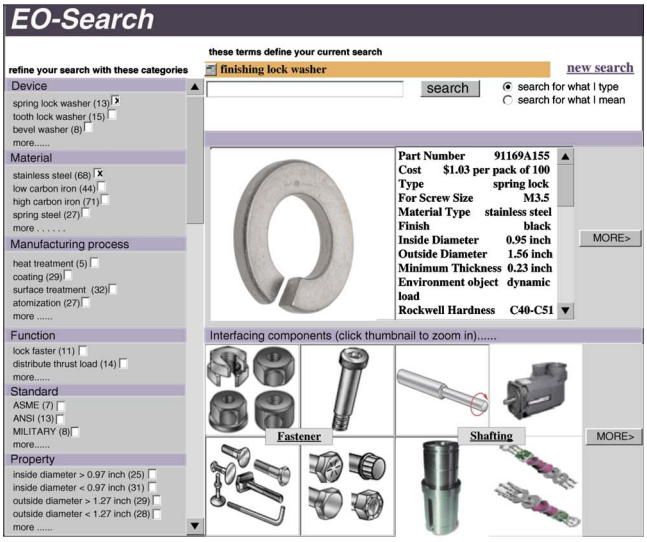
4.4.4 Context-Aware Dynamic Search Interface. Studies by [59] indicate that engineers who look for information want to browse and search documents by category. Teevan et al. [60] discovered the "orienteering" behavior in information seeking activities. This represents an evolving information-finding process where users interact with the search system and use the intermediate results to explore unexpected search directories or refine the search path to finally arrive at the target. It has been noticed that such an information seeking process is preferred to the directed

keyword search. However, to our limited knowledge, there is no search system that allows users to navigate engineering documents by dynamically generated *design orienteering previews*. Most of them return a flat list of relevant documents, or use predefined classification schemas as visible indices of the document collections. Design orienteering, on the other hand, is to dynamically update and organize the retrieval results based on the domain contexts in the EO that matches the query and the content of the relevant documents. Compared to the user interface proposed in Ref. [9], our approach (1) utilizes the specific inference structure of the EO corresponding to a query in order to illustrate the query intent; (2) employs the concept tagging and concept-based query analysis to categorize the relevant documents. For example, it illustrates how many documents are relevant to certain concepts as well as certain property-value ranges calculated on the fly; and (3) dynamically updates the matching inference structure and content of EO, the categories, as well as the returned documents according to users' subsequent interactions. This helps direct the user to the most relevant documents quickly.

We develop a search interface that allows users to navigate according to the query-dependent domain contexts or the concept categories in the EO. Figure 8(*a*) shows the first interface for navigating engineering catalog collections. It includes a text search box, the first level EO concepts, and the number of document that have the specific concepts. For each query, by either typing keywords or selecting concept categories, the system (1) returns a list of ranked documents, and (2) categorizes them based on the concept categories, which are present in the returned PartXMLs. Figure 8(*b*) shows the results categorized on the interface. The left panel lists the concept categories that apply to the returned documents. The search results are displayed on the right. The current query and the search box are shown on the top. For instance, suppose a query is "finishing lock washers with i.d. >1.25 cm (0.5 in.)," if the user then selects "stainless steel" from the material category, then stainless steel is added to the query. The new results will be further organized by the subconcepts of M-STAINLESS-STEEL. Therefore, specifying the query by adding concept categories helps users narrow down their search scope. Within the right panel, the upper left shows an image of the top-ranked searching component. Each catalog component is attached with an image for visualization purposes. By double clicking the image, users will see a separate window showing the original PDF catalog descriptions. The upper-right panel lists all the concepts in the PartXML. The bottom panel shows the types of components (in the collection) that interface with the current searching component.



(*a*)



(*b*)

Fig. 8 User interface for design orienteering: (*a*) search interface and first level of EO concepts for navigation and (*b*) search interface and returned results categorized

5 Preliminary Evaluations

The preliminary experiment uses an engineering catalog collection as the test bed and compares the retrieval performance of the EO-based search and keyword-based search. The effectiveness of using the design orienteering user interface will be part of our future studies. We have collected about 1000 components (with their PDF descriptions) from the online catalogs of 62 manufacturers. The length of the original PDF description ranges from 1 to 3 pages per component. Information in the tables is converted by using the table extraction method developed by Wei et al. [61].

We have implemented the most widely used keyword-based methods, the VS model. Note that the EO-based system indexes and retrieves PartXML documents, while the VS model acts on the full text, i.e., PartText documents. Standard IR measurements of recall and precision are used to measure the retrieval effectiveness.

$$\text{Recall} = \frac{\text{Number of retrieved relevant documents}}{\text{Number of relevant documents}}$$

$$\text{Precision} = \frac{\text{Number of retrieved relevant documents}}{\text{Number of retrieved documents}}$$

For instance, if five documents are retrieved, three of which are relevant, the total number of relevant documents is 10, then the recall is equal to 3/10, and the precision is 3/5.

The test was executed by the experiment committee, consisting of the first author, the last author, and two graduate students. Ten undergraduate students from the senior engineering design class and the SAE Formula-I design team in the Department of Mechanical Engineering were selected as subjects. They have various levels of design and manufacturing experience. The subjects were first briefed about the capabilities of the search systems. They searched for specific components, which satisfied certain requirements. Each of them provided the committee at least ten queries that they had generated during their tasks in the regular development process. They were also required to attach a short description as the context of each query such as why the component was needed and how it was used. The committee classifies these que-

Table 3 Average recall and precision of two retrieval methods

Type of queries		No. of queries	Average recall of EO model	Average recall of VS model	Average precision of EO model	Average precision of VS model
Query by concept disambiguation	General query	22	94%	28%	92%	31%
	Specific query	28	83%	77%	71%	81%
	Quantitative query	15	80%	36%	75%	41%
	Qualitative query	18	75%	38%	65%	35%
Query by concept abstraction		8	96%	25%	78%	30%
Average of total			85%	46%	78%	49%

ries based on criteria such as the level of complexity and the scope of search prior to applying them to querying the two search systems. The relevancy of each retrieved document is judged by the committee jointly.

There were a total of 100 queries generated. Nine of them were invalid and discarded; they were either duplicates or out of the scope of the EO and the catalog collection. The relations between the experience of the subjects and the variation of their queries were not analyzed. Instead, this will be part of future research. The valid sample queries were first classified into two types: "search for what I type" and "search for what I mean." These correspond to the type of queries that need concept disambiguation and the type of queries to be answered by concept abstraction, respectively. The first type of query is further classified as general queries, specific queries, qualitative queries, and quantitative queries. The general queries were associated with the upper-level concepts of the EO such as "search for electrical motors," while the specific queries were associated with the lower-level concepts, e.g., "find radius tab with threaded hole and made of steel" and "find parts made by SLA." Quantitative and qualitative queries refer to queries with quantitative and qualitative property-value pairs, respectively, for example, "dc motor of 6 V," and "ac motor that can resist high temperature."

Table 3 shows the results of the average recall and precision for the queries in each category and for all the queries taken together. The results demonstrate that the precision and recall of the EO-based model outperformed that of the VS model, especially for general queries. This is because for general queries, more relevant concepts are added to the selected concepts by query expansion, and therefore, recall is improved. For example, in the general query mentioned above, the selected concepts will be expanded from the matched D-ELECTRICAL-MOTOR to all of its descendent concepts such as D-STEPPER-MOTOR and D-SERVO-MOTOR. Meanwhile, precision is also improved significantly because concept disambiguation enables the system to search for the right lower-level concepts in the documents, which are also tagged by such concepts. In the VS model, however, the exact terms of the upper-level concepts are rarely used in documents or used differently from the query intent. Therefore, its retrieval performance is degraded.

For specific queries, the retrieval performance improvement over the VS model is less significant compared with the retrieval performance improvement for general queries. There are several reasons. First, it is more challenged in accurately recognizing lower-level concepts than upper-level concepts for the EO-based approach. Second, usually the recall is improved by query expansion. However, there is less space in the EO for query expansion with lower-level concepts. Third, in general, the VS model obtains fairly good precision when more exact keywords are provided.

For the rest of the query categories, the EO-based approach also achieves greater performance than the VS model because the concept-based document and query representations enable query reasoning based on meaning.

Note that for the retrieved results, the VS model is rank based, while the EO-based model is, in fact, Boolean retrieval. In order to make a reasonable comparison, a retrieved document is relevant if it has a similarity score greater than 0.15 in the VS model. The reported precision is when the maximum recall is achieved.

6 Conclusion

This research represents a first exploration of the possibilities of semantics-based engineering document analysis and retrieval, and its application. It develops an EO to represent the established design and manufacturing knowledge, both inside and outside of a company. We have described a new computational framework for the EO-based search system that aims at effectively retrieving unstructured engineering content.

The centerpiece of this framework is the EO and its associated EL. They are acquired semiautomatically by following a systematic ontological semantics approach. We have demonstrated the process used to conceptualize the EO and acquire it from various engineering knowledge resources. The EO can be easily extended to include new taxonomies, concepts, and relationships. We have developed algorithms for more efficient concept disambiguation, concept abstraction, and query by property values in order to (1) improve retrieval precision and recall, (2) detect users' search intents, and (3) satisfy users' information needs at different levels of detail. We have designed a novel context-aware user interface that (i) integrates EO-based search and category browsing, and (ii) dynamically updates and structures the retrieval results so that users can explore alternative solutions in a flexible manner and narrow their searches quickly.

The experimental results demonstrate that the EO-based search outperforms the keyword-based search. Using a test bed of 1000 engineering component descriptions from various suppliers, we found that the EO-based search improves the average recall by 39% and the average precision by 29%. More importantly, (1) it understands users' queries at the concept level when exact query terms are not available, and (2) it enables querying with quantitative as well as qualitative engineering specifications. All these query types are prevalent in engineering and design tasks but not handled properly by the traditional IR approaches.

The research suggests that PDM/PLM systems should take into account the importance of utilizing the established domain knowledge in order to achieve more effective engineering IR. Engineering design is a complex task. Therefore, understanding the context of the task, as well as obtaining information about this context in an effective manner, plays a crucial role in the success of engineers' decision making. By extracting dispersed contents and associating them with an explicit domain knowledge model in order to support the query inference on a meaningful and contextual level, our method has the potential of pursuing a more coherent design environment with future PDM/PLM systems.

Automatic ontology learning such as Refs. [62,63] aims at facilitating the ontology construction process by extracting knowledge from texts through NLP techniques and corpus statistics. It has potential to accelerate the ontology acquisition process. In addition, a large amount of engineering knowledge is already codified and available in engineering databases, design reposito-

ries, company-specific standards, etc. Each of these is either semi-structured or structured and has its underlying implicit ontologies. Therefore, it is feasible to develop NLP-based learning approaches to automate the knowledge extraction process from such documents complementary to the handcrafted acquisition process.

There are existing efforts to develop more general and upper-level ontologies, such as the suggested upper merged ontology (SUMO),[6] which provides definitions of general-purpose terms and acts as a foundation for more specific domain ontologies. It is desirable to merge the developed EO with SUMO in the future.

Future research will focus on the experiments that also include a large collection of internal documents such as CAD drawings, project reports, and notebooks provided by our industry partner. We will investigate the retrieval performance as well as the working performance gains by using (1) only the EO-based search system (text query+category browsing) and (2) the combination of the EO-based search and VS model, compared to combined base line approaches, such as Google desktop (to search the internal document collection), internet search engines (e.g., Google), and component warehouses.[7] Both tests will be executed in various product development environments or design tasks. In order to do these, the current EO needs to be expanded with more company-specific domain knowledge such as proprietary product classifications and various design and production processes.

Acknowledgment

Special thanks are extended to Imaginestics LLC for their help in preparing the test data and for donating the software, I-MIGRATE. Chris Bence, In Chul Chang, and Kaushik Mantri helped in acquiring the engineering ontology. We also want to thank the students who volunteered for our experiment. We would like to acknowledge the support of the 21st Century R&T funds and the National Science Foundation Partnership for Innovation Award (NSF-PFI) on ToolingNET. 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] Ertas, A., and Jones, J. C., 1993, *The Engineering Design Process*, Wiley, New York.
[2] Marsh, J. R., 1997, "The Capture and Utilization of Experience in Engineering Design," Ph.D. thesis, Cambridge University, UK.
[3] Lowe, A., McMahon, C., Shah, T., and Culley, S., 2000, "An Analysis of the Content of Technical Information Used by Engineering Designers," *Proceedings of ASME/DET Conference*, Baltimore, MA.
[4] Lowe, A., McMahon, C., and Culley, S., 2004, "Characterizing the Requirements of Engineering Information Systems," Int. J. Information Management, **24**, pp. 402–422.
[5] Radhakrishnan, R., 2006, "Information Retrieval at Boeing: Plans and Successes," *ACM SIGIR*, Seattle, WA.
[6] Ullman, D. G., 2001, *The Mechanical Design Process*, McGraw-Hill, New York.
[7] Ahmed, S., and Wallace, K. M., 2004, "Identifying and Supporting the Knowledge Needs of Novice Designers Within the Aerospace Industry," J. Eng. Design, **15**(5), pp. 475–492.
[8] Sivaloganathan, S., 1998, *Engineering Design Conference 98: Design Reuse*, ASME 98.
[9] McMahon, C. A., Lowe, A., Culley, S. J., Corderoy, M., Crossland, R., Shah, T., and Stewart, D., 2004, "Waypoint: An Integrated Search and Retrieval System for Engineering Documents," ASME J. Comput. Inf. Sci. Eng., **4**(4), pp. 329–338.
[10] Court, A. W., Ullman, D. G., and Culley, S. J., 1998, "A Comparison Between the Provision of Information to Engineering Designers in the UK and the USA," Int. J. Information Management, **18**(6), pp. 409–425.
[11] Hertzum, M., and Pejtersen, A. M., 2000, "The Information-Seeking Practices of Engineers: Searching for Document as Well as for People," Inf. Process. Manage., **36**(5), pp. 761–778.
[12] Uschold, M., and Grüninger, M., 2004, "Ontologies and Semantics for Seamless Connectivity," SIGMOD Record, **33**(4), pp. 58–64.
[13] Khan, L., McLeod, D., and Hovy, E., 2004, "Retrieval Effectiveness of an Ontology-Based Model for Information Retrieval," Int. J. Very Large DataBases (VLDB), **13**, pp. 71–85.
[14] Hernandez, N., Mothe, J., Chrisment, C., and Egret, D., 2007, "Modeling Context Through Domain Ontologies," J. Information Retrieval, **10**, pp. 143–172.
[15] Miller, G., 1995, "Wordnet: A lexical Database for English," Commun. ACM, **38**(11), pp. 39–41.
[16] Bhogal, J., Macfarlane, A., and Smith, P., 2007, "A Review of Ontology-Based Query Expansion," Inf. Process. Manage., **43**, pp. 866–886.
[17] Salton, G., 1988, *Automatic Text Processing*, Addison-Wesley, Wokingham.
[18] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R., 1990, "Indexing by Latent Semantic Analysis," J. Am. Soc. Inf. Sci., **41**(6), pp. 391—407.
[19] Ponte, J., and Croft, W. B., 1998, "A Language Modeling Approach to Information Retrieval," *Proceedings of ACM SIGIR 1998*.
[20] Robertson, S. E., Walker, S., and Hancock-Beaulieu, M., 1999, "Okapi at TREC-7: Automatic Ad Hoc Filtering, VLC and Interactive," *TREC-7*.
[21] Brin, S., and Page, L., 1998, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," J. Computer Networks and ISDN Systems, **30**(1–7), pp. 107–117.
[22] Ahmed, S., Kim, S., and Wallace, K. M., 2005, "A Methodology for Creating Ontologies for Engineering Design," *Proceedings of ASME/IDET&CIE Conference*, Long Beach, CA.
[23] Ferrucci, D., and Lally, A., 2004, "Building an Example Application With the Unstructured Information Management Architecture," IBM Syst. J., **43**(3), pp. 455–475.
[24] Hobbs, J. R., Appelt, D. E., Bear, J., Israel, D., Kameyama, M., Stickel, M., and Tyson, M., 1996, "FASTUS: A Cascaded Finite-State Transducer for Extracting Information From Natural-Language Text," *Finite-State Devices for Natural Language Processing*, MIT Press, Cambridge.
[25] Riloff, E., 1996, "Automatically Generating Extraction Patterns From Untagged Text," *Proceedings of the 13th National Conference on AI (AAAI-96)*, pp. 1044–1049.
[26] Farley, B., 2000, "Extracting Information From Free-Text Aircraft Repair Notes," Artif. Intell. Eng. Des. Anal. Manuf., **15**(4), pp. 293–305.
[27] Dong, A., and Agogino, A. M., 1996, "Text Analysis for Constructing Design Representations," Artif. Intell. Eng., **11**, pp. 65–75.
[28] Yang, M. C., Wood, W. H., and Cutkosky, M. R., 2005, "Design Information Retrieval: A Thesauri-Based Approach for Reuse of Informal Design Information," Eng. Comput., **21**(2), pp. 177–192.
[29] Song, S., Dong, A., and Agogino, A., 2002, "Modeling Information Needs in Engineering Databases Using Tacit Knowledge," ASME J. Comput. Inf. Sci. Eng., **2**, pp. 199–207.
[30] U.S. Patent No. 6,668,256.
[31] C3: Customization, Collaboration and Configuration Summit, 2005, Purdue University, West Lafayette, IN.
[32] Sudarsan, R., Fenves, S. J., Sriram, R. D., and Wang, F., 2005, "A Product Information Modeling Framework for Product Lifecycle Management," Comput.-Aided Des., **37**(13), pp. 1399–1411.
[33] Szykman, S., Sriram, R. D., Bochenek, C., Racz, J. W., and Senfaute, J., 2000, "Design Repositories: Engineering Design's New Knowledge Base," IEEE Intell. Syst., **15**(3), pp. 48–55.
[34] Kim, J., Will, P., Ling, S. R., and Neches, B., 2003, "Knowledge-Rich Catalog Services for Engineering Design," Artif. Intell. Eng. Des. Anal. Manuf., **17**(4), pp. 349–366.
[35] Patil, L., Dutta, D., and Sriram, R., 2005, "Ontology Formalization of Product Semantics for Product Lifecycle Management," *Proceedings of ASME/IDETC&CIE Conference*, Long Beach, CA.
[36] Nirenburg, S., and Raskin, V., 2004, *Ontological Semantics*, MIT Press, Cambridge.
[37] Kitamura, Y., and Mizoguchi, R., 2004, "Ontology-Based Systemization of Functional Knowledge," J. Eng. Design, **15**(4), pp. 327–351.
[38] Katranuschkov, P., Gehre, A., and Scherer, R., 2003, "An Ontology Framework to Access IFC Model Data," e-Journal Itcon, **8**, pp. 413–437; www.itcon.org.
[39] Patil, L., Dutta, D., and Sriram, R., 2005, "Ontology-Based Exchange of Product Data Semantics," IEEE. Trans. Autom. Sci. Eng., **2**(3), pp. 213–225.
[40] Grüninger, M., and Menzel, C., 2003, "The Process Specification Language (PSL) Theory and Applications," AI Mag., **24**(3), pp. 63–74.
[41] Borst, P., and Akkermans, H., 1997, "Engineering Ontologies," Int. J. Hum.-Comput. Stud., **46**, pp. 365–406.
[42] Grosse, I. R., Milton-Benoit, J. M., and Wileden, J. C., 2005, "Ontologies for Supporting Engineering Analysis Models," Artif. Intell. Eng. Des. Anal. Manuf., **19**, pp. 1–18.
[43] Liang, V. C., and Paredis, C. J. J., 2004, "A Port Ontology for Conceptual Design of Systems," ASME J. Comput. Inf. Sci. Eng., **4**, pp. 206–217.
[44] Nanda, J., Simpson, T. W., Kumara, S. R. T., and Shooter, S. B., 2006, "A Methodology for Product Family Ontology Development Using Formal Concept Analysis and Web Ontology Language," ASME J. Comput. Inf. Sci. Eng., **6**(2), pp. 1–11.
[45] Soininen, T., Tiihonen, J., Mannisto, T., and Sulonen, R., 1998, "Towards a General Ontology of Configuration," Artif. Intell. Eng. Des. Anal. Manuf., **12**(4), pp. 357–372.
[46] Fernández-López, M., Gómez-Pérez, A., and Sierra, J. P., 1999, "Building a

[6] http://www.ontologyportal.org/.
[7] www.McMaster.com.

Chemical Ontology Using Methonology and the Ontology Design Environment," IEEE Intell. Syst., **14**(1), pp. 37–46.
[47] Kuffner, T. A., and Ullman, D. G., 1991, "The Information Request of Mechanical Design Engineers," Des. Stud., **12**(1), pp. 42–50.
[48] Baya, V., Gevins, J., Baudin, C., Mabogunje, A., Leifer, L., and Toye, G., 1992, "An Experimental Study of Design Information Reuse," *Proceedings of the Fourth ASME/DTM Conference*, Scottsdale, AZ, Vol. 42, pp. 141–147.
[49] Pugh, S., 1997, *Total Design: Integrated Methods for Successful Product Engineering*, Addison-Wesley, Wokingham.
[50] Li, Z., Raskin, V., and Ramani, K., 2007, "A Methodology of Engineering Ontology Development for Information Retrieval," *Proceedings of the 16th International Conference on Engineering Design (ICED'07)*, Paris.
[51] Rothbart, H. A., 1996, *Mechanical Design Handbook*, McGraw-Hill, New York.
[52] Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S., and Wood, K. L., 2002, "A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts," Res. Eng. Des., **13**(2), pp. 65–82.
[53] Collins, J. A., Hagan, B. T., and Bratt, H. M., 1976, "The Failure-Experience Matrix-A Useful Design Tool," ASME J. Eng. Ind., **98**, pp. 1074–1079.
[54] Kutz, M., 2002, *Handbook of Materials Selection*, Wiley, New York.
[55] Kutz, M., 2005, *Mechanical Engineers" Handbook, Manufacturing and Management*, Wiley, New York.
[56] Glasgow, B., Mandell, A., Binney, D., Ghemri, L., and Fisher, D., 1998, "MITA, An Information-Extraction Approach to the Analysis of Free-Form Text in Life Insurance Applications," AI Mag., **19**, pp. 59–71.
[57] Reidsma, D., Kuper, J., Declerck, T., Saggion, H., and Cunningham, H., 2003, "Cross Document Ontology-Based Information Extraction for Multimedia Retrieval," *ICCS'03*, Desden.
[58] Baeza, R., and Neto, B., 1999, *Modern Information Retrieval*, Addison-Wesley, New York.
[59] Del-Rey-Chamorro, F. M., and Wallace, K. M., 2005, "Understanding the Search for Information in the Aerospace Domain," *Proceedings of 15th International Conference on Engineering Design (ICED'05)*, Melbourne, Australia.
[60] Teevan, J., Alvarado, C., Ackerman, M. S., and Karger, D. R., 2004, "The Perfect Search Engine Is Not Enough: A Study of Orienteering Behavior in Directed Search," *ACM Conference on Human Factors in Computing Systems*, pp. 415–422.
[61] Wei, X., Croft, W. B., and McCallum, A., 2006, "Table Extraction for Answer Retrieval," J. Information Retrieval, **9**(5), pp. 589–611.
[62] Maedche, A., and Staab, S., 2001, "Ontology Learning the Semantic Web," IEEE Intell. Syst., **16**(2), pp. 72–79.
[63] Shamsfard, M., and Barforoush, A. A., 2004, "Learning Ontologies From Natural Language Texts," Int. J. Hum.-Comput. Stud., **60**, pp. 17–63.