Representation and Reasoning of Empirical Engineering Knowledge

Zuhua Jiang, Bo Song, and Geng Li

***Abstract*—On the basis of analysis and review of the existing methods and problems of engineering experience knowledge acquisition, this paper proposes a method for the representation and reasoning of empirical engineering knowledge based on ontology, and a three-level architecture for acquiring knowledge from natural language is proposed, which makes use of unstructured knowledge bases to retrieve knowledge. The types of the attributes of empirical engineering knowledge are marked with the OWL language tags for implementation. The process of applying empirical engineering knowledge reasoning to solve potential problems is introduced.**

***Index Terms*—Empirical engineering knowledge, knowledge reasoning, knowledge representation.**

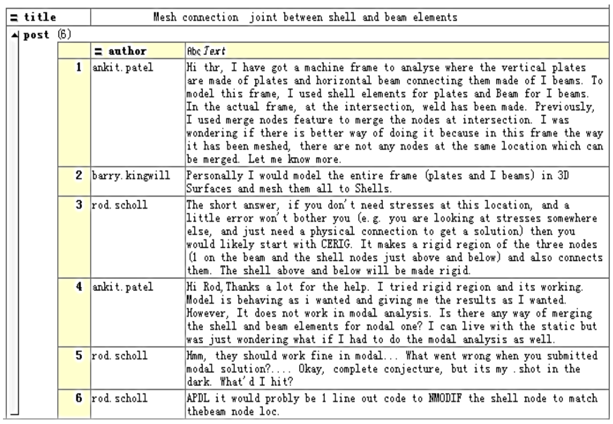
I. Introduction

Knowledge management and reuse have helped people reduce the time and effort for getting a solution, thus improving the capability, efficiency and creativity of both organizations and individuals. However, it is still insufficient for the current research to utilize the empirical engineering knowledge that is hard to be expressed by standard ways. As a popular way of conceptual association and organization, ontology is applied to the representation of empirical knowledge. Chen [1] designed a representation of empirical knowledge based on ontology and suggested a set of empirical knowledge inference rules regarding the four levels of empirical knowledge which are know-what, know-why, know-how and know-with. Chen realized this representation in the framework of OWL. Based on the natural language processing and domain ontology, Li [2] extracted concepts and the relations between concepts from the design documents and created the Concept Graph which can be applied to improving the retrieval performance of design information. Kamsu-Foguen *et al.* also applied the Concept Graph [3] for the description of empirical knowledge. Noh *et al*. combined Cognitive Map and Case Based Reasoning in their framework of reusing tacit knowledge, where a cognitive map which reflects the mutual influence between concepts [4]. Castro-Schez *et al*. used fuzzy rules to represent and acquire experiential knowledge [5]. Potes Ruiz *et al.* used Concept Map and rules to represent experiential knowledge [6]. More recently, Azmi and Adriman [7] proposed to use Repertory Grid Technique to acquire tacit knowledge. Chen and Luh [8] proposed to use practice, observation, and comments to speed up the process of tacit knowledge acquisition in freehand sketching learning.

It can be seen from existing research that there has been no certain way to represent and reason with empirical engineering knowledge. The incompleteness and uncertainty in empirical engineering knowledge prevent a systematic and formal representation of this kind of knowledge. In this paper, we use questions and answers (Q&A) as the source of empirical engineering knowledge, and propose an ontology-based, three-level architecture for representing, acquiring and reasoning with empirical engineering knowledge.

II. Sources of Engineering Experience Knowledge

Fig. 1 shows the Q&A for a certain topic acquired from the website XANSYS. XANSYS is a forum for users of ANSYS, a finite element analysis software, to exchange their problem-solving experience. Users of the forum must provide their real name, e-mail address and the name of their enterprise. The majority of topics in the forum are presented in Q&A format, which is generally a serious and professional discussion for specific cases. Participants rather than the issuer of the question will basically answer the question directly instead of throwing a new question, but the forum does not provide the function of scoring answers and marking the best answers. Currently, there are about 16,000 topics and 55,000 posts on XANSYS, and the English CAE forums similar to XANSYS include iMechanica, Physics Forum/Mechanical Engineering, Polymer FEM, etc.



title	Mesh connection joint between shell and beam elements

post (6)

	author	Text
1	ankit.patel	Hi thr, I have got a machine frame to analyse where the vertical plates are made of plates and horizontal beam connecting them made of I beams. To model this frame, I used shell elements for plates and Beam for I beams. In the actual frame, at the intersection, weld has been made. Previously, I used merge nodes feature to merge the nodes at intersection. I was wondering if there is better way of doing it because in this frame the way it has been meshed, there are not any nodes at the same location which can be merged. Let me know more.
2	barry.kingwill	Personally I would model the entire frame (plates and I beams) in 3D Surfaces and mesh them all to Shells.
3	rod.scholl	The short answer, if you don't need stresses at this location, and a little error won't bother you (e.g. you are looking at stresses somewhere else, and just need a physical connection to get a solution) then you would likely start with CERIG. It makes a rigid region of the three nodes (1 on the beam and the shell nodes just above and below) and also connects them. The shell above and below will be made rigid.
4	ankit.patel	Hi Rod,Thanks a lot for the help. I tried rigid region and its working. Model is behaving as i wanted and giving me the results as I wanted. However, It does not work in modal analysis. Is there any way of merging the shell and beam elements for nodal one? I can live with the static but was just wondering what if I had to do the modal analysis as well.
5	rod.scholl	Hmm, they should work fine in modal... What went wrong when you submitted modal solution?.... Okay, complete conjecture, but its my .shot in the dark. What'd I hit?
6	rod.scholl	APDL it would probly be 1 line out code to NMODIF the shell node to match thebeam node loc.

Fig. 1. Q&A containing empirical engineering knowledge.

III. Representation of Empirical Engineering Knowledge

Empirical engineering knowledge is a tetrad EEK = < C, P, Rcl, Rdf >, where C is the concept cluster of Empirical Engineering knowledge; P is the attribute set, including the

Manuscript received February 9, 2019; revised October 12, 2019.

Zuhua Jiang and Geng Li are with the Department of Industrial Engineering and Management, Shanghai Jiao Tong University, Shanghai, China (e-mail: zhjiang@sjtu.edu.cn, reaganli@sjtu.edu.cn).

Bo Song is with China Institute of FTZ Supply Chain, Shanghai Maritime University, Shanghai, China (e-mail: bosong@shmtu.edu.cn).

doi: 10.18178/ijke.2019.5.2.117

object properties that reflect the relations between concepts and the data properties that reflect the features of concept itself; Rcl is the rule set for classic reasoning defined on C and P; Rdf is the set of default rules extracted from the engineering Q&A. The three items C, P and Rcl constitute the ontology of EEK. Fig. 2 shows the attributes in the concept of EEK.

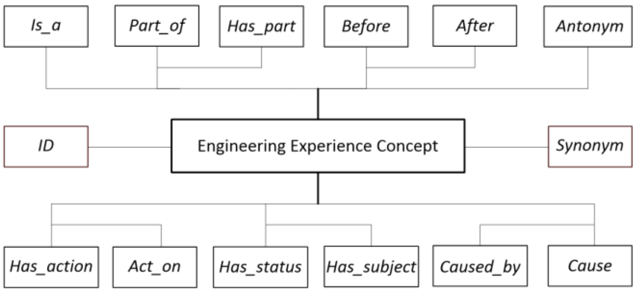


Fig. 2. Concept and its attribute relation of the empirical engineering knowledge.

The implications of the attributes are as follows, the types of the attributes are marked with the OWL language tags for implementation:

- ID: Data property (owl: DatatypeProperty) which is applied for the unique marking of every concept instance (owl: NamedIndividual). In the ontology of EEK established in this article, a concept of EEK that has been confirmed by the domain experts is defined as a class (owl: Class), the appearance of concept in specific case corresponds to the instance of the concept. The Volume, for instance, is a concept for the expression of a three-dimensional entity. Every time when volume or its synonym is encountered in a post for the discussion of an issue, a type (rdf: type) shall be established as the concept instance of Volume and offered with the unique value that can be applied for the accurate positioning of its source as its ID attributes, just like the website address added with the ordinal number for the appearance of the word.
- Is_a：Relationship of class and parent class (rdfs: subClassOf).
- Part_of: Object property that reflects the relationship of part and whole (owl:ObjectProperty).
- Has_part: Object property that reflects the relationship of part and whole, it is the inverse property of Part_of (owl:inverseOf).
- Has_action: Object property that reflects the movement of concept, its domain (rdfs:domain) is noun concept, its range (rdfs:range) is verbal concept.
- Act_on: Attribute that reflects the movement and its executor, it is the reverse property of Has_action.
- Has_status: Object property that reflects the status of concept, its domain is noun concept, its range is adjective concept.
- Has_subject: Attribute that reflects the status and its attached main body, it is the reverse property of Has_status.
- Synonym: Data property of character string, it lists the synonym for every concept.
- Antonym: Object property with the concepts being antonym to each other, for example *Antonym (fine mesh, coarse mesh)*, the attribute of Antonym is symmetrical (owl: SymmetricProperty).
- Cause: Object property that reflects the relationship of cause and effect, for example *Cause (divide volume, multiple volume)* indicates that the separation of solid mass may generate multiple solid masses, another example, *Cause (fine mesh, accurate result)* indicates that the fine segregation of mesh is likely to receive accurate result.
- Caused_by: Object property that reflects the relationship of cause and effect, it is the reverse property of Cause.
- Before: Object property that reflects the sequential order between movements, for example, *Before (build model, apply load)* indicates that the establishment of model is commonly conducted before the infliction of load.
- After: Object property that reflects the sequential order between movements, it is the reverse property of Before.

The concept of empirical knowledge is divided into three levels (Fig. 3), namely Nominal Concept, Status Concept and Action Concept, which correspond to nouns, adjectives and verbs parsed from the text. The concept of a nominal concept is a basic concept defined in an engineering field, such as materials, models, and loads in the field of finite element analysis. And these abstract concepts form a tree skeleton of EEK ontology through Is_a relationship. The concept of status is a common adjective used to modify the concept of name class, such as big, small, fast and slow, but it is difficult for an adjective term to help people judge the characteristics of Engineering problems, so a concept of status must be assigned with a Has_subject property and the corresponding value during its establishment. Similarly, a concept of action represented by a verb also needs to be specified the value of the Act_on property for each instance.

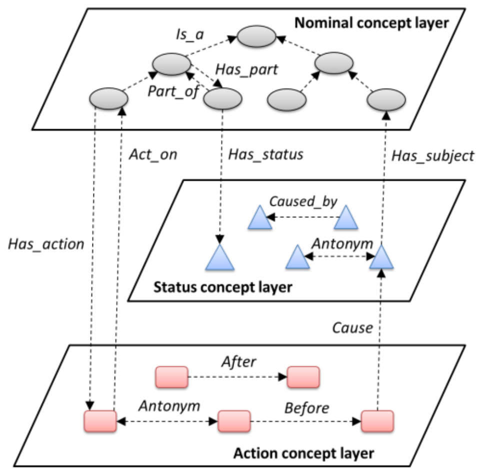


Fig. 3. Different levels of engineering empirical knowledge concepts.

With the concepts and attributes, we need to define a set of inference rules to reasoning out novel facts based on the ontology of empirical knowledge, which is Rcl of the tetrad <C, P, Rcl, Rdf>. In the following passage *X* in capital denotes the concept of the name *X*, it takes the question mark with the lowercase ?*x* to indicate any instance. *X* (?*x*) indicates that ?*x* is the instance of *X*, then the rules included in Rcl can be written as follows:

1) $X(?x) \wedge \text{Is\_a}(X, Y) \Rightarrow Y(?x)$
2) $\text{Part\_of}(?x, ?y) \wedge \text{Part\_of}(?y, ?z) \Rightarrow \text{Part\_of}(?x, ?z)$
3) $\text{Has\_action}(?x, ?y) \wedge \text{Part\_of}(?x, ?z) \Rightarrow \text{Has\_action}(?z, ?y)$

4) Has_ status (?*x*, ?*y*)∧ Part_of (?*x*, ?*z*)=> Has_status (?z, ?*y*)
5) Antonym (?*x*1, ?*y*1)∧ *X* (?*x*1)∧ *Y* (?*y*1)∧ *X* (?*x*2)∧ *Y* (?*y*2)∧ Has_subject (?*x*2, ?*z*1)∧ Has_subject (?*y*2, ?*z*2) ∧ *Z* (?*z*1, ?*z*2)=> Antonym (?*x*2, ?*y*2)
6) Antonym (?*x*1, ?*y*1)∧ *X* (?*x*1)∧ *Y* (?*y*1)∧ *X* (?*x*2)∧ *Y* (?*y*2)∧ Act_on (?*x*2, ?*z*1)∧ Act_on (?*y*2, ?*z*2)∧ *Z* (?*z*1, ?*z*2) => Antonym (?*x*2, ?*y*2)
7) Act_on (?*x* , ?*y*)∧ Cause (?*x*, ?*z*)∧ Has_subject (?*z*, ?*s*) ∧ Act_on (?*x*1, ?*y*1)∧ *X* (?*x*, ?*x*1)∧ *Y*(?*y*, ?*y*1)=> *Z* (?*z*, new ?*z*1)∧ *S* (?*s*, new ?*s*1)∧ Has_status (?*s*1, ?*z*1)
8) Has_status (?*x*1, ?*y*1)∧*X* (?*x*1, ?*x*2)∧*Y* (?*y*1, ?*y*2)∧ Cause (?*y*2, ?*z*1)∧ Has_subject (?*y*2, ?*x*2)∧ *Z* (?*z*1)∧ Has_subject (?*z*1, ?*k*1)∧ *K* (?*k*1)=> *Z* (new ?*z*2)∧ *K* (new ?*k*2)∧ Has_status (?*k*2, ?*z*2)
9) Before (?*x*, ?*y*)∧ Before (?*y*, ?*z*)=> Before (?*x*, ?*z*)

The process of diagnosing problems and giving answers from known facts belongs to the category of empirical tacit knowledge, and its reasoning process is non-monotonous, therefore we construct a set of default rules Rdf to store the empirical tacit knowledge obtained from the engineering Q&A. A default rule that determines knowledge invocation should has the following form:

$$\frac{\text{INCLUDE}(\gamma,\beta)\wedge\text{ANSWER}(\alpha,\beta):\text{CONSISTENT}(\alpha,\gamma)}{\text{SOLVE}(\alpha,\gamma)} \quad (1)$$

where α, β, $\gamma \in L$. L is a set of sentences, a sentence is composed of atomic propositions or atomic propositions connected by logical connectors ∧, ∨, ¬ (and, or, and NOT). In this paper, sentences are composed of relationships between concept instances in ontology. The implication of each mark in (1) is as follows:

- γ: The set of all known facts, including the basic facts of the domain declared when creating ontology, the facts of the user's current task environment obtained through context awareness, and the facts inferred from the rule set Rcl;
- β: A set of facts in a historical engineering problem, obtained from the problem description text of the Q&A;
- α: A set of facts in an answer, obtained from the answer description text of the Q&A;
- INCLUDE (γ, β): This condition is satisfied when γ contains all the facts in β. Since β describes a certain situation that occurred in the past, the facts in β are not involved in γ for reasoning. When checking whether γ contains β or not, if every conjunction of β is included in γ (assuming that both of them contain only conjunctive terms), then INCLUDE (γ, β) will be regarded as valid;
- ANSWER (α, β): α is the answer to β in a pair of Q&A.
- CONSISTENT (α, γ): α does not contain the facts that conflict with γ, that is, when the facts in α are declared, the system does not derive a new Antonym relationship;
- SOLVE (α, γ): When INCLUDE (γ, β), ANSWER (α, β), and CONSISTENT (α, γ) are all satisfied, then α is considered to be helpful for solving γ.

Since γ is derived from the task environment, the stored elements of the default rules in the knowledge base only include α and β.

IV. Inference Process of Empirical Engineering Knowledge

Fig. 4 shows the process of applying empirical engineering knowledge reasoning to solve potential problems in engineering task environments.

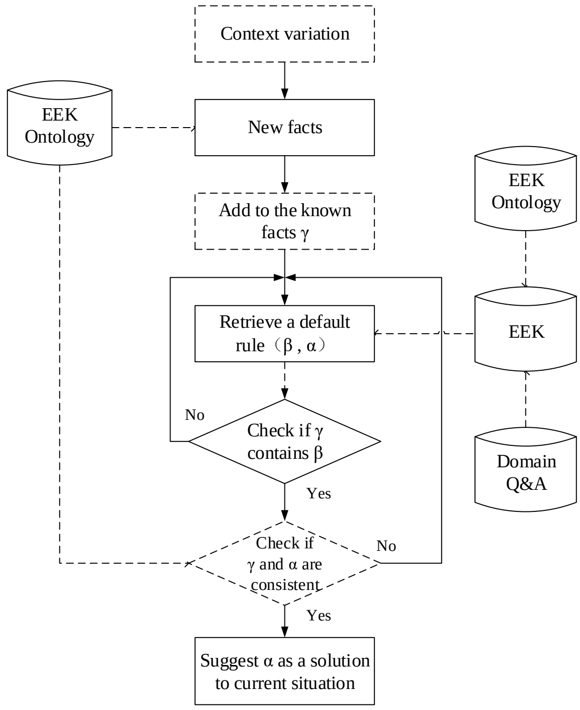


Fig. 4. The reasoning procedure of EEK.

- Context variation. The information obtained by sensor for the task processing environment and schedule variance of engineers, for example, the opening of new document, the receiving and sending of emails, the operation of specific software and the input of keyboard etc.
- New facts. Based on the original context information, the generation of new facts refers to form new concept instances and relation assertions with the integration of ontology concept mapping and ontology reasoning.
- Facts fusion. The fusion combines the facts newly generated in the previous step with the established facts in the engineering task environment to generate the default reasoning element γ, it mainly involves the oblivion for the old concept instance and the solution for the conflict of facts.
- Defaults retrieval. This phase extracts default rules from the empirical knowledge base that may be applicable to the current task environment. Due to the large quantity of default rules obtained from the Q&A, the item by item examination of its usability with the method of logical inference will cost large amount of time, thus the rule retrieval is added to reduce the candidate default rule sets by keyword matching.
- Prerequisite check. The prerequisite check of default rules checks whether the prerequisite β of a default rule is satisfied by γ. If satisfied, a conclusion sentence α corresponding to β is taken out, and if not, the next default rule is taken out for check.
- Justification check. Justification check of default rules, check the conclusion α to be exported for consistency with γ. If it conflicts with γ, then take out the next α

corresponding to β in the previous step, if there is no conflict, then enter the next step.

- Solution suggestion. A conclusion α passing the above two examination steps will be recommended to engineers in the corresponding task environment.

V. Conclustion

This paper proposes the definition, representation and reasoning mechanism of empirical engineering knowledge. In order to reflect both the explicit and tacit aspects of EEK, we suggest a formal representation of the knowledge based on ontology and a non-monotonous reasoning mechanism using default rules. The three-level architecture of the proposed ontology encodes concepts of object class, object status and object action that are commonly seen in the engineering field which brings convenience for the acquisition of EEK from natural language. A default rule represents the tacit knowledge extracted from engineering Q&A that signifies the perception of an answer provider of when to use what knowledge. It support non-monotonic reasoning with the prerequisites and consistency assumptions that are essential to knowledge application.

Acknowledgment

This work was supported by National Natural Science Foundation of China (No. 71671113, 71601113).

References

[1] Y. J. Chen, "Development of a method for ontology-based empirical knowledge representation and reasoning," *Decision Support Systems,* vol. 50, no. 1, pp. 1-20, 2010.

[2] Z. J. Li and K. Ramani, "Ontology-based design information extraction and retrieval," *Ai Edam-Artificial Intelligence for Engineering Design Analysis and Manufacturing,* vol. 21, no. 2, pp. 137-154, 2007.

[3] B. K. Foguem, T. Coudert, and C. Béler, "Knowledge formalization in experience feedback processes: An ontology-based approach," *Computers in Industry,* vol. 59, no. 7, pp. 694-710, 2008.

[4] J. B. Noh, K. C. Lee, and J. K. Kim, "A case-based reasoning approach to cognitive map-driven tacit knowledge management," *Expert Systems with Applications,* vol. 19, no. 4, pp. 249-259, 2000.

[5] J. J. Castro-Schez, J. M. Murillo, and R. Miguel, *et al.*, "Knowledge acquisition based on learning of maximal structure fuzzy rules," *Knowledge-Based Systems,* vol. 44, pp. 112-120, 2013.

[6] R. P. Potes, F. B. Kamsu, and B. Grabot, "Generating knowledge in maintenance from experience feedback," *Knowledge-Based Systems,* vol. 68, pp. 4-20, 2014.

[7] A. Azmi and R. Adriman, "Marketing practitioner's tacit knowledge acquisition using Repertory Grid Technique (RTG)," in *Proc. IOP Conference Series: Materials Science and Engineering,* 2018, vol. 352.

[8] S. N. Chen and D. B. Luh, "A social network supported CAI model for tacit knowledge acquisition," *International Journal of Technology and Design Education,* vol. 28, no. 3, pp. 1-18, 2017.

Zuhua Jiang is currently a full professor in the Department of Industrial Engineering and Management, Shanghai Jiao Tong University. His research works include knowledge management, preventive maintenance, ergonomics, and logistic planning.

Song Bo is currently a lecturer in Supply Chain Research Institute of China (Shanghai) Free Trade Zone, Shanghai Maritime University. His research works include knowledge management, knowledge engineering, artificial intelligence and data mining.

Geng Li is currently a postgraduate student in the Department of Industrial Engineering and Management, Shanghai Jiao Tong University. His research works include knowledge management and knowledge evolution.