Knowledge-Based Systems 13 (2000) 101–112
Knowledge-Based SYSTEMS
www.elsevier.com/locate/knosys

Towards the application of case based reasoning to decision-making in concurrent product development (concurrent engineering)

B.U. Haque, R.A. Belecheanu, R.J. Barson, K.S. Pawar*

School of Mechanical, Materials, Manufacturing Engineering and Management, University of Nottingham, Nottingham, NG7 2RD, UK

Abstract

This paper describes the development and application of case based reasoning (CBR) to provide decision support for project managers and engineers during the early phases of new product development in a concurrent engineering (CE) environment. The paper discusses the reasons for using CBR, focussing on issues such as case collection, maintenance, terminology, adaptation, and similarity; and how the final system could contribute towards achieving a CE conducive culture. The main issues in using CBR in a CE environment, that is characterised by ill defined and ill structured information during early phases of product development, are textual consistency of terminology, validity of case similarity, and the difficulty in automating case evaluation and adaptation. Additionally the paper concludes that using technology like CBR, which can be costly to develop and implement, requires the company to train considerably their managers and team members to document their experiences and knowledge in a manner with which the system can work with and team members can understand. There needs to a commitment to maintain and improve the knowledge base—a 'knowledge friendly' culture hence needs to be instilled for CBR tools to succeed.

Keywords: Case based reasoning; Concurrent product development; Decision support

1. Problem description

1.1. Background (concurrent engineering)

New product development (NPD), is an interdisciplinary activity requiring contributions from nearly all the functions of a firm, whether it is an upgrade/improvement of an existing product or a new concept either to the company or the market. Traditionally NPD has been viewed as an organisational activity, which was the result of various functional activities performed in stages from concept development to product delivery. The sequential operation of these functional stages resulted in long development times and many quality problems due to the lack of communication and understanding of the different product design, manufacturing and above all customer requirements. To avoid these problems *concurrent engineering* (CE) is being used by many companies and has resulted in companies making new products better and faster [1–4].

CE or CNPD is characterised by the early involvement of the different functional disciplines and parallelism of hitherto sequential activities (i.e. bringing downstream activities forward). Decisions made in the early (design) phases of product development are hence often based on incomplete, ill-structured, and poor quality information. This is why decisions are sometimes made in an empirical manner, using only personal knowledge and experience, gained during past problem solving processes. It is widely cited that most mangers/designers refer back to previous solutions to related problems as a first step in the design process [5].

1.2. The problem (end user needs)

When engineers or managers call upon past experience or the "experts" opinion, the information or knowledge is prone to bias of the particular experiences of that individual (or the so-called "expert"). The wider collective company (corporate) knowledge is not always readily available in a structured and consistent format. A detailed review of project management procedures and processes in NPD at selected companies [6] identified that there exists a case history of past projects contained in disparate 'data' or 'information' sources such as project files, databases and most importantly individuals memory. This information was not only restricted to major decisions concerning the continuation of the project, but also included data about specific products or components, design decisions that

* Corresponding author. Tel.: +44-1159514029; fax: +44-115951400.

0950-7051/00/$ - see front matter
PII: S0950-7051(00)00051-4

have worked well in the past and those that have not worked so well, problem solving approaches, etc. This data was usually difficult to access, especially where individual knowledge or memory was concerned, when making decisions in new projects. At the same time it was not always possible to find the most experienced or most knowledgeable personnel. Hence, there was the risk that a problem or difficulty that was found in an earlier project, and subsequently resolved, could be repeated in a new project.

There was a lack of structured support, not only in formal management reviews (decisions), but also in many decisions made by team members outside the formal reviews with respect to the detail design and development. These decisions could be equally critical to the success of the project. The basic requirement of the industrial partners was that the past experience was presented in a constructive way at the time of making the decision, and also indicate the relevance of the data for that particular decision. The decision support data or knowledge should also be examined by other team members to assess from their viewpoint the acceptability of the decision. The requirement for the system was hence to build a knowledge base in which complete ‘decision cases’ or scenarios could be entered and then recalled or reused when similar problems arose again.

Another issue was that design is typically carried out in an iterative manner in terms of generating the initial design and then testing. Product or component design, in mechanical engineering for instance, is evaluated under numerous interrelated criteria such as machinability, quality, reliability, structural integrity, assembly, maintenance and so on. This process is referred to as Design For x (DFx), with x as one of the criteria or constraints. Time delays and costs can be incurred if such evaluations result in red-design or take too long. Though the CE philosophy attempts to bring DFx issues to the attention of the designer as soon as possible, the process would benefit greatly if support could be provided through a what-if study based on past experiences.

Design changes or changes to specification arise from other sources too such as marketing, reacting to sudden changes in market needs or issues relating to industrial design; or purchasing, identifying supplier capability limitations, etc. Quite often engineers are not aware of the consequences of the changes. It would be quite useful if the consequences of such changes or similar changes could be identified or known in advance or prior to elaborate testing, simulations or waiting for the actual event to happen!

The above problems or issues called for a knowledge based decision support system, providing the managers and engineers (in design and development) with structured, consistent, comprehensive and accurate information and knowledge. This would enable the early phases of NPD and hence CE to be more productive. Additionally the success of CE depends upon collaboration between the different functional expertise to arrive at a mutually agreeable decision. The decision support should encourage this by providing viewpoints from different experiences of different people.

The development of the required system has been carried out in an EU funded project called CODESCO—A Practical *Co*mmunication and *De*cision *S*upport Environment for Managing *Co*ncurrent Product Development (ESPRIT project no. 25455) [7]. The overall objective of this research project was to develop and validate a communication as well as a decision support system for helping project managers and design/development engineers in their decision-making activities within a CE environment.

2. Application description

2.1. The solution—choice of CBR

The explicit request for reuse of knowledge and experience called for the application of *case based reasoning* (CBR). CBR is a computer technique, which combines the knowledge based support philosophy with a simulation of human reasoning when past experience is used, i.e. mentally searching for similar situations happened in the past and reusing the experience gained in those situations [8]. In the same way, in CBR, the knowledge cases are structured and stored in a database, which the user queries when trying to solve a problem. The system *retrieves* a set of similar cases and then *evaluates* the similarity between each case in the database and the query. The most similar case(s) are presented to the user as possible scenarios for the problem at hand. The user has to decide if the solution retrieved is applicable to the problem, i.e. the system does not make the decision, it only supports the decision making process. If it cannot be *reused*, the solution is *adapted* (manually or automatically). When the user finds a solution, and its validity has been determined, it is retained with the problem as a new case in the database (the case is “*learned*”), for future reuse.

The theoretical CBR cycle is, therefore, a *retrieve-evaluate-adapt-learn* process. However, a CBR system may very well implement only the *retrieval* stage of the process and does not need to implement the other stages. The *retrieval* stage is the basic stage and the expression of the concept of reuse of experience.

There were a number of reasons for choosing CBR over conventional rule based systems. The main requirement for the system was that it should be able to support a variety of product and business domains. Additionally the system was intended to deal with a fairly wide range of technical and managerial problems. Traditional rule-based knowledge approaches were not found suitable for this requirement, as they required strong domain knowledge and representation, whereas decision problems in CE are generally difficult to define and structure. In CBR, as opposed to rule-based approaches, knowledge about the domain is acquired and maintained through unrelated but similar cases and does not