Concurrent Engineering
http://cer.sagepub.com/

Petri Net-based Coordination Component for Collaborative Design

Hong-Zhong Huang, Huan-Wei Xu and Xu Zu

Concurrent Engineering 2010 18: 199 originally published online 9 August 2010
DOI: 10.1177/1063293X10379765

The online version of this article can be found at:
http://cer.sagepub.com/content/18/3/199

Published by:
SAGE
http://www.sagepublications.com

CONCURRENT ENGINEERING: Research and Applications

Petri Net-based Coordination Component for Collaborative Design

Hong-Zhong Huang,[1,*] Huan-Wei Xu[1] and Xu Zu[2]

[1]School of Mechatronics Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, China

[2]Beijing Forpetro Sino-rig Co., Ltd, Beijing, 100022, China

Abstract: Considering the lack of efficient coordination of task interdependencies in a collaborative design system, the temporal and resource coordination mechanisms for the problems in our investigation are established based on Petri Nets. The whole system could be expanded as a Petri Net for simulation and analysis. The architecture of reusable and pluggable components is also introduced to implement such mechanisms. The proposed approach offers strong support for the performance evaluation of the collaborative design.

Key Words: Petri Net, coordination mechanism, temporal interdependencies, resource management, component.

1. Introduction

With the rapid development of computer technology and the Internet, we have witnessed the rapid establishment of virtual society and realized the feasibility of remote interactions transcending the constraints of geographic location and time. Virtual software has proven to be an integral part of the design of complex systems in the field of mechanical engineering. When designing complex systems, it is necessary for the multidisciplinary design teams to use multiple sophisticated commercial and noncommercial engineering tools such as computer-aided design (CAD) tools, modeling, simulation and optimization software, engineering databases, and knowledge-based systems. The use of such modern computer technologies has been considered by scientists and engineers as the key to reducing cycle times and improving product quality and reliability [1,2].

Researchers from around the world have designed and developed many models and environments to meet the requirements mentioned above. Wang and Zhang [3] introduced a CAD and computer-aided manufacturing (CAM) integrated system to reduce the product developing cycle in order to respond to the changing market requirements. Some models were established based on technologies such as virtual environment (VE) and Java to access, explore and collaborate through the Internet [1]. Although these models or environments have met those requirements and improved the efficiency of design and development to a certain extent, none of them effectively coordinate interdependent tasks for design of larger systems.

Shirmohammadi and Georganas [4] presented a model that supported tightly coupled collaborative tasks to be performed efficiently in the design system; however, the rigidity of the collaborative protocols restricted possibility of collaboration in the complex system.

In this article, we establish two kinds of coordination mechanisms for temporal and resource interdependencies, respectively, and encapsulate the mechanisms into the coordination components to coordinate the task components, which represent the actual tasks in the collaborative design and development system.

2. Related Concepts

2.1 Design

Design is a process that involves members of many professions. Collaborative design involves teams of designers, engineers, and manufacturers from multiple areas and diverse geographical locations working together over networks [5,6].

Figure 1 shows the mechanism of the so called 'loosely coupled collaborative design.' The design activities are coordinated by social protocols and participants' abilities even without any explicit coordination mechanisms.

Figure 2 shows the activity called 'tightly (or closely) coupled collaborative design'. In this form of collaborative design, one task depends on another to start, to

*Author to whom correspondence should be addressed.
E-mail: hzhuang@uestc.edu.cn

1063-293X/10/03 0199–7 $10.00/0 DOI: 10.1177/1063293X10379765

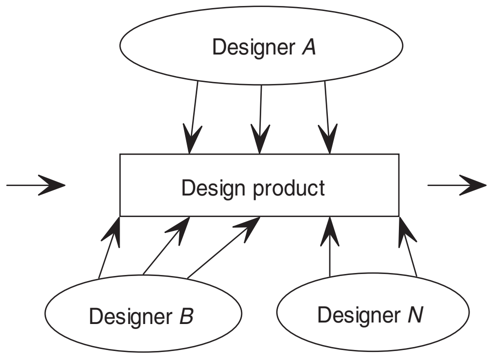

Figure 1. Loosely coupled collaborative design.

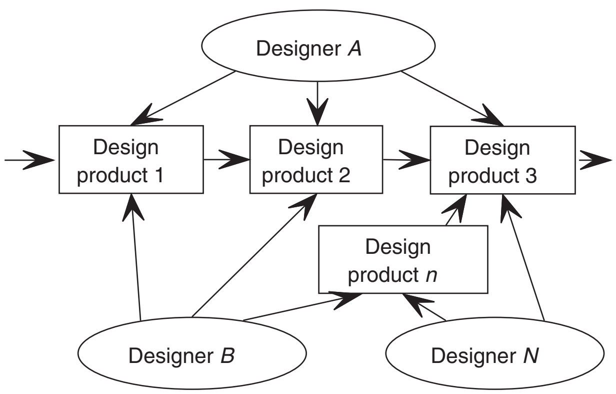

Figure 2. Tightly coupled collaborative design.

execute, or to end. Sophisticated coordination mechanisms are necessary for this kind of activity. During the process of tightly coupled collaborative design, each participant relies on the success of every other task and partner; however, such an 'assembly line' is not always possible. As a result, there must be coordination between these interdependent tasks to ensure that the whole system is executed successfully [7].

2.2 Computer Supported Collaborative Work

In collaborative design systems, individuals or individual groups of multidisciplinary design teams usually alternate between working in parallel and working separately with various engineering tools located in different sites. At any moment, individual members may be working on different versions of a design or viewing the design from different perspectives, at different levels of details. Computer supported collaborative work (CSCW) is used for the development of such environments.

CSCW systems offer a tool that could potentially enhance the productivity and effectiveness of teams through more effective communication. The primary goals of applying the CSCW technology include cost reduction, space optimization, improving performance, effectiveness, and satisfaction. There are also more specific goals, such as improving group cohesiveness and eliminating the influence of dominant figures [5,8].

2.3 Component

A component is a reusable software package or application. A standard application has two main parts: the implementation and the data. The interaction between two applications is through a database and is limited to data sharing. A component-based application provides operation services containing the data operation and the method operations in its implementation [9].

Cockburn and Jones [10] examined and recorded the major causes of component failure, and provided four principles of component design to address these causes and guide the designers. The four principles are personnel acceptance maximization, requirement minimization, constraint minimization, and external integration. These are the basic principles for establishing all kinds of components.

3. Coordination Mechanisms

The design and development of a product involves many tasks, and these tasks could be further divided into many subtasks. We can classify these tasks into two main categories: independent and interdependent. An independent task has no or little relation with others and therefore can be implemented separately, while an interdependent task involves interaction with other tasks in one or more aspects. Experience tells us that most tasks are interdependent. Interdependent tasks are divided into two types: temporal and resource dependencies [11,12].

3.1 Temporal Dependencies

Temporal dependencies establish the execution order for a series of tasks. Figure 3 shows the temporal dependencies between two tasks [7].

1. Task A equals Task B: Task A and Task B start together and have the same time interval.
2. Task A starts Task B: Task A and Task B start together, but have different time intervals.
3. Task A finishes Task B: Task A and Task B finish together, and have different time intervals.
4. Task A during Task B: Task A is totally contained in Task B.
5. Task A overlaps Task B: Task A starts before Task B, which starts before the end of Task A.
6. Task A before Task B: Task A finishes before Task B starts, and they do not overlap.

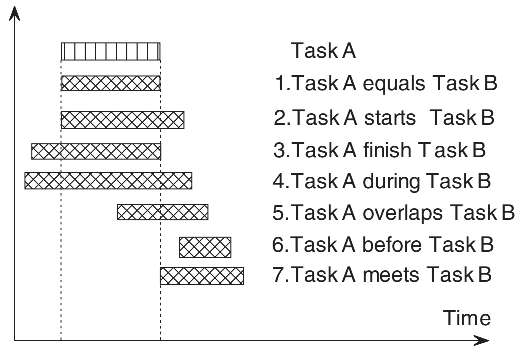

Figure 3. Temporal interdependencies.

7. Task A meets Task B: Task A finishes before Task B, which starts immediately after the end of Task A.

These conditions describe the possible relationships between two tasks. What we could do is to establish corresponding mechanisms to coordinate these dependencies.

3.2 Resource Management

The resources needed for tasks are managed by one or more resource managers. Resource managers control the allocation of resources to various tasks. Resource management interdependencies are complementary to temporal dependencies and may be used in parallel to them. This kind of interdependency deals with the distribution of resources among tasks. Three basic resource management dependencies are defined here [7,13].

3.2.1 SHARING

A limited number of resources may be shared among several tasks; for example, several designers editing a product drawing.

3.2.2 SIMULTANEITY

A resource is available only if a certain number of tasks request it simultaneously; for instance, a machine that may only be used with more than one operator.

3.2.3 VOLATILITY

A resource that is available again after being used; for example, a sheet of paper is a volatile resource while a printer is a nonvolatile resource.

3.3 Coordination Mechanisms

In this context, coordination is defined as the act of managing interdependent tasks to achieve a goal. Coordination is a highly dynamic process because of the re-negotiation during a collaborative effort. Without coordination mechanisms, the conflict between participants and repetition of components may result [14].

One coordination mechanism can only solve one kind of corresponding problem. Thus, in large systems for designing a complex product, multiple coordination mechanisms need to be established to solve a wide range of potential problems.

The coordination mechanisms built for temporal and resource management are based on the classical Petri Nets. Figure 4 shows the structure of a task which is based on the Petri Nets and Figure 5 shows the structure of resource manager [13]. As shown in Figure 4, each task is dependent on another task with five transitions (P1, P2, P3, P4, and P5) and four places (S1, S2, S3, and S4). The places titled request_resource, assign_resource, and release_resource connect the tasks with the resource manager. The places titled execute_task and finish_task connect the task with the temporal coordination mechanisms. Consequently, both the temporal and resource interdependencies are coordinated.

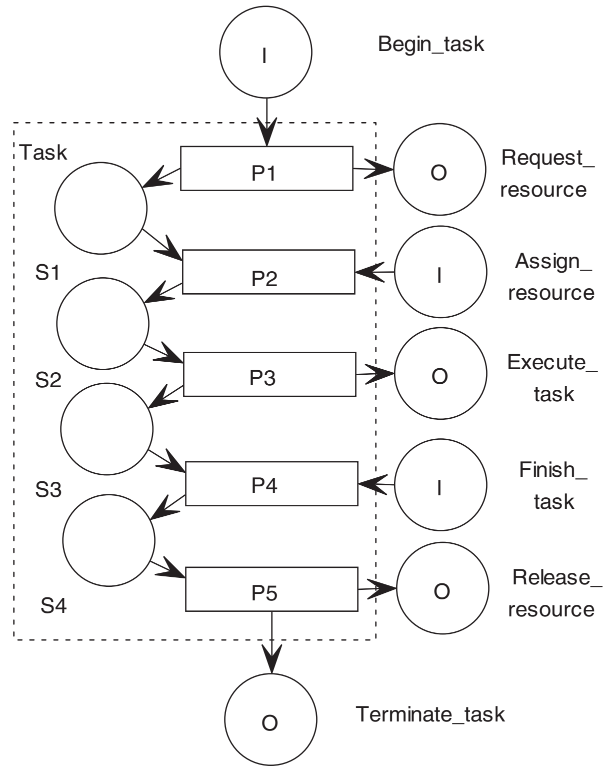

Figure 4. Structure of task [13].

The proposed coordination environment includes three distinct hierarchical levels: workflow, coordination, and execution [14]. The whole design system is delineated at the workflow level. All tasks are assigned to different participants, with simultaneous establishment of temporal and resource interdependencies between the tasks and the participants. Under the workflow is the coordination level where the interdependent tasks are coordinated by the coordination components. All the tasks of a system are actually executed at the execution level.

4. The Coordination Component

Figure 6 shows the components involved in the coordination of interdependencies between two tasks in the three level scheme. At the coordination level, there are three components, a coordination component and two task components. The two task components represent Task A and Task B, respectively and maintain the task's schedule. The coordination component implements the modeled coordination mechanisms, both for temporal and resource management dependencies.

In order to coordinate temporal relations (shown in Figure 3), several coordination mechanisms must be established. Figure 7 presents the model of the coordination mechanism for the temporal relation, Task A equals Task B. Between the two task components, Task A and Task B, is the temporal coordination component which ensures that the beginning and the completion of Task A and Task B occur at the same time. Transition T1 ensures simultaneous initiation of the two tasks and Transition T2 ensures simultaneous completion of the two tasks. Task A and Task B in the coordination component are noninstantaneous transitions, which mean that they are the actual execution of the two tasks in the execution level. Different coordination mechanisms are established to coordinate respective interdependencies, as shown in Figure 3.

In Figure 8, between the two tasks is the resource coordination component which manages the resources between Task A and Task B. This figure shows the sharing function, which is one of the basic resource management dependencies defined above. The place

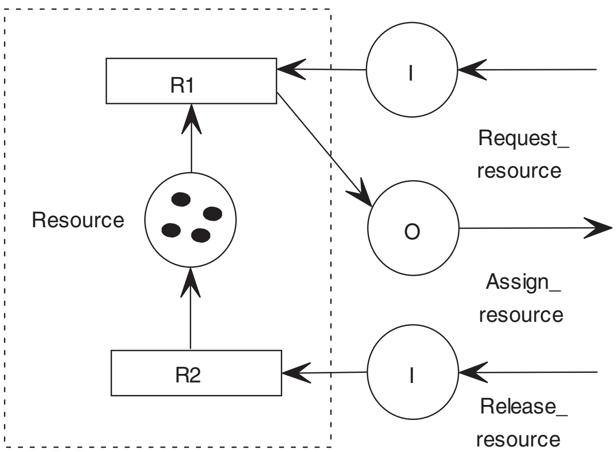

Figure 5. Structure of resource manager [13].

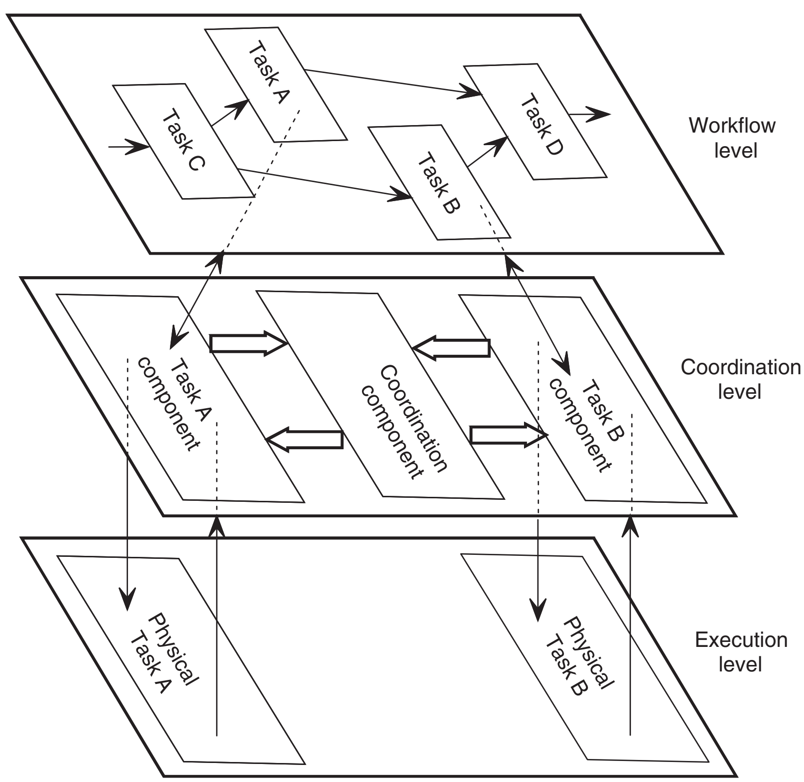

Figure 6. Three levels of collaborative design system.

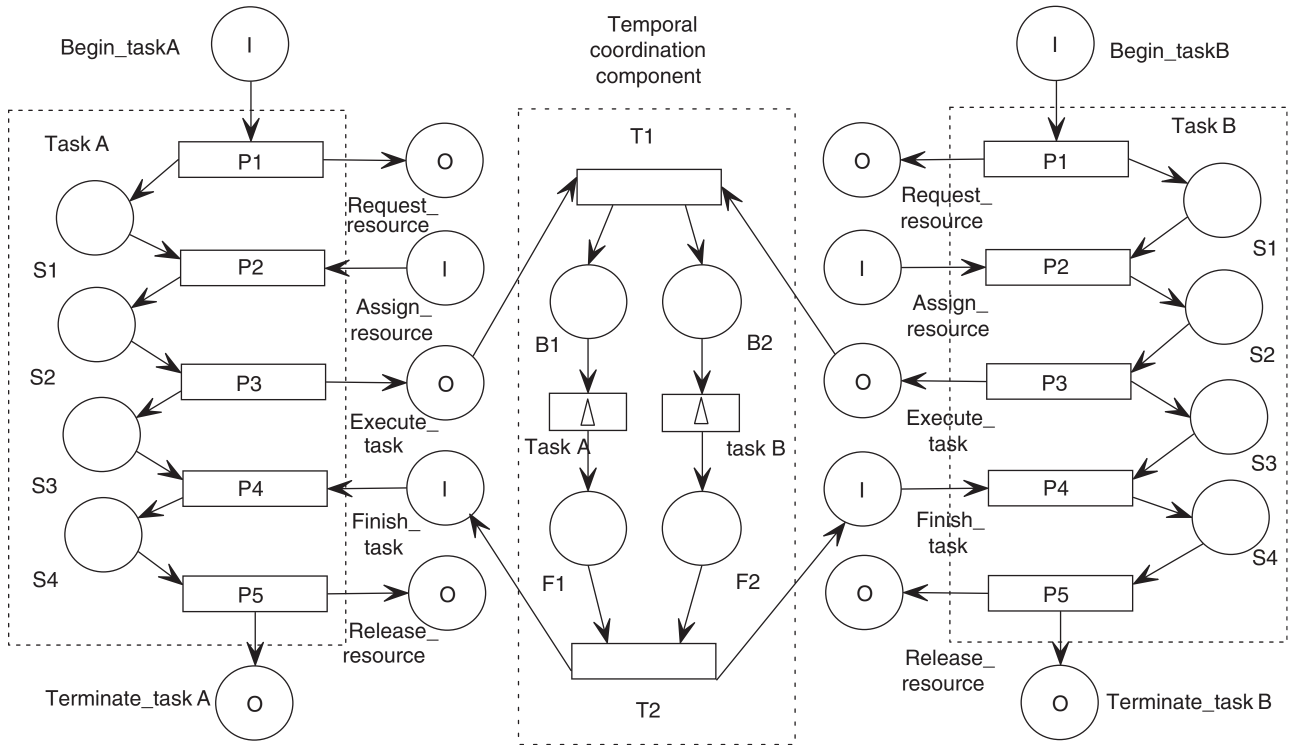

Figure 7. Coordination mechanisms for Task A equals Task B.

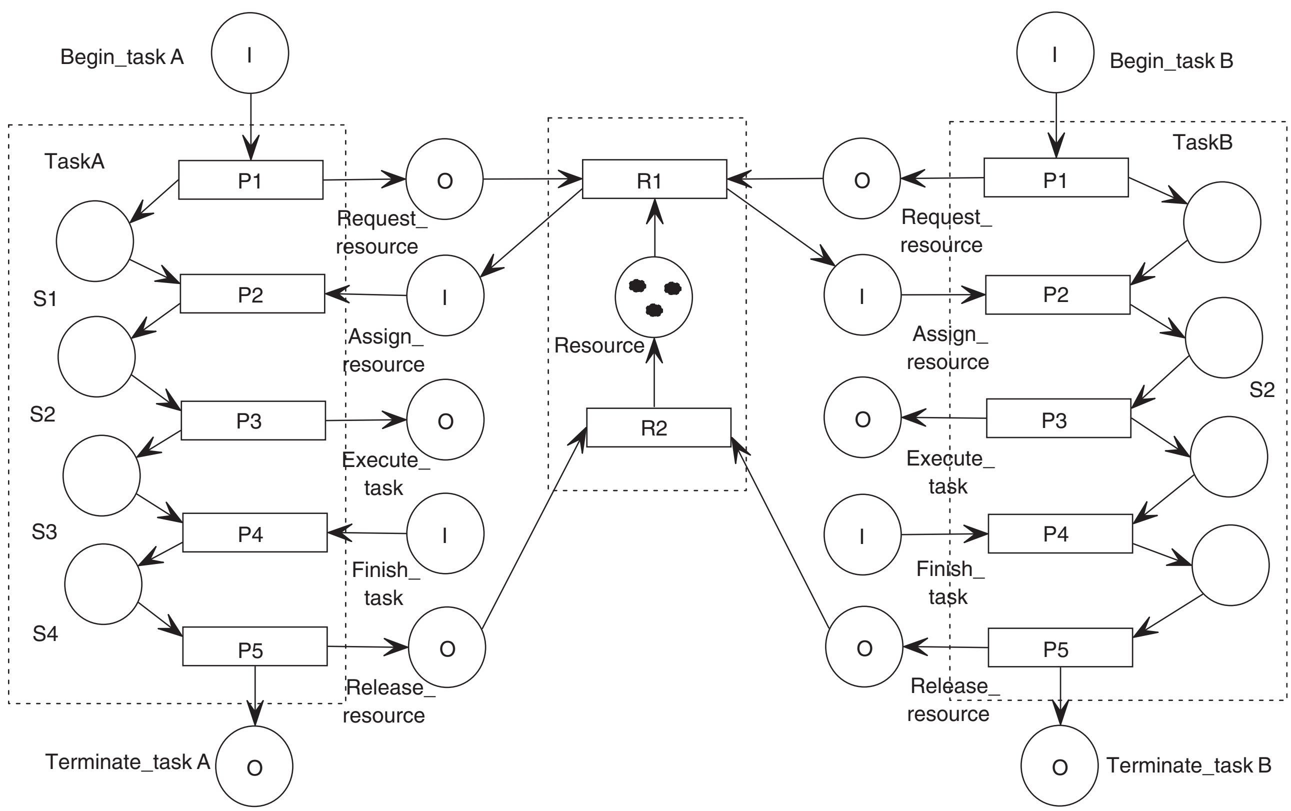

Figure 8.. Resource manager for two tasks sharing resources.

resource contains three tokens which represent the available resources. According to the conditions of the input places, namely, request_resource and resource, Transition R1 will determine whether to assign the corresponding resources to the task which has made the request. Transition R2 reclaims the resources which have been released.

In the design and development system, the coordination component is established to coordinate the interdependent tasks. Temporal and resource coordination mechanisms, as two main mechanisms of the coordination component, are encapsulated into the component to implement their functions. As a result, in the coordination component, besides the interaction between the coordination and tasks, there are interactions between these two mechanisms to ensure the component's efficiency. Because we have established all kinds of coordination mechanisms for the resource and temporal interdependencies, the coordination component could coordinate the CSCW system automatically and efficiently.

5. Conclusions

In this article, we have presented an approach for task coordination in the CSCW system. Task interdependencies include temporal and resource interdependencies, and there are several kinds of relations between tasks in these two kinds of interdependencies. Different coordination mechanisms have been respectively established according to these conditions. As all of these mechanisms are based on the Petri Nets, the whole system could be expanded as a Petri Net for simulation and analysis, and therefore some potential problems in the system could be considered in advance. The approach that encapsulates the two mechanisms into the component, which is called the coordination component makes the coordination in a modular way and the whole system more flexible.

In the future, we will combine the CSCW system with virtual environment. The system will be more complex, and the coordination mechanisms will be developed to meet the new requirements of the system. We will modify the mechanisms to improve the ability of coordination, and at the same time, improve the computing ability of the design system to reduce the chance of deadlocks.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China under the contract number 50775026, and Taiyuan Heavy Machinery (Group) Co., LTD, China.

References

1. Benford, S., Snowdon, D., Brown, C., et al. (1997). Visualizing and Populating the Web: Collaborative Virtual Environment for Browsing, Searching and Inhabiting Web Space, *Computer Networks and ISDN Systems*, **29**: 1751–1761.
2. Kan, H.Y., Duffy, V.G. and Su, C.J. (2001). An Internet Virtual Reality Collaborative Environment for Effective Product Design, *Computers in Industry*, **45**: 197–213.
3. Wang, H.F. and Zhang, Y.L. (2002). CAD/CAM Integrated System in Collaborative Development Environment, *Robotics and Computer Integrated Manufacturing*, **18**: 135–145.
4. Shirmohammadi, S. and Georganas, N.D. (2001). An End-to-End Communication Architecture for Collaborative Virtual Environments, *Computer Networks*, **35**: 351–367.
5. Kvan, T. (2000). Collaborative Design: What Is It? *Automation in Construction*, **9**: 409–415.
6. Xu, X.W. and Liu, T. (2003). A Web-Enabled PDM System in a Collaborative Design Environment, *Robotics and Computer Integrated Manufacturing*, **19**: 315–328.
7. Raposo, A.B., Magalhaes, L.P., Ricarte, I.L.M., et al. (2001). Coordination of Collaborative Activities: A Framework for the Definition of Tasks Interdependencies, In: *Proceedings of 7th International Workshop on Groupware*, Darmstadt, Germany, 6–8 September, pp. 171–179.
8. Monplaisir, L. (1999). An Integrated CSCW Architecture for Integrated Product/Process Design and Development, *Robotics and Computer Integrated Manufacturing*, **15**: 145–153.
9. Rosenman, M. and Wang, F.J. (2001). A Component Agent Based Open CAD System for Collaborative Design, *Automation in Construction*, **10**: 383–397.
10. Cockburn, A. and Jones, S. (1995). Four Principles of Groupware Design, *Interacting with Computers*, **7**(2): 195–210.
11. Raposo, A.B., Cruz, A.J.A., Adriano, C.M., et al. (2001). Coordination Components for Collaborative Virtual Environments, *Computers & Graphics*, **25**: 1025–1039.
12. Da Cruz, A.J.A., Raposo, A.B. and Magalhaes, L.P. (2002). Coordination in Collaborative Environments-A Global Approach, In: *Proceedings of the 7th International Conference on Computer Supported Cooperative Work in design*, Rio de Janeiro, Brazil, 25–27 September, pp. 25–30.
13. van der Aalst, W.M.P., van Hee, K M. and Houben, G J. (1994). Modeling and Analyzing Workflow Using a Petri Net Base Approach, In: *Proceedings of 2nd Workshop on Computer-Supported Cooperative Work, Petri net and Related Formalisms*, June, Zaragoza, Spain, pp. 31–50.
14. Raposo, A.B., Magalhaes, L.P., Ricarte, I.L.M., et al. (2000). Petri Nets Based Coordination Mechanisms for Multi-workflow Environments, *Computer Systems Science and Engineering*, **15**(5): 315–326.

Dr Hong-Zhong Huang

Dr Hong-Zhong Huang is a professor and the Dean of the School of Mechanical, Electronic, and Industrial Engineering at the University of Electronic Science and Technology of China in Chengdu, Sichuan, China. He received his PhD degree in Reliability Engineering from Shanghai Jiaotong University in Shanghai, China. He has published 150 journal articles and five books in the fields of reliability engineering, optimization design, fuzzy sets theory, and product development and has held visiting appointments at several universities in Canada, USA, and several countries in Asia. He is a Regional Editor of International Journal of Reliability and Applications, an Editorial Board Member for The International Journal of Reliability, Quality, and Safety Engineering, International Journal of Reliability and Quality Performance, International Journal of Performability Engineering, Advances in Fuzzy Sets and Systems, and The Open Mechanical Engineering Journal. In 2006, Huang was awarded the William A. J. Golomski Award from the Institute of Industrial Engineers, and he received the Best Paper Award of the 8th International Conference on Frontiers of Design and Manufacturing in 2008. His current research interests include system reliability analysis, warranty, maintenance planning and optimization, and computational intelligence in product design.

Dr Huan-Wei Xu

Dr Huan-Wei Xu is currently a lecturer at the School of Mechatronics Engineering at the University of Electronic Science and Technology of China in Chengdu, Sichuan, China. He received a PhD degree in Mechanical Engineering from Dalian University of Technology in Dalian, China. He has published 10 journal papers, and his research interests include reliability design, robust design, and multi-disciplinary design.

Dr Xu Zu

Dr Xu Zu is currently an engineer in Beijing Forpetro Sino-Rig Co., Ltd, China. He received a PhD degree in Mechanical Engineering from Dalian University of Technology in China. He has published 10 papers. His research interests include drilling rig research and development, concurrent engineering, and design optimization.