Chapter 7
Knowledge Engineering Support for Software Requirements, Architectures and Components

Muthu Ramachandran
Leeds Metropolitan University, UK

ABSTRACT
The demands of SE imply a growing need for using AI to support all aspects of the software development process. This chapter provides insights into the application of knowledge based approaches to the development of agile software development, software product line, software components and architecture. In particular, it presents three research systems that demonstrate the potential benefits of utilising knowledge based approaches to support agile methods. The first system, called SoBA, supports the use of a story card for agile software development; the second system, called .NET designer, provides design rationale for choosing appropriate architectural solutions, and the third system, called RAIS, provides reuse assessment and improvement for designing reusable software components.

INTRODUCTION
The term Artificial Intelligence (AI) was coined at Dartmouth in 1958 and nearly ten years later the term Software Engineering (SE) was coined at the same place in 1968. AI can be defined as “the study of how to make computers do things at which, at the moment, people are better” (AmI, 2008). Knowledge is gained from experience and highly related to all living things and Engineering is a human activity which aims to create, innovate, and produce a product. Therefore, there are common activities amongst AI, KE & KBS (Knowledge Engineering and Knowledge Based Systems), ES (Expert Systems), KM (Knowledge Management), Neural Networks, Fuzzy Logics, and SE. All these activities aim to solve a complex problem and follow a similar pattern, viz: identify the problem, identify common patterns, look up a similar problem which has been solved in the past, and produce a conclusion, product and result. Widely used AI methods include:

DOI: 10.4018/978-1-60566-758-4.ch007
• Rule-based systems, where knowledge is represented in the form of if-then production rules.
• Case-based systems, where past problem-solution pairs are stored and then subsequently used to help solve new problems by adapting the solution of past similar problems.
• Experience-based systems where the reasoning is by previous experience, however it is difficult to implement a full working system as knowledge is a continuous activity.

Software Engineering has emerged as a discipline that aims to develop and manage software development as a structured and systematic activity. One of its key challenges has been to address the software crisis, which is a situation where software projects are often late and over budget. Today SE is rich in terms of its achievements in modelling tools (UML), technology (CASE), programming languages (Java, C#), education, standardisation, and its impact on the global economy and workforce (world wide impact on establishing offshore software development and its contribution to local economy in places like India, China, and other countries). SE remains a highly skilled human intensive activity and relies on problem solving skills of human knowledge and experiences. Therefore AI, ES, and KE will continue to play a major role in automating numerous software development activities. Rech and Altoff (2008) say “The disciplines of Artificial Intelligence and Software Engineering have many commonalities. Both deal with modeling real world objects from the real world like business processes, expert knowledge, or process models”. Therefore, the interplay between these two areas is significant and it makes sense to take advantage of their mutual strengths.

This chapter aims to identify work on knowledge based support for agile based software requirements elicitation techniques, software architecture design and software components. The chapter is organised as follows. The next section summarises the relationship between intelligent systems and software engineering. This is followed by three sections, each of which describe research systems that explore particular aspects of the relationship: the first system, SoBa, explores how Story Boards can be used to support agile methods, the second system demonstrates how it is possible to support the design of .Net based systems, and the third system, called RAIS, illustrates how it is possible to automatically assess the reusability of a design and provide guidance on developing more reusable components.

INTELLIGENT SYSTEMS AND SOFTWARE ENGINEERING

This section presents a summary of the relationship between various aspects of SE and intelligent systems and then presents some of the main research that aims to develop intelligent tools for supporting software engineering. Figure 1 shows some of the current and futuristic research strategies based on integrating KBS and SE (KBSE). These are the building blocks for an experience based knowledge factory or learning organisation. The other techniques, such as capturing knowledge based best practices, software design guidelines, process improvement techniques, and knowledge based automated tools will continue to grow. The emerging areas, such as Software as a Service (SaaS), require additional domain expertise on service engineering that will make use of existing solutions, and enterprise knowledge management (Ramachandran 2008, Chapter 13).

Problem solving is one of the key skills that humans have acquired and experienced over the centuries. We can assimilate the problem solving process as a generic process as shown in Figure 2.

The generic process shown in Figure 2 is also generic to the SE problem solving strategy