Discussion paper for the ECAI Workshop W31 on Knowledge-Based Systems for Model-Based Engineering to be held in Berlin, 22.08.2000.

Rüdiger Klein

Knowledge-Based Engineering: Where we are – Where we go?

Information technologies have been of growing importance in this context over the last decades. Currently, this process is gaining dramatically speed and power in all areas. The main reason is the "internet revolution". The advent of internet and intranet technologies changes the way information is exchanged and used: inside companies as well as in their outer business. E-business and b2b commerce are only the newest buzz words. They complete the more "old fashioned" (just a few years old) slogans as concurrent engineering, computer supported cooperative work, knowledge management, and such "dinos" as CAD, CAE, CAM, PDM, etc. *They all* will be influenced substantially by the new-coming technologies.

What is exchanged in this way is information - but what is necessary is to share, communicate and use *knowledge.* With current information technologies, the meaning of data is mainly left to the human users. They are the (main) "sources" and "processors" of knowledge. When computers are used to support this information processing this is done mainly in a pre-defined and explicitly coded way. Automated knowledge processing by knowledge based systems is still more an academic issue than one of daily practice - though in the meantime a number of industrial knowledge based systems have been implemented. Knowledge based systems in engineering practice are still more an exception than the normal case. Such systems are mostly restricted to relatively small and isolated applications, and they need a large effort to get built and maintained.

This has to change – this is our confession – if the "internet revolution" shall be mastered effectively and efficiently. *Knowledge processing* has to become an *integrated part* of information technologies. Only in this way the *new complexity of information* in the internet age can be successfully managed. This is especially true in engineering with its complex and manifold tasks: engineering knowledge management, requirements engineering, specification-driven design, design-to-X, RAMS-engineering (reliability, availability, maintainability, safety), etc.

Knowledge based technologies have two main aspects in engineering (and not only there):

- knowledge modelling, and
- problem solving.

Knowledge modelling is important as "backbone" and "glue" of the different information systems used. The information processed by these systems has explicitly to be modelled including the underlying assumptions. This is essential for system interoperability, information integration, intelligent information retrieval, knowledge warehouses, etc. Ontologies emerged in recent years as a main field of knowledge modelling especially dedicated to this task

But knowledge modelling has a second focus – directly related to problem solving by knowledge based technologies. Knowledge based systems can be seen as two separated but interacting and interdependent components: the applied problem solving methods and the corresponding knowledge base. Both have to correspond to each other: the knowledge base has to be modelled according to the problem solving methods, and the problem solving methods have to chosen according to the task at hand (the problem solving competence) and the applied modelling assumptions in the knowledge base.

Currently, this process of knowledge modelling and knowledge based system design is a complicated and laborious process. It needs special skills in knowledge based technologies

as well as in the intended domain. Due to missing comprehensive domain models and powerful generic problem solving methods, various compromises and short cuts have to be made during modelling (which normally requires a deeper understanding of the domain). The resulting knowledge based systems are typically hard to reuse or to extend.
Let's describe the way out of this dilemma as a *vision of knowledge based engineering*:

> We have a library of comprehensive, generic, application independent domain models including explicitly described meta models, underlying assumptions, etc.; and
> We have libraries of problem solving methods each with a well formulated problem solving competence, and an explicit description of the assumptions the corresponding knowledge base has to observe.

Both knowledge libraries can be accessed through the internet using intelligent retrieval, and knowledge based systems can be configured automatically reusing such building blocks from the libraries.

In this way, much less effort will be needed in order to build much more complex knowledge based systems.

Visions are valuable: because they allow us to dream, they help us to get motivated, and they guide our search to turn them into a reality. Visions become a danger if one forgets the difference between dream and reality.

Both roles of this vision provide the ground for our workshop: which are the challenges on our way to knowledge based engineering, and what is realistic today or can become real on this way within the near future.

Of course, the vision described above is not restricted to engineering. But engineering has *some special aspects.* They make it challenging to discuss knowledge modelling and problem solving issues especially related to this domain. These aspects can be summarised as follows:

- Engineering knowledge tends to be quite *complex*: physical laws, technical principles, geometric shapes and relations, norms and directives, functions, structures, and behaviours etc. have to be modelled with all their facets and interdependencies.
- Though quite complex, engineering knowledge is *well structured* and has a *relatively clear* meaning.
- Knowledge based technologies will be important in engineering – but they are not everything. To the contrary: key to the success of knowledge based techniques in engineering is the *integration* of these techniques with conventional information processing techniques: geometric modelling, numerical simulation, etc. What is essential is that this integration can not be a one way street: both sides have to move towards each other.
- Engineering knowledge is dealing to a large extent with "object level" knowledge: system, component, and function descriptions, physical laws, behaviours, etc. But due to its complexity, this engineering knowledge is also highly *problem solving related*: how to attack a given task, which way to choose, how to manage conflicts, etc. Engineering *methodology* is an essential and indispensable part of engineering knowledge because this is the only way to manage the complexity in this domain.

The *papers* presented at this workshop are typical examples of the *state of the art* of knowledge based engineering today. They contain descriptions of successfully applied industrial cases or of academic research systems. They deal with model based diagnosis or with engineering design. Each of them concentrates on a special problem or problem class in the broad field of engineering applications, and follows a special approach to model this problem and the problem solving. There is a large amount of similar research activities and results described elsewhere dealing with special problems in engineering and special approaches to them.

In order to provide a common platform for the workshop, we suggest to focus the discussion onto the following issues related to the vision of knowledge based engineering outlined above:

1. What were the *main modelling principles* applied in each of the described systems: on the domain knowledge level as well as on the side of problem solving methods? On which *assumptions* are these models based?
2. What changes and/or extensions in the applied modelling approaches would be necessary in order to come to *more generic models*? Is that realistic? Do human engineers apply generic models in order to solve their problems? If yes – how do these models look like; if no – why not?
3. What should an *engineering ontology* (or a system of engineering ontologies) look like in order to support generic knowledge models in engineering domains?
4. How is problem solving controlled in engineering? If there are more than one *problem solving method* applied in a system, how is *their interplay* managed and controlled?

Of course, it is not an easy task to find comprehensive answers to these questions. Probably, this will not be achievable at this workshop alone. It will need time and much more substantial research efforts. But we should find answers to these questions – for the benefit of knowledge based engineering.

This is my personal view. Also counter positions are, of course, very welcome – if anybody does *not believe* in the visionary approach formulated above, or anybody has concrete experiences in contradiction to what has been suggested.

May this workshop provide a good and stimulating starting point for these discussions.