Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2001), *15*, 3–20. Printed in the USA.

A collaborative design process model in the sociotechnical engineering design framework

STEPHEN C.-Y. LU AND JIAN CAI
The IMPACT Research Laboratory, University of Southern California, Los Angeles, CA 90089-1111, USA

(RECEIVED January 12, 2000; ACCEPTED June 20, 2000)

Abstract

Collaborative engineering design involves various stakeholders with different perspectives. The design process is relatively complex and difficult to handle. Various conflicts always happen among the design tasks and affect the design team performance. Therefore, to represent the collaborative design process and capture the evolution of design perspectives in a structured way, it is critical to manage the design conflicts and improve the collaborative design productivity. This article provides a generic collaborative design process model based on a sociotechnical design framework. This model has a topological format and adopts process analysis techniques from Petri Nets. By addressing both the technical and social aspects of collaborative design activities, it provides a mechanism to identify the interdependencies among design tasks and perspectives of different stakeholders. Based on this design process model, a methodology of detecting and handling the design conflicts is developed to support collaborative design coordination.

Keywords: Collaborative Design; Conflict Management; Design Process; Petri Nets

1. INTRODUCTION

The increasing complexity of modern production makes the design process more and more difficult to handle since numerous technical and social issues are involved. The design activities are influenced not only by the technological factors, but also by the interactions among various stakeholders with different perspectives. To deal with this challenging problem requires an effective collaborative design process model that can clearly depict the characteristics of collaborative design activity and provide methodologies to improve design productivity.

There are many established approaches dealing with different aspects of engineering design process. They can be generally classified into three groups. The first group, which is mainly from the engineering discipline, focuses on generating formalized design methodologies by investigating how the technical design decisions are made. The design process models are often implied in these design theories and methodologies, such as the Systematic Design Model (Paul & Beitz, 1996), Axiomatic Design Model (Suh, 1990), QFD (Hauser & Clausing, 1988), General Design Theory (Yoshikawa, 1981), etc. Basically, these design methodologies provide the guidelines for a designer to make technical decisions more consciously and systematically (Jin & Lu, 1998). The second group views the design process as a workflow with task dependencies and information-exchange. The approaches in this category are mainly from the research of business operation and project management. From this aspect, engineering design is viewed as an information-driven process among design activities (Krishnan, 1997). The design organization is viewed as a stochastic processing network in which engineering resources are "workstations" and design tasks are "jobs" that flow among them (Sanvido & Norton, 1994; Adler & Mandelbaum, 1995). Accordingly, a set of techniques to manipulate the design activities has been developed, such as Signal Flow Diagram (Eppinger, 1997), Design Structure Matrix (Smith & Eppinger, 1997), and Design Process Network (Bras & Mistree, 1991). Besides the above two groups, several research approaches from CAD and CAE areas view collaborative design as individuals and groups accessing data and sharing the design information. Design process is accordingly specified as the managing of the product data in different abstraction levels. During this process, the technological, scientific, and interdisciplinary dependencies of the infor-

---

Reprint requests to: Jian Cai, The IMPACT Laboratory, Suite 101, Denney Research Building, University of Southern California, Los Angeles, CA 90089-1111, USA. E-mail: cai@usc.edu

mation could be established and maintained by handling the product model (Majumder et al., 1994). The information systems built by them are used to support the storage and processing of various types of data of interest to designers (Sriram et al., 1992; Krishnamurthy & Law, 1997).

These three classes of approach focus on different aspects of design and provide considerable contributions for understanding engineering design. Design theory research provides a clearer picture of design rationale and the decision-making process. Design activity manipulation characterizes design operations and identifies the dependencies of design tasks in the organization. Design data management supports information acquisition and storage typically in design automation. However, it is noticed that these established approaches have their own limitations when applied in collaborative design. They usually assume that the perspectives of different stakeholders (i.e., all of the human participants who have influences toward the design process and the product features) are independent and do not address the impact of their social interaction. They either ignore the social features of design or assume designers are purely rationale and simplify their preference as utility values. The design process is accordingly viewed as a series of pure technical activities, and the key issue "who" (e.g., the various people involved in the design, their distributed knowledge, their social networks, etc.) is not explicitly addressed. In fact, it is impossible to completely share knowledge and purpose among designers in collaborative design. Rather than being pure rationale, the stakeholders have an optimal or satisfied degree of consensus, which provides the desirable design result. Although the design methodologies and the workflow management techniques are applied, designers still face some failures of coordination due to their perspective differences and the inefficient design process management.

Therefore, a more comprehensive view is required to clarify the relationships among various technical and social aspects of collaborative design. A collaborative design process model based on this perception will generate effective coordination mechanisms for task planning, for scheduling and monitoring, for detecting and managing design conflicts, and for tracking and controlling design roles of stakeholders. This article describes a generic collaborative design process model based on a sociotechnical design framework, which is suitable to represent, analyze, and evaluate the collaborative design activities. We use Petri nets as topological process representation tools and adapt them for collaborative design process modeling. A methodology of design conflict management is developed with the design process representation model. The outline of this article is as follows. In Section 2, we discuss the fundamental issues of collaborative design process modeling and introduce a sociotechnical design process modeling architecture. Then the basics of the collaborative design process representation model are described in Section 3. Section 4 presents a methodology to manage design conflict by using the collaborative design process model. After that, a prototype collaborative design support system, which is a computer implementation of the methodology, is discussed. At the end, Section 6 offers conclusions and future research issues.

2. CHARACTERISTICS OF COLLABORATIVE DESIGN PROCESS

The central objective of engineering design is to achieve the prospective artificial objects having desired properties. An artifact can be thought of as a meeting point—an "interface" between an "inner" environment, the substance and organization of the artifact itself, and an "outer" environment, the surroundings in which it operates (Simon, 1996). During the design process, it is the design stakeholders' task to define the features within the inner environment of the product (e.g., form, structure, and behavior), which should be appropriate to its outer environment. Due to the involvement of human beings, the design process is not only based on the natural law of the artifact but is also affected by people's goals, skills, and circumstances. Therefore, within the collaborative design process, design information is driven by social, technological, scientific, and interdisciplinary dependencies. The design process therefore should be modeled by revealing the complicated relationships among these dependencies. We proposed a sociotechnical design framework to address the fundamental characteristics of a collaborative design process (Lu et al., 2000).

2.1. Design coordination in technical decisions and social interactions

In collaborative design, the stakeholders participate in the design campaign with both technical roles and social roles. Based on their roles, the ways stakeholders understand design and manipulate their activities are not uniform. They usually adjust the attitudes based on the feedback from others. The design process thus consists of not only technical decision making but also social interaction. By making technical decisions based on their technical roles, design stakeholders create, modify, and evaluate the product features. Because of the involvement of social roles, which are normally influenced by the organization structure, norm, and culture, technical decisions are coupled with social interactions during the design cooperation. The typical technical decisions in design include the activities to define the product characteristics, such as developing the function structure for the product, searching or generating the product infrastructure options, and assigning the values to the design parameters. While the technical decisions are dealing with "what" and "how," the social interaction, which is about "why" and "who," is indispensable to the negotiations among the collaborative design decisions. During social interactions, the stakeholders usually collectively define the item meanings and criteria, acquire the knowledge and experience, adjust their goals, and change their positions in the organization. Meanwhile, these interactions will change stakeholders' perspective and affect their technical decisions. Thus, the collaborative design process becomes more complicated than individual design.

To simplify the design problem, it is common to decompose it to small tasks, which are often assigned to different individuals separately. Although some design methodologies suggest that designers increase the probability of success by maintaining the independence of subproblems (e.g., Axiomatic Design Model), it is difficult to achieve this in collaborative design due to the various technical and social dependencies among tasks. On the other hand, individuals normally have limited capability to identify the influences of their decisions to others. Due to lack of coordination effort, the meanings about design objects might not be defined well, especially at the conceptual design stage. All of the above makes the decomposition and integration of design subproblems a rather complicated analyzing and synthesizing process. It is necessary to have a tool to support their coordination during the early design stages. In collaborative design, the task decomposition and integration must be achieved not only through the communication of contents, but also through the communication about the creation and evolution of shared meanings. The shared meaning is always defined by the interaction of design perspectives. That reveals one of the essential aspects of collaborative design process modeling, which is to represent and manage the interactions among the individuals' perspectives. In other words, design coordination relates to not only the dependency identification among the design decisions, but also the management of changing and interaction of the design stakeholders' perspectives. In collaborative design processes, the influence of one's decision making in a specific domain to others' decision making in different subproblems should be represented and evaluated. Furthermore, the design process representation model has to help design stakeholders to detect and evaluate the interdependencies among their design activities and to solve conflicts. Besides keeping the product data integrity, a design information system should provide the "language" or "medium" for design participants to declare and depict their perspectives and aid their communication. These will definitely affect the current way of organizing the design team and design process. To achieve these, it is critical to generate a design process representation model, which can facilitate the describing, tracing, and management of collaborative design interactions by referencing to design perspectives.

2.2. Design perspectives

Defining design perspective is one of the essential issues in design process modeling. In collaborative design, stakeholders' perspectives can be visualized as "lenses" they wear during the design process. Each stakeholder has his/her unique viewpoints and circumstances, which further define their roles in the design campaign. For the same object, they may have different perceptions and make different decisions. In the sociotechnical framework, a perspective of a stakeholder is defined as the combination of his/her purposes, contents, and contexts. The "purposes" involve one's intentions toward various issues in design. There are different levels of purposes in a person's mind, which are more complex than "design criteria" or "function requirement." The "contexts" are the circumstances (e.g., the stages of design, one's position in the organization, etc.) around a stakeholder during design. The "contents" contain the information (e.g., product specification, management decision, etc.) that a stakeholder will generate under his/her purposes and contexts. In a collaborative design process, the data that each individual produces, or exchanges through any medium (e.g. computer, lecture, text), is the external manifestation of his/her internal knowledge, appropriately filtered through his/her "perspective lens."

A collaborative design process is also a perspective evolution process. At the start of the design, the "what," "how," "when," "where," and "why" are interpreted differently by different perspectives. During designing, the participants and the organization interact together and build the shared reality (Berger & Luckman, 1966) (i.e., the institutional understanding of the world) in the social interaction process. While a technical design process model (e.g., Axiomatic Design Model) may serve as a basis or starting point for technical decision making, it is always dynamically adapted and modified by the participants during the course of the design campaign. The design perspectives of the stakeholders are affected and the shared reality is formed while the function, form, and behavior of the product are being defined. It should be pointed out that most of the conflicts in the collaborative design are caused by the discord among the stakeholders' perspectives. Hence, to represent the perspectives of the stakeholders and investigate their influence on the design process is indispensable in design process modeling and conflict management. Although "perspective" is critical for managing design interactions, traditionally it is in a person's mind and is not explicitly modeled in the design methods. To overcome this limitation, we need to have effective perspective models to help designers manage the design process and handle conflicts with respect to design perspectives.

2.3. Sociotechnical design process architecture

The sociotechnical design process architecture provides a more comprehensive view to model the collaborative design process (Lu et al., 2000). In Figure 1, the different elements and their relationships in collaborative design are clearly depicted. From the sociotechnical viewpoint, **technical decisions**, **social interaction**, and **conflict management** are the three essential components in the collaborative design process. Normally, the characteristics of the design problem and the existing design environment predetermine the stakeholders' technical and social roles. During the de-

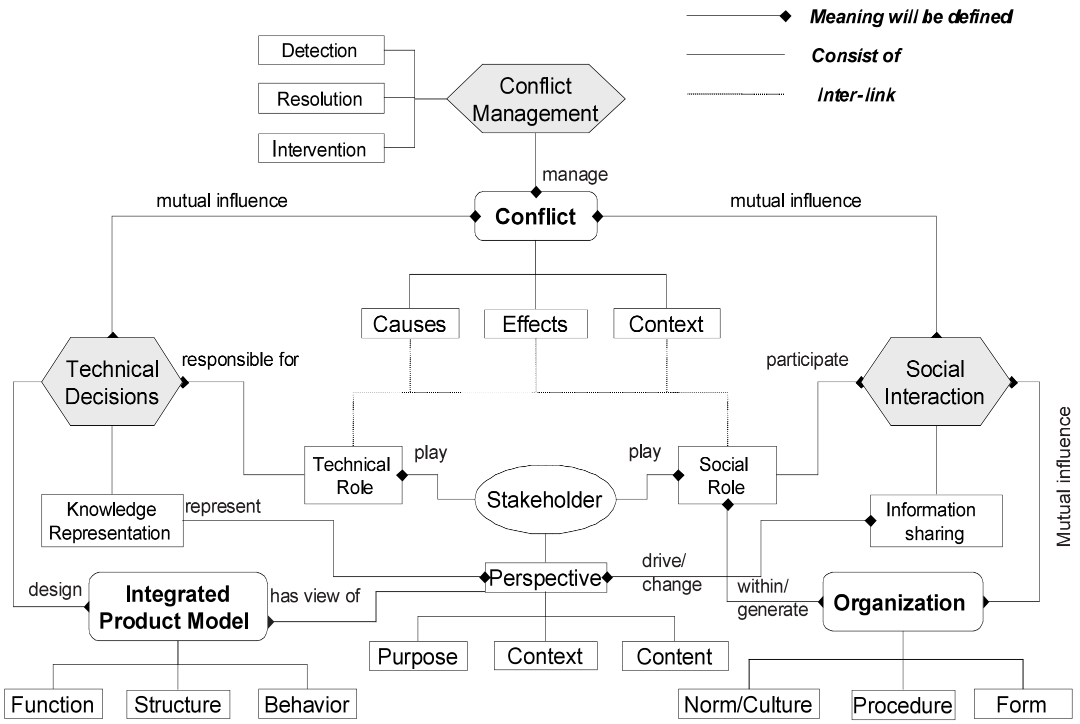


Fig. 1. The sociotechnical design process modeling architecture.

sign process, their roles are adaptive while the design perspectives evolve. That will also go back to influence the design process. Knowledge representation (e.g., CAD drawing, ruled-based system, etc.) is critical for designers to capture the understanding and reasoning behind technical decisions. Effective information sharing mechanisms (e.g., group discussion, brainstorming, information management system, etc.) accelerate the process of achieving agreement on the shared perspective. During technical decision and social interaction, various types of inconsistencies will occur. Inconsistency in the product data provided by different individuals can be viewed as conflict relating to the product specification level. That is one generic form of conflict focused on by most current conflict management approaches. Incompatibilities in the design perspectives may imply the knowledge conflict between different stakeholders. Since perspectives are inside humans' minds, this kind of conflict is relatively difficult to detect and represent. Handling conflict only in the technical domain without considering the design perspectives is insufficient since the critical causes are ignored. To manage conflict near its source, social interaction should be considered as a controllable infrastructure to affect and handle the design perspectives.

The sociotechnical architecture provides the guideline for us to develop models to represent the process of collaborative design. In the following sections, we first discuss a generic process representation approach (Section 3). It applies Petri Nets as the process modeling tools and thus has topological features suitable for calculation and analysis. Then, in Section 4, we describe the perspective models, which explicitly capture and represent the negotiation and interaction pattern among the stakeholders' design perspectives. When working with perspective models, the process model can represent and keep track of the state of collaborative design in the sociotechnical context (e.g., the solved problem and the coming problem, the changing product model, and the evolving perspectives of different stakeholders). Accordingly, conflict management methodologies can be further developed by utilizing these models.

3. REPRESENTATION OF THE COLLABORATIVE DESIGN PROCESS

There are various available tools for engineering process modeling, such as the Project Evaluation and Review Technique (PERT) (Wiest & Levy 1977), State-transition Diagram, and Signal Flow Graph (Eppinger, 1997). These tools have some limitations when they are used in collaborative process modeling. The PERT method is widely used for identifying the critical path of the process and estimating the completion time, but it does not support representation of iterations in the process. The State-transition Diagram is popular in logic design and object-oriented modeling. One of its major disadvantages is that one has to define all of the possible states of the system. The Signal Flow Graph provides a clear representation of design iterations, but it does not specify the presence of the stakeholders in the process. Our approach uses a modified Petri Nets model to represent the design activities and the coordination among stakeholders. Petri Nets have the unique advantage of supporting process specification, representation, and evaluation at the same time (David & Alla, 1992). Also, their mathematical properties help us in quantitatively analyzing the behavior of the design process. Furthermore, elementary Petri Nets have a simple graphical appearance, which can become a convenient and precise language for communicating among design stakeholders. However, it should be noticed that the collaborative design process is relatively complicated and unstructured compared with other process systems [e.g., computer code (Jenson, 1996), manufacturing system]. Some modifications are necessary to make Petri Nets more suitable and effective for design process modeling. In this section we introduce some basic definitions and their applications in representing the collaborative design process.

3.1. Definitions

A Petri Net graph represents a general process with two types of nodes named "places" and "transitions." Directed arcs join some places to some transitions. Each place may contain one or several tokens represented by dots. The following transitions of one place can only be executed when the required tokens are available. A weight can be associated with each transition, which is a positive number. The marking of the Petri Net is a vector that contains the values of marking in all places.

In the collaborative design process model, "place" and "transition" are equal to "event" and "task," respectively. A design process is represented by an organization of events and tasks. The weights of the tasks can be used to represent their resource consumptions. The default value of the weight is one. The arcs represent the transform directions between events and tasks during the design. The token denotes the state of each individual event. An event contains a token if and only if it is active (i.e., event is happening). Thus the whole state of the design process can be expressed by a marking M, which is a vector having the token numbers of each event in the design process. Since different stakeholders can conduct tasks, we introduce the "stakeholder" into the notation. Each task and event has a set of stakeholders associated. Formally, a Collaborative Design Process can be represented by a Petri Net graph with the following definitions.

DEFINITION 1. A **C**ollaborative **D**esign **P**rocess **N**et (**CDPN**) is a six-tuple $CDPN = (E, T, S, A, W, M)$ with a set of labels:

where

$E = \{e_1, e_2, \ldots, e_n\}$ is a finite set of design events,

$T = \{t_1, t_2, \ldots, t_q\}$ is a finite set of design tasks,

$S = \{s_1, s_2, \ldots, s_m\}$ is a finite set of design stakeholders,

$A \subseteq \{(E \times T) \cup (T \times E)\}$ is a finite set of directed arcs connecting event and task,

$W : T \to \{w_1, w_2, \ldots, w_p\}$ is a weight function attached to the design tasks,

$M_0 : E \to \{0,1,2,\ldots\}$ is the initial marking. ■

As shown in the example (Fig. 2), a portion of the building design process is represented in a graph with the above elements. To explicitly address the stakeholders in the design process, each event and task has a set of stakeholders associated. We use S_1, S_2, S_3, S_4 to denote project manager, design consultant, market surveyor, and architect, respectively, which are marked on top of the events and tasks. At the beginning of the design, the tokens are only contained in the beginning events (E1 and E2). After stakeholders

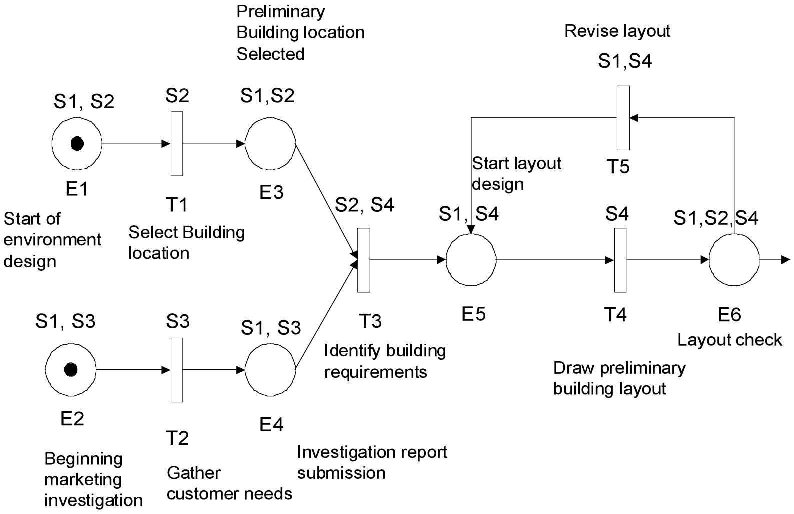


Fig. 2. An example of a Collaborative Design Process Net.

preform the tasks, the tokens from the upward events can be transferred to the downward events. M_0 is defined as the initial marking of a CDPN, which is a vector containing the token number for each event. For instance, at the beginning M_0 equals $[1\ 1\ 0\ 0\ 0\ 0]$, since only events 1 and 2 possess tokens. If M_0 equals $[0\ 0\ 0\ 0\ 0\ 1]$, that means that all of the tasks shown in the graph have been conducted, since the token is only presented in the last event.

The input and output relationships between task and events are denoted as

where

$^\circ t$ = the set of input events of task t, (i.e., the set of $\{e|(e,t) \in A\}$),

t° = the set of output events of task t, (i.e., the set of $\{e|(t,e) \in A\}$),

$^\circ e$ = the set of input tasks of event e, (i.e., the set of $\{t|(t,e) \in A\}$),

e° = the set of output tasks of event e, (i.e., the set of $\{t|(e,t) \in A\}$).

It is clear that finishing a task t consists of transforming the initial marking M_0 of the CDPN into a new marking M_{i+1}. Firing a task $t \in T$ includes two operations, which are removing a token from each $e \in {}^\circ t$ and adding a token to each $e \in t^\circ$ (assuming each arc has weight one). It could be formally defined as follows.

Definition 2. A task can be fired in a state M_i iff $\forall e \in {}^\circ t : M_i(e) > 0$. The firing of a task leads to the next state M_{i+1}, which can be calculated by

$$M_{i+1}(e) = \begin{cases} M_i(e) - 1 & \text{if } e \in {}^\circ t \\ M_i(e) + 1 & \text{if } e \in t^\circ \\ M_i(e) & \text{otherwise.} \end{cases} \tag{1}$$

■

Thus, the execution of a design process is represented by a task firing sequence $\sigma = \langle t_1, t_2, \ldots \rangle$, which relates to a transformation of the marking $M_0 \to M_1 \to M_2 \to \ldots$.

The process incidence matrix $U = [u_{i,j}]$ is defined to represent the relationship between tasks and events in a CDPN.

Definition 3. An incidence matrix $U = [u_{i,j}]$ is defined over all of the events $E = \{e_1, e_2, \ldots, e_n\}$, and the tasks $T = \{t_1, t_2, \ldots, t_q\}$ where

$$u_{i,j} = \begin{cases} 1 & \text{if } t_j \in {}^\circ e_i \\ -1 & \text{if } t_j \in e_i^\circ \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

■

For example the $(n \times q)$ incidence matrix of the above graph is

$$U = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}.$$

The relationship between state transformation and incidence matrix can be expressed in the following transformation equation:

Proposition 1.

$$M^T = M_0^T + U \cdot V_\sigma^T \tag{3}$$

■

In Eq. (3), $V_\sigma = [\nu_1, \nu_2, \ldots, \nu_q]$ is a counting vector for a task-firing sequence σ with the following definition.

Definition 4. The counting vector of firing sequence σ is defined as $V_\sigma = [\nu_1, \nu_2, \ldots, \nu_q]$, where ν_i is the number of tasks t_i included in σ. ■

Given the firing sequence $\sigma = \langle$T1, T2, T3, T4, T5, T4$\rangle$ in the example, its counting vector V_σ equals $[1\ 1\ 1\ 2\ 1]$. Based on Eq. (3), the final marking can be calculated as follows.

$$M^T = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$M = [0\ 0\ 0\ 0\ 0\ 1]$ shows that only Event 6 is active, which means the process shown in the graph might be finished.

The task dependencies are also easily identified from a CDPN, which is denoted by task dependence matrix $D = [d_{ij}]$. If one of the output events of task i is within the set of task j's input events (i.e., task i is immediately in front of task j), we call this situation "sequential dependency." Another situation is that two tasks are sharing the same input event or output event, which is named "joint dependency." In both cases, its dependency factor is set to 1. Otherwise, it is said that there is neither sequential nor joint dependency between the two tasks. In a task-dependency matrix, it is easy to identify the critical tasks (e.g., T3) that relate to many other tasks.

DEFINITION 5. A task dependency matrix $D = [d_{ij}]$ is defined over all of the tasks $T = \{t_1, t_2, \ldots, t_q\}$ where

$$d_{i,j} = \begin{cases} 1 & \text{if } t_i^\circ \cap {}^\circ t_j \neq \varnothing \\ 1 & \text{if } ({}^\circ t_i \cap {}^\circ t_j \neq \varnothing) \vee (t_i^\circ \cap t_j^\circ \neq \varnothing) \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

■

Also, to represent the assignment of stakeholders' tasks from the CDPN, we define a task assignment matrix as follows.

DEFINITION 6. A task assignment matrix $H = [h_{ij}]$ is defined over the stakeholder set $S = \{s_1, s_2, \ldots, s_m\}$ task set $T = \{t_1, t_2, \ldots, t_q\}$ with the value

$$h_{i,j} = \begin{cases} 1 & \text{if } t_j \in \{t \mid s_i \text{perform } t\} \\ 0 & \text{if } t_j \notin \{t \mid s_i \text{perform } t\}. \end{cases} \tag{3.6}$$

■

For example, the task dependence matrix and task assignment matrix of the above **CDPN** can be derived as:

$$D = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \qquad H = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

3.2. Concurrency and choice

It should be noted that although two tasks may have no direct linkage (i.e., $d_{i,j} = 0$), they might still have indirect dependencies since one may transfer its influence through other tasks. For instance, although the dependency factor $d_{2,4} = 0$, the output of T2 will still be embedded into the input of T4. Thus, the task dependency matrix only shows the direct linkage relationships among the tasks in the process. If two tasks are parallel (e.g., T1 and T2 in Fig. 2), they are allowed to be concurrently executed. The firing sequence σ is changed to $\langle(\text{T1, T2}), \text{T3, T4, T5, T4}\rangle$ to denote the concurrent execution of T1 and T2.

To represent the degree of concurrency of a design process, we define the concurrency ratio of a process as follows.

DEFINITION 7. The concurrency ratio of a design process is the proportion of the paralleled tasks in a firing sequence of **CDPN**.

$$Rc(\sigma) = \frac{N_p}{N_\sigma}, \tag{3.7}$$

where N_p is the number of tasks which are in parallel and N_σ is the number of tasks in σ. ■

For $\sigma = \langle(\text{T1, T2}), \text{T3, T4, T5, T4}\rangle$,

$$Rc(\sigma) = \frac{N_p}{N_\sigma} = \frac{2}{6} \approx 0.33.$$

In a CDPN, we prescribe that one event cannot simultaneously initiate two tasks. The situation of a design event e with more than one task in its output set e° indicates a "choice situation." For example, task E6 in Figure 2 may have two output tasks (one is T5 and the other is not shown in Fig. 2). In this case, only one of the tasks can get the token and be fired at one time. In design process, an event with "choice" represents a selection of following tasks (e.g., a decision point). The choice in a CDPN sometime implies design iteration, which might be caused by conflict or reworking. Given the possibilities of the options for each choice and the required time for each task, the execution time of the whole design process can be estimated by other process simulation techniques (e.g., signal flow analysis methods (Eppinger, 1997)).

3.3. Task decomposition

In collaborative design, concurrency is normally encouraged, since parallel task execution may reduce the design time and save resources. However, the perspective differences and communication faults will cause contradictions among the concurrent tasks. For instance, in original design, the lack of experiences and knowledge usually becomes a major obstacle of concurrent task execution. Even for routine design, as concurrency increases, failure of coordination will raise conflicts and damage the whole design process. Thus, there is a trade-off between task parallelism and minimizing coordination effort. There are two approaches dealing with this problem. One approach is to use design methodology to reduce the dependencies among design tasks. For instance, Axiomatic Design Theory suggests "decoupling" or "uncoupling" the product function requirements so that design tasks can be more independent. The other approach is to effectively support the communication and manage the conflicts among design tasks. The first approach is focusing on task decomposition and the second is considering task coordination.

A design task can be decomposed based on various issues, such as the features of the product, organization structure, or designers' disciplines. The sociotechnical design process architecture emphasizes the importance of three groups of activities in collaborative design, which are technical decision making, social interaction, and conflict management. If only the technical aspects of design are considered, the task of technical decisions could be carefully decomposed by selection of uncoupled functional requirements and design parameters so that the subtasks are relatively independent. However, tasks of social interaction and conflict management are relatively complex and are highly coupled. When conducting these tasks, people do not have precise predictions of the effects of their decisions. Then, in collaborative design, it is impossible to totally remove the interdependencies among the design tasks. Therefore, coordination among design tasks appears to be critical to support a successful design process.

A design process model can be derived at different abstraction levels. The stakeholders with expertise toward a certain design task can further decompose a task. Then, a hierarchy of process diagrams can be built. For example, T2 and T4 can be expanded to more specific tasks and events by different stakeholders (Fig. 3). Whether to expand a task or not depends on the complexity of the process and requirements of the stakeholders. The objective is to illustrate the process to a certain detail level so that the differences of stakeholders' perspectives are easy to identify and design conflict can be detected.

3.4. Process planning and scheduling

Design process planning is particularly critical to design collaboration, since the assignment and arrangement of design tasks will affect the quality of product and the cost of the design process itself. Various approaches are provided to address design process planning issues. They generate the design plan based on norm, by separating product parts, by identification of critical tasks, or by noticing information dependencies. One of the popular approaches adopted in engineering project management is using the product working structure as the basis for decomposing the task and organizing the design process (Kerzner, 1998). However, at the conceptual design stage, this product-driven planning is not sufficient. Its applicability is limited because product features are, in fact, defined and changed during design by group decision making. One could hardly work on a component of a product without interaction with others. The evolving perspectives of the stakeholders will always require the adaptation of the product and design process.

One of the essential objectives of design planning should be realization of stakeholders' perspectives (i.e., their purpose, context, and content during the design process). Especially when the stakeholders are not familiar with each other at the beginning of the design, to realize their roles in the design process is an indispensable step in design planning. Then the way in which perspective evolution affects the product specification process can be determined. After the realization of design perspectives, refining the design methodologies applied and evaluating resource consumption become possible. Besides the plan, short-term schedules are also necessary. They are different from the design plan since they specifically focus on the dependencies among subsystems. In the CDPN graph, the design schedule can be represented by interconnected Petri Nets with the coordination explicitly expressed. Information dependencies are implied in the task and event linkages. During the execution of the design process, individual stakeholders face more granular process networks, which are built based on the de-

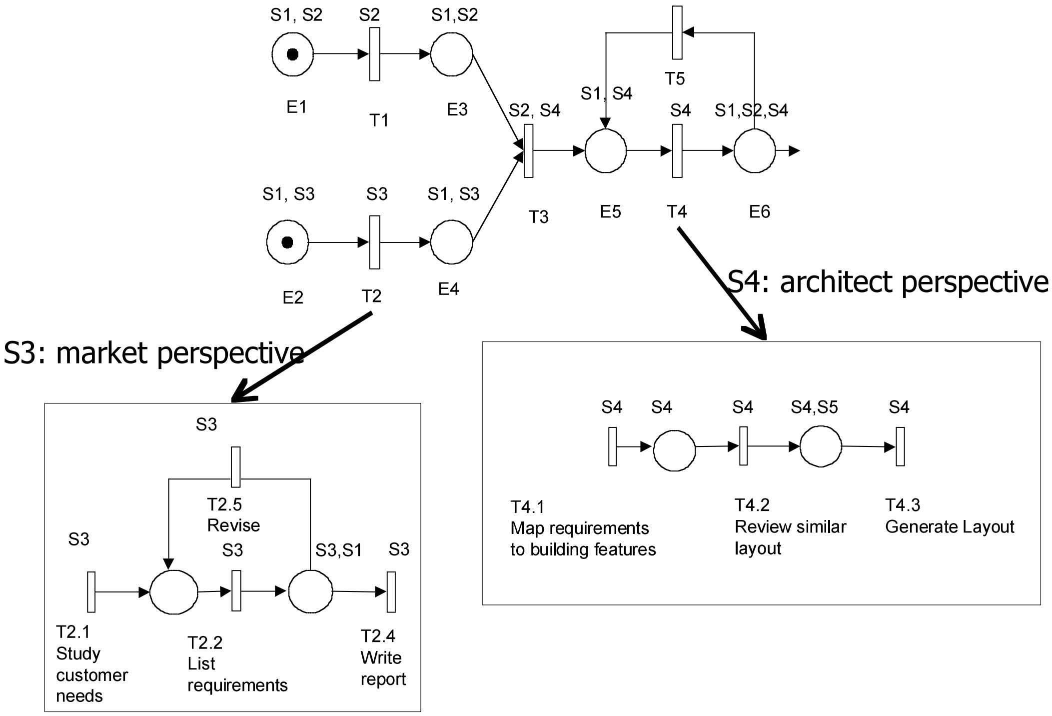


Fig. 3. Representation of task decomposition in the design process.

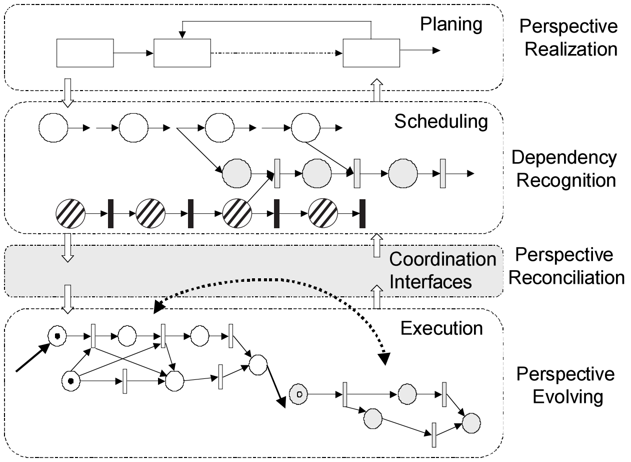


Fig. 4. Design plan, schedule, and execution.

sign schedules. Discovery of conflicts and failures of the previous design process reveals the deficiencies of the design plan and schedule. According to the feedback from the design task execution, the design plan and schedule for the next period are recomputed and applied. The design process continues in this “rolling-horizon” manner until the end.

Therefore, our model represents the collaborative design process in three levels with different considerations (i.e., planning, scheduling, and execution). The design perspectives should be seriously considered in each level (as shown in Fig. 4). Since we view the collaborative design process as a perspective-evolution process, a coordination interface to clearly capture and support perspective reconciliation is vital to design process management. In the following section, we introduce a conflict management methodology, which is to support perspective reconciliation in the design process.

4. SUPPORTING CONFLICT MANAGEMENT IN DESIGN PROCESS

Conflict can be treated as a significant issue to identify perspective dependencies, to drive idea interaction, and, therefore, to improve the design process. Traditionally, quite a few approaches are proposed to handle conflicts in design by modeling conflict as the multiobjective decision problem (Kannapan & Taylor, 1994; Kraus et al., 1995; Petrie et al., 1995; Lewis & Mistree, 1998). Most of them assume that design stakeholders are purely reasonable and their preferences can be represented by utility functions. However, utility theory has intrinsic limitations on conflict management and collaboration support (Binmore, 1987). The critical reason is that in collaborative design, the meanings and concepts are defined during the interaction rather than before the interaction. Many conflicts are actually caused by the insulated concepts in different perspectives. Only after the meanings are defined and shared among the stakeholders, can utility theory take effect to handle conflict. To address this issue, we take a sociotechnical approach to manage conflict by manipulating the design perspectives.

4.1. Overview of the methodology

Design conflict as a dynamic situation has its causes, contexts, and effects (Wall & Calister, 1995), which could be of a technical nature, a managerial nature, or a social-interaction nature. Conflicts of various types at different abstraction levels might occur when inconsistent local realities (i.e., individual meanings) are merging to a shared reality. To achieve a satisfying performance of the design team, conflicts should be effectively managed by investigating, understanding, and manipulating the perspectives of stakeholders. When treating engineering design as a purely technical process, conflicts are usually regarded as being abnormal and to be avoided as soon as possible at all costs (Kannapan & Taylor, 1994; Klein, 1995; Peña-Mora et al., 1995). On the other hand, when treating engineering design as a sociotechnical process, conflicts might be systematically and explicitly dealt with as a resource to drive the social construction process and design innovations. In the early design stage, conflicts can be treated as motivations to identify the deficiencies of the team and to generate new ideas, whereas at the late stage, conflicts should be prevented or resolved to achieve high efficiency.

A categorization of the different conflicts is derived from the sociotechnical framework. Its aim is to find mappings between the different types of conflicts in design and conflict management strategies that have been developed in the social, political, and organizational management literatures. It should be emphasized that conflict management may involve not just the detection, prevention, and resolution/extinction of conflict, but also the encouragement and control of conflicts in a desired manner. Of great significance is the development of tools to measure and monitor the “rate” at which conflict resolution occurs so that a confluence of viewpoints in the sociotechnical construction process can be achieved. Consequently, to manage conflict, we need to identify the roles of the stakeholder and understand their perspectives. Then the conflict could be diagnosed and its causes and context will be identified. By applying the intervention methods and adjusting stakeholders’ perspectives based on the analysis, the conflict process is controlled.

As shown in Figure 5, the methodology of conflict management in the design process can be viewed as a coordination interface between design plan and task execution. It has four basic steps:

1. Clarify design concepts and build the concept structure;
2. Generate the CDPN;
3. Create perspective state diagrams (PSDs) for each of the stakeholders;
4. Perform sociotechnical analysis and manipulate PSDs and CDPN.

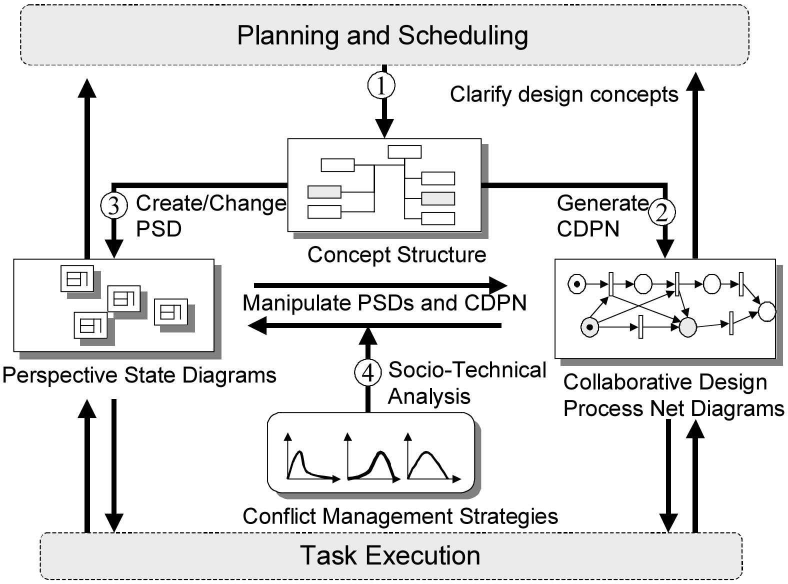


Fig. 5. Methodology to support conflict management in the design process.

In the following sections, the details of each step will be discussed by illustrating an architecture design scenario.

4.2. Building the design concept structure

To generate a "common language" in collaborative design, the stakeholders first collectively build a Concept Structure. The Concept Structure is a model to organize the social and technical ontologies perceived by the group. Building the Concept Structure begins with reading a concept structure template (e.g., the collaborative design architecture model shown in Fig. 1 can become a preliminary template). The template depicts most of the abstract concepts involved in the design group. It can be viewed as the preliminary shared perspective model structure organized by "contents" and their relations at the beginning of design. It also serves as a guideline for stakeholders to further specify their design concepts.

After identifying the critical concepts, the complicated relationships among these concepts can also be organized. A list of the most important concepts and their relationships is generated and viewed by all stakeholders. When an individual proposes a new concept, he/she should first consider whether there are similar concepts in the structure. Thus, only novel concepts can be specified and added. When stakeholders propose new concepts in the design process, the concept structure (Fig. 6) is updated and is used to systematically organize these concepts and their relationships. The concepts are often best generated by individuals, whereas the concept selection and enhancement are often best performed by the group. Therefore, we classified the concepts into two types. "Shared concepts" are those that have been well defined and have widely accepted meaning among the stakeholders. "Private concepts" are perceived only by some individual stakeholders. Their names or meanings are not expressed around the group. Most of the concepts in the concept structure template are shared concepts. As the stakeholders propose new concepts to the concept structure, more domain-dependent concepts are involved and are viewed only at the individual level. Whether a concept is shared or not is relative to the purpose of a certain design group. If a group of people have similar purposes regarding a concept, it would be better to share it. Sometimes, a concept is not shared between two groups, but may be shared within one group. After the concepts are identified, the dependencies among these concepts can be further clarified. For instance, the concept "function requirements" in a technical decision will influence the "function" of the product. The "structure" of the product is decided by the "design parameter" of the design methodology. During design, the statuses of concepts are evolving. When many stakeholders have interest about a private concept, they can update it to a shared concept. That means their perspectives are converging.

4.3. Developing a process model

A design process model (i.e., CDPN diagrams) can be generated from a preliminary design plan or a process template, which can be simple descriptive words or PERT diagrams. At the beginning of the design, a plan could be very informal and may omit many real-world necessary activities. Additional work has to be done to transfer the informal description to structured forms. Consulting the design group about detailed information is sometimes necessary to clarify important issues.

In the architecture design scenario, there are seven types of stakeholders considered: Sponsor, Client, Design Con-

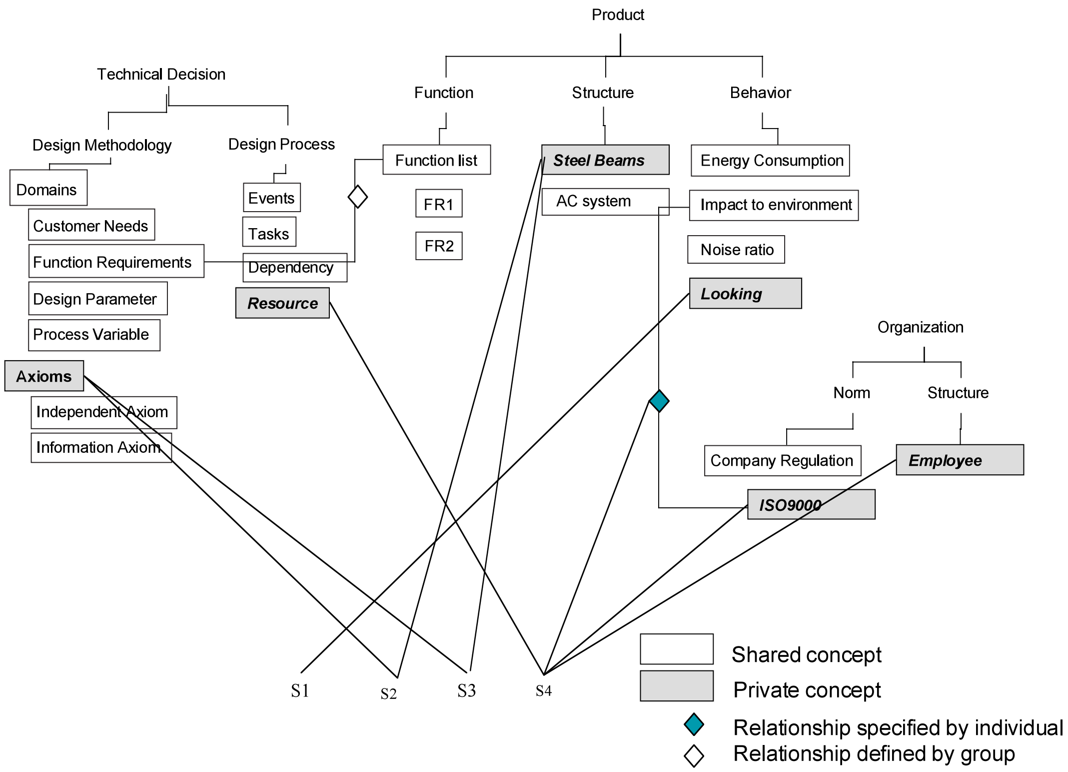


Fig. 6. A concept structure built by stakeholders.

sultant, Architect, Engineering Consultants, Building Authority, and Building Users. They have various perspectives and play various roles in the design process. Their perceptions of design tasks and events are intensively different at the beginning of the design. Figure 7 shows the CDPN diagram covering the schedule of preliminary design and the early part of conceptual design. To make the stakeholder more explicit, tasks and events are arranged in rows with each stakeholder assigned. Shared tasks and events are shown in all of the related stakeholders' rows and indicated by dotted-line linkages. It is clear that stakeholders S2 and S3 conduct most of the preliminary design tasks, while others' roles are to provide related information for their decision making. The incidence matrix, task dependencies matrix, and task assignment matrix are generated from the above CDPN.

In the CDPN, design iteration is depicted as a circular linkage from a choice event to a previous task. In practice, a design iteration might also imply the occurrence of conflicts. If the opportunity of conflicts has been considered during scheduling, the possible process iterations should be explicitly expressed in the graph. Task dependencies and iterations are represented more clearly in the incidence matrix. As shown in Figure 8, the design iteration could happen if a loop dependency square exists in the incidence matrix. Three iterations are easily identified by finding loop dependency squares in that matrix. The critical choice events (e.g., E8 E15, E16) in the design process can be detected after recognition of design iteration. These events are viewed as the key nodes within the process graph.

4.4. Perspective states diagram construction

To analyze the evolution of the perspectives, we applied a systematic approach to capture the purposes, contexts, and contents of the different stakeholders. "Purpose" relates to the stakeholders' goals or concerns about a design concept. "Content" is the contained information of the perspective, which generates messages to be communicated. "Context" places the information within the overall product life cycle and stakeholders circumstances. The principal objective of design perspective tracking is the recognition of the stakeholder perspectives and the identification of the means by which they transfer the content of these perspectives (i.e., "communicate") among themselves. A design perspective model can be represented by a structured format. For example, the Design Consultant has a perspective about forecasting the facility usage requirement, which can be described as in Table 1.

Our approach organizes and presents stakeholders' perspectives in collaborative design by using PSDs. A PSD represents the "view" of a stakeholder in a particular time of collaborative design. It consists of all of the perspective models of a stakeholder and can be visualized as a picture of the perspective status of one stakeholder. By referencing the concept structure, it is possible to ask the stakeholders

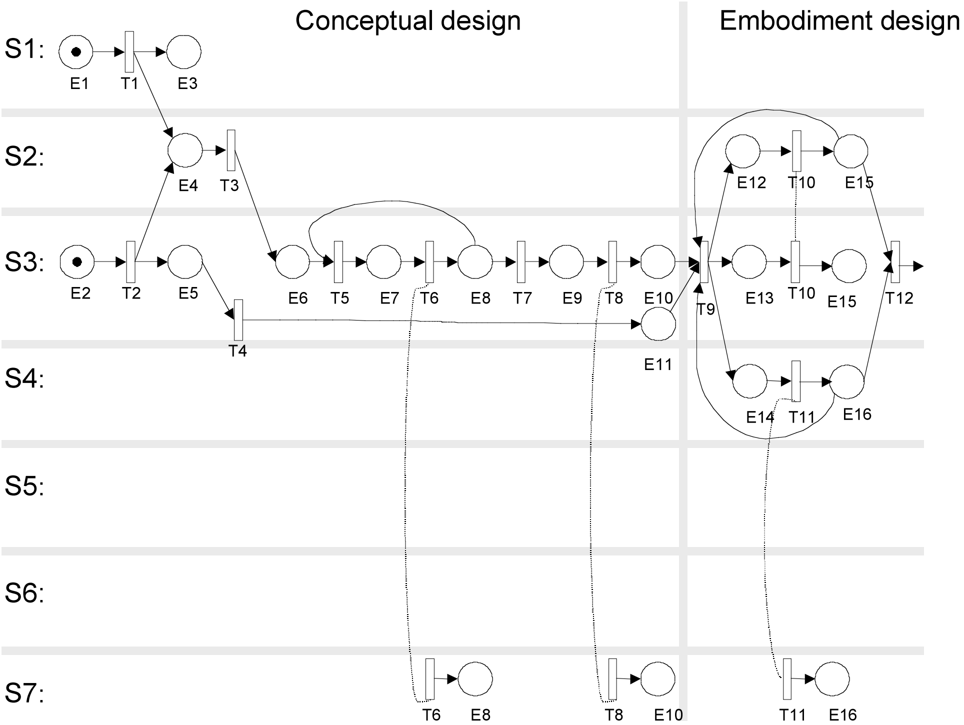
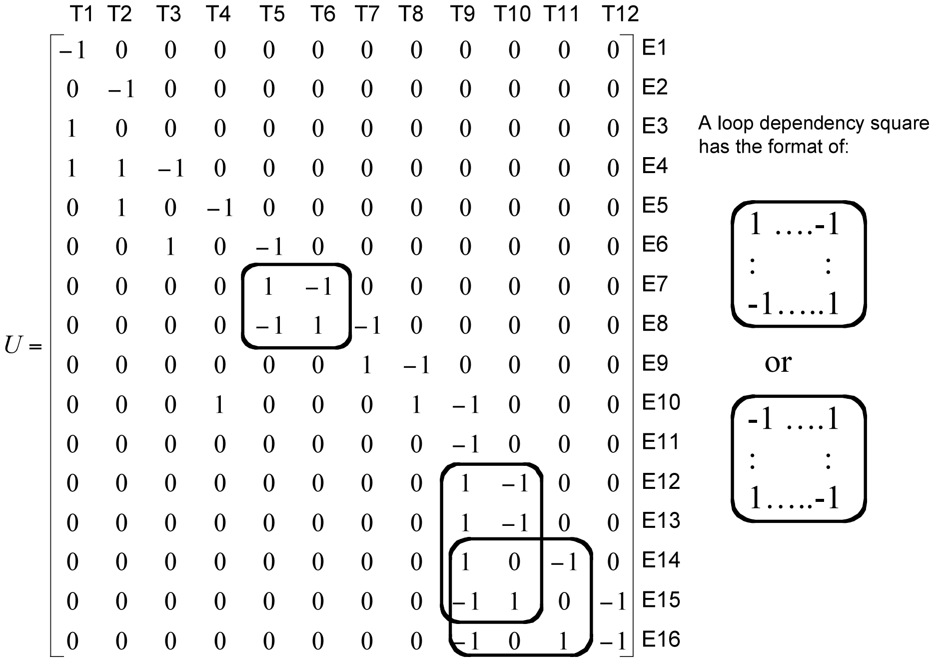


Fig. 7. CDPN of the design example.

to build the PSDs at a certain time. A stakeholders' PSD depicts the relationships among his/her concepts (including the shared concepts and private concepts) and his/her purpose and context. The concepts listed in the PSD are categories of perspective contents relating to stakeholders. They are not all information of the design stakeholders' perspectives. In fact, using concept structure in the PSD provides a way for us to systematically compare and exam-



$$U = \begin{array}{c|cccccccccccc|l}
 & T1 & T2 & T3 & T4 & T5 & T6 & T7 & T8 & T9 & T10 & T11 & T12 & \\
\hline
 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & E1 \\
 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & E2 \\
 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & E3 \\
 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & E4 \\
 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & E5 \\
 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & E6 \\
 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & E7 \\
 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & E8 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & E9 \\
 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & E10 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & E11 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & E12 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & E13 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & E14 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & -1 & E15 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & -1 & E16
\end{array}$$

A loop dependency square has the format of:

$$\boxed{\begin{array}{cc} 1 \ldots\ldots & -1 \\ \vdots & \vdots \\ -1 \ldots\ldots & 1 \end{array}} \quad \text{or} \quad \boxed{\begin{array}{cc} -1 \ldots\ldots & 1 \\ \vdots & \vdots \\ 1 \ldots\ldots & -1 \end{array}}$$

Fig. 8. Identification of design iteration of the design process in an incidence matrix.

Table 1. *An example of a perspective model*

Concept:	Product :: Functional Requirement :: Facility usage requirement
Purpose:	Project the usage schedule, demand, and use of the facility by the clients based on historical data and usage estimations of the units using the facility.
Content:	• Historical personnel load and mission data • Analysis procedures and parameters • Forecasts
Context:	• Lifecycle: requirements analysis • Information type: requirements data (demand forecast data), historical usage data

ine the perspective differences among stakeholders. For each of the concepts, there is a set of a purpose, a context, and a content associated with it. Figure 9 shows two stakeholders' PSDs and their relationships with the concept structure. Each of the boxes in the PSD shows the perspective information for a given concept.

In the collaborative design process, each stakeholder has serial PSDs that describe the adjustment and evolution of their perspectives. When looking through the boxes of different stakeholders' PSDs, related issues and inconsistencies can be noticed. The related issues of perspectives reveal the dependencies and links of the local realities of the stakeholders, while their inconsistencies imply conflicts. The dependencies can be used as anchor points to integrate the individual perspective models and to form a larger meaning community. The conflicts can be managed to reconcile the design perspectives. By tracking perspective state evolution, our approach provides a systematic and operational way to analyze the design process by identifying the dependencies and conflicts.

4.5. Sociotechnical analysis to manipulate design process and perspectives

After the PSDs and the CDPNs are derived, one can analyze and manipulate them to support perspective evolution and manage design conflict. Examining the PSDs between/through adjacent points in time provides a way to identify/reconcile conflicts in perspectives. By rearranging the design process to reconcile the contents of the perspectives, the design activity is handled to control the conflict manner. Thus, the efficiency of human design is improved by providing support to their negotiation.

The detail analysis methods are depicted in Figure 10. Given the process diagram, the concept structure, and the PSDs, their dependencies can be represented as several matrices (e.g., T-T, T-S, C-S, etc.). Controlling the interplay among these

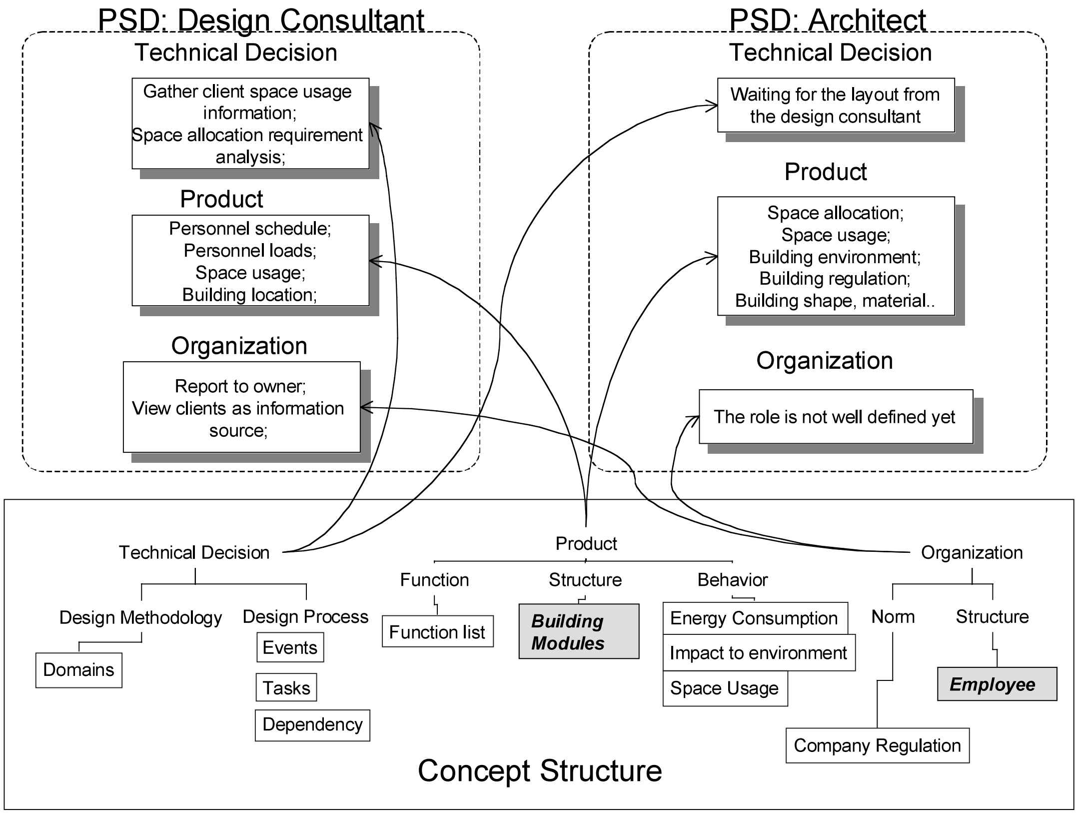


Fig. 9. Generation of design perspective models by concept structure.

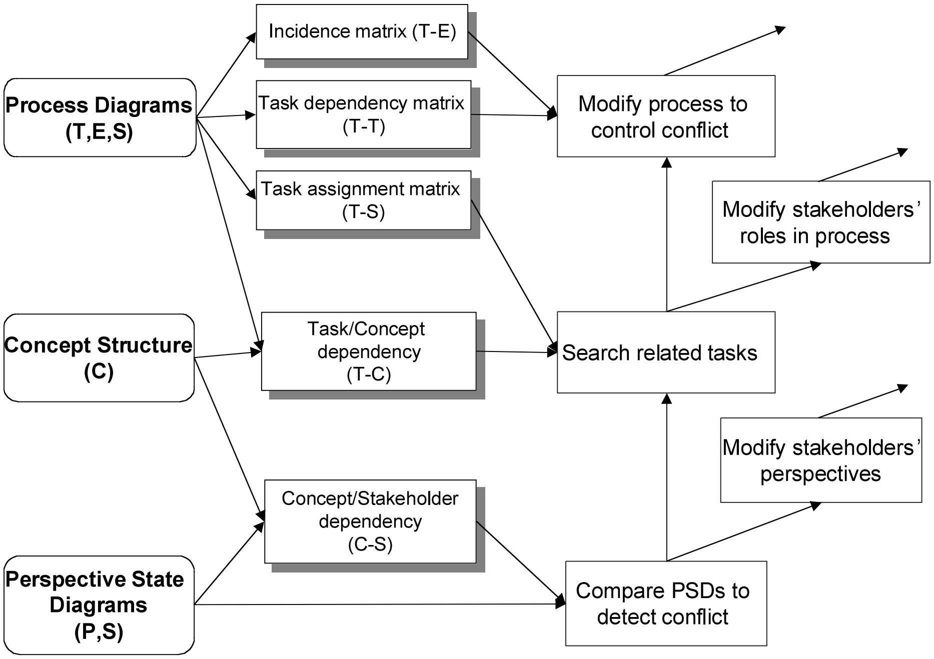


Fig. 10. The analysis methods.

three models provides various conflict management methods. At a certain design stage, the design conflict can be detected by tracking and comparing the "perspective states" of different stakeholders. If design perspectives are not tracked, due to the loss of coordination, the chances of noticing the inconsistency and dependence are relatively small. Then, some design deficiencies are not noticed until conflicts occur. For example, if tasks are executed according to the old schedule shown in Figure 7 and design perspectives are not tracked, the Architect (i.e., S4) and the Design Consultant (i.e., S3) might realize a fatal conflict on the location selection after they discuss detail features of the building (after Task 9 in the schedule). Since numerous jobs have already begun, that conflict will cause a lot of rework and waste time and money. By using the sociotechnical analysis, the perspectives models are tracked and the PSDs of the Design Consultant and Architect are compared (Fig. 11). It is easy for the stakeholders to notice the inconsistency in that design concept and treat it as a conflict. Although there is still no direct meeting between the stakeholders, this potential conflict on building location selection is identified much earlier.

The basic objective of these methods is to manipulate the PSDs in a way to converge them more quickly and therefore to resolve conflict. As shown in Figure 10, to manage the conflict, we can modify stakeholders' perspectives, change their roles, or rearrange the design process according to the attributes of the conflict. Since the patterns of PSDs will largely depend on the interactions among the design tasks, arranging the design process in a desired manner is an effective approach to coordinate the perspectives of the stakeholders. By searching on the dependency matrices, it is possible to identify the tasks that will affect this conflict. Then, the process can be modified to change stakeholders' perspective states. For instance, a new schedule can be proposed to let the Architect become involved in the design campaign earlier so that he can identify the key concepts in layout design. After analyzing the task assignment matrix and task dependency matrix, a new task T8.1 is added to let the Architect join the design and declare his concerns. Thus, the location decision conflict will be prevented. A comparison of the old and new design process is shown in Figure 12. The design iterations are reduced while the concurrency of the process is increased.

5. AN INFORMATION SYSTEM TO SUPPORT COLLABORATIVE DESIGN

Several critical issues that arise in developing collaborative design support systems can be identified based on the above discussion. First, it is necessary to have a system that can explicitly capture the perspectives of the stakeholders and assist their interactions. The representation of their views of product and organizations should capture individual interests. Stakeholders' goals, contexts, and contents should be modeled in the system in a structured way and be communicable to other stakeholders. Secondly, it is important for the system to trace the merging of perspectives in the design process. Furthermore, by referencing to the system knowledge base, it might be able to evaluate the potential consequences of stakeholders' decisions for collaborative design. Besides,

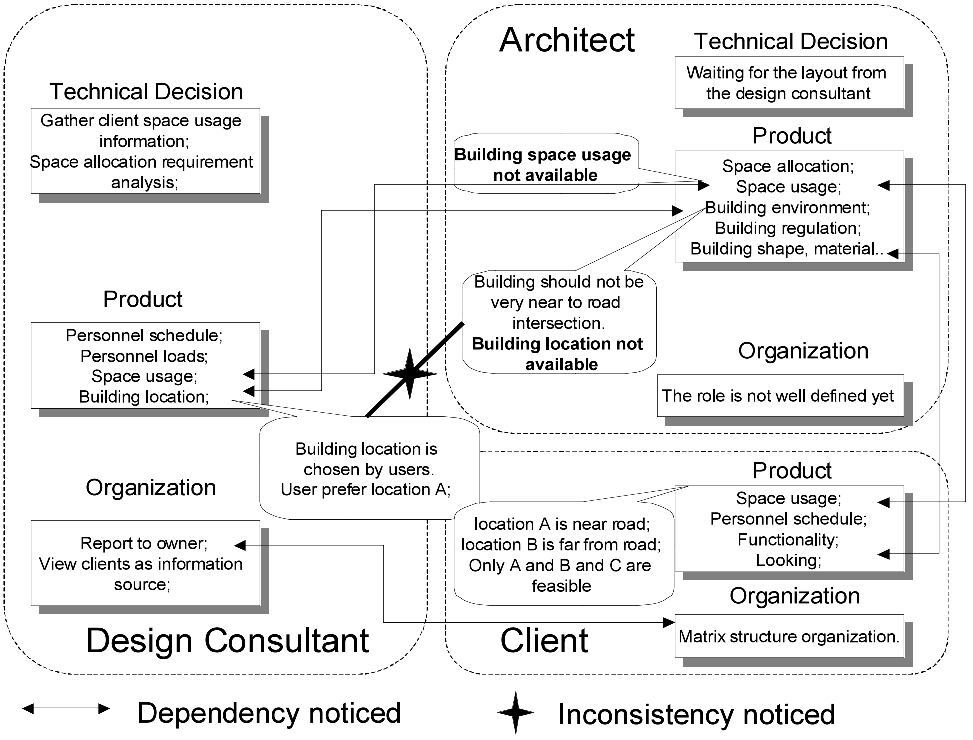


Fig. 11. Comparing perspective models of three stakeholders.

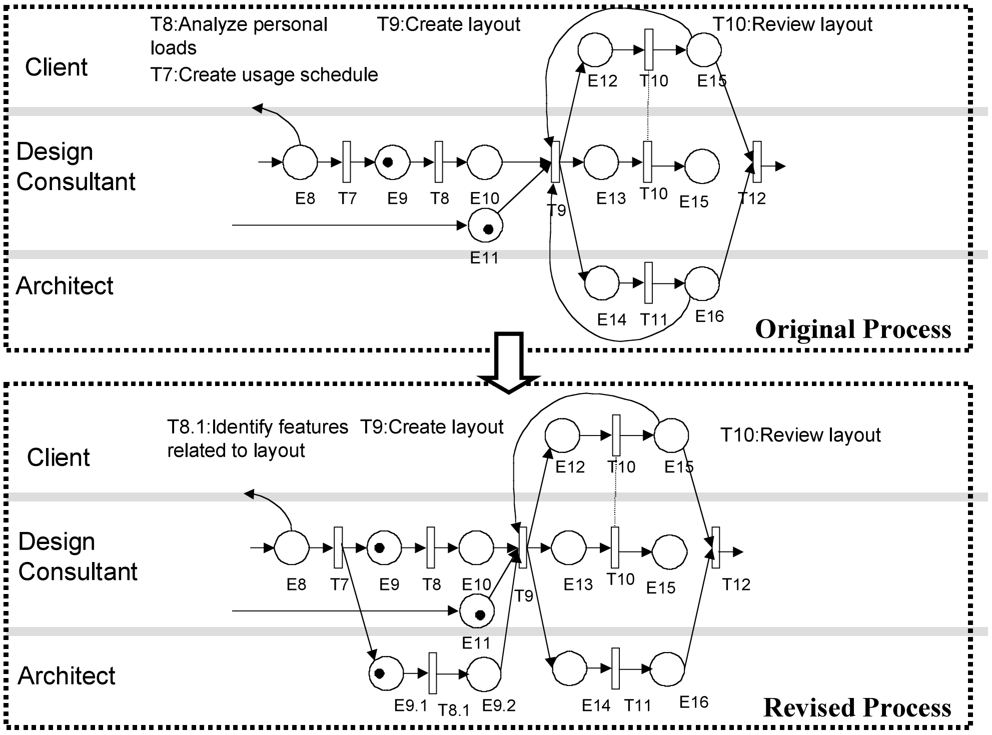
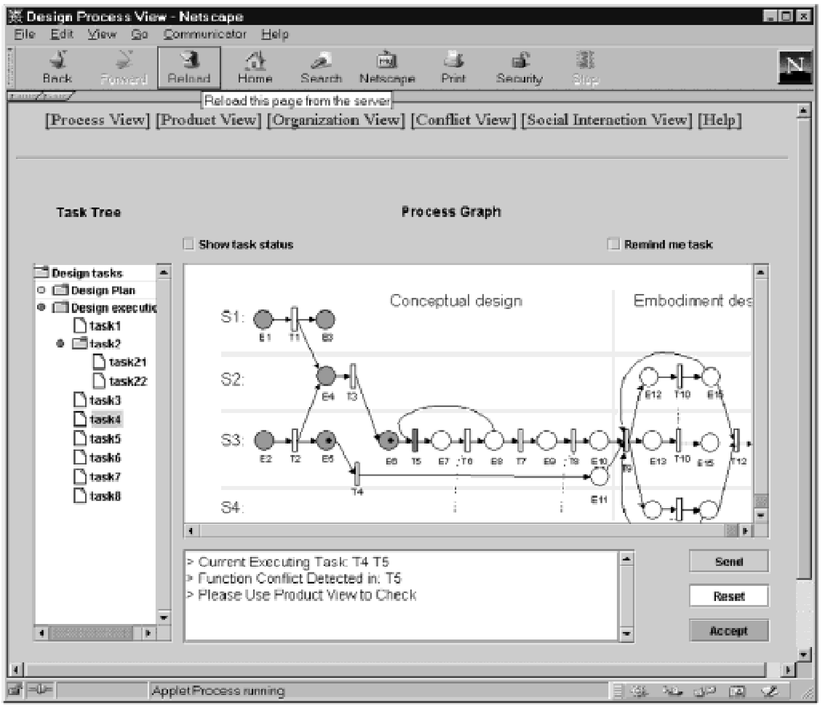


Fig. 12. Rearranging the design process to manage conflict.

it is also critical to support design conflict management by facilitating both technical and social negotiations.

The Sociotechnical Design Process Management system (ST-DPMS) is a prototype implementation of the methodology of manipulating the collaborative design process and controlling conflicts. It is quite different from the available information systems, which manage conflict by using the exception-handling approach (Klein, 1995) or by eliminating data inconsistencies (Sriram et al., 1992). The objective of the ST-DPMS is to provide a computerized environment that supports the sociotechnical coordination among stakeholders during conceptual design. During the design process, the system maintains stakeholders' design perspective models and depicts their various roles. Communication tools with network and server-client database access functions enlighten the stakeholders located in different places to notify his and others' perspectives. Several subsystems (e.g., Conflict Management, Process Management, Product Management, and Organization Management) are provided to support design interaction and manage design information. The system knowledge repository tracks the evolutions of product and organization data. These changes will become feedback to the perspective models of the stakeholders and influence the design process in the future.

The ST-DPMS has several unique characteristics. First, the integrated product model and process model are built on the information structures represented by the perspective models of the stakeholders. To explicitly capture the perspectives of the stakeholders and assist their interactions, interfaces with different contents are provided. Each stakeholder uses a set of unique Web-based interfaces to declare his/her perspective and access design information. They access the product data, the organizational data, and others' perspective models when they operate their own workspaces. Second, the ST-DPMS can help the group to refine the design process by referring to the conflict management strategies. As shown in Figure 13, the information of design tasks and the state of the design process are explicitly shown to the individual stakeholders. After the conflict management model analyzes the causes, effects, and contexts of detected conflicts, the stakeholder can apply the strategies (e.g., remove task dependencies, add new tasks, rearrange schedule, etc.) to manipulate the design tasks and control the conflicts. During the design process, the system maintains a socionetwork model to help people realize the dependencies among their perspectives (Fig. 14). It also forces the stakeholders to communicate the effects of their own decisions to the others and the group. Third, the ST-DPMS will record and trace the merging of perspectives in the design process. As shown in Figure 14, after conflict is detected, the history of interaction and the conflict profile is displayed. The evolving history of the concept structure is captured by the system. That will become very helpful to the future design. Fourth, when it is fully developed, the system can support the learning of the rationale of the technical design decisions and conflict resolutions during the design process, like some proposed systems such as iDCSS (Klein, 1995), and SHRAE-DRIM (Peña-Mora et al., 1995).



> Current Executing Task: T4 T5
> Function Conflict Detected in: T5
> Please Use Product View to Check

Fig. 13. The collaborative design process management interface.

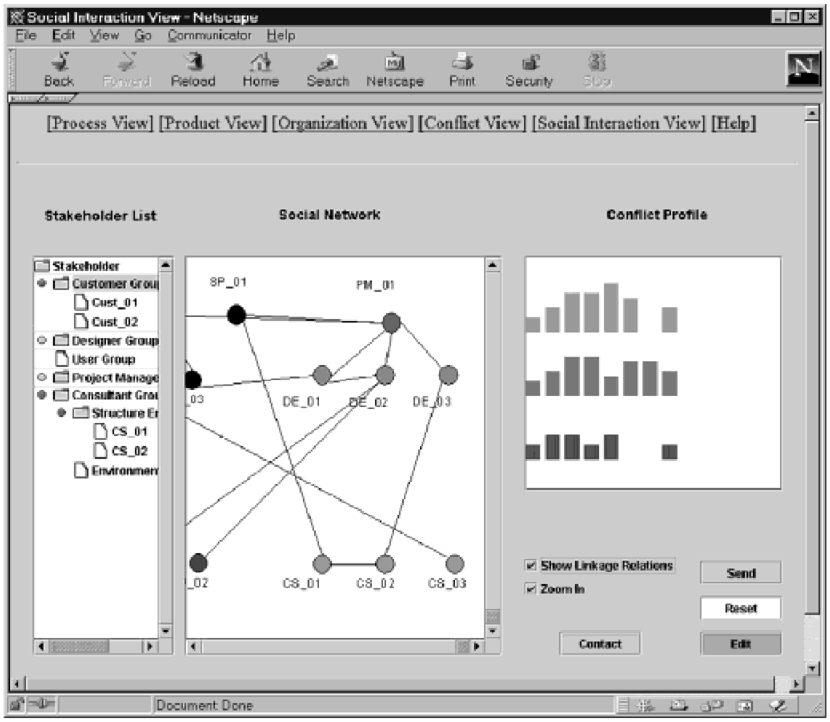


Fig. 14. The social interaction and conflict profile interface.

The dynamic perspective models are supposed to capture the stakeholders' local realities, while their evolutions in the design process can be also captured and proposed to the team. That means the system is not only able to learn the design expertise and the design rationale, but also can improve designers' recognition of the organization structure, norm, and culture.

6. CONCLUSIONS

This article presents an original methodology for representing and managing the collaborative design process in the sociotechnical framework. By using Petri Nets as convenient tools for topological and computational process illustration, a systematic representation method of the collaborative design process is developed. Several essential issues in collaborative design, such as design state transformation, task dependencies, and task decomposition, can be clearly expressed by applying the techniques of process modeling. This representation method provides the basics for the collaborative design process and conflict analysis. By investigating the relationship between perspective evolution and the structure of the design process, conflicts can be effectively detected and analyzed. Managing design conflict becomes a coordination infrastructure among design stakeholders to support the refinement of the design process. By using conflict management to identify the deficiencies of the design process, this methodology realizes a feedback control mechanism to manipulate the collaborative design process, while the existing approaches view it as an open loop system. The methodology also provides a framework for information system development for collaborative design support. The aim of the ST-DPMS is to provide not only the design process management facility but also an integrated information support system for collaborative design by capturing the perspective states of the stakeholders and systematically handling conflicts. Following this direction, a series of design process management methodologies can be derived and implemented.

Our future research work will further refine the collaborative design process model by applying the advanced analysis techniques of Petri Nets. Also, we hope to gain deep understanding of social interactions and their relations to technical decisions occurring in more real-life collaborative design cases. Based on the further understanding of the characteristics of design perspective interaction, the collaborative design process model and the design support system can be improved significantly.

ACKNOWLEDGMENTS

Partial funding for this research was provided by the Construction Engineering Research Laboratories of the U.S. Army, Corps of Engineers. We thank Dr. Mike Case, Mr. Eric Griffth, and Mr. Blessing Adeoye for their continuous support. We are also grateful for the help of Dr. F. Udwadia and Mr. W. Burkett, who have given us many incentives to write this article.

REFERENCES

Adler, P. & Mandelbaum, A. (1995). From project to process management: An empirically-based framework for analyzing product development time. *Management Science 41(3)*, 458–484.

Berger, P. & Luckman, T. (1966). *The Social Construction of Reality A Treatise in the Sociology of Knowledge*. Doubleday, New York.

Binmore, K.G. (1987). Why game theory "doesn't work"? In *Analyzing Conflict and Its Resolutions: Some Mathematical Contributions*, (Bennett, P.G., Ed.), pp. 23–42. Oxford University Press, Oxford.

Bras, B.A. & Mistree, F. (1991). Designing design process in decision-based concurrent engineering. *SAE Transactions Journal of Materials & Manufacturing 100*, 451–458.

David, R. & Alla, H. (1992). *Petri Nets and Grafcet: Tools for modeling discrete event systems*. Prentice Hall, New York.

Eppinger, D.S. (1997). Generalized models of design iteration using signal flow graphs. *Research in Engineering Design 9*, 112–123.

Hauser, J.R. & Clausing, D. (1988). The house of quality. *Harvard Business Review*, May–June, 63–73.

Jenson, K. (1996). *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*, Vol. 1, Springer, New York.

Jin, Y. & Lu, S.C.-Y. (1998). Toward a better understanding of engineering design models. In *Universal Design Theory*, (Grabaowski, H., Rude, S. and Grein, G., Eds.), pp. 71–86. Shaker Verlag, Aachen, Germany.

Kannapan, S. & Taylor, D. (1994). The interplay of context, process, and conflict in concurrent engineering. *Journal of Concurrent Engineering Research and Applications 2*, 183–196.

Kerzner, H. (1998). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. John Wiley & Sons, Inc., New York.

Klein, M. (1995). Conflict management as a part of an integrated exception handing approach. *AIEDAM 9*, 259–267.

Kraus, S., Wilkenfeld, J., & Zlotkin, G. (1995). Multiagent negotiation under time constraints. *Artificial Intelligence 75*, 297–345.

Krishnamurthy, K. & Law, H.K. (1997). A data management model for collaborative design in a CAD environment. *Engineering with Computers 13*, 65–86.

Krishnan, V. (1997). A model-based framework to overlap product development activities. *Management Science 43(4)*, 437–451.

Lewis, K. & Mistree, F. (1998). Collaborative, sequential and isolated decisions in design. *ASME Journal of Mechanical Design 120(4)*, 643–652.

Lu, S.C.-Y., Cai, J., Burkett, W., & Udwadia, F. (2000). A methodology for collaborative design process and conflict analysis. *The CIRP Annals 49(1)*, 69–74.

Majumder D., Rangan, M.R., & Fulton, E.R. (1994). Information management for integrated design environments. *Engineering with Computers 11*, 227–245.

Paul, G. & Beitz, W. (1996). *Engineering Design—A Systematic Approach*, 2nd ed., Springer, London.

Peña-Mora, F., Sriram, R., & Logcher, R. (1995). Conflict mitigation system for collaborative engineering. *AIEDAM 9*, 101–124.

Petrie, C.J., Webster, T.A. & Cutkosky, M.R. (1995). Using Pareto optimality to coordinate distributed agents. *AIEDAM 9*, 269–281.

Sanvido, V.E. & Norton, K.J. (1994). Integrated design-process model. *Journal of Management in Engineering 10(5)*, 55–62.

Simon, H.A. (1996). *The Sciences of the Artificial*, 3rd ed. MIT Press, Cambridge, Massachusetts.

Smith, P.R. & Eppinger, D.S. (1997). Identifying controlling features of engineering design iteration. *Management Science 43(3)*, 276–293.

Sriram, D., Ahmed, S., & Logcher, R. (1992). A transaction management framework for collaborative engineering. *Engineering with Computers 8*, 213–232.

Suh, N.P. (1990). *The Principle of Design*. Oxford University Press, Oxford.

Wall, J.A. & Calister, R.R. (1995). Conflict and its management. *Journal of Management 21(2)*, 515–558.

Wiest, J.D. & Levy F.K. (1977). A management guide to PERT/CPM. Prentice-Hall, Englewood Cliffs, NJ.

Yoshikawa, H. (1981). General design theory and a CAD system. In *Man-Machine Communication in CAD/CAM*, (Sata, T. and Warman, E., Eds), p. 35. North Holland, Amsterdam.

Stephen C.-Y. Lu is the permanent holder of the David Packard Endowed Chair in Manufacturing Engineering at the School of Engineering at the University of Southern California (USC). He is a tenured full professor at the Department of Aerospace and Mechanical Engineering, the Department of Industrial and Systems Engineering, and the Department of Computer Science. He heads the IMPACT research laboratory, and serves as the Director of USC's Center for Automation and Manufacturing Systems. Dr. Lu received his B.S. degree (1978) from the National Taiwan University, Taiwan, and M.S. (1982) and Ph.D. (1984) degrees form the Carnegie-Mellon University, Pittsburgh, PA. He has written or coauthored 185 technical publications, including 70 journal papers and 3 books. He is an active member of ASME, SME, IEEE, and AAAI. Presently, he serves as the Vice-Chairman of the Design Scientific and Technical Committee at CIRP.

Jian Cai is a research assistant in the IMPACT laboratory at USC. He received his B.S. degree in Precision Instrument and Mechanical Design from the Tsinghua University, China, in 1997 and is currently working on his Ph.D. in Mechanical Engineering at USC. His research interests include design process modeling, design conflict management, and Web-based design information system.