Proceedings of IDETC/CIE 2005
ASME 2005 International Design Engineering Technical Conferences &
Computer and Information in Engineering Conference
September 24-28, 2005, Long Beach, California, USA

DETC2005-85107

SEMANTICS-BASED DESIGN KNOWLEDGE ANNOTATION AND RETRIEVAL

Zhanjun Li **Min Liu** **David C. Anderson** **Karthik Ramani**

Purdue Research and Education Center for Information Systems in Engineering (PRECISE)
School of Mechanical Engineering
Purdue University, West Lafayette IN 47907-2024, USA
Emails: {liz, liu66, dave, ramani} @ecn.purdue.edu

ABSTRACT

Nowadays computer aided tools have enabled the creation of the electronic design documents on an unprecedented scale, while determining and finding what can be reused is like searching a "needle in a haystack." One of the primary reasons for this is that the design knowledge behind the physical design is not properly represented and indexed. With the large amount of designs available, design engineers need to retrieve suitable ones, so that a knowledge-based unified reuse environment can be realized.

In this paper, we describe our approach to intelligently annotating and retrieving designs by using ontology engineering and natural language processing. We use the design documents from an engineering design class as the first case study.

KEYWORDS Ontology, Design knowledge annotation, Design document, Retrieval, Reuse

1. INTRODUCTION

Rational mechanical design is the process of materializing the product life cycle information into a physical prototype driven by the design knowledge. In current practice, textual files and 3D CAD models are two of the most prevalent data formats and information resources in which the design knowledge for constructing the product is embedded. Examples of the design knowledge are requirements, specifications, functions, behaviors, structures, manufacturing rules, tolerances, and material selections. For design knowledge retrieval and reuse, the extraction of such knowledge from the existing textual and geometric representations is necessary.

The popularity of computer-aided tools in product design and manufacturing industry generates a large amount of design knowledge which is embedded in the textual or geometric design documents. For example, as reported by Marsh [1, 2], there are approximately 40,000 documents produced in the design of a single engine in an aerospace company. The availability of such extensive knowledge resources creates new challenges as well as opportunities for research on how to retrieve and reuse the knowledge from the design documents rather than index the documents with simple attributes, similar to what is done by most general document management tools.

There are many benefits to reuse design knowledge. For example, reusing a good existing design reduces efforts as well as risks at the design stage and downstream stages, because the proven products preserve the validated design knowledge. Also, not using knowledge that has been found unworkable reduces making similar errors. Design by re-use can save up to 75% of non-recurring costs since most of them are committed by the end of the design process [3]. It can reduce the lead time by taking "short-cuts" and eliminating many downstream activities as well as iterations [4]. By reducing part proliferations, design knowledge reuse lowers product variability and improves inventory efficiency.

Despite the benefits, industries find that design knowledge reuse has only met limited success in practice [5]. Design engineers always find it hard to locate the previous designs for their needs. An empirical study conducted in [1] shows that design engineers are reluctant to access documents to which they did not directly contribute. The reasons for this are that there is no mechanism for engineers to be aware of the contents or semantics of the documents, and retrieve them. In short, the design documents and the knowledge embedded in them are not proper represented and not well indexed.

Design knowledge retrieval is the first step towards enabling design knowledge reuse. In order to retrieve design knowledge from unstructured design documents, one first needs to agree on what kinds of design knowledge should be extracted and represented, and how to explicitly model this knowledge.

Experiments conducted by [2, 6-9] show the importance of design knowledge about functions, behaviors, and structures for design reuse. Such importance also is recognized in a project-based senior engineering design class, which we use as a case study. During the initial design stage, students look for past designs which have some specific functions, mechanical behaviors, or structures. For project management, the functional and physical significance of each design are also of interest.

All these design knowledge descriptions such as functions, behaviors, and structures can be abstracted and represented as concepts and relationships. Example of the functional, behavioral, and structural concepts are phrases such as “rotate,” “rotation clockwise,” and “gears,” respectively; the structural relationships are phrase such as “part A is aligned with the top surface of part B,” and “part B is connected with part C through a pivot joint;” the functional relationships are phrases such as “the rotation of part B is controlled by part C,” and so on.

In this paper, we propose to use the ontology model to guide the extraction of design knowledge from the textual documents. Ontology is a knowledge representation model to explicitly conceptualize a domain by defining and characterizing the concepts, as well as various relationships among the concepts. It thereby systematizes the “ill-structured” design knowledge to make the design knowledge accessible by both humans and computers. The details of our ontology model are given later.

On the other hand, we found that the sentences which describe the function, behavior, and structure of a design often present a limited set of linguistic patterns as well as a limited scope of lexical terms. For example, the function of a part is described as verb phrases such as “A second DC motor turns a pulley that allows the turret to rotate 360 degrees clockwise or counter-clockwise.” The structure concepts or the name of the parts are termed as nouns such as “gears” and “the strut of the front suspension.” The structural relationships between two parts are always represented by prepositional phrases such as “the mechanisms show the movement of all gears in the power train to drive the rear wheels.”

With these observations, we believe that such linguistic patterns and the domain specific concepts can be extracted from the texts as domain semantics by using natural language processing (NLP). Ontology engineering provides the fundamental concepts and relationships acting as a basis to assist the automatic parsing.

Our long-term goal is to develop a theory and method for automatically extracting and indexing the design knowledge from unstructured design knowledge resources based on a structured representation model. The method should be intuitive to design engineers so that they can retrieve relevant design knowledge with effectiveness and efficiency. In this paper, we focus on combining ontology engineering and NLP to extract the design knowledge from the textual documents, as well as the methods of annotating, indexing, and retrieving the existing designs based on the ontology model. The prototype is implemented with the initial data from the engineering design course.

The rest of the paper is organized as follows. Section 2 describes some definitions used in this paper. In Section 3, we give a brief overview of our approach. The details about the ontology modeling and the meta-knowledge acquisition are discussed in Section 4. Section 5 describes the document parsing and knowledge extracting algorithm. The ontology-based personalized retrieval algorithm is discussed in Section 6. In Section 7, we summarize the prototype implementations. The case study is discussed in Section 8. Current limitations and potential improvements of our approach are discussed in section 9.

2. DEFINITIONS

To allow the reader to follow the remainder of the paper, several fundamental terms are defined formally here.

Document: Design document or project document. It refers to all the digital works produced during the product life cycle, including textual files, 3D CAD models, 2D drawings and sketches, analysis and simulation models, video and audio files, etc. In this paper, document means only the textual document.

Component: An assembly, a sub-assembly, or a part. It is often referred to by its name, such as “DC motor” and “spur gear.”

Structure or form: The shape of the components and mating relations between components, or the geometry and topology of features, of one part, or between two parts.

Behavior: The response action of the component to the input from other components or from the environment. Behavior can be described either qualitatively such as clockwise rotation, or quantitatively such as the parameters and the analytical equations for the representation of the behavioral attributes.

Behavior attribute: The physical and geometric attributes which characterize the performance or action of the structure. For example, the output speed of a DC motor.

Function: The abstraction of the behavior. It is how humans perceive the purpose of a component and is represented qualitatively as a transitive verb (+ objects). Function includes performance function, design function, manufacturing function, maintenance function, etc. We pursue the function definition from the design perspective.

Meta-Knowledge: The elemental concepts of the ontology model and the relationships among these concepts.

3. OVERVIEW

Our knowledge representation model is represented by function-behavior-structure-ontology (FBSO). It systematizes the three fundamental aspects of the design knowledge, and it shows how the functions and behaviors are achieved. It directly answers many questions about the design intent and sets the stage for further analysis as well as design by reuse.

Given that the input documents are in proper English, we seek to obtain concise and complete descriptions of the design knowledge. The descriptions should be both qualitative and quantitative. Qualitative descriptions characterize the design functions and working behaviors in an abstract manner. They allow design engineers to design and modify complex mechanisms before delving into details. On the other hand,

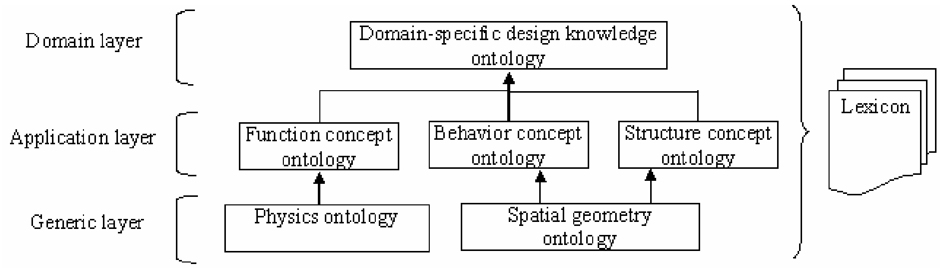

Figure 1. THE LAYERED FBSO MODEL.

quantitative descriptions provide the details for tasks which qualitative descriptions lack. The complete description combines both methods to avoid the deficiencies of using only one method. It allows the design engineer to understand, evaluate, and reuse previous designs under various conditions.

Automatically extracting design knowledge from the document requires recognizing the syntactic structure as well as the semantic meaning of the text which describes the knowledge. Meta-knowledge is also needed to fulfill the inference. This is a nontrivial task even for a simple document with several sentences. In-depth analyzing and fully understanding the syntax as well as semantics is practically impossible due to the complexities of natural language [10]. However, because we are only interested in specific aspects of the domain knowledge, the corresponding linguistic patterns and terms found in the documents are quite limited. Furthermore, we can solve the domain-specific natural language problem with a general framework where the methodology as well as the primary modules can be applied to different domains.

The FBSO model is not only used to assist the automatic design knowledge extraction, but provides a dynamic concept space to annotate and index the documents. We name the whole process as design knowledge annotation, which is to use computer-operatable natural language phrases to describe the contents of various segments of design knowledge resources, such as documents. We use "simplified" natural language processing to parse the text and generate the metadata. These metadata are then mapped to the corresponding meta-knowledge in the application ontology and form a concept network to be added into the domain-specific design knowledge ontology. With this procedure, we reduce the amount of in-depth linguistic analysis and human intervention required for building a large knowledge base. The FBSO model is organized in a modular as well as a layered manner, thus reducing the complexity involved in the ontology maintenance and improving the portability as well. An ontology-based retrieval algorithm is developed to support the design knowledge retrieval.

The document analysis algorithm consists of tokenization, filtering, part of speech (POS) tagging, reference resolution, and semantic analysis. The ontology-based retrieval algorithm includes query analysis, query by disambiguation, and personalized retrieval.

At the document analysis stage, the inputs are the documents in each project folder; the outputs are the design knowledge regarding functions, behaviors, and structures extracted from the texts. At the retrieval stage, the input query is a list of keywords about the function, behavior, or structure of the designs which a user searches for; the outputs are existing designs ranked by their relevance to the query.

4. ONTOLOGY MODELING

Ontology is the abstraction of a domain. It can be seen as a realization as well as a practical argument for knowledge-based processing [11]. The ontology model defines a set of representational terms which we call concepts, the attributes and constraints of each concept, and the relationships among concepts. These concepts must be expressed in a language that is sufficiently expressive and free from the problems of imprecision and ambiguity [12]. The ontology model represents the domain at both the syntactic and semantic level, and integrates different inference mechanisms within one structure through relationships. In our research, it acts as the data structure as well as the representation model to systemize the design knowledge.

Figure 1 shows the layered FBSO representation of design knowledge. The ontology models at each layer are distinguished by different levels of generality [13]. The concepts in a certain layer are described in terms of the concepts in the lower layer. At the generic layer, the spatial geometry ontology defines the fundamental concepts and relationships of the spatial geometry such as the terms of plane, axis, horizontal, and vertical. The physics ontology describes the physical principles of the domain such as the principles of motion, force, and electricity.

The application layer includes function concept ontology, behavior concept ontology, and structure concept ontology. The function concept ontology specifies the functional concepts, which are the purpose of the structural concept, and the hierarchy among these functional concepts. Although the functional representation of "transitive verb (+ objects)" is domain specific, the definition of each functional verb is general. It is a transitive verb with the semantic relationships defined with other verbs. Examples of these relationships are superordinate and subordinate. The function concept ontology is obtained by extracting the verbs and their relationships from the general linguistic ontology such as KBAE, where each verb is defined as an event [14]. The behavior concept ontology describes the fundamental concepts about behaviors and behavior attributes, and the relationships among these concepts as well. The behavior includes motion behavior, material behavior, electric behavior, etc. The concepts about motion behavior are also associated with the corresponding concepts in the function concept ontology. For example, rotation->rotate. The behavior concept ontology is shown in Fig. 2. The structure concept ontology includes geometry feature ontology, mating relation ontology, and component ontology. The

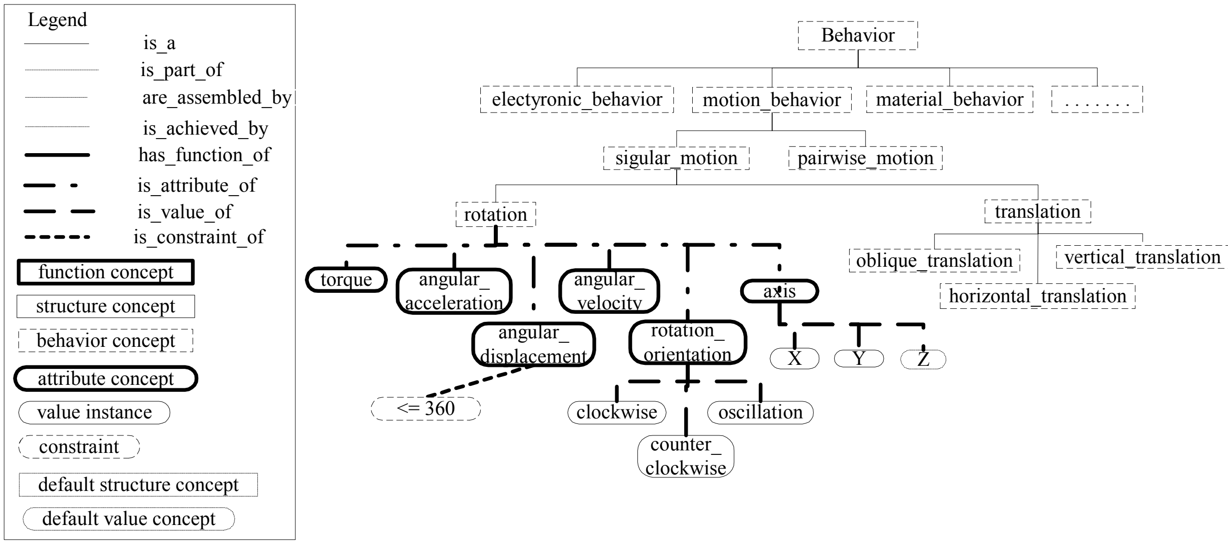

Figure 2. EXAMPLE OF BEHAVIOR CONCEPT ONTOLOGY.

structures of the former two ontology models are similar to the feature taxonomy and mating relation taxonomy, respectively. They are hierarchical organizations of generic shapes as well as spatial mating relationships between two parts. The geometry definitions of each concept in the hierarchy are expressed as a set of rules and facts. Please refer to [15] for the details of these two ontologies. The component ontology involves the definition, characterization and hierarchical classification of the mechanical and electrical elements, which are produced and named by the company, such as bolts, bearings, and motors. Currently, the concepts in the component ontology are obtained by manually analyzing all the documents in the repository. These could be done using the automatic corpus-based approach [16]. An example of the component ontology is showed in Fig 3.

The focus of this paper is the model and the automatic generation process of the domain-specific design knowledge ontology. Figure 4 illustrates an example of the ontology representation for the design knowledge extracted from the document of a design project: mobile rocket launcher. The concepts and the relationships formulate the domain specific design knowledge embedded in the documents. In the following sections, the term of the ontology refers to the domain-specific design knowledge ontology.

Each ontology is represented as a directed graph structure with root, each node represents a concept, and each arc

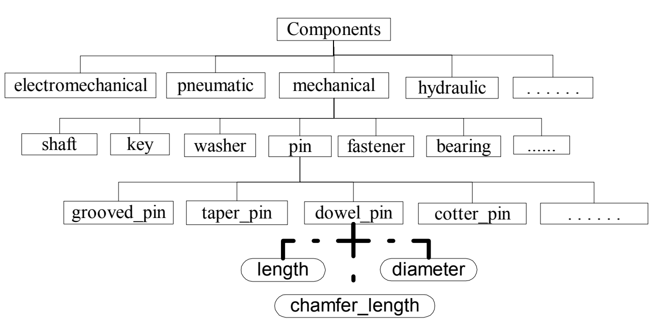

Figure 3. EXAMPLE OF COMPONENT ONTOLOGY

represents a specific relationship. In general, each concept in application ontology as well as the domain-specific design knowledge ontology at the top layer contains a unique reference/pointer, which refers to its comprised term(s) in the lexicon. The lexicon is described in the next section.

Concepts are connected by relationships. Eight types of relationships are used to create our ontology: is_a, is_part_of, are_assembled_by, has_function_of, is_achieved_by, is_attribute_of, is_constraint_of, and is_value_of. We argue that these relationships correspond to the key abstraction primitives in typical ontological models for engineering design systems.

Is_a: It also can be named as is_a_kind_of. This relationship is used to represent concept specialization. A concept represented by C_j is said to be a specialization of the concept represented by C_i if and only if C_j is a kind of C_i. For example, a mobile rocket launcher is a_kind_of toy. In other words, the toy is the generalization of the mobile rocket launcher.

Is_part_of: A concept C_j is_part_of a concept C_i if and only if C_i has a C_j (as a part), or C_j is a part of C_i. For example, the rocket mount is_part_of the mobile rocket launcher.

Are_assembled_by: Two or more structure concepts can be assembled together either by a mating structure concept such as a pivot joint or by a function concept, e.g. the phrase of "the mount plate is *connected* to the turret" can be represented as the mount plate and the turret are_assembled_by connect.

Has_function_of: Each structure concept may have several functions and each function concept can be achieved by multiple structures. The relationship refers to the connection between a structural concept and a functional concept. For example, a DC motor has_function_of turning a pulley.

Is_achieved_by: It specifies how the function of a structure is achieved either through applying some physical principles, such as transfer heat by radiation, or through the function provided by the other structures.

Is_attribute_of: In general, each structure concept has several numeric or symbolic attribute concepts characterizing its behavior and geometry, and each behavior concept has the

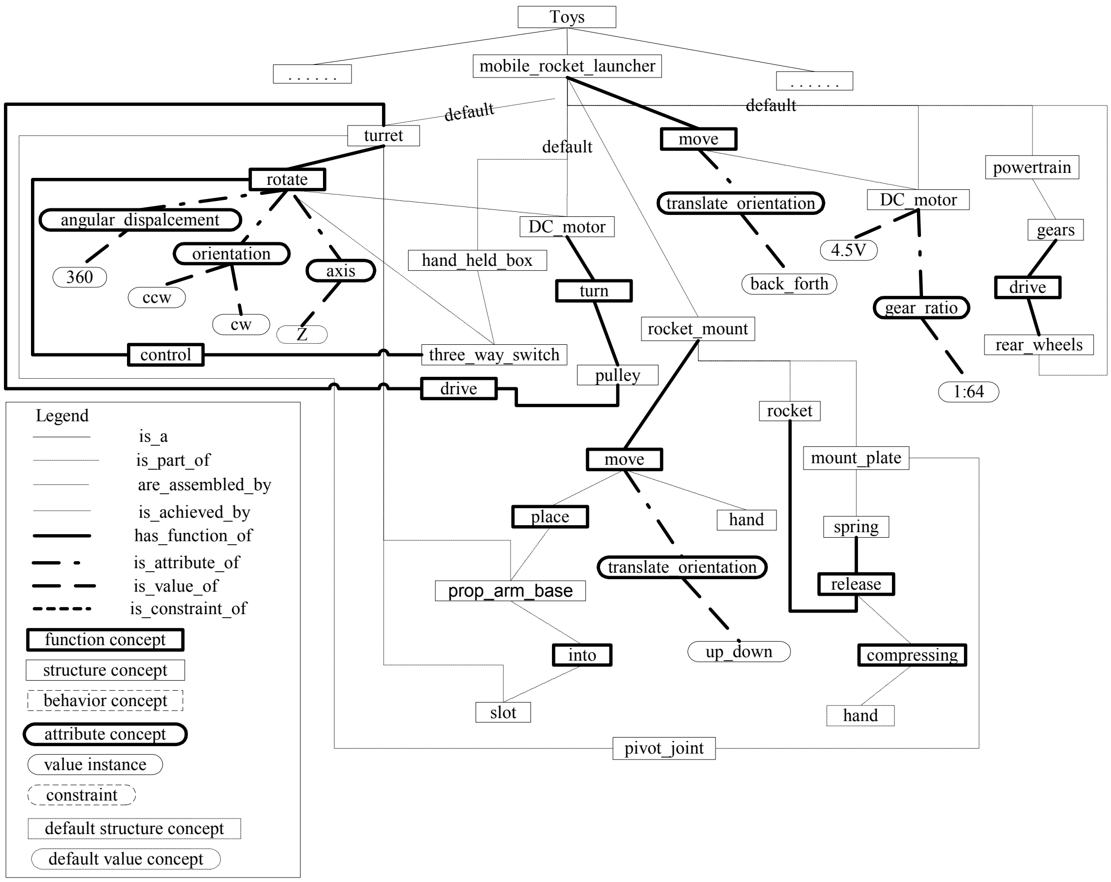

Figure 4. DOMINA-SPECIFIC DESIGN KNOWLEDGE ONTOLOGY.

attributes for detailing. For example, "length" is_attribute_of "pin", and "orientation" is_attribute_of "rotation".

Is_constraint_of: A constraint concept specifies the type (numeric/symbolic) and range for the value of an attribute concept. The relationship between the constraint concept and the attribute concept is defined as is_constraint_of.

Is_value_of: The value concept can be recognized as an instance of an attribute concept.

There are also default relationships and default concepts generated by the system. Although they do exist in a specific product, some instance of the concepts and relationships are not explicitly described in the document. The generation of the default concepts as well as the default relationships will be discussed in the next section.

5. DOCUMENT ANALYSIS

Analysis of free-form text has been studied mainly from three perspectives [17]: information retrieval (IR) or keyword-based [18], information extraction (IE) [19], and in-depth natural language processing [11, 16].

In the classic IR approaches, such as Boolean [20], vector space [21], and probabilistic [22], documents are represented through a list of representative keywords, which are assumed to be highly relevant to the application or the user's query. These keywords are used to index the documents as well. They are relatively easy to implement. However, the extracted keywords are unable to represent the desired semantics which many documents contain. For example, the following are three queries with which user want to investigate the designs which

i. Lock a car plate with a curvilinear slot sliding along a cylindrical pin in the assembly;
ii. Have a spur gear assembled with a step shaft using a woodruff key; or
iii. Have a DC motor with output speed of 100 rpm to 1000 rpm.

The first query is for a specific design intent or design function of a mechanism and its significant components. The second is for the detailed assembly structures. In the last example, the main concern is the exact value of a behavior attribute from its structure. It is impossible to extract these semantic knowledge descriptions accurately by using keywords. Keywords are syntactic units, which are unable to reflect meanings as well as relationships.

The in-depth natural language analysis approach fully analyzes the input document at both the syntactic and semantic levels. It is usually very complex and expensive. We chose not to take this approach because it does not satisfy the need of our application to focus on the limited domain-specific aspects of the text, instead of the general linguistic contexts.

IE approaches integrate NLP tools with domain knowledge models or machine learning to extract the concepts in the texts and form a framework based on the pre-defined template [17].

The IE approach is the closest to our approach regarding the fundamental NLP techniques and the principle of using

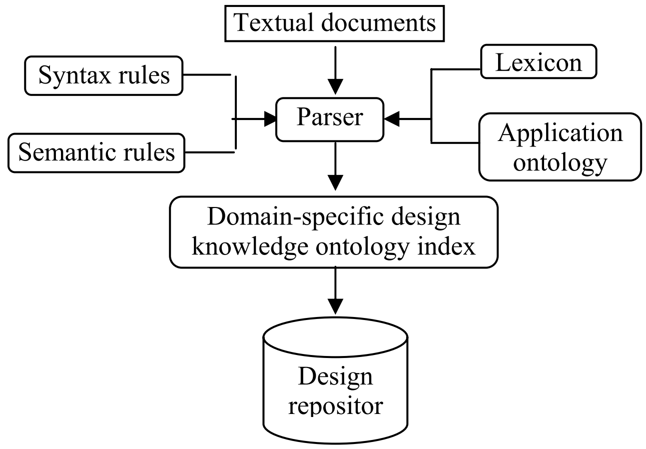

Figure 5. DOCUMENT ANALYSIS PROCEDURES

domain knowledge to assist the extraction process. However, there are two major differences. First, our method extracts not only concepts, but also the relationships among these concepts. We believe the representation of design knowledge is not complete without identifying the relationships. Second, the extracted design knowledge ontology model in our method is also used to index the documents and support the retrieval procedure.

Our method makes use of NLP techniques (parsing, tagging, lexicon, and rules) as well as the application ontology to identify the concepts and relationships contained in the documents. It then helps in the engineering of the domain ontology, represented as a network of these concepts connected by the relationships. The application ontology models, i.e. the function concept ontology, the behavior concept ontology, and the structure concept ontology are used to represent the context of the domain to be analyzed in terms of important concepts. Simplified NLP algorithms are developed to process texts in order to extract the concepts and the relationships. Figure 5 shows the modules and procedures of document analysis.

The lexicon includes terms such as functional verbs, nouns (domain-specific structure concepts and behavior concepts), prepositions, pronouns, adverbs, and the synonyms and inflections of each term. Given the lexicon as well as the syntax rules and semantic rules, which are currently manually coded, the document analysis algorithm extracts the functional, behavioral, and structural knowledge from the textual document. The algorithm works as follows:

1. Tokenization: The input character stream is parsed into tokens/words and punctuation marks. Sentences are then formed.

2. Filtering: each sentence passes through the lexicon filter, which looks up each token in the lexicon. If the sentence does not have any token matching with the function, behavior, or structure concepts/terms in the lexicon, it is removed.

3. POS tagging: each token is tagged with its part of speech [23].

4. Reference resolution [24]: the passive sentences are re-formatted into the active, pronouns in the noun clauses are replaced by the subject it refers to.

5. Semantic analysis: sentence constituents are grouped into nouns, verbs, prepositional phrases, etc (This sub-step was named as "bracketing" in [17]). Further analysis of their adjacent tokens is required to map from the sentence constituents to the concepts and relationships in the ontology model:

a) Nouns are mapped to the corresponding structure concepts in the structure concept ontology if they are either subjects or objects. Nouns with cardinal numbers together are mapped to the attribute concept of the corresponding behavior concept, etc. For example, in the sentence, "the pulley drives the turret to rotate 360 degrees clockwise or counter-clockwise" where "pulley" and "turret" are mapped to the structure concepts, "360 degrees" is mapped as a value instance to the attribute concept of angular_displacement under the behavior concept of rotation.

b) Verbs are mapped to the function concepts in the function concept ontology only if they are syntactically between two structure concepts (i.e. the subject and the object) in the sentence. If the verbs represent the assembly meanings, such as connect, joint, and touch, the functional relationship between the subject and object is mapped to are_assembled_by in the ontology. Otherwise the relationship is has_function_of. For example, the same sentence used in a) is mapped to pulley (has_function_of) drive turret.

c) The prepositional phrases may represent the relationships of is_part_of, has_function_of, is_attribute_of, or is_value_of, depending on the constituents at the left and right sides of the prepositions. Table 1 shows some examples.

As we mentioned before, these extracted concepts and relationships are represented in a directed graph structure with a root node. Each graph is used to automatically extend the structure concepts of the ontology, which is either an abstracted/generalized concept of all the products a company produces, or a concept of an assembly and subassembly. For example, car is a generalized concept of all models of cars of an automotive company. A reference/pointer to the project folder is assigned to the root node of the graph. The project folder includes all the documents of a design.

Some concepts or relationships may be omitted by the author in the document, for example, "the cam rotates (around the Z axis)," "the output shaft (in the gearbox) transmits the rotation from the motor." However, readers can identify them according to their background knowledge and the contexts of the documents. In practice, we handle these implicit contents by defining the default value instance of the attribute concept in the behavior ontology, and by identifying the most relevant concept in the structure concept ontology. For example, "Z axis" is the default value to the attribute of "rotation axis."

Table 1. PREPOSITIONAL PHRASES ANALYSIS

Sentences	Prepositional phrases	Extracted relationships
The mechanisms show the movement of all gears in the power train to drive the rear wheels	gears in the power train	gears is_part_of power train
The rotation of the cam results in the cam followers and the brake shoes being forced radially outward	rotation of the cam	cam has_function_of rotate
The slider moves along the slot with the length of 5 inch	slot with length	length is_attribute_of slot
	length of 5 inch	5 inch is_value_of length

6. QUERY PROCESSING

6.1 Concept Disambiguation and Abstraction

The user's query is a list of keywords representing query intents. The ontology provides concepts as index keys that are matched by the keywords at both syntactic and semantic levels. Tokens are generated from the query. These tokens are matched with concepts in the ontology through the terms and their synonyms in the lexicon. As mentioned in the preceding section, each concept in the FBSO model has references to the corresponding terms in the lexicon. Each term has a list of synonyms. The term also is a synonym of itself. Denote the synonyms list as (S_1, S_2, S_3……S_i, …… S_n). A keyword in the user query is matched with an element S_i in the list. A term may be shared by multiple concepts.

In the ontology model, ambiguity occurs at both the syntactic level and semantic level. In the first case, several concepts, each of which is composed of multiple terms, may share common term(s), such as the concepts of "gear shaft" and "motor shaft." At the semantic level, the same term may represent different meanings in different contexts. For example, a shaft of a hammer means a handle, and a shaft of a wheel means an axle.

On the other hand, the system needs to understand the true meaning of each keyword, as well as the query intent when a set of keywords is combined. For example, with the query of "rotation, drive follower up and down," the algorithm should return some design like a cam even though the description of the cam does not have the exact match with all the keywords. We call such capability concept abstraction.

Concept disambiguation and concept abstraction show the potential for bridging the "semantic gap" between the user query and the system representation. For example, a user's query of "output speed 1000 rpm and motor" has semantic meanings, i.e. output speed is an attribute of the motor and 1000 rpm is a value instance of the output speed. However, since the query is treated as a list of separated keywords, the semantic meanings are lost in the general keyword-matching methods. In our approach, however, because the ontology characterizes the concepts and the relationships, the semantics of the query can be recovered at the system side. For the same example, "motor" and "output_speed" are the structure concept and the related attribute concept respectively, while "1000 rpm" satisfies the constraint of the attribute concept "output_speed" in the behavior concept ontology.

False matches, which cause the loss of retrieval precision and recall [25], result from the inability of resolving concept disambiguation and concept abstraction. Therefore, we need a metric that measures which concept is intended from the presence of the keywords.

Our *scoring region* metric which is adapted from the algorithm in [25] is to measure the relevance of the concepts to the user's query. The metric is based on the observation from linguistics: the way to disambiguate the meanings of a word in the sentence is by referring to its context, such as the neighboring words, sentences or paragraphs. Each scoring region refers to the sub-graph including all the sub-concepts and the relationships under the same parent concept in the ontology. Usually, the parent concept represents the designed product, assembly, or subassembly. The metric includes two separate measurements: number of hits (NOHs) and concept distance (CD). NOHs measures the number of unique matches between the keywords and the concepts in a region. The region with more matches has a higher score. To calculate the CD, we give positive weight to each relationship in the ontology model. Suppose the two keywords used in the query are "cam" and "translate," and further assume that there are only two concepts in each region which include these two keywords respectively. The CD of each region is then calculated by traversing the graph from one concept to the other concept and adding the weight of each traversed arc. In general, the retrieved results are first ranked by NOHs, and then the CD is used to order the results which have the same NOHs. The concept that is the parent node of the region with the maximum NOHs and minimum CD represents the most relevant design. The project folder attached with this concept node is retrieved first.

Table 2. USER PROFILE

	Query keywords	Concepts in the ontology model			Project folder references		
1	Shaft	Handle, hammer	Axel2	……	label 101	label 32	……
2	Connect	Fastner_attachment	Welding_attachment	……	label 33	label 01	……
3	……	……	……	……	……	……	……

6.2 Personalized Retrieval

As the number of deposited designs grows, it is more and more difficult for engineers to find the desired ones which reflect what they want. We build a user taxonomy by tracking behaviors during retrieval and browsing. It is also important for a text-based retrieval system to establish the relations between the system taxonomy and the user taxonomy in order to improve the precision and recall and make the system adaptive to the user's context as well [26]. The personalized retrieval first collects the user's selections when the user browses the retrieved results. These selections are text documents or 3D CAD models in the project folder, and the corresponding concepts in the ontology model. Then it adds the concept to a queue of the query keywords. This procedure forms the user profile, illustrated in Tab. 2 as a dynamic table: the second column records all the different keywords used by a specific user; the third column lists the selected concepts. The references of the associated project folders are in the fourth column. During each retrieval, if the keywords are identified in the user profile, all the project folders referenced by the listed concepts of the keywords will be displayed followed by the results further processed with the disambiguation algorithm. The user profile is updated whenever the new concepts in the ontology are selected for the existing keywords.

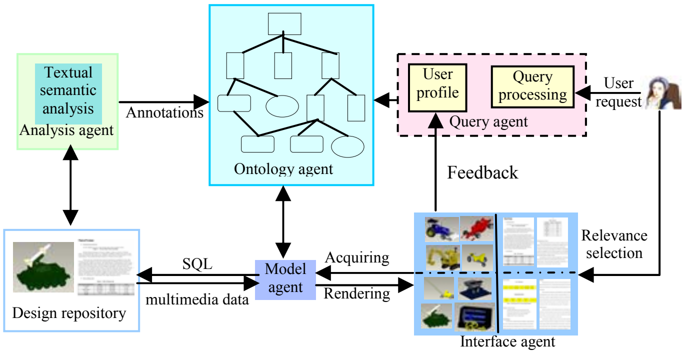

Figure 6. SYSTEM FUNCTION ARCHITECTURE

7. IMPLEMENTATION

Figure 6 presents the functional architecture of the prototype: a Design-Knowledge-On-Demand Dispatcher (DKODD).

Design document repository: In many industries, an engineering database is used to store all the design documents based on projects. However, to simulate that scenario on a small scale in the prototyping class, the system stores all the documents in a file directory for easy implementation. Each design group submits its final design documents and CAD models into its project folder.

Analysis agent: The textual documents are the input to the analysis agent for design knowledge annotation. The analysis agent first acts as a parser to do document analysis. It learns concepts and relationships automatically from the documents and uses them to build the domain-specific design knowledge ontology. The ontology is dynamic in that it is continuously and automatically expanded as new projects are submitted.

Query agent: A user query with a list of keywords is processed by the query agent and sent to the ontology agent. A user profile in between is to record the user's preference and later filter out unwanted retrieval results lately.

Ontology agent: The ontology agent maps the user query to the concept space of the ontology. It selects the concepts with a better scoring region metric (i.e. higher NOHs and lower CD) by the disambiguation algorithm and sends the attached project folder references to the model agent.

Model agent: The model agent generates the SQL query to retrieve the design documents from the repository. It takes the reference names of the project folders passed from the ontology agent. The tessellated data (of the CAD models) and the textual documents are then sent to the interface agent for rendering.

Interface agent: The interface agent is a search result navigator and a document viewer. The user can browse and select the projects, and zoom into the retrieved text documents or CAD models.

The system is implemented in Microsoft .Net framework using Visual C++.

8. CASE STUDY

This research uses a senior engineering design class for the requirement analysis and the test-bed. The course ME444, Computer-Aided Design and Prototyping, in the School of Mechanical Engineering at Purdue University is a senior undergraduate level course. Each team in the class implements a semester-long project by designing and prototyping a real product, such as a toy, with computer-aided design software as well as prototyping equipment. The only initial constraints for the toy are the weight and volume, and it should have some kinematic actions. During the project, teams submit their design documents online in a folder. They are also required to include a project summary report, which describes the significant function, behavior, and structure of their design. The report starts with a statement of the designed product.

Crane Rocket launcher Racing car

Figure 7. EXAMPLES OF PAST PROJECTS

After several years, hundreds of these projects have been completed together with thousands of CAD models, analysis files, and Word documents. The pictures shown in Fig. 7 are some examples of past projects. All these projects can be reviewed by later students. Typically, students start the project by first locating previous designs by using a random search. Then they look at the project summary report before they generate their own concept and detail it. They may also use internet search tools, e.g. Google, to search for current products and solutions. However, there are some limitations:

i. Past projects are organized by years. It is impossible for both the instructor and the student to retrieve by the design knowledge in which they are interested.

ii. Product concepts are implied as file names in the project folder but are not structured and are not easy to interpret by new students.

iii. Google search helps in the initial concept search, but lacks the engineering contexts such as functions and behaviors. The students always have to do a second time search among many unrelated retrieved results.

Therefore, it is valuable to have a course project platform which is capable of interpreting past design knowledge and organizing previous projects in a structured as well as knowledge-based manner, and which enables fast retrieval. More value will be added if such a platform can incorporate the newly created knowledge in an open fashion.

In practice, the project summary report is used as the input for document analysis. An illustrative example is provided to explain the analysis and the mapping procedure from the metadata of the document to the meta-knowledge in the ontology. The final output is the extracted semantic knowledge shown in Fig. 4.

Example:

Our action toy is a mobile rocket launcher. The physical prototype is capable of moving forward or backward by using a 4.5V DC motor with a 1:64 gear ratio. A second DC motor turns a pulley that allows the turret to rotate 360 degrees clockwise or counterclockwise. These motions are controlled by 3-way switches mounted in the handheld control box. The rocket mount can be raised and lowered by hand by placing the base of the prop arm into the slots on the top of the turret. The rocket can be fired by pulling it back against the plate on the *rocket mount to compress the spring and releasing the rocket. The virtual prototype of the rocket launcher was modeled in Pro/E and used to construct the preliminary mechanisms of the toy. The mechanisms show the movement of all gears in the power train to drive the rear wheels. It also is capable of showing the rotation of the turret and the elevation of the rocket mount.*

1. Tokenization: produces tokens and forms sentences. Only the first three sentences are listed.

Sentence 1: our action **toy** is a mobile **rocket launcher**

Sentence 2: the physical prototype is capable of <u>moving</u> *forward* or *backward* by <u>using</u> a *4.5V* **DC motor** with a_*1:64 gear ratio*

Sentence 3: a second **DC motor** <u>turns</u> a **pulley** that <u>allows</u> the **turret** to <u>rotate</u> *360 degrees clockwise* or *counterclockwise.*

. . . . . .

2. Filtering: in this example, the first three sentences are retained because each of them has words matched with some terms of function verbs (underline), structure nouns (bold), or behavior terms (italic) in the lexicon.

3. POS tagging: each word is tagged with its part of speech. Consider sentence 3 as an example. The words are tagged as follows. The naming conventions of the tags can be found in the Penn Treebank tag set [27].

a/DT second/JJ DC/NN motor/NN turns/VBZ a/DT pulley/NN that/TDT allows/VBZ the/DT turret//NN to/TO rotate/VB 360/CD degrees/NNS clockwise/RB or/CC counterclockwise/RB

4. Reference resolution: the pronoun (that) in the noun clause is replaced by the subject it refers to (pulley). The sentence is separated into two sentences.

Sentence 3.1: a/DT second/JJ DC/NN motor/NN turns/VBZ a/DT pulley/NN

Sentence 3.2: pulley/NN allows/VBZ the/DT turret/NN to/TO rotate/VB 360/CD degrees/NNS clockwise/RB or/CC counterclockwise/RB

5. Semantic analysis: sentence constituents are grouped into nouns, verbs, prepositional phrases, etc. The algorithm further analyzes their surroundings to map from the sentence constituents to the concepts and relationships in the ontology model:

DC motor→ structure concept

Pulley → structure concept

Turret → structure concept

360 degrees → attribute value concept, is_value_of → *rotate_angle_range* (is_attribute_of → rotate)

Clockwise → attribute value concept is_value_of → *rotate_orientation* (is_attribute_of → *rotation*)

Counter clockwise → attribute value concept is_value_of → *rotate_orientation* (is_attribute_of → *rotation*)

DC motor (has_function_of) → <u>turn</u> → **pulley**

Pulley (has_function_of) → <u>drive</u> → **turret**

Turret (has_function_of) → <u>rotate</u>

There is no prepositional phrase in sentence 3.

9. DISCUSSION AND FUTURE WORK

In this paper, we assume the input documents are in proper English; in practice, however, errors and bias exist in many subjective descriptions. Design engineers may neglect the descriptions of some important aspects of the design. For

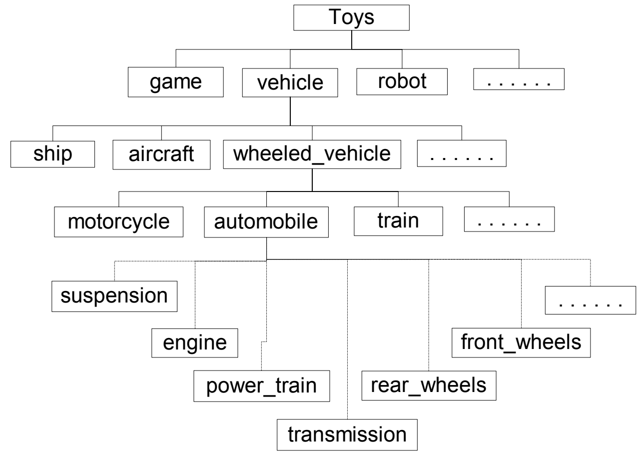

Figure 8. ACTION TOY ONTOLOGY

example, in the initial design document of the knowledge descriptions shown in Fig. 4, as well as in the example in Section 8, there are no descriptions about how the wheels of the prototype function, what their mechanical behaviors are, and what the structures are. The integration of the design knowledge annotation based on design documents as well as 3D CAD models has the potential capability of producing a more complete descriptions and deeper analysis as well. Reader can refer to [15] for an initial study about design knowledge representation and annotation for 3D CAD models based on structure analysis.

We also found in the case study that it would be helpful for retrieval purposes if the documents could be classified according to some pre-coded classification schema. For example, in the domain-specific design knowledge ontology, all the projects which are related with automobiles can be grouped together under the concept of automobile rather than the concept of toy, i.e. the concept of automobile will be the sub-concept of toy. To achieve this goal, a more general domain-specific ontology model with an abstract description of each product concept is needed. This ontology will be located in the application layer of the FBSO model. It is basically a domain concept classification schema with an “is_a” relationship between the non-leaf nodes. We only need one level of the leaf nodes to specify the composition (i.e. is_part_of relationship) of their parent concept. An example of the ontology is shown in Fig. 8. The concepts in the leaf node are used to match with the metadata generated from the document analysis. The scoring metric is the same as the one described in Section 6. The concept in the parent node with the maximum NOHs and the minimum CD is selected as the concept category which the submitted design belongs to. The concept is automatically added onto the domain-specific design knowledge ontology if it does not exist yet. Its sub-concept is represented as the name of the design, such as the mobile_rocket_launcher.

10. CONCLUSION

We reported here an approach to the management of domain-specific knowledge specialization, namely, a framework for practical knowledge representation and retrieval that aims at being effective, adaptive, and efficient with respect to the concepts and relationships encountered in unstructured textual design knowledge resources. The centerpiece of our method is the layered ontology model, where the domain-specific design knowledge ontology is automatically constructed using natural language processing based on the application ontology. The method detects the mappings between the linguistic patterns and the domain semantics. Query intents are discovered by concept disambiguation and concept abstraction. Personalized query interface makes the system adaptive to the user's interests.

Our approach requires minimal pre-coding: the generic ontology models and the application ontology models are all reusable modules. Some of them make use of the current linguistic ontology model.

The automatic domain ontology acquisition method is another important contribution to the research into engineering design and knowledge management. Besides the design knowledge regarding function, behavior, and structure, the contents of the application ontology can be extended to other applications in the product lifecycle, such as customer requirement analysis and cost estimation.

The input documents provided by the engineering design class are written in proper English with moderate linguistic complexities. We plan to test our model in the future with a larger set of documents from industries.

ACKNOWLEDGMENTS

Our special thanks go to Professor Victor Raskin, Department of English, Purdue University, for his insightful suggestions in ontology and natural language processing. We would also like to thank the members of PRECISE Lab: Natraj Iyer and Subramaniam Jayanti, for their valuable comments. Natraj Iyer also provided the design documents as test data.

REFERENCES

[1] Marsh, J. R., 1997, "The Capture and Utilization of Experience in Engineering Design," PhD. Thesis, Cambridge University, Cambridge, UK.

[2] Ahmed, S. and Wallace, K., 2003, "Reusing Design Knowledge," 14th CIRP Design Seminar, Cairo, Egypt.

[3] Ullman, D. G., 1997, *The Mechanical Design Process*, 2nd ed., McGraw-Hill, New York, USA.

[4] Dixon, L. A. and Colton, J. S., 1998, "An Anchoring Adjustment Process Model for Redesign," Journal of Engineering Design, **9**, pp. 297-314.

[5] Busby, J. S., 1999, "The Problem with Design Reuse: An Investigation into Outcomes and Antecedents," Journal of Engineering Design, **10**, pp. 277-296.

[6] Hmelo-Silver, C. E. and Pfeffer, M. G., 2004, "Comparing Expert and Novice Understanding of a Complex System from the Perspective of Structures, Behaviors, and Functions," Cognitive Science, **28**, pp. 127-138.

[7] Baya, V, Gevins, J, Baudin, C, Mabogunje, A, Leifer, L., and Toye, G., 1992, "An Experimental Study of Design Information Reuse," Proceedings of the ASME 4th International Conference on Design Theory and Methodology, Scottsdale, AZ, USA, Vol. 42, pp. 141-147.

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

[9] Chovan, J. D. and Waldron, M. B., 1990, "The Use of Function and Form when Reading Mechanical Engineering Drawings by Expert and Non-experts," Proceedings of the ASME Design Theory and Methodology Conference, Chicago, IL, USA, pp. 137-144.

[10] Hwang, C. H., 1999, "Incompletely and Imprecisely Speaking: Using Dynamic Ontologies for Representing and Retrieving Information," Technical Report MCC-INSL-043-99, Microelectronics and Computer Technology Corporation, Austin, TX, USA.

[11] Nirenburg, S. and Raskin, V., 2003, *Ontological Semantics*, MIT Press, Cambridge, MA, USA.

[12] Ciocoiu, M., Nau, D. S., and Gruninger, M., 2001, "Ontologies for Integrating Engineering Applications," Journal of Computer Information Science in Engineering, **1**, pp. 12-22.

[13] Studer, R., Benjamins, V.R., and Fensel, D., 1998, "Knowledge Engineering: Principles and Methods," Data and Knowledge Engineering, **25**, pp. 161-197.

[14] The Center for Education and Research in Information Assurance and Security (CERIAS): Knowledge Base Acquisition Editor, Http//:kbae.cerias.purdue.edu:443/.

[15] Li, Z., Anderson, D. C., and Ramani, K., 2005, "An Initial Study of Ontology-based Design Knowledge Modeling for Product Retrieval and Reuse," International Conference on Engineering Design, ICED 05, Melbourne, Australia.

[16] Brill, E. and Mooney, R. J., 1997, "An Overview of Empirical Natural Language Processing," AI Magazine, **18**, pp. 13-24.

[17] 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 magazine, **19**, pp. 59-71.

[18] Baeza, R. and Neto, B., 1999, *Modern Information Retrieval*, Addison Wesley, New York, USA.

[19] Cowie, J., and Lehnert, W., 1996, "Information Extraction," Communications of the ACM, **39**, pp. 80-91.

[20] Wartick, S., 1992, "Boolean Operations, in Information Retrieval: Data Structures and Algorithms," edited by Frakes, W. B. and Baeze-Yates, R., Prentice Hall, Englewood Cliffs, NJ, USA, pp. 264-292.

[21] Salton, G., 1989, *Automatic Text Processing*, Addison Wesley.

[22] Fuhr, N., 1992, "Probabilistic Models in Information Retrieval," The Computer, **35**, pp. 243-255.

[23] Daelemans, W., Zavrel, J., Berck, P., and Gillis, S., 1996, "MBT: A Memory-based Part Of Speech Tagger Generator," Proceedings of 4th Workshop on Very Large Corpora, edited by Ejerhed, E. and Dagan, I., pp. 14-27.

[24] Jurafsky, D. and Martin, J. H., 2000, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice-Hall, Upper Saddle River, NJ, USA.

[25] Khan, L., McLeod, D, and Hovy, E., 2004, "Retrieval Effectiveness of an Ontology-based Model for Information Retrieval," VDLB Journal, **13**, pp. 71-85.

[26] Liu, J., Wong, C. K., and Hui, K. K., 2003, "An Adaptive User Interface Based on Personalized Learning," IEEE Intelligent Systems, **18**, pp. 52-57.

[27] Marcus, M., Santorini, B., and Marcinkiewicz., M. A., 1994, "Building a Large Annotated Corpus of English: The Penn Treebank," Computational Linguistics, **19**, pp.313–330.

