Analysis of requirements and approach to support decision making processes in Concurrent Engineering

Roxana Belecheanu[1], Dr. Kulwant S Pawar[2], Badr Haque[3], Dr. Richard Barson[4] and Frithjof Weber[5]

[1]epxrmb@epn1.maneng.nottingham.ac.uk
[2] Kul.Pawar@nottingham.ac.uk
[3]epxbuh@nottingham.ac.uk
[4]Richard.Barson@nottingham.ac.uk
All at the Division of Manufacturing Engineering and Operations Management, University of Nottingham, University Park, Nottingham, NG7 2RD, UK
[5]web@biba.uni-bremen.de
Bremen Institute of Industrial Technology and Applied Work Science (BIBA), PO-Box 33 05 60, D-28335 Bremen, Germany

This paper presents an analysis of a decision support approach for engineers and managers in Concurrent Product Development. The work was carried out as part of an on-going research project entitled CODESCO (A Practical Communication and Decision Support Environment for Managing Concurrent Product Development - ESPRIT project no. 25455), involving two industrial partners, two academic partners and two software companies. Firstly, the need for implementing a knowledge based decision support system in one of the companies involved in the project is identified. Secondly, several issues regarding the implementation of the system are discussed (e.g. what range of problems the knowledge base should cover, what kind of information it should contain and how this information should be structured). Finally, results of a preliminary evaluation of the system are discussed, in terms of advantages and disadvantages, and emerging research issues.

1. Introduction

Increasing competition in world markets as well as new regulations and laws concerning environmental and quality issues have lead to immense structural and technological changes within industry. The majority of today's industrial products have increased complexity e.g. mechanical components, electronic hardware, software as well as value added services. Co-operative product development within multidisciplinary design teams is one way to obtain competitiveness in such an environment.

This is the framework of a typical Concurrent Engineering (CE) project, where different tasks are executed in parallel and collaboratively. CE is being used nowadays by many companies and has resulted in companies making new products better and faster (Riedel and Pawar, 1991, Shina, 1994, Hanssen, 1997, Trygg, 1992). In CE projects, knowledge and information sharing is one of the key prerequisites for achieving the objectives. The efficiency of communication processes and the correctness of decisions depend significantly on the immediate availability of information. Besides, the quality, consistency and structure relevance of this information are equally important influencing factors for these processes. Due to parallelism and early involvement of different disciplines in early development stages, these requirements are rarely met. In early stages design decisions are ill-structured and often supported with scarce information. Multiple potential solutions and limited predictability all contribute to the design complexity (Lambright and Ume, 1996). Besides, significant functional and technical barriers often prevent the free flow of the necessary knowledge and information (Forgionne, 1994). This is why, most designers refer back to previous solutions to related problems as a first step in the design process (Walsh, V. *et al.*, 1992). Past relevant design solutions suggest how other designs were approached, carried out, manufactured and how they performed in their operational environment (Lambright and Ume, 1996) and augment the decision making capability of the designer or technical manager.

In the light of these findings, a natural approach for decision support in any environment is to build a knowledge base of past lessons and experiences. This database of problems and solutions should be sufficiently rich so that an automated reasoning system can use that knowledge to analogically find solutions to problems or answer to what-if questions.

This paper tries to answer the question of "what does the implementation of such a system mean for a company", based on a case study with one of the companies involved in the project which funded the research. The methodology for the research and development of the system implied intensive interviews, observations and data collection effort, in order to analyse the industrial environment, to determine the need for a knowledge based decision support system and to perform the knowledge acquisition task required by the implementation.

2. The need for a knowledge based decision support environment

The company which this research addressed (called here *Company A*) is a 'one-of-a-kind' firm from high-tech industry, producing electrical and electronic hardware, and information technology systems for military and commercial applications. Its business is always contract-oriented. Products are designed for special environments (e.g. marine or air applications) and contracts include often long-term maintenance and life cycle support service for proprietary or external products.

From interviews and discussions with engineers and managers in this company, it was found that a certain procedure is commonly undertaken when technical problems have to be solved (see Figure 1).

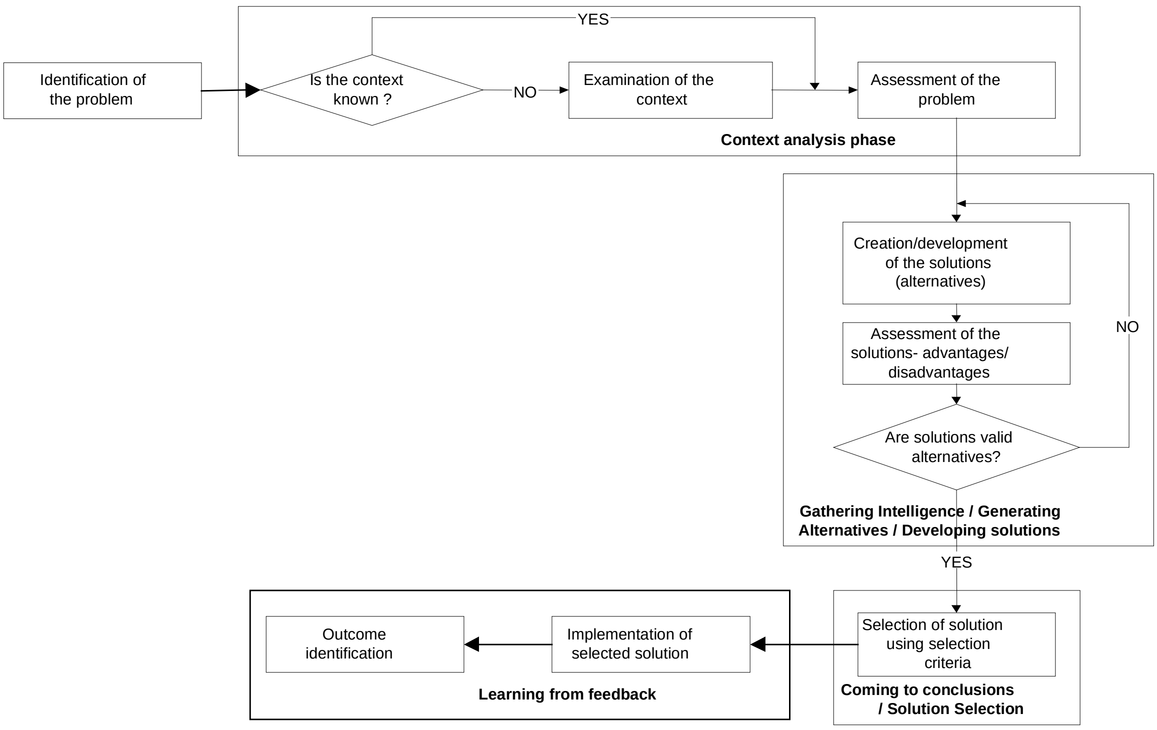

Figure 1. Decision making process.

A problem is usually reported to the Program Manager or Design Manager (*problem identification phase*), either by the client, or as a failure occurred in the test or production phases. The manager analyses the problem with design engineers qualified for that kind of task (*context analysis phase*), trying to identify all the constraints imposed to that problem. For instance, it is usually important to consider the relationship and type of contract they have with the client (long or short term contract), the production and maintenance cost incurred for the company, the test equipment available, the life cycle of technology in-house developed, etc. Sometimes the faulty component goes to a test in this stage, with in-house or client test system, to identify and analyse the cause of the problem. In the next step, the design engineers have to develop alternative solutions (*solution development phase*) which are design changes that eliminate the problem. These technical solutions are proposed to the manager who, in accordance with the contract restrictions, the company policies, and the prototype test results, selects one for implementation (*solution selection phase*). Sometimes, when possible, the solution selection decision is left for the client, situation which limits the company's responsibility and therefore is preferred. Otherwise the subsequent implementation effects need to be renegotiated with client (usually product performance and cost). If all solutions (sometimes only one available!) would be rejected, the engineer would start to develop and test a new one, and the process is repeated until a trade-off design is accepted or time is no more available for design changes.

In order to identify the need for a knowledge based decision support environment in this company, a few issues had to be looked at while analysing the general decision making process of engineers and managers:

- *How are engineering or managerial decisions supported in the design stage? Do current development efforts make use of past experience?*

All through the process described before, designers and managers had to make decisions: either when new designs had to be found, as the old ones had proved unsatisfactory, or even more critical, when a definite design solution had to be accepted to go to production. To support their decisions, they would usually use checklists or guidelines, and, most usefully, personal experience and knowledge about past problems, as much as one's memory could recall. For instance, technical guidelines and procedures would be used particularly when harsh environmental constraints would affect the design (e.g. humidity resistance, electromagnetic resistance, etc.). In no case the designer or manager would use a case base of past problems and solutions. However, they would try to remember similar situations, and when they could, not with all the necessary details. As they have also reported themselves, because the guidelines they were using were too technical, they could not possible capture all the parameters of the problem context (e.g. financial limitations, development or supplying time limitations, knowledge about the client, repair difficulties, etc.) and the implications of the design on these parameters.

- *Are the rationale and the outcome of decisions documented?*

People did not consider documenting the rationale of the decision making process, mainly because of having to handle usually complex ill-defined problems, where the steps to a solution were not clear, or because of lack of time (many times decisions were made under time pressure), or even because of not being used to. In this way, even though design solutions would have been recorded, they might have not been reusable, and if the knowledge and reasoning that went into developing those solutions might have been applicable for the new problem, they would have been lost.

- *Are the downstream consequences of choices made in development recorded?*

Problems sometimes recurred (or, better said, they had not been properly solved), or other problems would appear, as result of undetectable errors during the test phase, or unforeseen operational issues. Design solutions were released for production based on the tests performed in company laboratories, which, even though costly and time consuming, were sometimes less accurate than expected. Consequently, when the product would go into its operational environment, it would prove less reliable or its functional performance unsatisfied.

It was the case when a failure was detected during the installation, after two years from the product was supplied to the customer. The same problem had appeared in the prototype stage, due to the adaptation of an old design to satisfy a new product specification. The designer had changed the product internal configuration and eliminated the fault, as the laboratory test has wrongly proved. However, by the time the product was delivered, (typical development and production phases can go up to 3 years), the designer has left the company and his experience with that product was lost. The product had to go again through the same design problem solving process, with a new designer.

In such cases, every iteration through design results in long de-facto lead times, higher product cost, and not less importantly, affects company's reputation. To record downstream effects of decisions would be useful for the trade-off process of available design alternatives, especially when a similar past design implementation can show the undesired consequences.

- *Are the downstream consequences of choices made in development communicated across disciplines?*

A lack of communication support has also been identified in the company under research, both between design and production, and with the technical management level. A knowledge-based system used by various functions in the company would fill in the gap created by lack of communication support.

Our hypothesis is, therefore that, keeping record and sharing the successful or bad experiences of designers and managers through computerised support, would certainly improve the individual decision making, would raise the quality of arguments in the group of decision makers and would make opinions converge on better solutions. However, an analysis of the implementation issues and of the cost and benefits of such a system is required.

3. Implementation issues

Interviewed about their requirements for the system, engineers and managers of *Company A* opined that "the past experience should be 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 knowledge should be accessible and understandable by other engineers and managers in the company, to access from their viewpoint the acceptability of the decision". Consequently, problems and solutions were collected through interviews with engineers and managers and stored in the form of *decision cases*. They were categorised according to the type of decision they accounted for.

The design and implementation of the knowledge base implied consideration for the following matters:

- It is known that companies, especially of the large types of business, have built engineering knowledge repositories and integrated them in the product development process. However, they have usually focused on one particular type of technical problems, which for the advantage of having more precise answers to queries, they limited the usability to the engineering functions (usually design) of the company. In CODESCO, in the light of the requirements, the knowledge based decision support system addresses *engineering and engineering-management decisions*, by storing past lessons from both types of activities. Implementing such a knowledge base is not being a practice yet adopted.
  The types of decision cases collected determined a tree-type categorisation of cases, which was later on used by the automatic reasoning mechanism to retrieve the relevant knowledge. Most of the technical decision cases were related to design decisions, either occurring in design stages, or maintenance, or raised as production malfunctions or needs for improvement. The most common types of technical decisions were regarding reuse of old designs, with modifications due to inherent incompatibilities, design modifications to correct prototype failures or maintenance failures, design changes requested as improvements in cost of production, design adaptation for the use of a certain test technology, etc.
  The main categories of managerial decisions addressed in *Company A* concerned production cost reduction, resource allocation (e.g. in-house vs. outsource development), make vs. buy decisions, tooling investment, automatic test equipment investment. These decisions, as much as strategic some may sound, they were always strictly related to a particular product or product family, which brought many technical constraints into consideration, required usually collaboration of the technical manager with the engineer, and made the cases reusable only on similar product contexts. Therefore they were categorised as technical-management (or engineering-management) decisions.

- The wider range of cases to be collected called for a high-level structure (or low-level detail of representation): cases were structured in *problem*, *solution* and *outcome*.
  Within the problem section, a description of the problem is required, together with its context (i.e. product development stage, product and project, engineers or managers responsible for problem resolution, etc). All the constraints imposed to the solution must also be documented here. This section maps into the problem identification stage and context analysis stage of the decision making process identified in *Company A.*
  Within the *solution* section, the user is asked to provide details about the alternative solutions he developed, how he developed them, the advantages and disadvantages of each alternative, which alternative he selected for implementation (*the decision made*) and why (selection criteria). This section maps into the solution development phase.
  Lastly but equally importantly, the *outcome* of implementing the solution must be documented: positive or negative downstream consequences, related problems occurred, to what extent the problem was solved, etc.

- The relevance of the data was modelled through weighted fields, estimated by the user according to the case he was describing. However, they could not be accurately estimated (usually mathematical decision models and techniques are used to determine criteria weighting). Moreover, the weights of the fields could not be generalised, even within a class of problems.

The system was implemented using an Artificial Intelligence technique and is currently under test in *Company A*, where engineers and managers are using it in their Concurrent Engineering projects. The detailed software implementation and its positive and negative effects on performance have been discussed in previous articles (Belecheanu et. al, 1999). The costs and benefits of the system are discussed here, based on a comparison of the feedback from the end users with the expected results.

4. System evaluation – costs and benefits

An early evaluation of the system was performed both at practical (user test) and research level. Firstly, *Company A* became aware of the fact that, as useful as it may seem at the first glance, building a knowledge based decision support system is at least cost risky, if not useless, if the user does not have an understanding of his knowledge needs and does not realise the knowledge capabilities of his company. The effective use of the system lies in several technical and strategic aspects like:

- The company should provide the system with a sufficient number of cases
- There should be sufficient information in each decision case, to overcome the lack of detail due to the general structure of cases
- The cases should represent decisions with practical relevance, i.e. decisions which, either of usual type or more peculiar, required analysis, combined expertise and even creativity (the more creative the cases the better for future reuse, of course, if successful). Also, decisions should have temporal relevance (e.g. cases older than 5 years are unlikely to be useful for future, especially in high-tech industries, where technology evolves rapidly).
- Both successful and unsuccessful stories are equally important for a knowledge repository, as avoiding past mistakes is as useful as trying to copy old successes.

Satisfying these requirements guarantees that the system will function according to its capability. However, the pros and cons of the system are dictated by the system's nature and any user must be aware of them before deciding to have it in the company.

The following remarks highlight the advantages and disadvantages of such an investment:

- Due to the simple and general structure of the cases, the system can be used by engineers and technical managers involved in various activities throughout the product development process, in a commonly structured and understandable form (e.g. design engineers, production engineers, cost engineers, maintenance engineers, and the corresponding functional managers). In this way, similar problems, at general level, can be found in the past record of a different discipline. Unfortunately, due to the same generality of the information, the level of uncertainty in the decision support is decreased very little.

- The system proves its utility not only in regular problem solving tasks, when past solutions to similar problems could be replicable, but also when, in order to select a course of action and make a decision, an engineer needs to "see" the consequences of his action; in this case, a similar decision taken in the past and documented shows its outcome and underpins the reasoning. Also, the system allows for multidisciplinary trade-off, i.e. each decision case can capture and document various constraints, imposed by co-involved disciplines.
  As in the CE projects, the problems are highly unstructured, the causal relationship models implementations for what-if scenarios or the automated DFx design evaluations would be computationally much more expensive than our system.

- The system 'learns' continually in the realm of the user problem: as the decision-maker uses the system to make more and more decisions, the system becomes incrementally 'more intelligent' in the domain of problems is being used for. The downside of this is that, since the system is not tailored to a narrow particular domain of engineering design or management problems, the user will need to solve several different problems before the system can really be of major use. The time or effort associated with this 'start-up' cost is directly related to the diversity of the problems that the system is exposed to (Bardasz and Zeid, 1991).

- Using such a knowledge base reduces information gathering time in the process of decision making and improves communication with the technical manager. Unfortunately, as practices prove, most managers still make decisions based on intuition, despite the risk. Most managerial decisions are still disturbingly immune to technological and conceptual advances (Shoemaker and Russo, 1993). The success of the system, which can be itself costly to develop and implement, depends also on the people willingness to use and improve the system through their use. The system also requires training of managers and team members, both at technical level, and towards documenting and sharing their experiences and hence knowledge in a manner which others can use.

5. Open research issues

A prime open research issue is to determine a valid procedure of evaluating the system performance. Therefore, performance measures will be defined and used during the pilot projects, in order to see how much the decision–making process has been improved by using our decision support system. A few examples of performance measure are given below:

For instance, improvement in 1) the quality of decision made, can be measured through:

- the number of undesired consequences occurred as result of the decision made, and which could have been avoided without causing other bad consequences (a counter-measure of performance);
- the number of constraints satisfied by the solution found, in correlation with their user-estimated importance (a weighted sum of the constraints)

Other measures can be: 2) the time saved for making the decision in rapport with previous situations (this variable can only be roughly estimated, through surveys of the users), or 3) the cost of making wrong decisions, as for example, by measuring the number of redo activities performed as consequences of that decision.

6. Conclusions

Designers and managers need to use knowledge repositories of past lessons in their daily decision making tasks. Such a system not only can bring past knowledge into present, but also helps people sharing their experiences, and make better decisions, individually or together. However, the system can be costly to develop and implement, depending on the scale of the problem addressed. In order for the system to work and to be worth its cost, people must trained towards documenting and sharing the knowledge in a manner which others can use. Additionally there needs to a commitment to maintain and improve the knowledge base. An 'information sharing and knowledge friendly' culture hence needs to be instilled for such decision support tool to succeed.

7. References

Bardasz, T. and Zeid, I. (1991) Applying analogical problem solving to mechanical design, *Computer-Aided Design*, **23**, 3, pp. 202-212.

Belecheanu, R., Pawar, K.S., Haque, B.U., Barson, R. (1999) A framework for supporting decision making in early phases of product development in concurrent engineering, Proceedings of the 15th International Conference on Production Research, Limerick: Hillery & Lewis, **1**, pp.201-204

Belecheanu, R., Haque, B., Pawar, K.S., Barson, R. (1999) Decision Support Methodology for Early Decision Making in New Product Development – A Case Based Reasoning Approach, *Proceedings of the 5th International Conference on Concurrent Enterprising (ICE)*, The Hague, 15-17 March, pp. 111-120.

Forgionne, G.A. (1994) A Decision Technology System to Deliver effective Concurrent Engineering, *Concurrent Engineering: Research and Applications (CERA)*, **2**, pp. 67-76.

Hanssen, R.W. (1997) Reducing Delivery Times in Engineer-To-Order Firms by Using the Concepts of Concurrent Engineering, *Proceedings of the 4th International Conference on Concurrent Enterprising (ICE'97)*, The University of Nottingham, **8-10**, October, pp. 495- 508.

Lambright, J.P. and Ume, C. (1996) A Flat Composite panel Design Advisory System Using Knowledge Based and Case Based Reasoning, *Journal of Mechanical Design*, **118,** December, pp. 461-475.

Riedel, J. and Pawar, K. S. (1991) The Strategic Choice of Simultaneous Versus Sequential Engineering for the Introduction of New Products, *International Journal of Technology Management*, Special Issue on Manufacturing Strategy.

Shina, S.G. (1994) Successful Implementation of Concurrent Engineering Products and Processes, New York: Van Nostrand Reinhold.

Shoemaker, P.J.H. and Russo, J.E. (1993) A Pyramid of Decision Approaches, *California Management Review*, Fall, pp. 9-31

Trygg, L. (1992) Simultaneous Engineering: A Movement or an Activity of the Few? *International Product Development Management Conference on New Approaches to Development and Engineering*, Brussels, 18-19 May.

Walsh, V. *et al.* (1992) Winning by Design, Blackwell Publishers.