CONCURRENT ENGINEERING: Research and Applications

Towards a Computer-Supported Cooperative Environment for Concurrent Engineering

Biren Prasad*

Delphi Automotive Systems, Electronic Data Systems (EDS), General Motors Account, Automated Concurrent Engineering (ACE), P.O. Box 250254, West Bloomfield, MI 48325-0254, USA

Fujun Wang and Jiati Deng

Institute of Manufacturing Systems, Beijing University of Aeronautics & Astronautics, Beijing 100083, China

Received 10 March 1996; accepted in revised form 25 June 1997

Abstract: Computer-supported cooperative work has been the focus of attention in a concurrent engineering environment for some time. This paper studies this cooperative team environment from three perspectives: (a) the infrastructure of communication and information model viewpoint; (b) the planning and control of workflow viewpoint, and (c) some implementation viewpoints. Implementation viewpoints include constraints' management, negotiation, and memory capture and management. A set of enabling technology in each perspective area and its rationale are discussed. Finally, a structure of the whole system environment including definitions of an architecture and a global user interface are introduced.

Key Words: concurrent engineering, cooperation, computer supported cooperative work, collaborative environment, work-flow, CE architecture, product development.

1. Introduction

The traditional product developing process is mostly serial. This is often due to absence or lack of a firm team organizational structure, absence of a real-time communication environment, no shared information model for product realization and also due to lack of essential cooperating tools [1]. Those factors force teams in work-groups to do their tasks in isolation from each other. This gives teams less opportunity to exchange new ideas and assess their implications (Figure 1). For instance, the tasks of an upstream phase are completed long before feedbacks from a downstream phase are considered. This product realization mode of operation is prone to errors. It does not allow problem areas (during product realization) to be detected in earlier stages. Further, this can cause the product development cycle to take more time to complete the phases than necessary. Concurrent Engineering (CE) can be used to alleviate those problems. CE is a systematic approach to product design and development including the integration of design, production and related processes. CE considers all aspects of a product life-cycle at its outset. This new approach focuses on close cooperation amongst the work-group members to accomplish the product development tasks. Multidisciplinary team members work together in a computer networked environment [1–3], on a shared basis, towards a common set of consistent goals [1]. Using CE, there is a potential to improve the design efficiency, to avoid undetection of defects during earlier phases and thus shorten the product development life-cycle time.

Cooperation has been the lynch pin of Concurrent Engineering. Previously, a number of papers have looked into exploiting cooperation among distributed experts [4–8]. In those papers and projects, a number of technology areas as they relate to communication, information modeling, AI, strategic management etc., were studied. For instance, M. Klein [8] explains how an integrated cooperation can take place in a cooperative design.

A successful CE development environment needs a seamless communication channel, and also a mechanism to ensure that the team members do cooperate in a heterogeneous and distributed environment. The authors propose such a computer cooperative environment in this paper. The Section 2 discusses what types of cooperation modes are relevant to facilitate Concurrent Engineering. Section 3 introduces a set of supporting cooperative technologies to employ in this environment. Section 4 describes a system configuration of the whole system, and the last section wraps up this paper.

*Author to whom correspondence should be addressed. Email: bprasad@cmsa.gmr.com., (248) 265-7453.

1063-293X/97/03 0233–20 $10.00/0

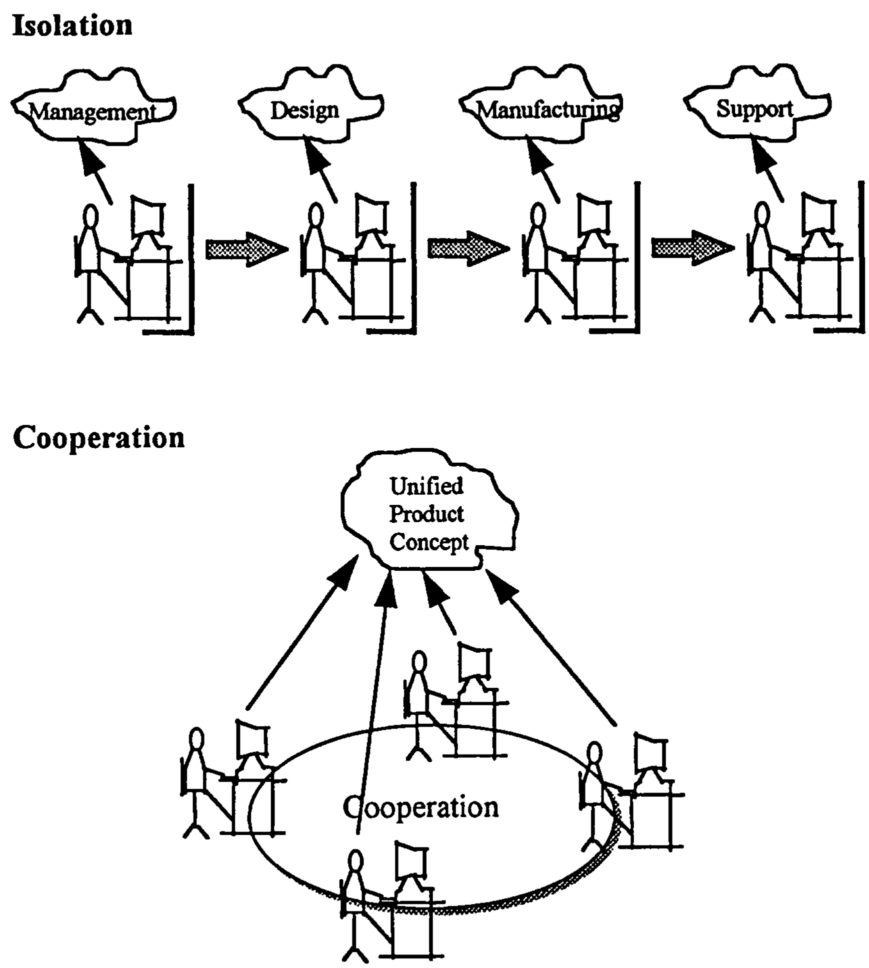

Figure 1. Isolation versus cooperation.

2. Cooperation in CE

Product Development Team (PDT) is an essential component of CE. The next section describes it briefly including the modes of cooperation that may take place amongst the PDT's sub-teams.

2.1 Product Development Team (PDT)

CE is commonly structured around multifunctional teams that bring in specialized knowledge necessary for the completion of a product or process development program [1]. A multidisciplinary setup—called a product development team—is composed of several distinct technical work-groups specializing in a variety of disciplines with titles such as Product planners, Engineers, Designers, Managers etc. (Figure 2). Each work-group is responsible for developing an aspect of the product and integrating it to the rest of the product's life-cycle aspects as the program develops. Figure 2 shows an organization of a PDT structure. The PDT structure is supported by the customers on the top and a company infrastructure (organization) at the bottom. It is sandwiched between a product and process layout on one side, and tools and technology support on the other side.

PDT replaces the traditional functional department into a cross-functional team structure organized along a set of goal-oriented principles. Experts coming from various fields work together as a coherent team of teams. For example, experts from a volume production area get involved in a prototype production to identify opportunities to improve process reliability as early in the realization process as possible. By using this multifunctional team of teams' approach to merge design and manufacturing, General Electric (GE) engine division, for instance, had reduced design and fabrication lead-time for some aero-engine components from twenty-two weeks to three weeks [1].

2.2 Cooperation Mode Among PDT Sub-Teams

Cooperation is the key lynch-pin of achieving team work. In a team, the project leader (PL) is responsible for the project planning, coordinating with the customers and the work-group members, and resolving conflicts. While experts—from the multifunctional work-groups—work with each other and share results among themselves and with the project leader (PL). Such interactions and cooperation are considered part of their job responsibility (Figure 3).

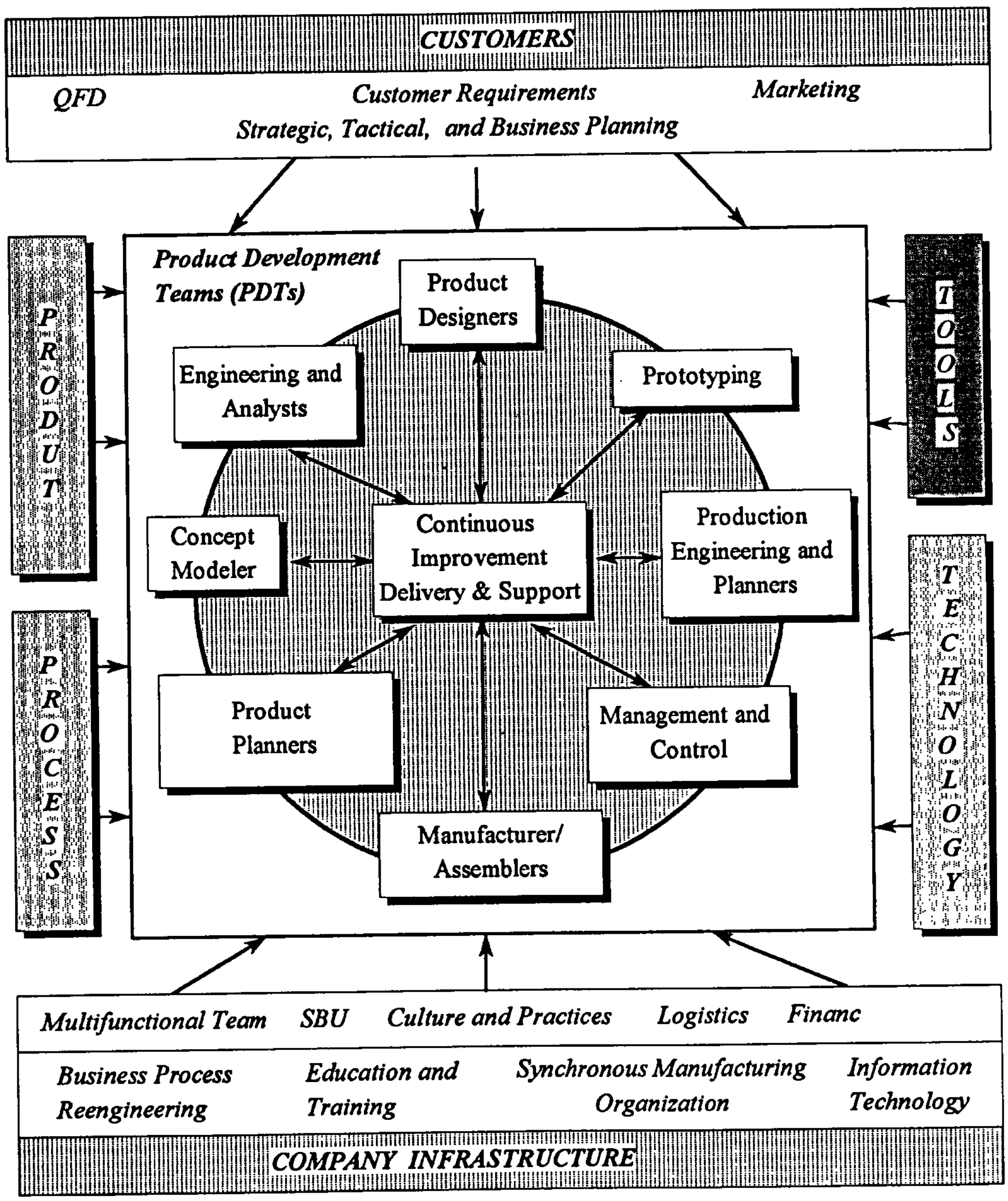

Figure 2. A typical product development team.

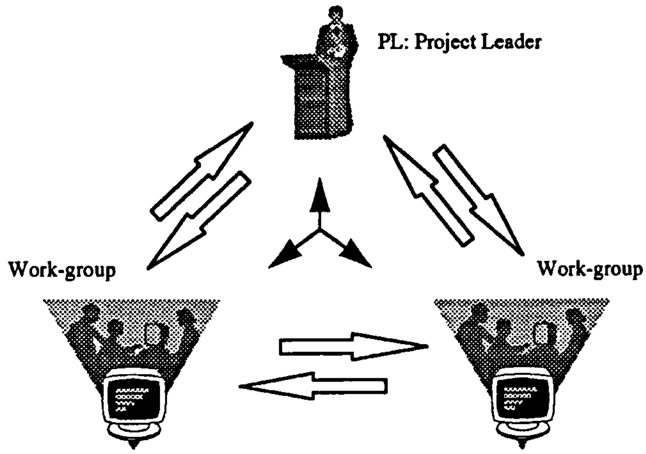

Figure 3. Cooperation among team and teams.

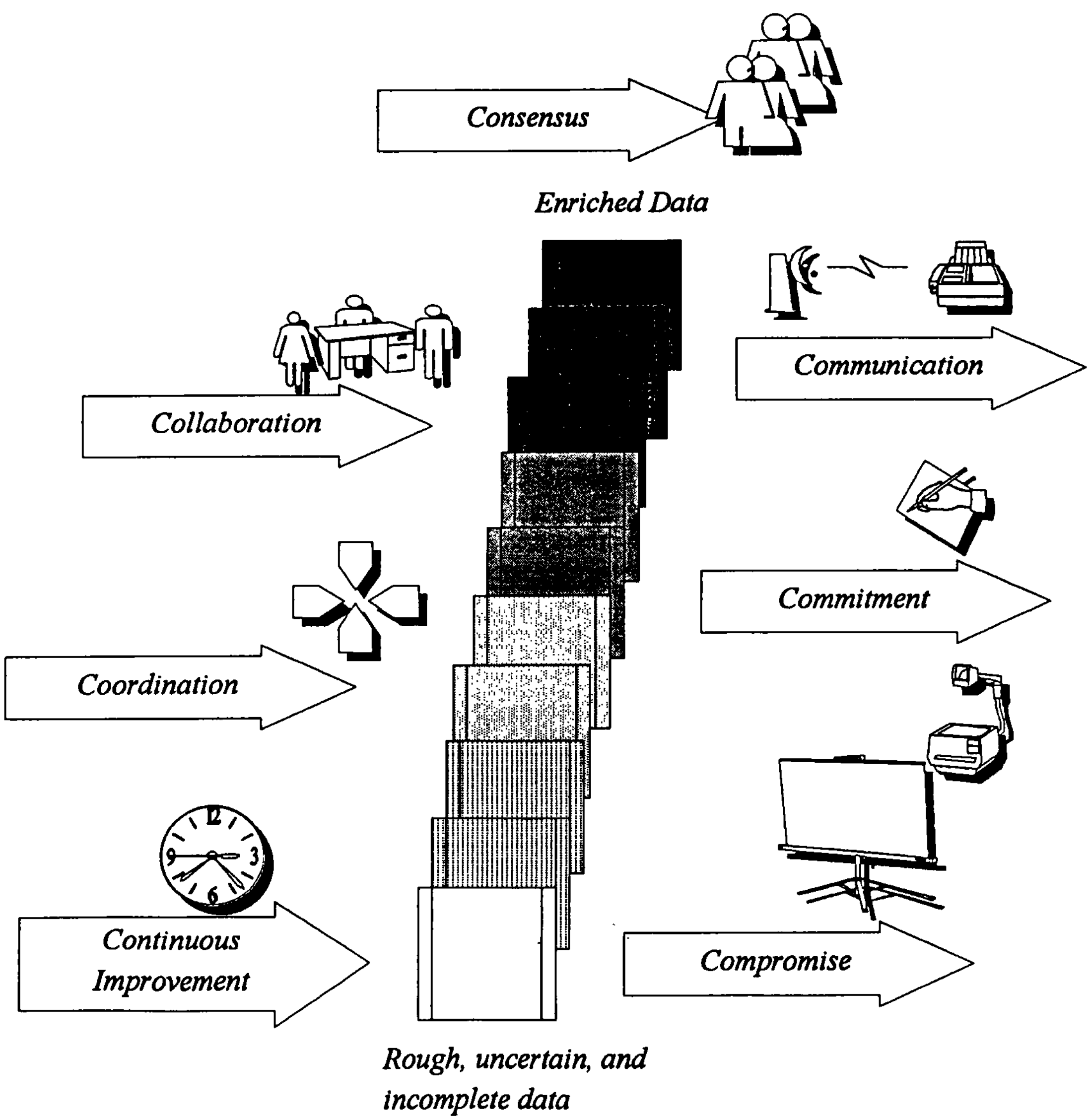

Figure 4. 7C's in cooperation.

According to Reference [1], there are seven elements (called 7Cs) to this team cooperation philosophy (Figure 4):

- *Communication*: Participants transfer information or exchange ideas. A seamless communication channel helps to realize the "virtual team."
- *Collaboration*: Participants of the team work collaboratively on some shared object to seek out the unplanned and unpredictable.
- *Compromise*: There is compromise and input from every discipline so that simultaneous development of the product, process, and associated tooling can be achieved.
- *Commitment*: Empowered teams define the tasks and prioritize areas to make breakthrough opportunities. Goals and objectives, duration, utility, complexity, expected results, and key success factors are outlined as much as possible. Management is fully committed to meeting the goals.
- *Continuous Improvement*: Product or process re-engineering teams work toward a total elimination of wastes. The concept focuses on enhancing productivity and profitability through the improvement of product quality and reduction in product development cycle-time.
- *Consensus*: Project team and management members may disagree, but team support of the requirements and a commitment to objectives from the very outset of a project is essential. These common objectives are reinforced throughout the life of the concurrent design project.
- *Coordination*: It employs collaborative actions of work-group participants to achieve the desired results as efficiently as possible. Coordination replaces "actors" by "agents" through performing interdependent activities that achieve goals, and its analysis includes goal decomposition, resource allocation, synchronization, group decision making, and preparation of common objectives.

The aforementioned 7Cs describe the level of interactions that take place in a CE process. How to realize such interactions effectively in a distributed environment is a key question that teams need to address. The next section studies a systematic approach and addresses this question in more detail.

3. Computer-Supported Cooperative Environment

Figure 5 shows a pyramid-based computer-supported cooperative environment (CSCE) for a team work. Three layers are involved in this pyramid: Infrastructure, Execution, and Planning and Scheduling.

1. *Planning and Scheduling* is a project leaders (PL's) task and forms the top layer of the CSCE environment. Responsibility includes planning the product development process, providing a set of clear and consistent goals for the work-groups and the team-members, and resolving any conflict that may arise due to such interactions.
2. *Execution* is the middle-layer. Execution layer puts the multidisciplinary experts and the PDT in touch with the infrastructure layer. Memory capture and management provides a virtual environment, and is full of previously captured historical working designs (so called technical memory).
3. *Infrastructure* is the CSCE's foundation—the bottom layer. This provides a seamless communication environ-

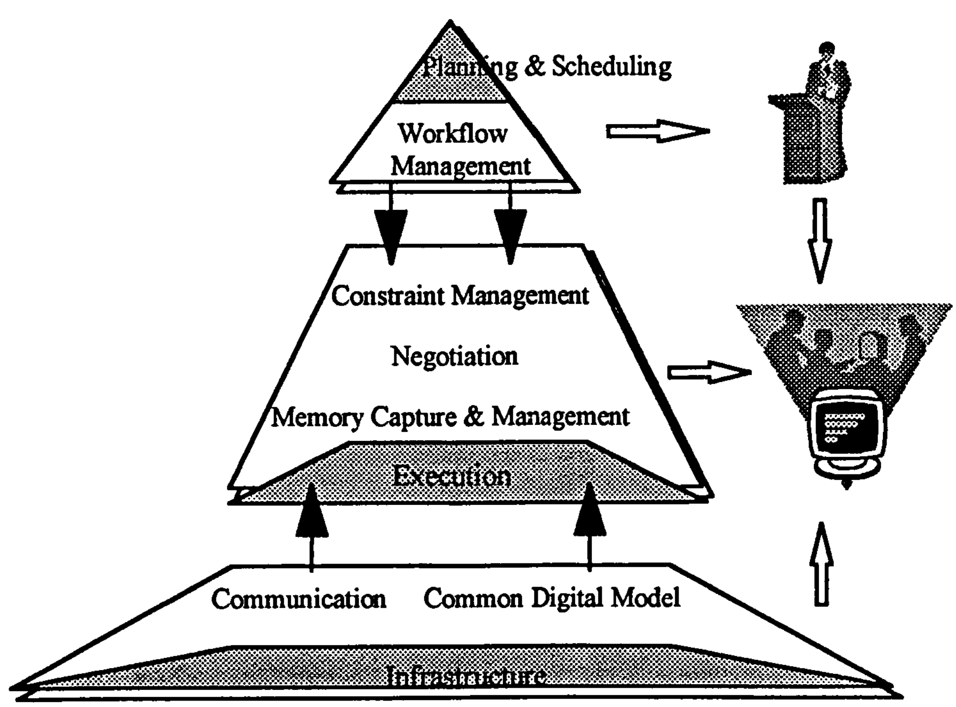

Figure 5. Computer-supported cooperative environment.

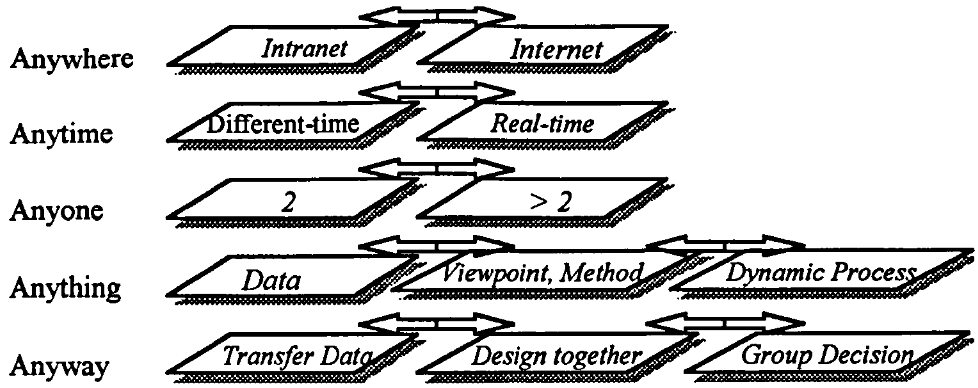

Figure 6. Status switch to each other.

ment for facilitating interactions among the team-members and a common digital model for sharing the information, if needed.

3.1 Communication

Today, the development of network technology has provided concurrent teams with many effective communication tools such as E-mail, network file system (NFS), Lotus-Notes, Tele-conference, etc. [37]. The emergence of Internet as an acceptable collaborative tool at the workplace has brought geographically separated divisions and departments spanning across international boundaries much closer to each other. Such tools have also made it possible for the team members to cooperate in a distributed environment. The question remains can we be able to provide a seamless communication with this proposed cooperative environment. This is discussed next.

3.1.1 SEAMLESS COMMUNICATION

A seamless communication environment ensures quality in the work tasks and are performed and improvement in the output efficiency [11]. The word "Seamless" is used here to signify a set of "five-anys"—meaning how to make members communicate anything in anyway (with respect to a desired data format), with anyone located anywhere in the world at anytime [38] (Figure 6). Anything refers to the contents exchanged including data, knowledge, viewpoint, method, or a dynamic process. Anyway refers to the manner in which this communication is carried out including ways of transferring data, designing parts together, or making group decision. Anyone refers to one or more groups or persons. Anywhere emphasizes the geographical regions or zones around the world where members are distributed. While anytime refers to the timing of communication—communication can occur between the work-groups, each at different time or all at once (the same time), even though the time zones of the members' location may be different.

3.1.2 COMMUNICATION LANGUAGE

Though, multi-agents technology has been applied in Concurrent Engineering and is being consistently improved in recent years, timely communication among the agents is still an issue. An agent is a system or software that needs a readable language to communicate [9,10]. The speech actions (such as "tell," "ask if," "achieve") and speech contents are generally not standardized [38] (Figure 7). The ambiguities still exist and have not been completely overcome. Thus, multi-agent components are not able to communicate effectively with each other.

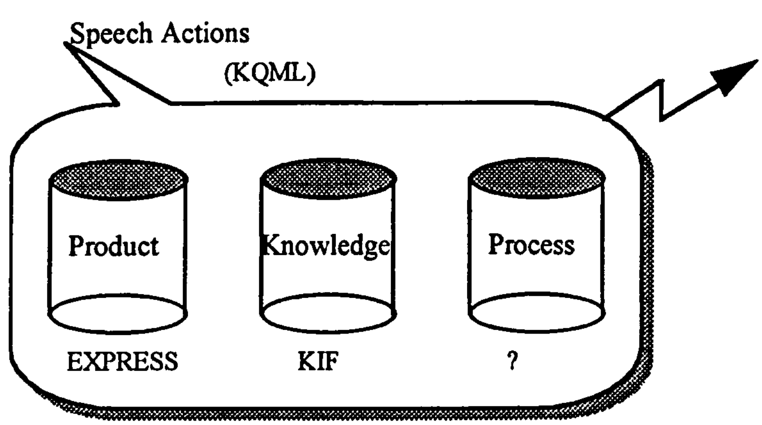

Figure 7. Communication using shared language.

KQML (Knowledge Query and Manipulation Language) is one such protocol language that contains speech action Performa and context related information. KQML can be used to represent a member's speech actions.

Three kinds of communication or speech contents are common in a language: product, knowledge, and process. Product's expression can be based on EXPRESS/STEP standard (AP203). Knowledge can be expressed in Knowledge Interchange Format (KIF). KIF is a neutral specification language. While using KIF, it may seem difficult to describe the process (as-is flow diagram, design rationale) in some rational (open standard) way, many researchers have found this reasonable. Another resort is to define one's own language at least on a short-term basis.

3.2 Information Model

In order for the members of a PDT to work at a high degree of concurrency in a distributed environment, there must be a shared information model on which basis the teams can communicate and collaborate. In the DICE project [4], a shared model of the product, process (activities in all life-cycle phases), and organization (resources of all types) were introduced [13]. In MIT's Dice environment [6], a product's multi-dimensions model was studied. In this paper, an organization model, a resource model, a product model, a process model, and their relationships are studied.

3.2.1 A 3-D ORGANIZATION MODEL

In traditional organizational structure, an enterprise is divided into distinct groups or departments. For example, a design department is separated from a manufacturing department. Even within a department, such as an aero-engine department, there are a number of work-groups, which have distinct assigned responsibilities. For instance, analysis groups are responsible for analysis functions (such as strength or aerodynamics jobs); design groups are responsible for design of components (such as fan, camber, and turbine) and so on. The connections within these groups are loose and the communication channels between these diverse groups are limited or non-existence.

As mentioned earlier, PDT is an important vehicle to facilitate Concurrent Engineering. Usually there are a number of teams present in a PDT for the purpose of designing and developing a product. An organization model describes this PDT's structure for an enterprise. Figure 8 indicates how to configure a PDT when there are more than one work-group or team. A complex product, such as aero-engine, is made up of many different parts. To design a part (such as blade), requires a team including experts from aeroelasticity, structure, analysis, and manufacturing work-groups. The organization can be viewed in a 3-dimensional model-form spanned along three axes. The vertical dimension represents a product breakdown structures (PtBS)—decomposition of the product into its hierarchy. The horizontal axis is a life-cycle functional distribution—feasibility study, perspective design,

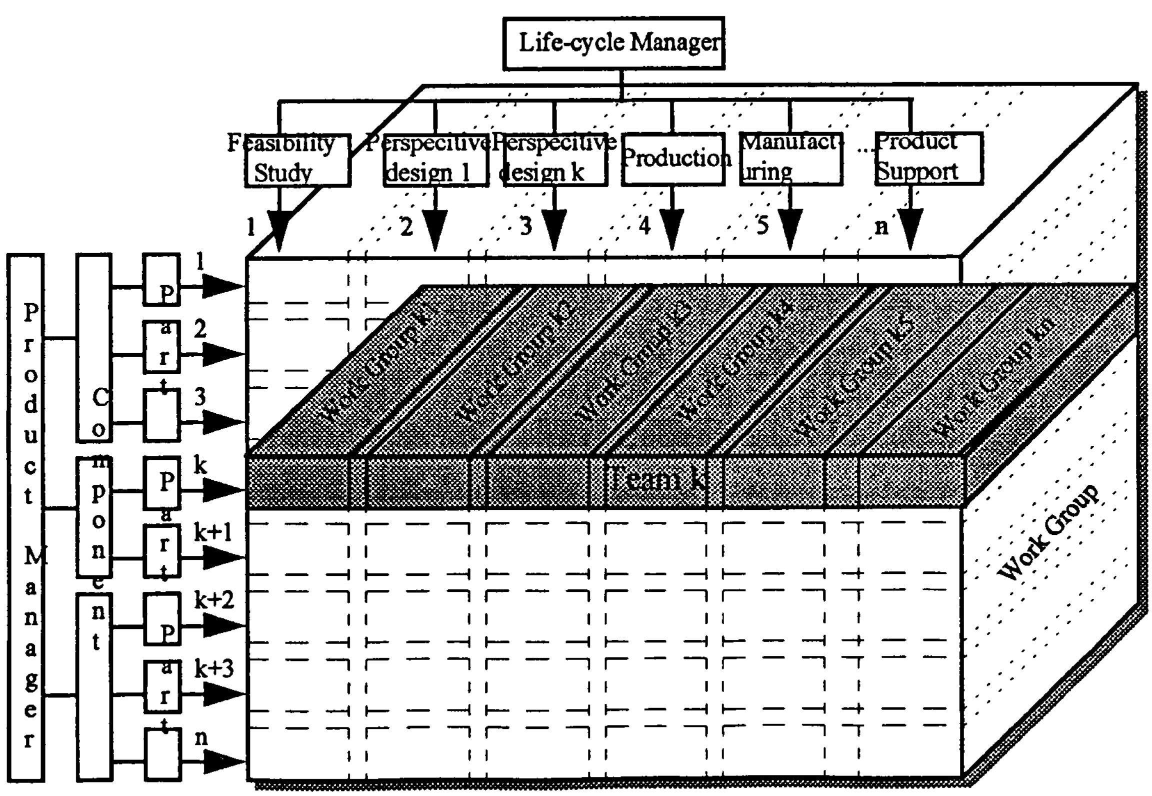

Figure 8. A three-dimensional structure for an organization structure.

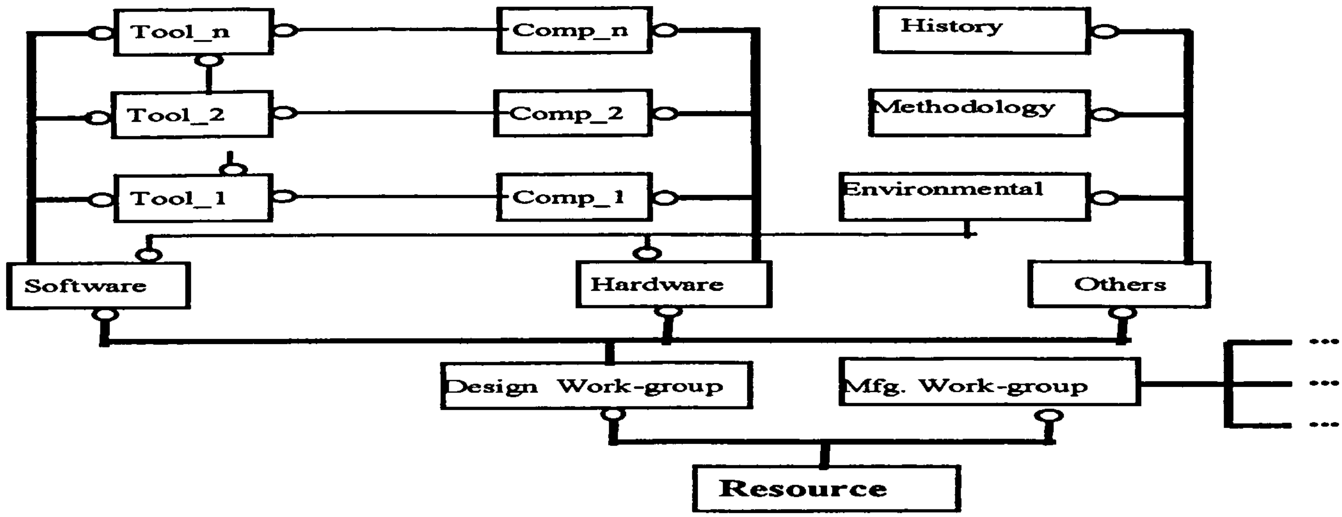

Figure 9. Resource taxonomy.

production planning, manufacturing, product support, etc. The axial dimension represents the work-breakdown structures (WBS)—the distribution of work-tasks to the work-groups. The organization model shows that all members of a CE enterprise are deployed into a set of teams of teams. One set of teams is responsible for a set of parts of the products based on PtBS. Another set of teams is responsible for its (part's) life-cycle function based on life-cycle hierarchy. And a third set of teams is responsible for performing their distributed works based on the WBS.

In a PDT, each area manager is responsible for its own work-groups. For instance, a life-cycle manager is responsible for choosing the right experts for appropriate multi-disciplinary teams to ensure that the goals of those sub-groups are consistent. Similarly, a team leader in a product hierarchy's layer (say a component or a part) should focus on the corresponding performance such as assemblability of those parts or components, or the product overall performance.

$$\text{Organization (O)} = f\ (\text{Work-groups, Life-cycle, Product-line}) \quad (1)$$

where:

O stands for organization

f denotes a function:

Product-line ∃PtBS (Product Breakdown structure)

Work-groups ∃WBS (Work Breakdown Structure)

Life-cycle ∃PcBS (Process Breakdown Structure)

And ∃ denotes an element

(2)

3.2.2 RESOURCE MODEL

Resource is used here to mean all available things, such as 4Ms—Money, Machine, Manpower and Management—[1], in an enterprise that can be leveraged to support life-cycle development activities of a product. The two common resources in the product development area (shown in Figure 9) are the design work-groups and the manufacturing work-groups. Apart from the above two work-groups, there are other resources that link or influence the various life-cycle functions. One such resources is the process resource. Examples include design methodology, historical data, experience, solutions, etc., since they influence the process of design and development. For example, a methodology—such as DFX (Design for X)—may improve the performance quality of a design. To develop a new car, a major portion of parts may be carried over from an old car's. It is therefore necessary to manage those historical design documents (data, methods, and experiences) that contain such information.

Capability or characteristic of a resource is an important element to manage. With regard to application resource—which may refer to a software—the data exchange standard has an important bearing on the tasks or jobs performed. If a software in use for that application follows the same set of standards (say an IGES or STEP), many unnecessary translation jobs could be eliminated.

In order to integrate resources, their dependence on each other or constraints that limit their usefulness need to be identified. For example, information such as some materials can't be cut by some tools could help a mechanical designer identify an alternate tool or alternatively select a suitable material for that part.

$$\text{Resource } (R) = f\ [\text{Work-groups, 4Ms}] \quad (3)$$

$$\text{And Work-groups} \ \exists\ \text{WBS} \quad (4)$$

where:

R stands for Resource

WBS stands for work-breakdown structure
f denotes a function
$\exists$ denotes an element of.

3.2.3 PRODUCT MODEL

In a traditional design, different participants create their own variation of a product model. It is, therefore, very difficult to keep consistency among different versions of the product model. Not only more time is wasted in trying to reformat data required for each version, but also some errors could slip through. A version could be eliminated or minimized using a multi-view product model. A product model provides a shared object for multiple participants to work together as a team [6]. As such this model is also called a total or global product model. Information for each aspect of product design scenario or its functions is developed from the same master model. Each variant design is obtained by constraining their functional dependence or by limiting the constraints imposed on the attributes describing the total product model. The STEP-AP214 supports this model. Figure 10 shows an example of an information model of a blade, which contains the requirements, information of aerodynamics, structural body, withstanding loads, and production as view-windows. Views of the child models shown in separate view-windows are interdependent. Meaning, a change in one view-aspect affects the other views dynamically (depending on their constraints) and vice-versa.

3.2.4 WORKFLOW PROCESS MODEL

A workflow process model contains all life-cycle activities towards a product design development and delivery aspect, and their precedence on each other. There are many ways (e.g., IDEFx, PERT chart, PETRI Net, and State Machine) to describe a workflow process [36]. For Concurrent Engineering, it is more important for the teams to study the inter-dependencies of those work activities and discuss these with the work-groups performing them. B. Palmer [33] had summarized life-cycle activities of a product development process into five primary classes according to their relationship. However, the interdependencies among various classes were not stressed. Understanding of inter-dependencies requires development of a workflow process model. This is discussed next.

Information flow is the first step towards creating a workflow process model. In this paper, authors have introduced a new type of activity model, which is based on the type of information flow. Information could flow within its own work-group or could flow across work-groups. Figure 12 shows four basic elements of an information flow for an activity-model: Inner-input, Inner-output, Outer-input, and Outer-output. The word "Inner" refers to the information flow between participants of the same work-group. The "Outer" refers to the exchanges among participants of two different work-groups. The outer flow influences the collaboration among participants of two different work-groups. For example, a process planner would put emphasis on the outer flow, not on the inner flow. This is because in process planning, information flow takes place between design work-group and the process planning work-group, which normally are two different work-groups. If an activity contains inputs or outputs, the types of flow between work-groups could be distinguished using Figure 11. If a flow is within a group (say work-group B), it is referred as

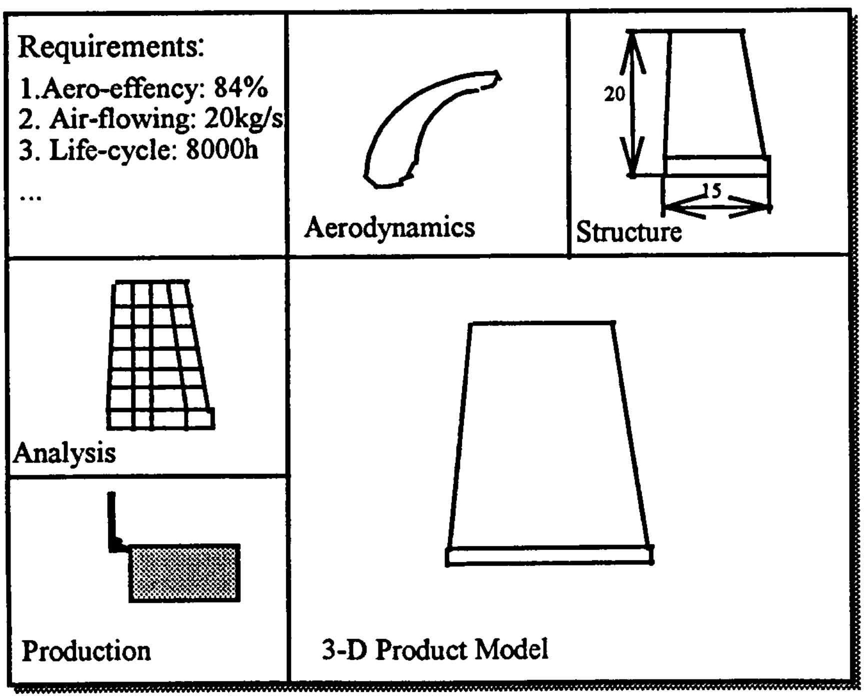

Figure 10. A multi-view product model.

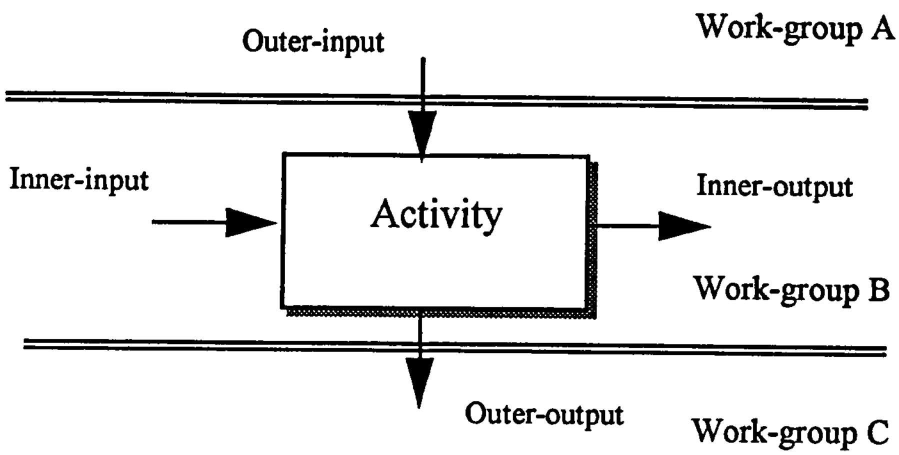

Figure 11. Information flow of an activity.

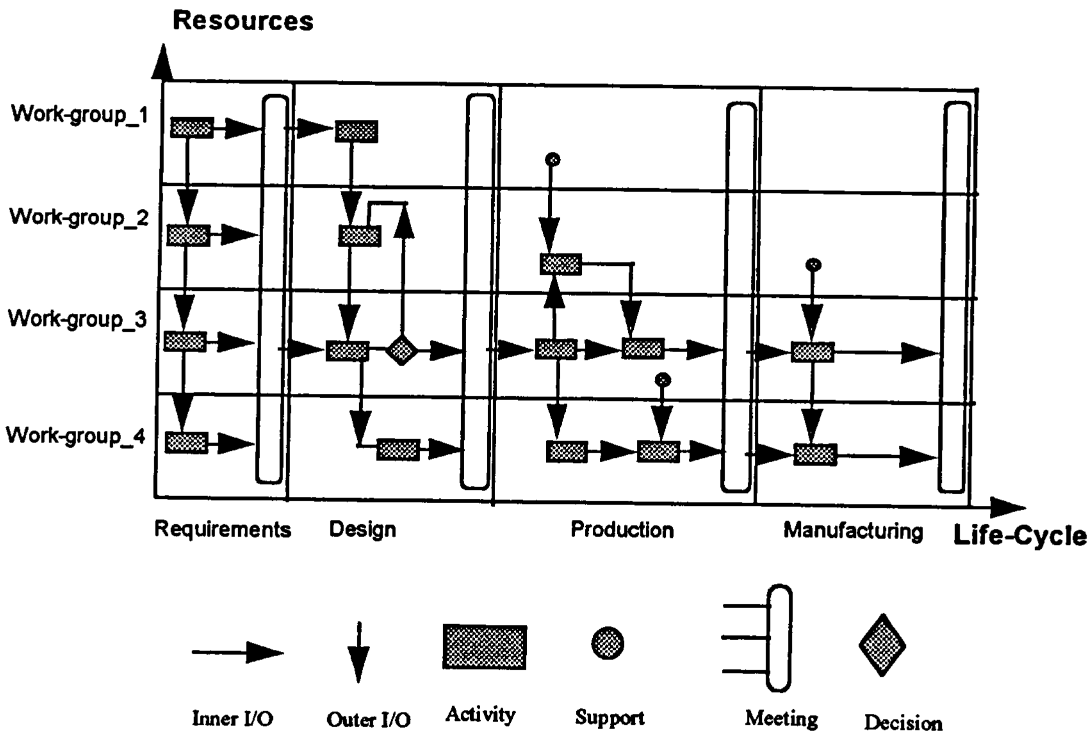

Figure 12. Work-flow charting (Life-cycle versus resource).

Inner-input (if the flow is pointing inwards) and inner-output (if the flow is pointing outwards) as shown in Figure 11. Conversely, if the flow is pointing inwards from one work-group, say A, to another work-group, say B, it is called "outer-input." Similarly, if the flow is pointing outwards from one work-group, say B, to another work-group, say C, it is called "outer-output."

Workflow Charting (WC) is a work method for describing a life-cycle process for a work-group and distinguishing that process from one work-group to another. WC is different from other flow-charting tools in that it shows who (a person or a work-group) performs each activity and the time or workflow sequence in which the activities are performed (Figure 12). It simplifies the process because the interactions (Up and Down arrows) are limited to one life-cycle phase. This WC has been used in DICE *GE Project Case Study* [31]. The method is found quite effective for capturing both the "*as-is*" and the "*To-be*" process.

$$W = f\ [\text{Life-cycle, Resource}] \tag{5}$$

where:

W stands for workflow process
R stands for Resource
PcBS stands for Process breakdown structures.

3.2.5 INTEGRATION OF O, R, P AND W MODELS

The four elements of organization (O), resource (R), product (P), and work-flow process (W) are related to each other, and could be represented as:

$$P = f\ [O, R, W] \tag{7}$$

where:

workflow process (W) is related to life-cycle processes (PcBS) and resource (R), that is

$$W = f\ [R, PcBS] \tag{8}$$

Figure 13 interprets this relationship. The end product or the output, P, is the goal of the system (a physical unit). The organization and resource, O and R, are the foundations for W. The element W is the workflow process to develop a physical end product P. There is no unique process for representing W. There could be many workflow modes to develop a product, depending on the current status of an organization, available resource, and the choice of a scheduling method used. Generally, the "As-is" and "To-be" process flowcharts are two popular workflow methods to describe the current and improved modes of product development, respectively.

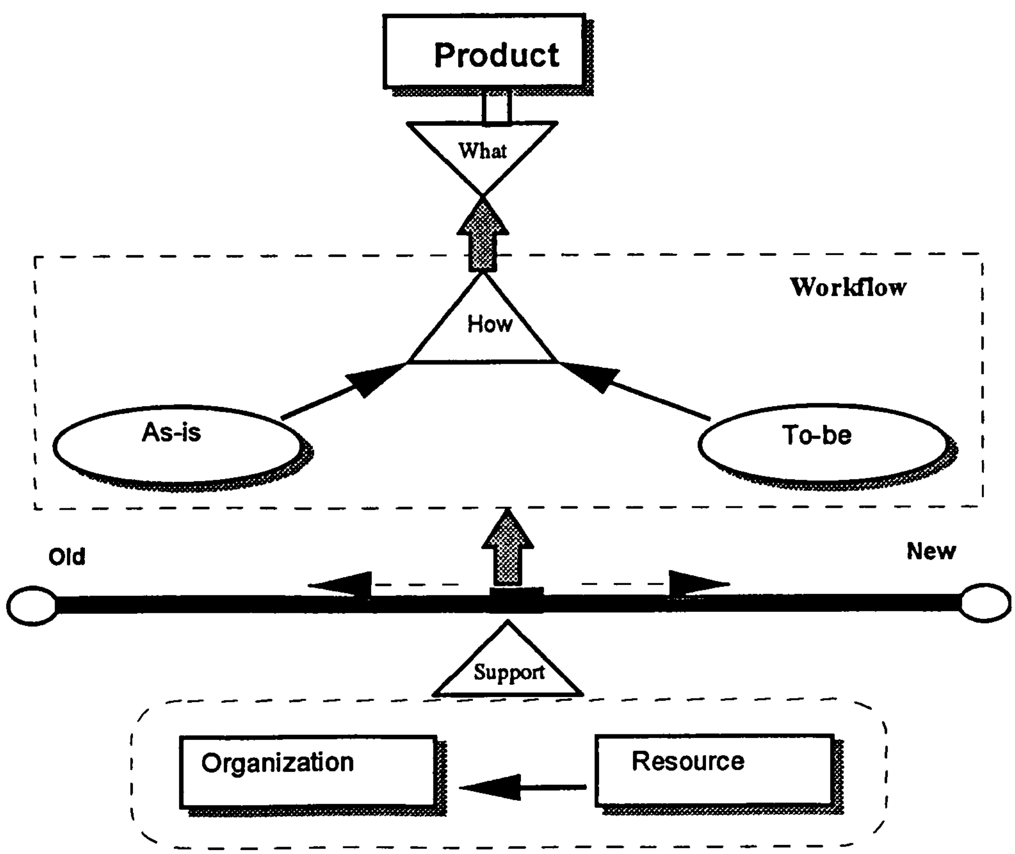

Figure 13. Relationship among four models.

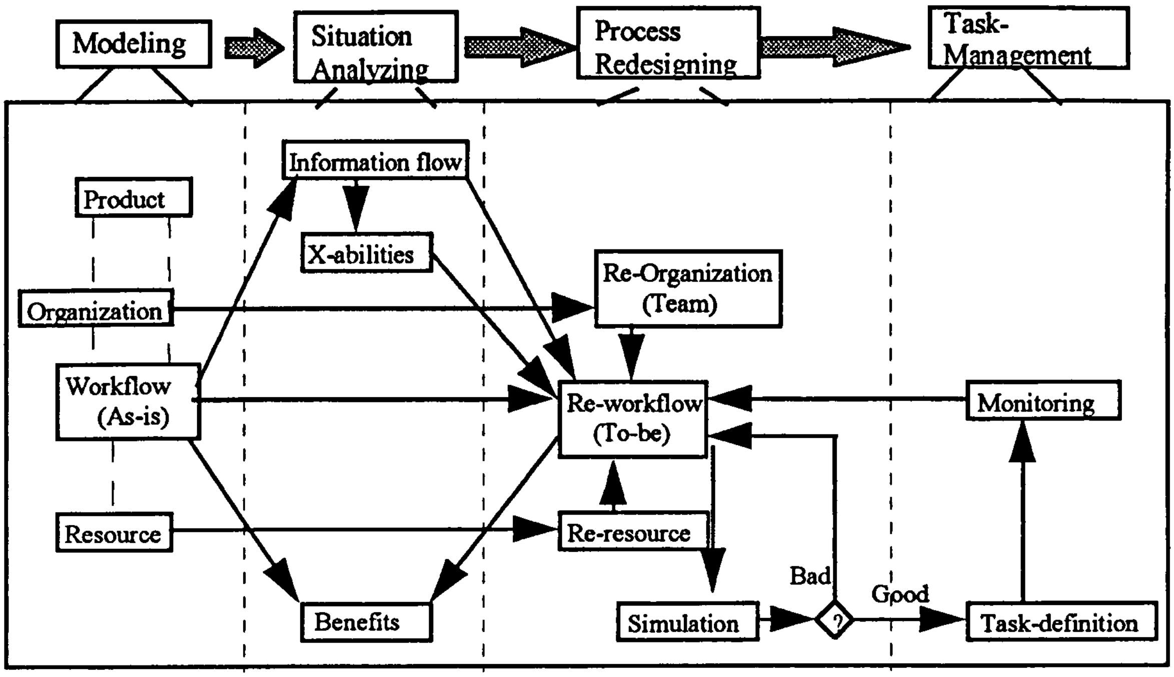

Figure 14. Stages in a typical workflow management.

3.3 Workflow Management Styles

There are two workflow management styles for a PDT: Team Leader-centered and Team Self-directed [30]. Leader in the Leader-centered team is generally responsible for carrying out most business and for giving day-to-day direction to the team's activities. While in a self-directed team, the members are clear on their destiny, and interactions take place freely among the team. The authors feel that an effective team should be a binding of the two management styles. The project leader (PL) should schedule and manage only the global business and should give team members more and more freedom. In a *Workflow Management* (WM), the project leader focuses on their interaction points among members as discussed above. The process model studied in this paper supports this work style.

There are a number of papers [32–35] studying WM. In this paper, an approach to WM is introduced. Generally, there are four stages to WM described as follows (Figure 14):

1. *Stage 1: Workflow Process Modeling*
   Process model is the infrastructure of WM. It should contain all information required and could express the feature of CE. The model in this paper is decomposed into 4 sub-models: Product, Workflow Charting, Organization, and Resource. The main task at this stage is to model the current work process (As-is), which is the integration of the four sub-models.
2. *Stage 2: Situation or Performance Analyzing*
   A workflow process is made up of numbers of activities that are related to each other. In order to redesign the workflow process, we must make clear the dependence and constraints among activities. In addition, the performance (says X-abilities, Benefits) of a process are also important parameters.
3. *Stage 3: Workflow Process Redesigning*
   As discussed earlier, Re-process can be realized by Re-organization, Re-resource alignment, or Re-workflow charting. The process's dynamic simulation is also required at this stage.
4. *Stage 4: Task Management*
   After defining a task for each work-group contributor, the real-time monitoring and controlling of rework tasks are carried out at this stage. This system should have the capability to adjust the work process—re-workflow (to-be)—once some unexpected case arises.

3.4 Constraints Management

Constraints exist in terms of relationship amongst parameters, features, which limit functional performance or a product's behavior. Figure 15 shows a constraint network of a system having three perspectives. Besides the intra-contraints (due to constraint interactions inside an ellipse), constraints among perspectives (called here as inter-constraints) also exist [17,18]. Each member may know clearly his own constraints (intra-constraints), but may have difficulties considering the influence from other perspectives (inter-constraints). The computer supported cooperative environment for CE should be able to resolve the problem. The environment should have constraint management (CM) tools to avoid possible conflicts by cooperation among multidisciplinary teams. Such CM tools should be able to develop a constraint network, and detect conflicts. This requires *Constraints Management* (CM) as a part of a CE domain.

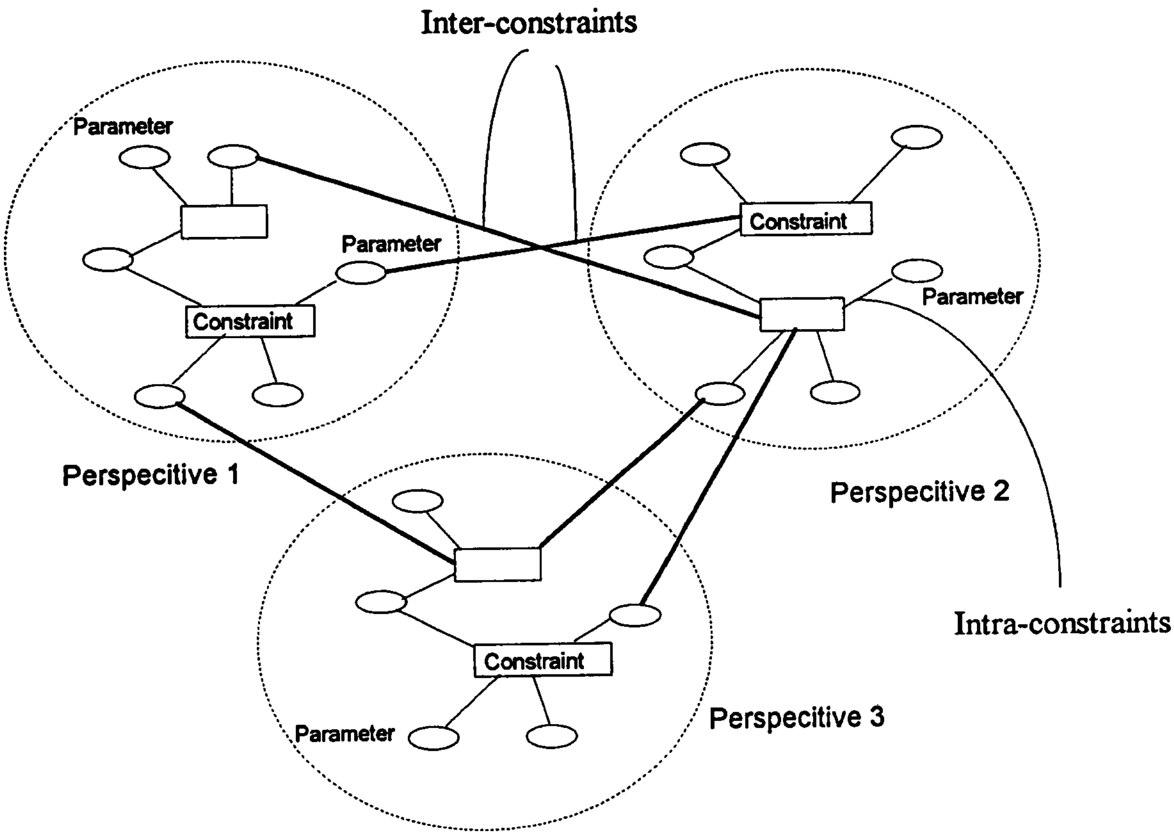

Figure 15. Constraints network.

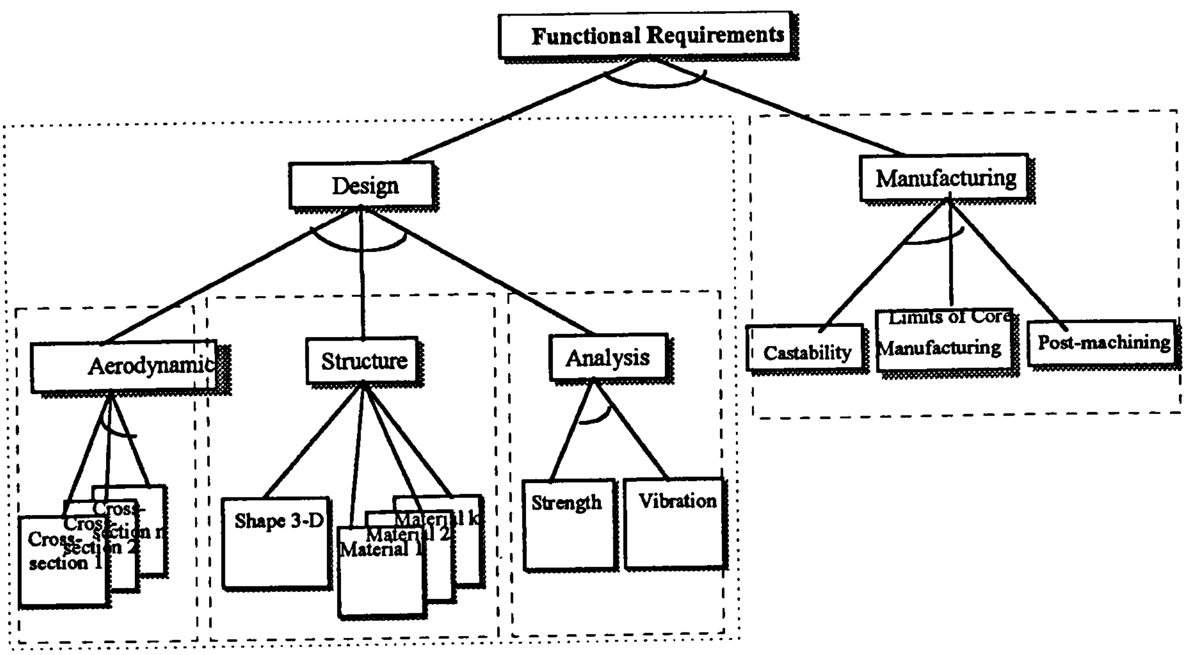

Figure 16. Constraints hierarchy for blade.

There have been a number of research developments on CM (as [17,18]). Some of the related technologies are Constraints Represent, Propagation, Detection, etc. However, the constraints network might be very monolithic and complex in a large system, which could confuse CE team members. Peter J. O'Grady [19] has introduced a hierarchical approach to resolving this problem. Figure 16 is an example of a turbine blade. The arc represents a constraint. This hierarchy makes each member interact with the network readily and allows the members to concentrate more easily on the area under consideration. Each module can be modified and updated based upon the aggregation of the network elements that are its children.

3.5 Negotiation

Once a conflict between multi-disciplinary groups arises, a related resolving method must be devised. There are two main classes for negotiation: Team-member Centered and Knowledge-based systems (KBS) Centered. An expert with experiences in a number of disciplines could be a valuable resource, but such experts are in short supply, and are not readily available. Artificial Intelligence or knowledge-based techniques might then be used to support decisions, and to resolve conflicts among the multidisciplinary groups, or over the distributed networks. A knowledge-centered negotiation can take place in three different ways: (a) through a local KBS, (b) through an integrated KBS, or (c) through a distributed KBS.

(a) **Local KBS** is limited to the knowledge content belonging to a single group or a member. That is,

$$KL_i = f[k_i(m_i)] \quad (9)$$

Where k_i is the knowledge content of a member-group m_i. KL_i is the local KBS knowledge content.

(b) **Integrated KBS** means a global knowledge content. Integrated KBS is the result of intersections of all knowledge contents from each of the local groups, which are collaborating.

$$KI = [k_1 \cap k_2 \cap k_3 \cap k_4 \cap k_5 \cap \ldots, k_i \cap \ldots, \ldots, k_n] \quad (10)$$

or

$$KI = [k_1(m_1) \cap k_2(m_2) \cap k_3(m_3) \cap k_4(m_4) \cap \ldots, k_i(m_i) \ldots, k_n(m_n)] \quad (11)$$

Where k_i is the knowledge content of a member-group m_i. KI is the integrated KBS knowledge content.

(c) **Distributed KBS:** This is when the knowledge content of each group is distributed over the network so that each member can be able to access somebody else's knowledge. The knowledge is not merged into a single source like in those "integrated KBS."

$$KD = [(k_1 \cup k_2 \cup k_3 \cup k_4 \cup k_5 \cup \ldots k_i \cup \ldots k_n)] \quad (12)$$

$$KD = [k_1(m_1) \cup k_2(m_2) \cup k_3(m_3) \cup k_4(m_4) \cup \ldots k_i(m_i) \cup \ldots k_n(m_n)] \quad (13)$$

Where k_i is the knowledge content of a member-group m_i. KD is the distributed KBS knowledge content. ∪ represents a Union-of the individual knowledge contents. Depending upon whether the knowledge is derived from a KBS source and who is accessing the KBS, a number of interactions can take place. Figure 17 lists six such interaction modes based on the possible types of KBS and teams.

These six modes of interactions amongst members and KBS are listed in Table 1. The word "local" means teams are working in isolation from each other. The word "distributed" means that the teams may be geographically separated from each other or may be linked remotely through a computer network.

(a) *Distributed Members with no help of KBS*: Often through the help of networks, such as Teleconference, etc. [12], distributed members or domain experts are linked or contacted to provide relevant knowledge. Such knowledge elements are pooled (collected) from the experts to solve a common problem. The solution may depend on the knowledge (quantity) collected from the experts and on the quality (relevancy) of the knowledge pooled. Domain experts do not participate in the decision-making; project leaders make all the decisions. However, this mode requires a large amount of time because it is not easy to pool-in all the relevant knowledge and satisfy multi-experts' requirements in a short span of time.

$$D = f(m_1, m_2, \ldots, m_n) \quad (14)$$

Where, D represents an output of the decision-making, f denotes a decision-making action, m_i signifies an ith member of the team.

(b) *Distributed Members with the help of a local KBS*: The knowledge may be collected in the same way as above, however, in this case, some domain experts do take part in the decision-making, but they are isolated to each other. The final negotiation depends upon the project leaders. The knowledge content used for decision-making is member-centered.

Table 1. Interactions between members and KBS.

	None KBS	Local KBS	Integrated KBS	Distributed KBS
Local Members	X	X	(c)	(d)
Distributed Members	(a)	(b)	(c)	(f)

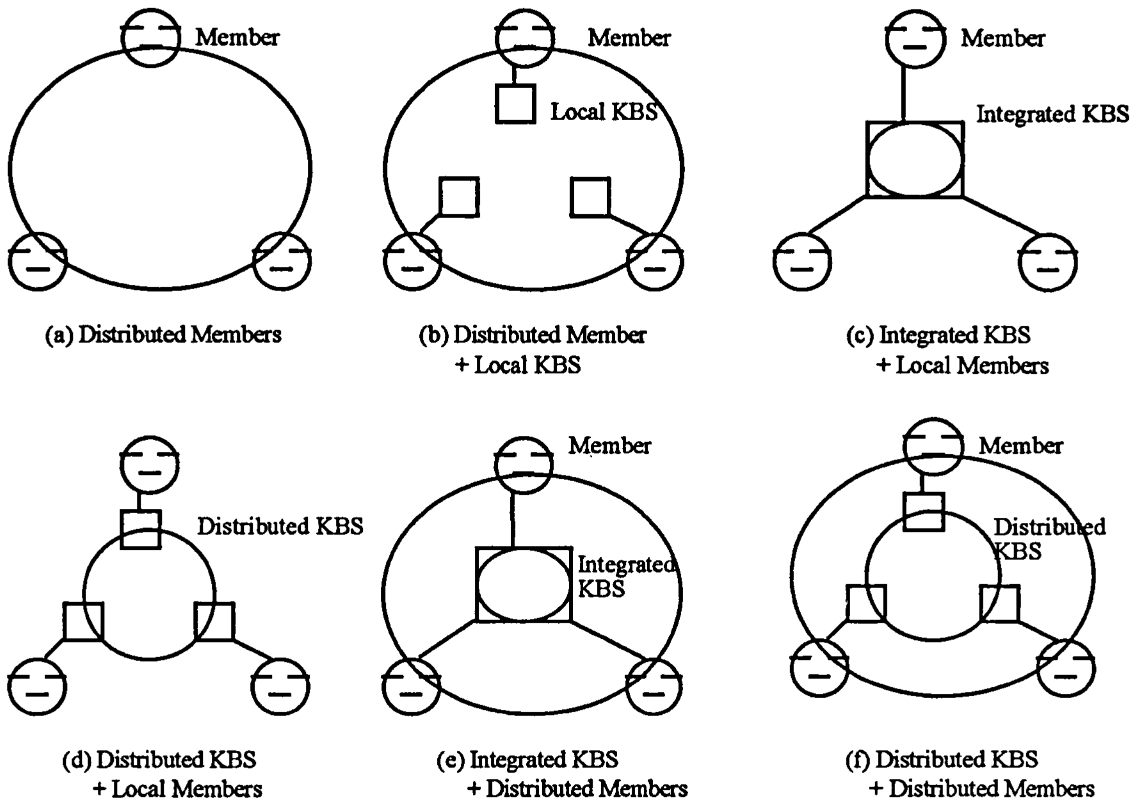

Figure 17. Decision-making modes.

$$D = f[m_1(k_1), m_2(k_2), \ldots, m_n(k_n)] \tag{15}$$

Where, k_i means a local KBS belonging to an ith team.

(c) *Local members and Integrated KBS*: Developing an integrated KBS implies bringing in knowledge elements from individual team-members or domain experts, and merging these knowledge elements into an integrated KBS. The hierarchical analysis could be used here, which considers the global situation, analyzes all influenced factors. R. F. O'Connor studied the selection of automobile electronic inter-connection architectures using this method [20]. However, an integrated KBS is difficult to develop [1].

$$D = f(KI) \tag{16}$$

Where, KI is an integrated KBS—defined in Equations (10) and (11).

(d) *Multi-agents based Collaborative Expert system*: This method considers the global constraints, based on multiple knowledge bases, which are distributed [23–26]. It uses the concept of Distributed Artificial Intelligence (DAI), needing collaborative reasoning. It is more effective, and difficult to develop than the above approaches.

$$D = f(KD) \tag{17}$$

Where KD is defined in Equations (12) and (13).

(e) *Distributed Members and Integrated KBS*: Member-centered decision is flexible, while the KBS is an automated decision-support system. This mode benefits from both the member-centered decision-making and integrated KBS powers. The decision-making may take place between member-member or member-KBS.

$$D = f(m_1, m_2, \ldots, m_n, KI) \tag{18}$$

Where KI is defined in Equations (10) and (11).

(f) *Distributed Members + Distributed KBS*: This mode might be the most ideal one.

$$D = f(m_1, m_2, \ldots, m_n, KD) \tag{19}$$

Where KD is defined in Equations (12) and (13).

In this case, decision-making actions may take place between member-member, member-agent (KBS), or agent-agent itself. The above six modes are useful in different collaborative situations. When individual member (m_i) is not effective or knowledge content (k_i) is poor, group's participation is important. When a conclusive answer can't be arrived at, the leader's decision is required. Therefore, an effective decision or negotiation should include both a human-element and a knowledge-content element mined from the experts, if not available analytically. As the level of collaboration increases, it is important to set up a seamless negotiation environment for PDTs.

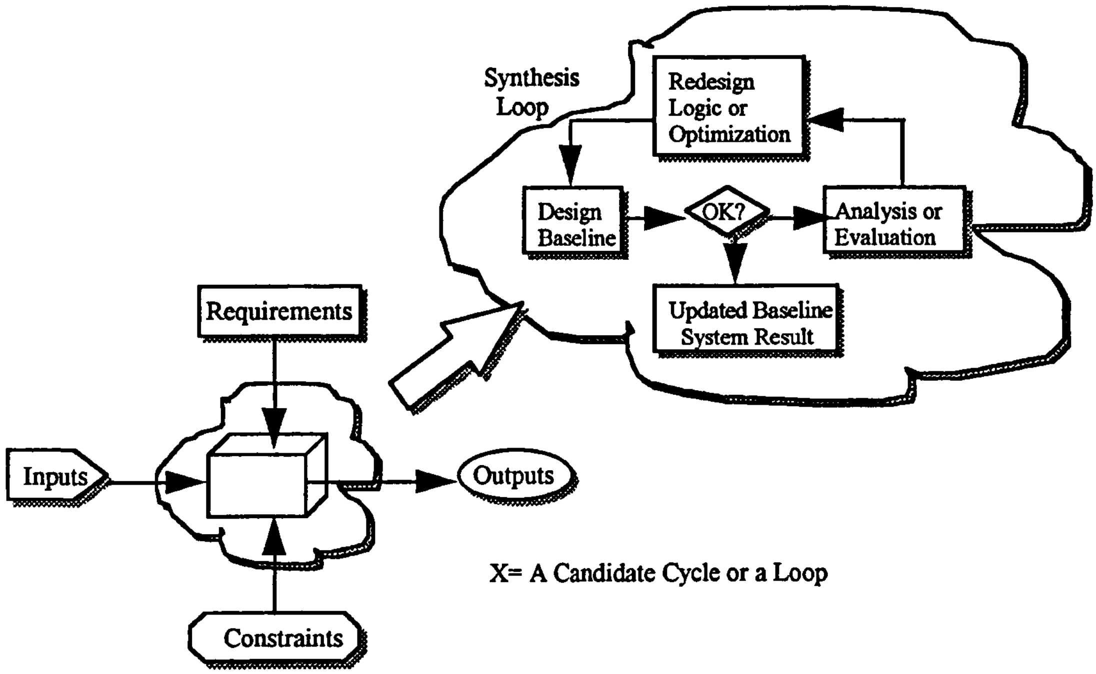

Figure 18. Rationale record.

3.6 Memory Capture and Management (MCM)

Collaboration among team members has two modes in which one can operate with respect to time: collaboration at the same time or at different times [8, 37]. Collaboration occurs at different times if you want to know how to resolve the same matter as you did at an earlier time, or if you did not know since your collaborator was absent last time, or maybe he was busy with some other things. How would you proceed now? Who would you ask for help? *Memory Capture and Management* (MCM) may come to rescue. MCM allows those useful resources (experience, historical data, or other's rationale) to be captured in a form that can be interrogated or can be visited any number of times at any time.

Figure 18 represents a unit of design process. It contains the Inputs, Outputs, Requirements, Constraints, and the process. The traditional documents do record the former 4 elements, but with such static means it is difficult for the visitors to know how to use the information or how to reach the final result. MCM is a dynamic means to capture the process and to lead you to the final conclusion [27–29]. The key technology is how to represent different disciplines' knowledge and rationale. References [28, 37] give some good advice.

With MCM, team members need not be required to be present or forced to work simultaneously at any given time. This improves the degree of cooperation and flexibility. A virtual "designer" replaces the human presence with captured MCM knowledge.

4. The Systems

The systems configuration and a user interface will be introduced in this section.

4.1 Systems Configuration

The software configuration of the proposed cooperative environment for CE is shown in Figure 19. The PDT members are linked through communication channels. The software configuration provides both a distributed and integrated environment for members to communicate. Team members may work in their own *Distributed Workspace* (DW). In addition, there is a server called *Shared Workspace* (SW) for all members including project leaders (PL). It is made up of two applications: *Product Multi-views Model* and *Memory Capture and Management.*

An *Access to CE Environment* is located at each DW, which provides an entrance to visit the SW, the PL, or other participants. The *Distributed Constraints Management, Distributed Negotiation* are provided as a part of this DW. In PL's workspace, in addition to DW (distributed workspace) elements, *Workflow Management* is configured. *Global Constraints Management, Global Negotiation* could also be configured here as a part of PL's workspace, if this PL is responsible for making final decision on conflicts.

4.2 User Interface

Access to CE Environment provides a window to the distributed team members, including PL to the workflow, resources, and the organizations. This is realized by a friendly user interface (Figure 20). It is made out of four areas:

- *Member list area,* this is the place, where all CE project participants are listed. Anyone in the team can preview other team member's messages (such as Perspective, E-mail, Telephone, etc.).

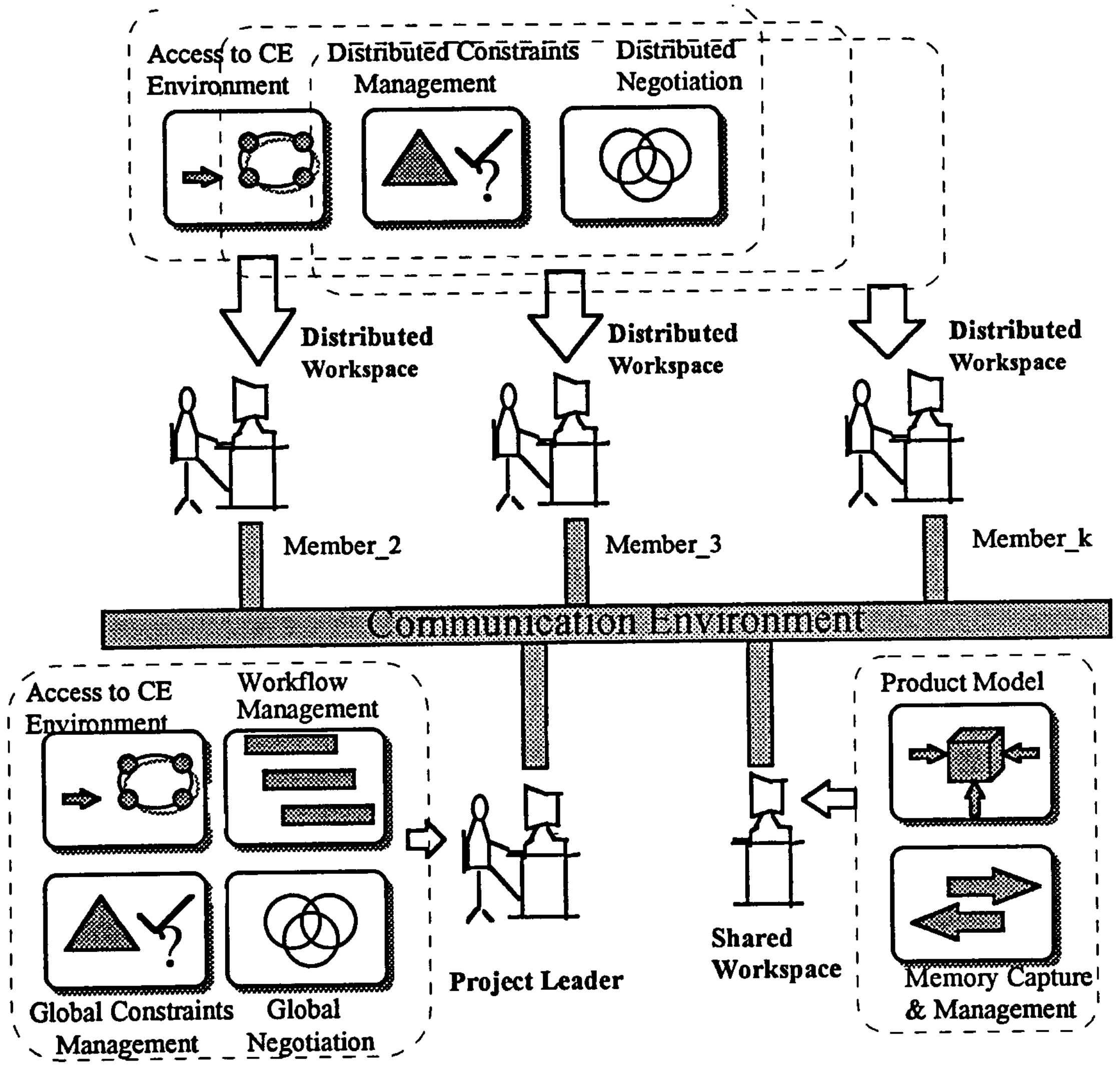

Figure 19. Systems configuration.

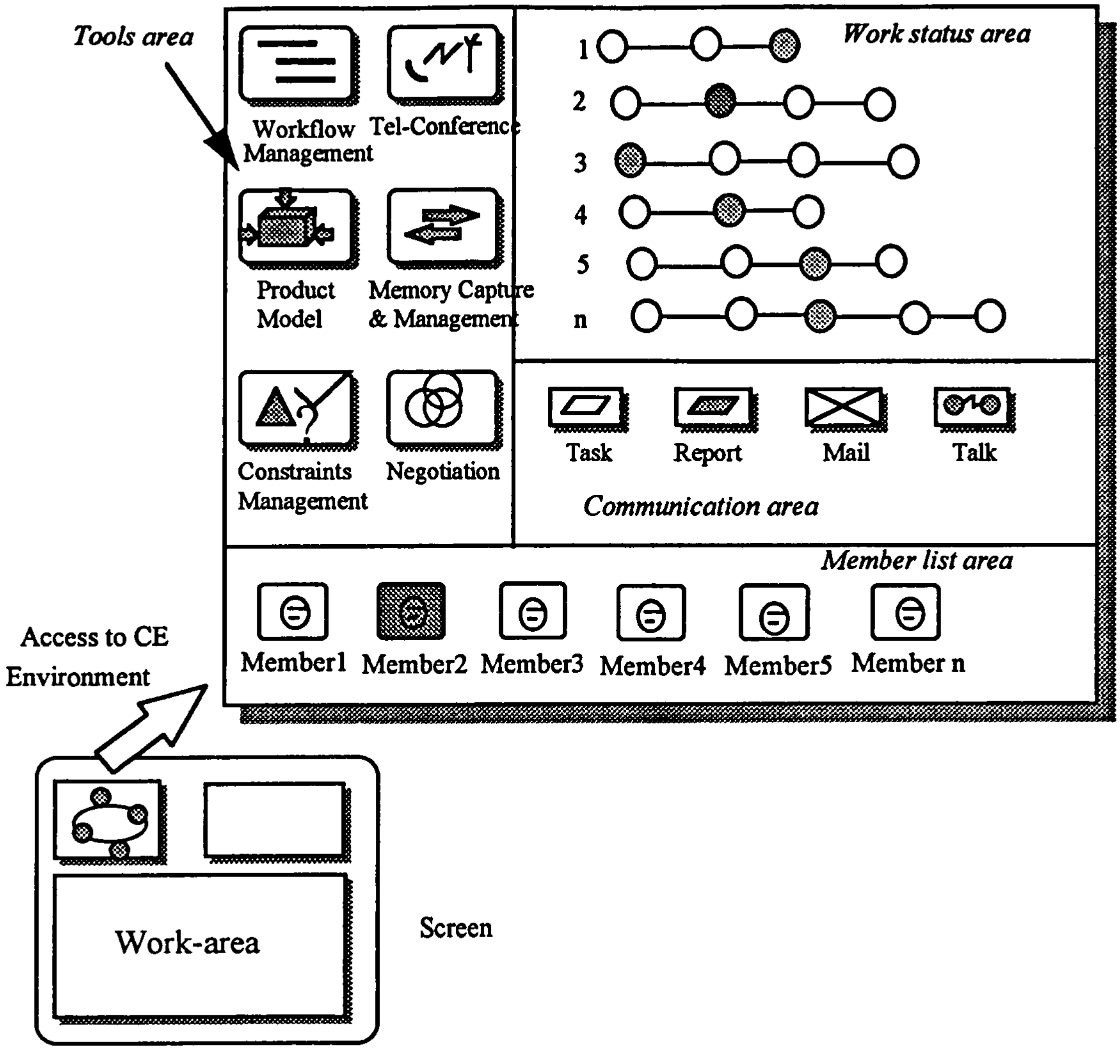

Figure 20. Access to CE environment.

- *Tools area,* where an access to a list of common and supported applications is provided to the team members. These applications are Workflow Management, Product Multi-views Model, Constraints Management, Negotiation, Memory Capture and Management, and Communication Tools. Workflow Management is available only to the team's project leader. The CE participants may visit these tools and the tool's area at any time.
- *Work status area,* is where all teams members' work progress are posted. With the help of this, any one would know what other team members are doing at any time.
- *Communication area,* is where a member can preview his or her task, report his or her work status, talk to anyone on the team, and send messages to others.

All together, with this access to CE environment, any participant may know who are the members on the project team, state of their work completion, what tasks should a member do next, and the resource (tools, 4Ms, etc.) available at any time.

4.3 Status of Current Implementation

The technical team at Beijing University of Aeronautics and Astronautics is developing a software environment based on heterogeneous workstation platforms. The different platforms used are SGI, SUN, and HP, using TCP/IP network protocol, based on UNIX systems. Object-Oriented technology is leveraged to describe the system model (entities) in EXPRESS language. The plan is to translate the entities into classes of C++ objects, which will make the system development more flexible.

5. Concluding Remarks

This paper introduces architecture for a flexible computer environment supporting cooperation among concurrent team members. This architecture is made out of three layers. In the infrastructure layer, a set of communication and information models is grouped. A seamless communication concept is introduced and the shared language is proposed for multi-agents. The information model includes organization, resource, product, production process, and workflow process models. Their characteristics are analyzed. In the control layer, workflow management is discussed. It has four stages: process modeling, situation or performance analyzing, process re-designing and task management. To ensure satisfaction of constraints the authors summarize briefly methods to manage constraints and resolve conflicts. Memory capture and management leg of this architecture provides designers with a set of historical data and a rationale basis for decision-making and product improvement. Finally, the whole system configuration and a global user interface to the CE environment is described.

References

1. Prasad, B. 1996. "Concurrent Engineering Fundamentals: Integrated Product and Process Organization," *Volume I,* New Jersey: Prentice Hall PTR.
2. Dewan, P. and J. Riedl. 1993. "Toward Computer-Supported Concurrent Software Engineering," *IEEE Computer,* Jan.
3. Bayliss, D. C. 1995. "Concurrent Engineering Philosophy Implemented Using Computer Optimized Design," *IMechE.*
4. CERC Report. 1989. "Red Book of Functional Specifications for the DICE Architecture," Feb.
5. Lu, S. C-Y and S. Subramanyam. 1990. "A Cooperative Product Development Environment to Realize the Simultaneous Engineering Concept," pp. 9–18.
6. Sriram, D. and R. Logether. 1993. "The MIT Dice Project," *IEEE Computer,* Jan.
7. Cutkosky, M. R. 1993. "PACT: an Experiment in Integrating Concurrent Engineering Systems," *IEEE Computer,* Jan.
8. Klein, M. 1995. "Integrated Coordination in Cooperative Design," *International Journal of Production Economics.*
9. Olsen, G. R. 1995. "Collaborative Engineering Based on Knowledge Sharing Agreement," *Concurrent Engineering: Research and Applications,* June.
10. Branki, C. NE. 1995. "The Acts of Cooperative Design," *Concurrent Engineering: Research and Applications,* September.
11. Jones, R. M. and E. A. Edmonds. 1995. "Supporting Collaborative Design in a Seamless Environment," *Concurrent Engineering: Research and Applications,* September.
12. Huang, M., F. J. Wang and J. T. Deng. 1995. "Colorboard for Network Supporting of Concurrent Engineering," *INCOM '95,* Beijing.
13. Kinstrey, M. 1990. "PPO Management," CERC Report.
14. Aguiar, M. W. C. and R. H. Weston. 1995. "A Model-Driven Approach to Enterprise Integration," *Int. J. Computer Integrated Manufacturing,* 8(3).
15. Chan, A. 1993. "Concurrent Engineering Using an Enterprise Modeling System," *Concurrent Engineering: Methodology and Applications.*
16. Majumder D. 1994. "Information Management for Integrated Design Environment," *Engineering with Computer,* p. 11.
17. Tiwari, S. and A. Gupta. 1995. "Constraints Management on Distributed Design Configurations," *Engineering with Computers,* p. 11.
18. Chandra, N., M. S. Fox and E. S. Gardner. 1993. "Constraints Management in Design Fusion," *Concurrent Engineering: Methodology and Applications.*
19. O'Grady, P. J. 1994. "A Hierarchical Approach to Concurrent Engineering Systems," *INT. J. Computer Integrated Manufacturing,* 7(3).
20. O'Connor, R. F. and G. B. Williams. 1995. "Standardizing the Design of Automobile Electronic Inter-Connection Architectures Using the Analytical Hierarchy Process," *Concurrent Engineering: Research and Applications,* June.
21. Londono, F. 1990. "A Blackboard Problem Solving Model to Support Product Development," CERC Report.
22. Abdalla, H. S. 1994. "An Expert System for Concurrent Product and Process Design of Mechanical Parts," IMechE.
23. Bowen, J. 1992. "A Constraint-Based Approach to Negotiation in Concurrent Engineering," *The 4th Annual National Symposium on Concurrent Engineering.* Washington.
24. Werkman, K. J. 1993. "Using Negotiation in DAI to Support Concurrent Engineering," *Concurrent Engineering: Methodology and Applications.*
25. Berker, I. and D. C. Brown. 1996. "Conflicts and Negotiation in Single Function Agent Based Systems," *Concurrent Engineering: Research and Application,* March.
26. Bengu, G., B. Prasad and A. Dhar. 1994. "A Multi-Agent Based Cooperative Framework to Support Concurrent Engineering," *CE94, Proceedings of the Concurrent Engineering: Research and Applications Conference,* Pittsburgh, Aug. 29–31.
27. Pohl, K. and S. Jacobs. 1994. "Traceability between Cross-Functional-Teams," *Proceeding of CERA '94,* Pittsburgh, Aug. 29–31.
28. Klein, M. 1993. "Capturing Design Rationale in Concurrent Engineering Teams," *IEEE Computer,* Jan.
29. Uejio, W. H. 1990. "Electronic Design Notebook for the DARPA Initiative in Concurrent Engineering," CE Enabling Technology Selected Technical Papers, CERC.
30. Taylor, G. L. 1995. "Self-Directed R&D Teams: What Makes Them Effective," *Research—Technology Management,* Nov.–Dec.
31. DICE Report. 1991. "DICE GE Project Case Study," CERC.
32. Wang, F. J. and J. T. Deng. 1996. "Work Process Management in Concurrent Engineering," *Advances in Concurrent Engineering, CE96,* Proceedings of the Concurrent Engineering: Research & Applications Conference, August 26–28, Toronto, Canada.
33. Palmer, B. 1990. "Modeling the Concurrent Engineering Process," CERC Report.
34. Kusiak, A. and J. Wang. 1993. "Decomposition of the Design Process," *Journal of Mechanical Design,* Dec.
35. Rolstadas, A. 1995. "Planning and Control of Concurrent Engineering Projects," *International Journal of Production Economics,* 38.
36. Sum, R. N. 1992. "Activity Management: A Survey and Recommendations for the DICE," *General Electric Research & Development,* Jan.
37. Prasad, B. 1997. "Concurrent Engineering Fundamentals: Integrated Product Development," *Volume II,* New Jersey: Prentice Hall PTR.
38. Wang, F. J. 1997. "The Systems of Computer Supported Communication, Coordination, and Control for Team Work," Ph.D. thesis, Beijing University of Aero. and Astronautics, June.

Biren Prasad

Biren Prasad is on the Senior Engineering staff at Electronic Data Systems (EDS), a past subsidiary of General Motors where he holds a Senior Engineering Consultant Position. Dr. Prasad earned his Ph.D. from Illinois Institute of Technology, Chicago, a Degree in Engineering from Stanford University, California. He has written and co-authored over 80 technical publications, including 50 archival papers and nine books. His most recent text is Concurrent Engineering: Fundamentals, Prentice Hall—a two-volume set. He is the founding member of the International Society for Productivity Enhancement (ISPE). Dr. Prasad has served on numerous committees for ASCE, ISPE and SAE. He is the Chairman of the SAE's Computer Attendees' Committee. His professional honors include: Associate Fellow of AIAA, senior member of ISPE, ASCE, SAE, AAAI, and fellow and life member of ISPE. He is listed in numerous Who's Who biographies. He is also a founder and Managing Editor for the Concurrent Engineering: Research & Applications (CERA) Journal—published by Technomic Publishing Inc., since 1990.