Towards a Methodology for Appropriate Acquisition of Design Software in Concurrent Engineering Design Organisations

Author/Contributor:
Reidsema, Carl

Publication details:
Proceedings of the Ninth International Conference on Manufacturing Excellence

Event details:
The Ninth International Conference on Manufacturing Excellence
Melbourne

Publication Date:
2003

DOI:
https://doi.org/10.26190/unsworks/429

Towards a Methodology for Appropriate Acquisition of Design Software in Concurrent Engineering Design Organisations

C. Reidsema

[*]*The University of New South Wales, Australia*

Keywords

Concurrent Engineering, Design, Design Software, Expert Systems.

Abstract

Concurrent Engineering (CE) attempts to process as many product development tasks in parallel as possible while at the same time maximising the use of downstream life-cycle knowledge at early stages of the design process. An effective implementation of this strategy can reduce the duration of design projects, save development costs, and provide better quality products. The successful management and execution of CE projects within such a highly distributed, knowledge intensive environment requires an appropriate selection of Knowledge Based Engineering (KBE) software as well as Computer Aided Engineering (CAE) tools for product and process modelling and analysis, and includes tools for such functions as data management, decision-making and communications between stakeholders. Because such tools and technology play a significant role in the successful implementation of CE, an important challenge to modern industry is the ability to quickly and accurately assess and acquire the most appropriate software tools to support product development. A framework that utilises the Analytic Hierarchy Process (AHP) technique and incorporates the unique requirements of the CE product development domain is proposed.

1. Introduction

Concurrent Engineering (CE) provides a systems engineering approach to achieve significant reductions in time-to-market whilst incorporating downstream life-cycle considerations upstream where early decision making has the most profitable impacts. The proliferation of technological software systems to support the high rates of information and data that is necessary in leveraging the benefits of CE, presents a real problem for managers and owners of small to medium size engineering companies. It is only in recent times that companies are beginning to concern themselves with the problem of employing a methodology for the

[*] School of Mechanical and Manufacturing Engineering, Sydney, New South Wales, Australia, 2052

rational acquisition of technological systems and computer based tools necessary to operate a business in today's competitive environment. Some of the questions that they face are:

1. What software capabilities do we need?
2. Are the claims of vendors realistic?
3. How and when should the company invest in new software, and
4. What benefits can be expected and are they measurable?

While the answers to these questions are not readily apparent, a means of addressing the issues that they present is needed if a company is to remain competitive. Such a methodology should not overlook the real benefits of implementing a low technology strategy if it is shown to be better in terms of reducing risk and more effectively utilising resources within the tight project schedules that are part and parcel of CE. The focus of the methodology should be towards assessing the impact that a software tool has on at least the most dominant drivers of a successful Product Development Program (PDP). Such drivers include:

1. Reduction in design iterations
2. Reduction of rework,
3. Enhanced integration of data, information and knowledge, and
4. Improvements in coordination of tasks and resources.

Although many vendors use CE terms such as integration, design for X, and so on, to claim that their tools are designed for CE, it is not clear that this is the case, nor is it clear what advantage some of these tools have for CE. For example, many applications have profound difficulty in the simple sharing of data with other legacy systems. What is clear though is that organisations are faced with an ever increasing initial cost for tools to improve their competitive advantage. A rational decision would include an analysis of whether this initial cost and the ensuing maintenance fees needs were justified against well understood and measurable outcomes.

Research indicates that the lack of suitable tools for handling the high quantities of information that CE requires can have serious deleterious effects on the quality of the design, schedule and team morale [1]. Conversely though, the incorrect selection of support software has the added disadvantage of consuming precious resources for little or no benefit. On the other hand, a careful and well-considered approach to the problem can enable design personnel at every point in the PDP [2,3]. A methodology for acquiring KBE and other engineering software tools for CE PDP decreases the risk of negative outcomes that result through "non-rational" approaches to the selection problem such as decisions that are based on:

1. Corporate image,
2. incorrectly perceived return on investment,
3. incorrectly perceived technical capabilities,
4. poorly understood requirements, and
5. over-exaggerated vendor claims.

An example of the type of claims that some vendors make is:

Such software (simulation) predicts product life cycle attributes before a product is manufactured, including development cost, performance, fuel economy, product lifespan, *crashworthiness, strength, safety, noise, reliability, comfort, manufacturability, maintainability, warranty costs, time to market, profitability and more. When a company purchases simulation software, services, and computers, it is making an investment [4].*

Claims such as these are commonplace in trade literature and on vendor websites, and have a tendency to obscure the real issues that should form the basis of rational acquisition policy. Issues such as:

1. What design aspect requires the most support from software tools?
2. What software will give the best return on investment for our company?
3. Is this acquisition in alignment with our corporate strategy?

Similarly, an acquisition method should support the often hidden but nevertheless important motivations or strategies that underlie a company's need to acquire engineering software including:

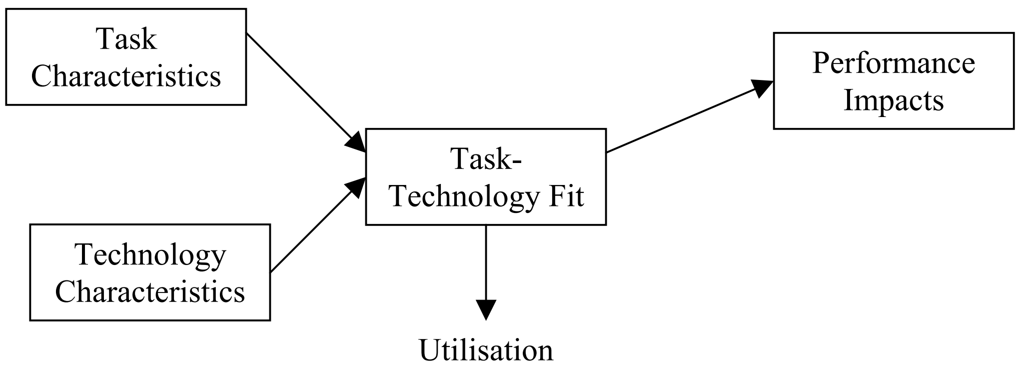
1. The company is forced to adopt a software system as part of a contractual requirement or to align themselves with a client,
2. The company is seeking to leverage their core processes (knowledge assets) to position themselves in the market as a provider of specialised design services, and
3. The company is seeking to acquire software to improve their competitiveness in design on the basis of time-to-market, performance and product cost improvements.

This research addresses the needs of the last strategy and is focused on those aspects of design that are performance related. The acquisition problem is concerned fundamentally with matching the functions and characteristics of the software to the activities of a successful CE design organisation. The acquisition decision is then based on whether or not a particular software alternative is likely to contribute to the effectiveness of a concurrent product development strategy.

2. Software Support for CE Design

The Task-Technology Fit (TTF) model depicted in Figure 1 focuses on the fit between tasks and technology and the link they have to user performance [5,6]. The TTF model is useful in representing the important elements of the software acquisition problem. While any performance assessment is dependent on the degree of utilisation and user attitudes towards the software, this can be assumed to be a secondary focus to that of matching technology to task requirements. The primary reason for this is that the utilisation of engineering design software can for the most part be assumed as the user is forced by virtue of his/her work role to utilise the software. The TTF model consists of the following aspects:

1. Task characteristics
2. Technology characteristics
3. Performance Impacts



In determining the appropriate tasks to focus on, the philosophy of Design Coordination (DC) can act as a useful framework [8]. Design Coordination suggests that providing the right information at the right time, in the right format, to the right person, for the right reasons and is a useful means of avoiding an over emphasis on the benefits of task concurrency. Within the DC framework, the different dimensions of a computational design support system can be given by the complexity of the:

1. Artefact,
2. Decision making,
3. Actors, and
4. Knowledge and sources.

These dimensions offer a top level for decomposing into tasks that can be assessed against the technology in accordance with the TTF model. The resulting task list will consist of the actions of the software users in transforming inputs into outputs within the design process [5].

Technology characteristics refer to those features of the software that fit the requirements of the tasks [5]. Such features include but are not limited to:

1. Data representations
2. File handling
3. Speed
4. Vendor issues
5. Ease of Use,
6. Reliability
7. Upgrades

Performance impacts are the expected consequences or outcomes of using the software within a PDP. Some of the more important performance impacts would be:

1. Reduction in iterations and rework,
2. Improved quality of analysis, information and decision making
3. Improved flow of information (right information at the right time implies effective storage and retrieval of data.
4. Inter-operability

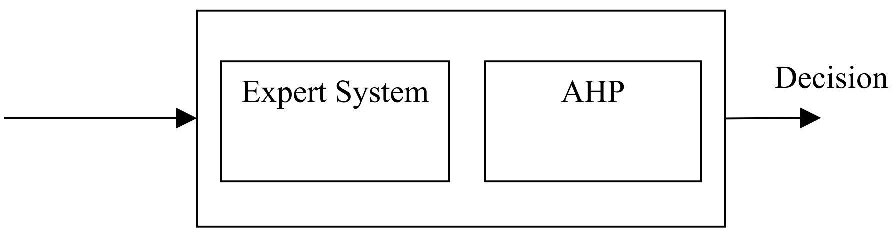
5. Appropriate information format for different users of design information, such as manufacturing engineering, (to avoid costly conversions),
6. Reuse of design knowledge (links to previous product data)
7. Job simplification
8. Speed up of existing tasks.
9. Improved product quality

The first eight of these performance factors centre on improving the responsiveness of actors in the PDP through integration [9]. A model for acquisition should encompass all of these areas as each contributes to achieving the effective integration necessary for CE to deliver its benefits. KBE and other design software tools play an important role in effecting integration in product design because they have the potential to provide higher levels of information and knowledge as well as an improved flow and quality of information. Improvements in this area can lead to a reduction of errors that when made early in the design process lead to additional rework [10]. This rework can be quite costly as it accounts for as much as 40% of avoidable design revisions.

While the TTF model is helpful in decomposing the problem into important elements, it is not in itself a decision making framework. A realistic solution of the problem requires us to ascertain the importance of particular tasks to an organisation. In fact, one of the more important variables in the design of a methodology for acquiring software for CE design projects is the relative impact particular CE strategies such as co-location and so on, have on schedule, cost (budget) and performance (quality) objectives. A suitable framework for the acquisition problem must first establish the relative importance of each attribute.

3. Proposed Framework

Having discussed some of the more important elements of the software acquisition problem, we turn our attention toward the specification of a suitable framework to represent the solution approach to the problem. While still highly conceptual, the problem appears to lend itself to a framework that combines the strengths of an Expert System (ES) with that of the Analytic Hierarchic Process (AHP) technique (Fig 2).



An ES architecture is designed to allow systems developers to develop applications that can emulate the problem-solving behaviour of a human expert within a particular discipline. An ES is one of a class of AI techniques that is able to capture the knowledge and reasoning of an experienced expert for re-use in assisting the less experienced in making decisions [11,12]. The role of the ES within this framework is to interrogate the user to establish the context of the decision making process. This approach takes into consideration the differences between organisations and allows for comparison and ranking of important CE attributes that influence the selection of the best software tool. These differences include the products that they design (machinery, aircraft, automobiles, white goods), the functions that they perform (innovative design versus derivative design) and the organisational structure that exists to support the design process and so on.

This context would include information that assists in defining a particular organisational domain such as:

1. Organisational structure;
2. commercial strategies;
3. core processes;
4. employee/resource characteristics;
5. design/product data; and
6. budget allocation for software.

The ES would interrogate the user via a question and answer session to establish a profile of the important contextual information needed to make a suitable acquisition decision. For example, an organisation that performs derivative design is more likely to benefit from generative product model capabilities than an organisation that is engaged in highly innovative design. It is this information that is supplied by the ES that can be used to rank criteria in the evaluation of software alternatives.

This decision making criteria can be structured by means of the Analytic Hierarchic Process (AHP) technique. The AHP technique is used to evaluate discrete options formulated from subjective and intangible criteria [13]. Using AHP, the decision problem can be clearly analysed and structured into a hierarchy that reflects the values, goals, and preferred attributes of the decision-maker .

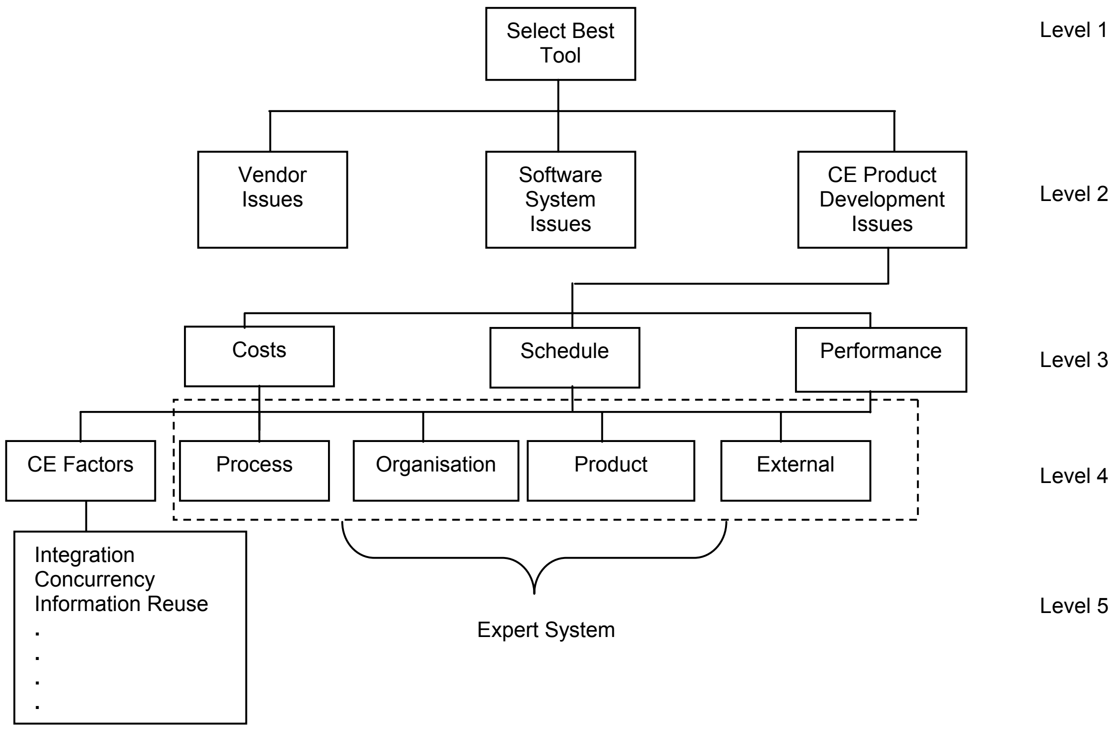


Figure 3   Preliminary Decision Hierarchy

A preliminary hierarchy, such as that given in Figure 3 is structured from the overall objective of selecting the best software tool at Level 1 and is further decomposed into the criteria that require consideration to accomplish this objective [14,15]. In this particular case, the CE Product Development Issues on Level 2 represent criteria specific to the organisation and the tasks that it engages in, while the Software System and Vendor Issues on the same level represent the technology criteria as discussed in relation to the Task-Technology Fit model. The cost, schedule and performance criteria at level 3 represent the primary determinants of a successful PDP. These may be further decomposed into the domain specific criteria represented by the Product, Process and Organisation categories. These criteria are expected to be handled by the ES and will be further decomposed as the research progresses. The CE Factors that reside at Level 4 represent those techniques prescribed by a CE philosophy that also influence the criteria on Level 3.

4. Conclusions and Future Directions

A conceptual framework for solving a software acquisition problem within the CE PDP domain has been presented. The differences between design organisations, the processes that they employ, and the products that they design, suggests a hybrid approach in constructing a framework that combines the strengths of an Expert System (ES) with that of the Analytic Hierarchic Process (AHP) technique.

We envisage that the next phase of research will focus firstly on the generation of questions for the expert system to establish the decision context. An equally important next step is a further refinement of the hierarchical structure with regard to software issues and tasks/technology attributes. Finally, an investigation to establish a method to link the ES to the AHP structure will be required.

5. References

[1] Zirger B.J., Hartley J.L., "The Effect of Acceleration Techniques on Product Development Time", *IEEE Transactions On Engineering Management*, Vol. 43, No. 2, May 1996, pp. 143-152

[2] Prasad B., "Concurrent engineering fundamentals: integrated product and process organisation", *Prentice Hall International Series in Industrial and Systems Engineering*, 1996

[3] Irwin J.G., Hoffman J.J., Lamont B.T., "The effect of the acquisition of technological innovations on organizational performance: A resource-based view", *Journal of Engineering and Technology Management*, Vol. 15, 1998, pp. 25-54

[4] Toupin, L.A., "Software spells profits", www.designnews.com, May, 21, 2001, pp. 31-32

[5] Goodhue D.L., Thompson R.L., "Task-Technology Fit and Individual Performance", *MIS Quarterly*, June, 1995, pp. 213-236

[6] Shirani A.I., Tafti M.H.A., Affisco J.F., "Task and Technology fit: a comparison of two technologies for synchronous and asynchronous group communication" *Information and Management*, Vol. 36, 1999, pp. 139-150

[7] Mark T. Dishaw, Diane M. Strong. (1999) Extending the technology acceptance model with task-technology fit constructs. Information and Management. Vol. 36 pp. 9-21

[8] Duffy A.H.B., Andreasen M.M., MacCallum K.J., Reijers L.N., "Design Coordination for Concurrent Engineering", *Journal of Engineering Design*, Vol. 4, No. 4, 1993, pp 251-265

[9] Ettlie J.E., "Integrated design and new product success", *Journal of Operations Management*, Vol. 17, 1997 pp. 33-55

[10] Molina A., Al-Ashaab A.H., Ellis T.I.A., Young R.I.M., Bell R., "A Review of Computer-Aided Simultaneous Engineering Systems", *Research in Engineering Design,* Vol. 7, No. 38, 1995 pp. 38-63

[11] Reidsema C.A., Szczerbicki E., "Review of Intelligent Software Architectures for the Development of Intelligent Decision Support Systems for Design Process Planning in

Concurrent Engineering", *Cybernetics and Systems: An International Journal*, Vol. 33, No. 6, 2002, pp. 629-658

[12] Klein M., "Expert Systems: A Decision Support Approach", *Addison-Wesley Publishing Company*, 1990

[13] Saaty T.L., "Fundamentals of Decision Making and Priority Theory with the Analytic Hierarchy Process", *RWS Publications*,1994

[14] Wei C.C., Wang M.J.J., "A comprehensive framework for selecting an ERP system" *International Journal of Project Management*, (In Press) (2002)

[15] Lai V.S., Trueblood R.P., Wong B.K., "Software selection: a case study of the application of the analytical hierarchical process to the selection of a multimedia authoring system", *Information and Management*, Vol. 36, 1999, pp. 221-232