

# Knowledge Engineering-an overview

<sup>1</sup>D.S Darai, <sup>2</sup>S Singh, <sup>3</sup>S Biswas

*Department of Information Technology, SOA University*

<sup>1</sup>[deepikal505@gmail.com](mailto:deepikal505@gmail.com), <sup>2</sup>[shravanisingh21@gmail.com](mailto:shravanisingh21@gmail.com), <sup>3</sup>[sitanath\\_biswas2006@yahoo.com](mailto:sitanath_biswas2006@yahoo.com)

**Abstract**— Knowledge Engineering is an important concept in recent years. Knowledge Engineering is the aspect of system engineering which addresses uncertain process requirements by emphasizing the acquisition of knowledge about a process and representing this knowledge in a Knowledge-based System. The purpose of this paper is to discuss how to use the basic principles of knowledge engineering to solve the real-world problems.

**Keywords**—Knowledge; Knowledge Engineering; Principles; Knowledge Acquisition.

## I. INTRODUCTION

In the fast changing information world of today, the nature of global economic growth has been changed by the development of science and technology, which has been made possible by rapidly evolving communication models, shorter product lifecycles and a higher rate of new product development. Knowledge engineering is a field within artificial intelligence that develops knowledge-based systems. Such systems are computer programs that contain large amounts of knowledge, rules and reasoning mechanisms to provide solutions to real-world problems.

The early years of knowledge engineering were dogged by problems. Knowledge engineers found that acquiring enough high-quality knowledge to build a robust and useful system was a very long and expensive activity. As such, knowledge acquisition was identified as the bottleneck in building an expert system. This led to knowledge acquisition becoming a major research field within knowledge engineering. The aim of knowledge acquisition is to develop methods and tools that make the arduous task of capturing and validating an expert's knowledge as efficient and effective as possible. Experts tend to be important and busy people; hence it is vital that the methods used minimize the time each expert spends off the job taking part in knowledge acquisition sessions.

A major form of knowledge-based system is an expert system, one designed to emulate the reasoning processes of an expert practitioner. Typical examples of expert systems include diagnosis of bacterial infections, advice on mineral exploration and assessment of electronic circuit designs.

At present, it refers to the building, maintaining and development of knowledge-based system. It has a great deal in common with software engineering and is used in many computer science domains such as artificial intelligence including database, data mining, expert system, decision support system and geographic information system. Knowledge engineering is also related to mathematical logic, as well as strongly involved in cognitive science and socio-cognitive engineering where the knowledge is produced by socio-cognitive aggregates (mainly humans) and is structured according to our understanding of how human reasoning and logics works. Thus Knowledge Engineering has become a major technique for information integration. And, last but not least the fast growing World Wide Web generates an ever increasing demand for more efficient knowledge exploitation and creation techniques.

## II. KNOWLEDGE AND KNOWLEDGE ENGINEERING

### A. Knowledge

Knowledge is defined (Oxford English Dictionary) variously as (i) expertise, and skills acquired by a person through experience or education; the theoretical or practical understanding of a subject, (ii) what is known in a particular field or in total; facts and information or (iii) awareness or familiarity gained by experience of a fact or situation .

Knowledge acquisition involves complex cognitive processes: perception, learning, communication, association and reasoning. The term knowledge is also used to mean the confident understanding of a subject with the ability to use it for a specific purpose. By definition, knowledge can be divided into two kinds: tacit and explicit knowledge.

Tacit knowledge is knowledge that people carry in their minds and is, therefore, difficult to access. Tacit knowledge is considered more valuable because it provides context for people, places, ideas, and experiences. Effective transfer of tacit knowledge generally requires extensive personal contact and trust .Tacit knowledge is not easily shared. Tacit knowledge consists often of habits and culture that we do not recognize by ourselves.

On the other hand Knowledge that is easy to communicate is called explicit knowledge. The process of transforming tacit knowledge into explicit knowledge is known as codification or articulation. Explicit knowledge is knowledge that has been or can be articulated, codified, and stored in certain media. The most common forms of explicit knowledge are manuals, documents and procedures.

Knowledge also can be audio visual, works of art and product design can be seen as other forms of explicit knowledge where human skills, motives and knowledge are externalised.

### B. Knowledge Engineering

**Knowledge engineering (KE)** was defined in 1983 by Edward Feigenbaum and Pamela McCorduck as follows: KE is an engineering discipline that involves integrating knowledge into computer systems in order to solve complex problems normally requiring a high level of human expertise. Knowledge is a vital component of engineering design. Significant reductions in costs and development time can be made if knowledge is captured from specialists and stored in a knowledge base. The contents of knowledge base can be used in a number of ways: (i) to disseminate knowledge to other people in an organization; (ii) to re-use knowledge in different ways for different purposes; (iii) to use knowledge to develop intelligent systems that can perform complex design tasks. Although the benefits of capturing and using knowledge are manifest, it has long been recognized in the artificial intelligence (AI) community that knowledge is hard to acquire from specialists. First, specialists are not good at recalling and explaining ‘tacit knowledge’ which operates at a subconscious level and is thus hard, if not impossible the difficulties stem from a number everything they know. They have what is termed, to explain. Second, specialists have different experiences and opinions that require aggregating to provide a single coherent picture. Third, specialists develop particular conceptualizations and mental shortcuts that are not easy to communicate. To deal with such difficulties, the field of Knowledge Engineering was born some 25 years ago, and the role of the knowledge engineer was created. Since that time, knowledge engineers have developed a number of principles, methods and tools that have considerably improved the process of acquiring, using and implementing knowledge.

## III. PRINCIPLES OF KNOWLEDGE ENGINEERING

*A. Principle 1: “Recognize that there are different types of knowledge.”*

Philosophers have been thinking about knowledge for thousands of years. Part of their endeavor has been the identification of different types of knowledge. One distinction is between declarative knowledge and procedural knowledge. A popular way of thinking about these is the difference between ‘knowing that’ and ‘knowing how’. In knowledge engineering, these two types are often referred to as static knowledge and dynamic knowledge: .Another well-known classification of knowledge is that of tacit knowledge and explicit knowledge. A particularly important way of classifying knowledge is to what extent it is abstract or specific. Developing ways in which specific knowledge can be made more abstract and abstract knowledge can be made more specific has been a major effort in knowledge engineering. The field of logic has also inspired other

important knowledge types: concepts, attributes and values. For instance, within the statement ‘it is a fast car’, we refer to ‘car’ as being a concept, and ‘fast’ as being a value of the attribute speed. Concepts, attributes and values can be linked together to form another important class of knowledge rules.

*B. Principle 2: “Recognize that there are different types of experts – and expertise.”*

Not only are there different types of knowledge, but there are different types of experts. The Psychologists and knowledge engineers have found that experts vary in how well they can articulate their knowledge. In addition to differences in ability to articulate knowledge, experts vary in how well they recall information in a given context. Studies in psychology have repeatedly shown that experts are not able to remember the same things during interviews as they can when they are performing a task. In addition, the ability to recall the same information in different tasks can vary between individuals. Again, people may vary in the extent to which they do this, and it is important to recognize this. Knowledge engineers have devised a number of methods to overcome such problems, such as aggregating knowledge from various sources and validating knowledge across sources.

*C. Principle 3: “Recognize that there are different ways of representing knowledge.”*

The field of artificial intelligence may not have produced fully intelligent machines, but one of its major achievements is the production of a range of ways of representing knowledge. However, producing different knowledge representations is a vital part of artificial intelligence, since the ease of solving a problem is almost completely determined by the way the problem is conceptualized and represented. A well-chosen analogy, anecdote or diagram can make all the difference.

*D. Principle 4: “Recognize that there are different ways of using knowledge.”*

People use knowledge in different ways depending on the task they are performing. From a knowledge perspective, it is useful to be able to classify tasks. One classification, adapted and refined from ideas in psychology, is a hierarchy of knowledge-intensive tasks based on the type of problem being solved. Knowledge engineers have created models that define the types of knowledge that form the inputs and outputs of the task, as well as how the knowledge is transformed for each of the tasks. Using these models, the knowledge involved in a task can be defined in terms of the way it is used to satisfy a goal or set of goals. This can not only increase the efficiency of task analysis and process modeling, but also enables one to identify how the same piece of knowledge is used differently depending on the context in which it is used.

### E. Principle 5: "Use structured methods."

This principle leads on from the first four. There are different types of knowledge, different types of experts, different ways of representing knowledge and different ways of using knowledge. Therefore, it needs a way of relating these types of knowledge, experts, representations and tasks together to perform a knowledge-oriented activity. To do this, requires well-defined methods: methods that are structured such that the right techniques and tools are used depending on the situation and, especially, the goals involved. For that we need a well range of technique and software tools.

## IV. KNOWLEDGE ENGINEERING as a MODELLING PROCESS

Nowadays there exists an overall consensus that the process of building a KBS may be seen as a *modeling activity*. Building a KBS means building a computer model with the aim of realizing problem-solving capabilities comparable to a domain expert. It is not intended to create a cognitive adequate model, but to create a model which offers similar results in problem-solving for problems in the area of concern. This knowledge is not directly accessible, but has to be built up and structured. The modeling view of the building process of a KBS has the following consequences:

- i) Like every model, such a model is only an *approximation* of the reality. In principle, the modeling process is infinite, because it is an incessant activity with the aim of approximating the intended behavior.
- ii) The modeling process is a *cyclic* process. New observations may lead to a refinement, modification, or completion of the already built-up model.
- iii) The modeling process is dependent on the subjective interpretations of the knowledge engineer. Therefore this process is typically *faulty* and an evaluation of the model with respect to reality is indispensable for the creation of an adequate model.

## V. Problem-Solving Methods and Ontologies

### A. Problem-Solving method

Knowledge-based systems are computer systems that deal with complex problems by making use of knowledge. This knowledge is mainly represented declaratively rather than encoded using complex algorithms. This declarative representation of knowledge economizes the development and maintenance process of these systems and improves their understandability. Therefore, knowledge-based systems originally used simple and generic inference mechanisms. Making the control knowledge explicit and regarding it as an important part of the entire knowledge contained by a knowledge-based system is the rationale that underlies *Problem-Solving Methods (PSMs)*. PSMs refine the generic inference engines mentioned above to allow a more direct control of the reasoning process. Since this control knowledge is specified independently from the application domain, *reuse* of this strategic knowledge is enabled for different domains

and applications. PSMs abstract from a specific representation formalism in contrast to the general inference engines. Finally, PSMs decompose the reasoning task of a knowledge-based system in a number of subtasks and inference actions that are connected by knowledge roles. As such, PSMs are a special type of software architectures, software architectures for describing the *reasoning* part of knowledge-based systems.

### B. Ontologies

Ontologies have become a popular research topic and have been investigated by several Artificial Intelligence research communities, including KE, natural-language processing and knowledge representation. Ontologies meet a major demand in these fields: they establish shared and common understanding of a domain that can be communicated across people and computers.

Depending on their level of generality, different types of ontologies may be identified that fulfill different roles in the process of building a knowledge-based system. Among others, we distinguish the following ontology types:

- i) *Domain ontologies* capture the knowledge valid for a particular type of domain (e.g. electronic, medical, mechanic, digital domain).
- ii) *Generic or commonsense ontologies* aim at capturing general knowledge about the world and provide basic notions and concepts for things like time, space, state, event etc.). As a consequence, they are valid across several domains.
- iii) *Representational ontologies* do not commit themselves to any particular domain. Such ontologies provide representational entities without stating what should be represented.

A well-known representational ontology is the *Frame Ontology*, which defines concepts such as frames, slots and slot constraints allowing knowledge engineers to express knowledge in an object-oriented or frame-based way.

## V. TECHNIQUES AND TOOLS

### A. Techniques

A number of techniques have been developed in knowledge engineering to embody the principles. Here we shall focus on one particular set of techniques that constitutes a methodology for acquiring knowledge from an expert during 'knowledge acquisition'. Two types of techniques are used. One is natural techniques and the other is 'contrived' techniques. The process of moving from natural to contrived techniques. It is summarized in the following steps:

- i) Conduct an initial interview with the expert to (a) scope what knowledge should be acquired, (b) determine to what purpose the knowledge should be put, (c) gain some understanding of key terminology, and (d) build a rapport with the expert. This interview should be recorded on either audiotape or videotape for later analysis.
- ii) Transcribe the initial interview and analyse the resulting document i.e. protocol, to produce a set of questions that cover the essential issues across the domain.

- iii) Conduct a second interview with the expert using the pre-prepared questions to provide structure and focus i.e. semi structured interview.
- iv) Transcribe the semi-structured interview and analyse the resulting protocol, looking for knowledge types: concepts, attributes, and values, classes of concepts, relationships between concepts, tasks and rules.
- v) Represent these knowledge elements in a number of formats, for example, hierarchies of classes, hierarchies of constitutional elements, grids of concepts and attributes, diagrams and flow charts.
- vi) Use the resulting representations and structured documentation with contrived techniques to allow the expert to modify and expand on the knowledge you have already captured.
- vii) Repeat the analysis, representation-building and acquisition sessions until the expert is happy that the goals of the project have been realized.
- viii) Validate the knowledge acquired with other experts, and make modifications where necessary. It does not assume that any previous knowledge has been gathered, or that any generic knowledge can be applied.

### B. Tools

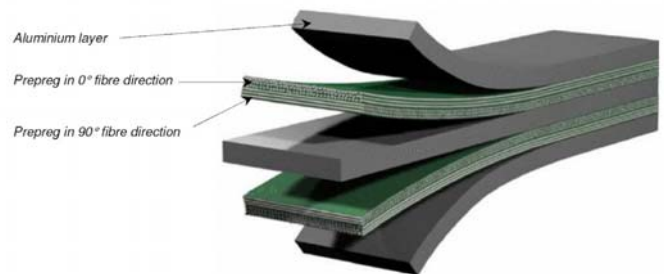
Knowledge engineers make the process of acquiring, modeling and storing knowledge more efficient and less prone to errors by using dedicated software tools. PCPACK is the comprehensive of such tools that is a commercially available set of knowledge engineering tools that is developed for research on a variety of project. The aim is to identify the important aspects of the domain. The tool simulates the way someone would mark a page of text using different coloured highlighter pens, for ex., green for concepts, yellow for attributes. When the user has completed highlighting a document, the marked text can be automatically saved in the PCPACK database for use in all other tools. PCPACK is the support tool for the MOKA and CommonKADS methodologies. Hence, it is fully consistent with the methods and techniques of knowledge engineering. PCPACK is a suite of tools that include the following:

- i) Protocol tool: this allows the marking-up of interview transcripts, notes and documentation to identify and classify knowledge elements to be added to the knowledge base.
- ii) Ladder tool: this facilitates the creation of hierarchies of knowledge elements such as concepts, attributes, processes and requirements;
- iii) Diagram tool: this allows the user to construct compact networks of relations between knowledge elements, such as process maps, concept maps and state-transition diagrams;
- iv) Matrix tool: this allows grids to be created and edited that show relations and attributes of knowledge elements;
- v) Annotation tool: this allows sophisticated annotations to be created using dynamic html, with automatically generated links to other pages and to other knowledge models in PCPACK;

- vi) Publisher tool: this allows a web site, or other information resource, to be created from the knowledge base using a template-driven approach that maximizes re-use. Knowledge engineering methodologies, such as MOKA, CommonKADS and the 47-Step Procedure, provide ways of conducting a project from start to finish and ensuring best practice along the way.

## VI. A PRACTICLE KNOWLEDGE ENGINEERING EXAMPLE

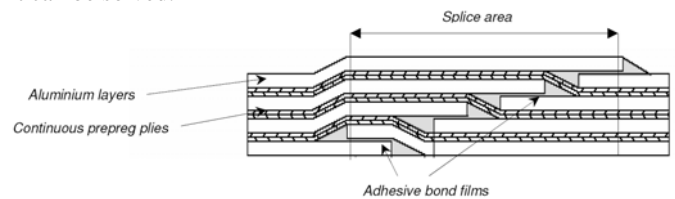
This section describes a practical application of knowledge engineering using the tools and methods are discussed. The practical application is performed at Stork Fokker AESP in The Netherlands and focuses on the design process of Fibre Metal Laminates (FML) panels. Fibre Metal Laminates are a member of the hybrid materials family, currently being developed for applications in primary aircraft structures. A FML consisting of alternating layers of aluminum and unidirectional glass fibers pre-impregnated with epoxy resin is called Glare (Fig.1). Glare is developed for the skin panels of the front and aft fuselage section of the Airbus A380. Some of the biggest panels measure 10 by 3.5 meters.



**Fig. 1 Built-up of Glare Fibre Metal Laminate**

From an engineering point of view FML structures are like composite structures, where, besides the structure, the material itself has to be designed. The laminate parameters can be tailored to achieve lally an optimum structure. The constraints to this tailoring process are various. Among it are manufacturing constraints which set limits on the dimensions of the aluminium and prepreg layers in the FML. The region where aluminium sheets are overlapped is called the splice area. The splice area induces many requirements to the manufacturing process.

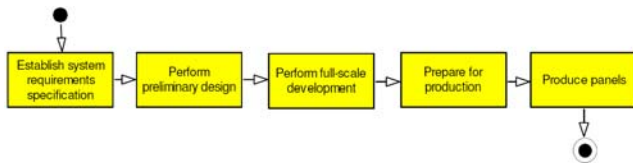
For example, requirements describe the riveting of the FML skin panel to the aluminium back-up structure. For these splice areas a minimum distance between a rivet and a free aluminium edge should be accounted for. Since the position of the back-up structure is fixed by higher level design constraints, the locations and the dimensions of the splice areas will have to be altered until these rivet requirements are satisfied. In turn, shifting the position of the splice areas to satisfy the constraints for one frame may result in violated constraints in other frames. This results in an iterative process that needs to converge before it can be solved.



**Fig. 2** In the splice area, two aluminium sheets are coupled together to form a continuous Glare panel and this area is driven by lot of (production) requirements.

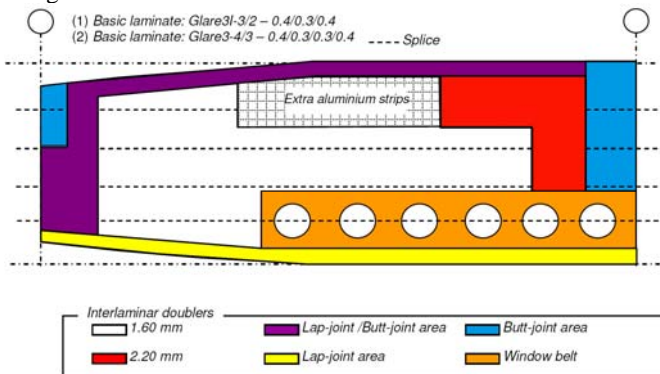
Current FML Skin Panel Engineering Process:

The current engineering process at Stork Fokker for Glare panels consists of several consecutive phases and is shown in Fig. 3.



**Fig.3** Process diagram of the Glare fuselage panel design process.

After establishing the requirements on the Glare panel an initial preliminary design (Fig.3) is made to check the placement of the various splice areas. After the preliminary design a full-scale development phase is started resulting in a detailed Glare panel design (Fig.3, Fig.4), followed by a production preparation phase. The output from the detailed design and production preparation phase results in a product definition dossier. The product definition dossier is the baseline for starting the series production of the specific Glare panel. In the next section some tasks within the full-scale development phase are analyzed on the applicability for providing solutions using KBE methods.



**Fig.4** A practical preliminary design of an FML panel.

## VII. CONCLUSION

Principles, techniques and tools, developed for knowledge engineering, can be modified and extended to help academic institutions with knowledge management. Knowledge engineering has been dealing with knowledge for the past 25 years, from both a theoretical and practical perspective, and has built up a stock of principles, techniques and tools that have a good track record in a wide range of applications. There is a substantial overlap between the concerns and problems encountered in knowledge engineering because, in many areas, this is trying to do the same thing—using technology to “transport” knowledge from an expert in a domain to those who are not experts.

## ACKNOWLEDGEMENT

To our guide Sitanath Biswas, for wisdom and patience and encouragement. I wish also to acknowledge the support and help to my friends who helped us a lot.

## REFERENCE

- [1]. Stokes, M., “Managing Engineering Knowledge, MOKA: ‘Methodology and Tools Oriented to Knowledge Based Engineering Applications’”, *Professional Engineering Publishing Ltd*, London, United Kingdom, 2001.
- [2]. Schreiber, G., H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de Velde and B. Wielinga, “Knowledge Engineering The CommonKADS Methodology”, *MIT Press*, Boston, MA, USA, 1999.
- [3]. Milton, N., “Knowledge Acquisition in Practice: A Step-by-step Guide”, London Ltd., London, UK, 2007.
- [4]. PCPACK, Software Package, Version 1.4.4R, Release 5, *Epistemics*, Nottingham, United Kingdom, 2006.
- [5]. Elst, S.W.G. van der, “Design of a Knowledge Based Prepreg Sub-Division Process: FML Engineering Process KBE Principles”, Delft University of Technology, Netherlands, 2007.
- [6]. A. Abecker, S. Decker, K. Hinkelmann, and U. Reimer, *Proc. Workshop Knowledge-based Systems for Knowledge Management in Enterprises, 21st Annual German Conference on AI (KI’97)*, Freiburg, 1977. URL: <http://www.dfki.uni-kl.de/km/ws-ki-97.html>
- [7]. M. Aben, *Formal Methods in Knowledge Engineering*, PhD Thesis, University of Amsterdam, 1995.
- [8]. J. Angele, S. Decker, R. Perkuhn, and R. Studer, Modeling Problem-Solving Methods in New KARL, in: *Proc. of the 10th Knowledge Acquisition for Knowledge-based Systems Workshop (KAW’96)*, Banff, 1996.
- [9]. J. Angele, D. Fensel, and R. Studer, Developing Knowledge-Based Systems with MIKE, to appear in *Journal of Automated Software Engineering*, 1998.
- [10]. J. Angele, D. Fensel, and R. Studer, Domain and Task Modeling in MIKE, in: A. Sutcliffe et al., eds., *Domain Knowledge for Interactive System Design*, Chapman & Hall, 1996.
- [11]. H. Akkermans, B. Wielinga, and A.Th. Schreiber, Steps in Constructing Problem-Solving Methods, in: N. Aussenac et al., eds., *Knowledge Acquisition for Knowledge-based Systems, 7th European Workshop (EKAW’93)*, Toulouse, Lecture Notes in AI 723, Springer-Verlag, 1993.
- [12]. T. Berners-Lee, M. Fischetti, and M. Dertouzos, *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. San Francisco: Harper, 1999.
- [13]. T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web,” *Scientific American*, p. 279, May 17 2001.
- [14]. H. Beyer and K. Holtzblatt, *Contextual Design: Defining Customer-Centred Systems*. San Francisco: Morgan Kaufmann, 1998.
- [15]. J. Blythe, J. Kim, S. Ramachandran, and Y. Gil, “An integrated environment for knowledge acquisition,” in *Int. Conf. on Intelligent User Interfaces*, San Francisco, CA, 2001, pp. 13–20.
- [16]. R. Brachman, D. L. McGuinness, P. Patel-Schneider, L. Resnick, and A. Borgida, “Living with CLASSIC: When and how to use KL-one-like language,” in *Principles of Semantic Networks*, J. F. Sowa, Ed. Morgan Kaufmann, 1991, pp. 401–456.
- [17]. B. Buchanan and E. Shortliffe, *Rule-Based Expert Systems: The MYCIN experiments of the Stanford Heuristic Programming Project*. Reading, MA: Addison-Wesley, 1984.
- [18]. C. Cheng, Y. Shahar, A. Puerta, and D. Stites, “Navigation and visualization of abstractions of time-oriented clinical data,” Stanford Medical Informatics, Tech. Rep. SMI-97-0688, 1997.
- [19]. L. Chung, B. Nixon, and E. Yu, “Using non-functional requirements to systematically select among alternatives in architectural design,” in *1st International Workshop on Architectures for Software Systems*, Seattle, 1995, pp. 31–43.
- [20]. P. Clark, J. Thompson, K. Barker, B. Porter, V. Chaudhri, A. Rodriguez, J. Thomere, S. Mishra, Y. Gil, P. Hayes, and T. Reichherzer, “Knowledge entry as the graphical assembly of components,” in *1st International Conference on Knowledge Capture (K-Cap ’01)*. Victoria, BC: ACM Press, 2001, pp. 22–29.