PERGAMON

Computers & Industrial Engineering 44 (2003) 435–459

computers & industrial engineering

www.elsevier.com/locate/dsw

Decomposition of interdependent task group for concurrent engineering☆

Shi-Jie (Gary) Chen[a,*], Li Lin[b,1]

[a]*Department of Mechanical and Industrial Engineering, 303 Roberts Hall, Montana State University, Bozeman, MT 59717, USA*

[b]*Department of Industrial Engineering, 342 Bell Hall, State University of New York at Buffalo, Buffalo, NY 14260, USA*

Abstract

In concurrent engineering, project tasks generally involve the establishment of multifunctional teams in which team members from different functional departments interact in every phase of development tasks to design the products and processes concurrently. However, the increasing complexity of product development and design process often come up with large interdependent task groups due to the nature of the concurrent strategy. The large size of interdependent task groups makes it difficult for team organization and thus delays the project completion. This calls for the intention of this research to develop an effective model to: (1) transform the binary task relationships into the quantifiable task coupling strengths; and (2) to decompose the large interdependent task group into smaller and manageable sub-groups. Design structure matrix (DSM), analytic hierarchy process and cluster analysis are used to represent task relationships, quantify task couplings and decompose large size of task groups. Clustering performance between numerical DSM versus binary DSM is evaluated using a simulation experiment. The experimental results show that the clustering performance of using numerical DSM is better than the use of binary DSM. The effectiveness of this model is then demonstrated by an illustrative example. The result shows that our proposed model is capable of decomposing the large coupling task group that helps team organization for concurrent engineering project.

Keywords: Concurrent engineering; Analytic hierarchy process; Design structure matrix; Cluster analysis

1. Introduction

Complex products or projects with long development times always involve multiple disciplines and a great deal of efforts. For example, the design of an automobile can involve the coordination of hundreds

☆ Processed by area Editor Tony C Woo.

* Corresponding author. Tel.: +1-406-994-5942; fax: +1-406-994-6292.

E-mail addresses: gchen@ie.montana.edu (S.-J.(G.) Chen), lilin@eng.buffalo.edu (L. Lin).

[1] Tel.: +1-716-645-2357x2119; fax: +1-716-645-3302.

PII: S0360-8352(02)00230-9

or thousands of engineers who make more than a million design decisions over months or years (Eppinger, Whitney, Smith, & Gebala, 1990). The difficulties in designing complex products do not simply arise from their engineering complexity, but also stem from the organizational sophistication necessary to manage the design process. Especially in a concurrent engineering environment, the design needs to simultaneously consider various downstream activities throughout the entire product life cycle, in addition to meeting the functional requirements of the products. To accomplish this, multifunctional teams are usually organized with team members from different functional departments interacting in every phase of development tasks. It has been suggested that teams are more successful if their members are able to fully communicate with one another, because interactions among team members strongly affect the success of product development (Barczak & Wilemon, 1991; Girffin & Hauser, 1992; Prasad, 1996). Further, the effectiveness of communication depends on the number of communication links among team members. Therefore an effective and efficient communication will be difficult as the team size increases (Carmel, 1994; Chung & Guinan, 1994; Clark & Wheelwright, 1992; Johnson & Johnson, 1991; Lanigan, 1994). However, the increasing complexity of the product development and design process often results in large interdependent task groups due to the nature of the concurrent strategy. The large size of interdependent task groups usually makes it difficult for team organization and thus delays the project completion. This calls for the intention of this research to develop an effective model to decompose the large interdependent task group into manageable sub-groups. The objectives of this research are two folds:

(1) To transform the binary form of task relationships into the quantifiable task coupling strengths.
(2) To decompose the large interdependent task group into smaller and manageable sub-groups.

2. Related literature review

2.1. Identification of coupling information—design structure matrix

To identify the independent (uncoupled), dependent (decoupled), and interdependent (coupled) information flows inherent in complex processes, design structure matrix (DSM) proposed by Steward (1981a,b) is a useful tool. Much research has demonstrated its effectiveness in the last decade (Eppinger et al., 1990; Eppinger, Whitney, Smith, & Gebala, 1994; Gebala & Eppinger, 1991; Guo, Cha, Fang, & Woo, 1995; Kusiak, Larson, & Wang, 1994; Kusiak & Wang, 1993; Kusiak, Wang, He, & Feng, 1995; Rogers, 1989; Tang, Zheng, Li, Li, & Zhang, 2000). DSM divides the design project into n individual tasks in an $n \times n$ matrix. Similar to adjacency matrix, DSM is a square matrix with n rows and columns, and m non-zero elements, where n is the number of nodes and m is the number of edges. If there exist an edge from node i to node j, the value of element ij is a unity or a marked sign in the matrix, otherwise the value of the element is zero or empty. Information links among individual tasks are clearly shown by the systematic mapping, regardless of number of tasks. The advantages of using DSM are: (1) DSM overcomes the size and complexity limitations of digraphs; (2) DSM is easy to understand and able to handle the processes in their entirety; and (3) the matrix format is suitable to program and calculate using computers.

Fig. 1(a) shows a binary DSM. However, other than the input or output relationship between tasks, no structure can be seen in this matrix. According to Steward's (1981b) partitioning algorithm, the original