



# Dynamic configuration and collaborative scheduling in supply chains based on scalable multi-agent architecture

Fu-Shiung Hsieh<sup>1</sup>

Received: 6 April 2018 / Accepted: 1 September 2018 / Published online: 14 September 2018  
© The Author(s) 2018

## Abstract

Due to diversified and frequently changing demands from customers, technological advances and global competition, manufacturers rely on collaboration with their business partners to share costs, risks and expertise. How to take advantage of advancement of technologies to effectively support operations and create competitive advantage is critical for manufacturers to survive. To respond to these challenges, development of a dynamic scheme to better manage collaborative workflows is urgent. In this paper, we will study how to develop a flexible and scalable framework to dynamically and coherently configure workflows that can meet order requirements based on multi-agent systems (MAS). Configuring and scheduling collaborative workflows is a challenging problem due to the computational complexity involved, distributed architecture and dependency among different partners' workflows. To achieve flexibility and reduce the cost and time involved in configuration of a supply chain network, we propose an approach that combines MAS, contract net protocol, workflow models and automated transformation of the workflow models to dynamically formulate the scheduling problem. To attain scalability, we develop a solution algorithm to solve the optimization problem by a collaborative and distributed computation scheme. We implement a software system based on industrial standards, including FIPA and the Petri net workflow specification model. In addition, we also illustrate effectiveness and analyze scalability of our approach by examples. Our approach facilitates collaboration between partners and provides a scalable solution for the increasing size of supply chain networks.

**Keywords** Supply chain · Configuration · Scheduling · Scalability · Multi-agent system

## Introduction

Frequently changing demands, global competition and technological advances pose new challenges to the manufacturing sector. Traditional centralized hierarchical organizations cannot effectively respond to rapidly changing demands, innovative production processes and highly dynamic business partnership in supply chains. How to take advantage of the advancement of technologies to effectively support the operation and create competitive advantage is critical for enterprises to survive. To reap the potential technological benefits, new organizational structure and strategy must be developed to effectively manage the business processes/workflows,

resources and changes in business environment to support inter-enterprise collaboration.

Manufacturers rely on cooperation and collaboration of business partners to share costs, risks and expertise as no single company has all the expertise needed. The partners need to collaborate to achieve a business goal by forming a supply chain network. A supply chain is a system of organizations, people, activities, information and resources involved in transforming resources, materials and components into a finished product that is delivered to the end customer. Please refer to the book by Hugos (2018) for an introduction to supply chain management and the recent works by Moktadir et al. (2018a, b) on sustainable supply chain management and robustness of supply chains by Monostori (2018). The systemic, strategic coordination of the traditional business functions and the tactics across these business functions within a particular company and across businesses within the supply chain for improving the long-term performance is the essence of supply chain management (Mentzer et al. 2001).

---

✉ Fu-Shiung Hsieh  
fshsieh@cyut.edu.tw

<sup>1</sup> Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung, Taiwan

Lambert et al. defined supply chain management as, “the integration of key business processes from end user through original suppliers, that provides products, services, and information that add value for customers and other stakeholders” (Lambert et al. 2006). The concept of virtual enterprises (VE) makes it possible to achieve business goals through dynamic coalition and sharing of core competencies and resources in supply chains. It also poses new challenges and issues (Petrie and Bussler 2003). A wide variety of research issues and topics of VE have been studied, including cooperation/coordination by Camarinha-Matos and Pantoja-Lima (2001), formation by Afsarmanesh and Analide (2009), Camarinha-Matos et al. (2009), Hoffner et al. (2001), partner selection by Camarinha-Matos and Cardoso (1999), Hsieh and Lin (2012), planning and control by Soares et al. (2000), McFarlane and Bussmann (2000), dynamic network process management by Grefen et al. (2009), dynamic process composition by Hsieh and Chiang (2011) and design and implementation of automated procurement systems by Jagdev et al. (2008) in VE. Recently, Samdantsoodol et al. (2017) studied how to predict the relationships between VE and agility in supply chains. Kovács and Kot (2017) studied economic and social effects of novel supply chain concepts and VE. However, scalability of supply chains and VE is not addressed. Scalability is an important issue but there is little study in the context of supply chains and VE. Scalability is defined by Putnik et al. (2013) as the design of a system with adjustable structure to enable system adjustment in response to market demand changes. Scalability is a system’s feature that provides potentials for resolving a number of problems in supply chains design and operation of VE.

In the existing literature, several issues of supply chain management have been studied. Liu and Chung (2017) considered a two-stage supply chain problem in which the first stage is to produce jobs by several suppliers and the second stage is to transport those jobs by a number of vehicles. A centralized mathematical model is established to describe the problem and develop a solution algorithm. However, the scheduling problem for multistage supply chain was not addressed in [a]. Ivanov et al. (2016) studied a two-stage supply chain with job shop processes at each supplier stage. Ivanov et al. introduced a robust analysis of schedule coordination in the presence of disruptions in capacities and supply to derive managerial insights for scheduling problem and dynamic control methods for supply chain coordination. However, an extended study of Ivanov et al. (2016) is required for multistage supply chain. To respond to business opportunities, an important issue is to develop a flexible, reconfigurable and scalable technology for integration of collaborative workflows or processes of partners in supply chains.

In this paper, we will focus on dynamic configuration and collaborative workflow scheduling in VE to attain flexibility, reconfigurability and scalability. The problem is

to develop a solution methodology for the partners to configure their resources and create workflow schedules to fulfill customers’ orders timely under workflow and resource constraints. An effective scheme for managing collaborative workflows in supply chain networks should provide a methodology that is flexible, reconfigurable and scalable to respond to business opportunities. We will propose architecture and design methodology to reduce cost and time in the development of software for managing collaborative workflows. In addition to develop software to support workflow scheduling in supply chains, we also study the scalability of our approach in terms of response time. Response time is an important performance index in measuring scalability of a supply chain management method. To be applicable in supply chains, response time should be acceptable as the number of partners in the network grows. We exploit recent advancements in multi-agent systems, scheduling theory and algorithms to propose a scalable method to dynamically and collaboratively configure and schedule workflows in supply chains.

To achieve flexibility, reconfigurability and scalability, several requirements must be met. First, the workflow of a company in supply chains must be described and specified by a standard format. Second, the multi-agent system platform used for implementation must also support information infrastructure and interaction protocols/mechanism defined by industrial standard organization to attain interoperability between agents. Third, the scheduling algorithm must be developed based on dynamic publication and discovery of services. Fourth, the scheduling method must be scalable by taking advantage of distributed computing architecture based on a divide-and-conquer strategy. Fifth, to achieve interoperability, all information in the negotiation processes, including call for proposals, proposals, awarding of contracts and establishment of contracts, must be described based on a standard format such as XML.

To propose a pragmatic, sustainable, flexible and scalable methodology for solving the workflow management problem, a proper architecture and suitable models must be adopted. Nilsson (1998) and Ferber (1999) indicated that the distributed architecture of multi-agent systems (MAS) and agents’ characteristics of autonomy and cooperation make MAS a potential model for managing collaborative workflows. Cooperative distributed problem solving (CDPS) by Durfee et al. (1989) is a technique for loosely coupled network of problem solvers to work together to solve problems that are beyond their individual capabilities. It is an approach for solving a problem based on coordination and cooperation in multi-agent systems (MAS). Our solution methodology combines MAS architecture by Ferber (1999) and Nilsson (1998) with Petri net models (Murata 1989) and CDPS.

In our architecture, the workflow to be performed by an agent is represented by a workflow agent and each resource



is modeled by a resource agent. The study by Wang et al. (2007) shows that MAS provide a flexible architecture for capturing the main features of VE and agent-based computer integrated manufacturing systems. In MAS, the most well-known protocol for coordination and negotiation is the contract net protocol (CNP) by Smith (1980). There are a lot of works by Parunak (1987), Ramos (1996), Neligwa and Fletcher (2003), Hsieh and Lin (2014a, b) on distributing tasks in MAS with CNP. Our approach takes advantage of CNP and the service publication/discovery capabilities of MAS defined by FIPA. To facilitate representation of workflows, a modeling tool or specification language is required to represent atomic services. In the existing literature, many workflow specification languages have been proposed, for example XPDL by Workflow Management Coalition (1999) and Web Services Business Process Execution Language (WS-BPEL) by OASIS (2009). However, these workflow specification languages lack formal analysis method. The works by van der Aalst (1998), van der Aalst and Kumar (2001) and Weske et al. (2004) indicate that Petri net is an effective model for modeling and analysis of workflows. To endow each agent with the knowledge to perform operations in the workflows, we construct the timed Petri net (TPN) model for each workflow agent and resource agent. The Petri Net Markup Language (PNML) by Weber and Kindler (2002) and Billington et al. (2003) is an XML-based interchange standard for Petri nets. Therefore, we adopt PNML as the format for representing the Petri net models. Our approach uses a software module to formulate the scheduling problem based on the Petri net models and the order requirements. The cost and time involved in the development of scheduling software can be significantly reduced. The collaborative workflow scheduling problem can be decomposed into a number of interrelated workflow scheduling subproblems that are solved by individual agents. To schedule workflows, we first transform the TPN models into network models and then develop a scheduling algorithm by combining network models, Lagrangian relaxation and subgradient algorithm. We illustrate our dynamic configuration and collaborative workflow scheduling method by examples.

Based on the proposed dynamic configuration and workflow scheduling method, we will present the analysis to show that our method is scalable in terms of response time as the size of supply chain network grows. The response time is the longest response time of all directed paths that start with a leaf node and end with the final node of a supply chain network. We also illustrate scalability of our approach by examples. We have compared our approach with an industrial centralized problem solver used in the existing literature. In the work by Liu and Chung (2017), a two-stage supply chain is considered and a

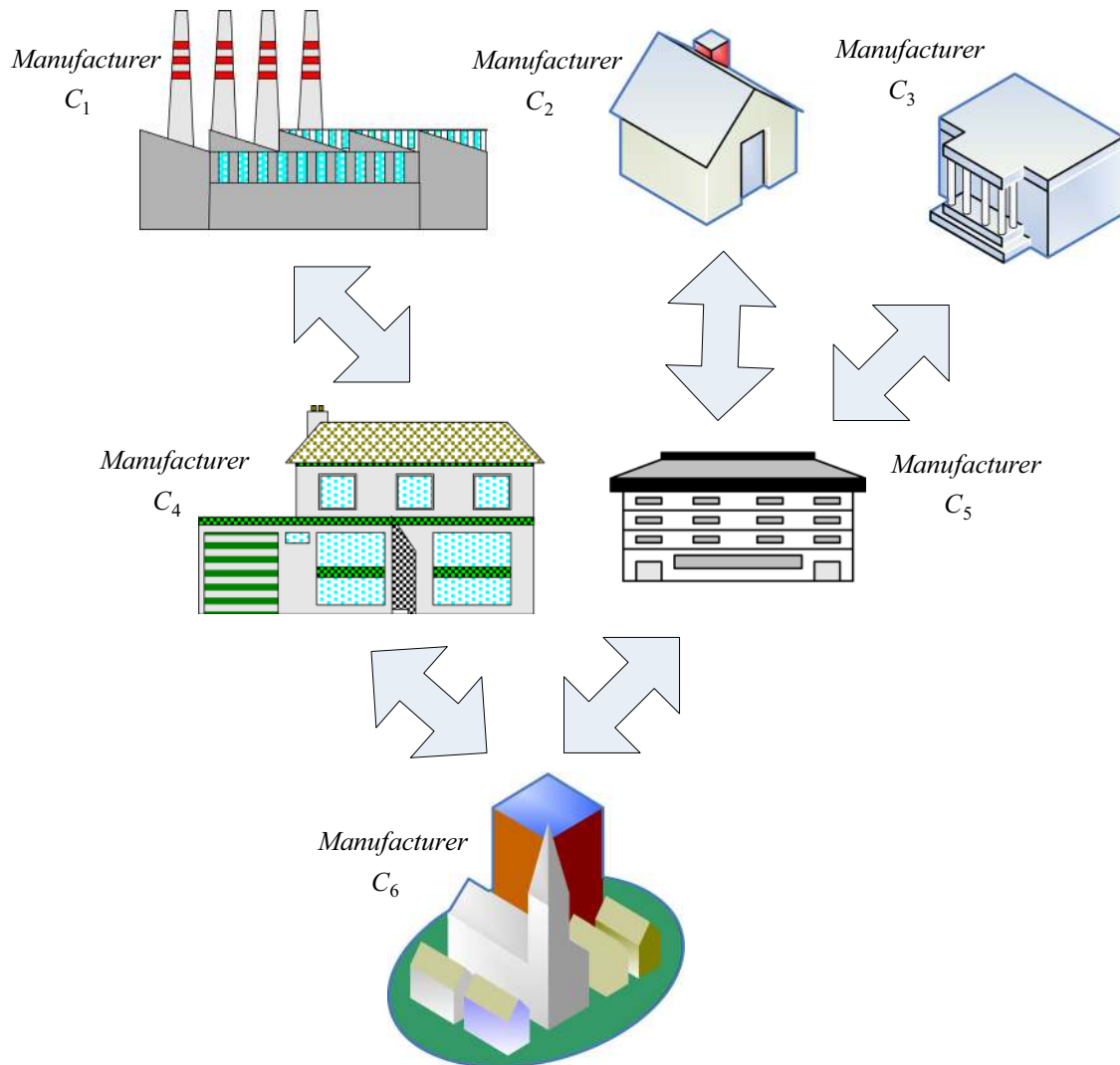
centralized mathematical model is established to describe the problem and develop a solution algorithm. Both our analysis and numerical results in “[Scalability analysis and verification by examples](#)” section indicate that our approach is much more efficient than the centralized industrial problem solver as the supply chains grow.

The remainder of this paper is organized as follows. In “[Dynamic configuration and workflow scheduling of supply chain networks](#)” section, we describe the dynamic configuration and workflow scheduling problem in supply chains and introduce our approach in “[A model-based collaborative scheduling approach](#)” section. In fourth section, we propose a “[Subgradient method for scheduling collaborative workflows](#).” In “[Agent interaction model for collaborative scheduling](#)” section, we introduce our agent interaction model for configuring and scheduling collaborative workflows. We present the experimental results by examples in “[Numerical results](#)” section. In “[Scalability analysis and verification by examples](#)” section, we conduct scalability analysis and verify our analysis by examples. We conclude this paper in “[Conclusions](#)” section.

## Dynamic configuration and workflow scheduling of supply chain networks

Figure 1 illustrates a supply chain formed by six companies. For a manufacturer, usually there are several activities performed in their operations, from order management, product design, process design, manufacturing to delivery. Although there are a variety of tools that support these activities individually, there is still a lack of methodology that addresses how to effectively configure supply chain networks and manage relevant workflows of partners. As the above-mentioned activities are usually distributed, an urgent need is to develop a framework to support dynamic configuration of supply chain networks and optimization of workflows in collaborative and distributed environment. In this paper, we focus on the development of a software platform to support dynamic supply chain network configuration and collaborative workflow management.

Note that formation of a supply chain network and scheduling of workflows in the supply chain network are two related problems. In practice, a two-stage process is usually adopted to find a solution. At the first stage, generation of a supply chain network is done first. At the second stage, scheduling of workflows is then done for the generated supply chain network. However, such a two-stage process may not lead to satisfactory results as the configuration of a supply chain network will influence the performance of workflows. In addition, a supply chain network formed at the first stage may not be able to generate feasible schedules to meet the order requirements due



**Fig. 1** Manufacturers in a supply chain

to insufficient capacity of the resources provided by the partners. Therefore, an effective approach should generate the supply chain network and schedules simultaneously and dynamically. Workflow scheduling in a supply chain is a complex issue. In supply chains, the requirements of an order are specified by the product demands, price and due date. The decisions of a company depend on those of its upstream partners and have influence on its downstream partners. The decisions of different companies in a supply chain must be coherent so that the customers' order requirements can be met timely and cost effectively. Development of an effective workflow scheduling method is an important research issue in supply chain management.

Figure 2 illustrates the collaboration of six companies  $C_1$  through  $C_6$ . Company  $C_n$  must create an associated workflow schedule  $S_n$  to meet the order requirements. To develop an effective scheme to schedule workflows based on collaboration of partners, a problem formulation is

required. Let  $\{1, 2, \dots, N\}$  be the set of companies involved in the scheduling decisions. We use  $w_n$  to represent workflow agent corresponding to  $C_n$ . The workflow of company  $n \in \{1, 2, \dots, N\}$  is described by a workflow model. Let  $Q$  denote the product demand of order  $o$  placed to a supply chain. Let  $\psi$  denote the due date for completing an order  $o$ . The set of all workflow agents is denoted by **WA**. The operations in the workflows need to be performed by some resources. Let  $\mathfrak{R}$  denote the set of all resources in the system. The activities of a resource are described by a resource activity model  $a_r$ ,  $r \in \mathfrak{R}$ . In our workflow scheduling system, we use  $a_r$  to represent resource agent corresponding to resource  $r \in \mathfrak{R}$ . Let **WR** denote the set of all resource agents in the system.

A supply chain network for handling an order  $o$  is denoted by a digraph  $\text{SCM}(\text{WA} \cup \text{RA}, E)$ , where  $\text{WA} \subseteq \mathbf{WA}$  is the set of nodes of workflow agents in SCM,  $\text{RA} \subseteq \mathbf{WR}$  denotes the set of nodes of resource agents that take



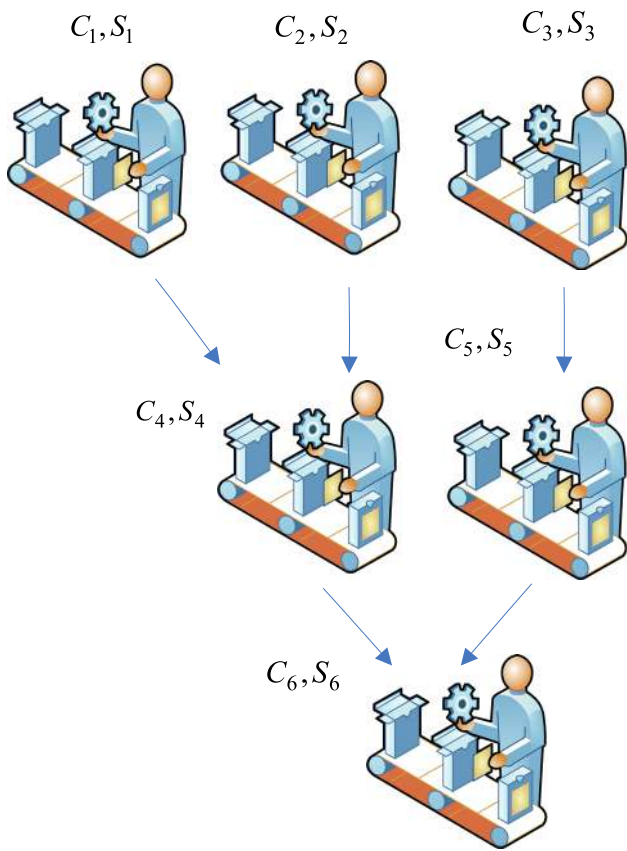


Fig. 2 Dependency of workflow scheduling decisions in a supply chain

part in the activities in SCM and  $E$  is the set of arcs connecting nodes. An arc in  $E$  represents the dependency between two workflow agents. Figure 3 shows the digraph representation of a supply chain network associated with Fig. 2.

Note that  $SCM(WA \cup RA, E)$  only defines the structure of a supply chain. A supply chain must be constructed

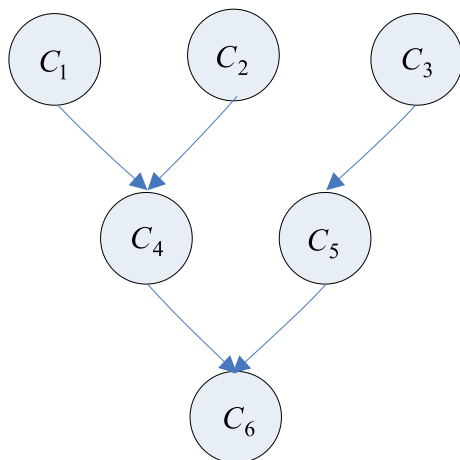


Fig. 3 Digraph representation of a supply chain

dynamically to respond to business opportunities. In addition, the operations of SCM must be scheduled properly for each  $a \in WA \cup RA$ . Therefore, the workflow management problem in SCM can be broken down into two related subproblems: (1) configuration/formation of  $SCM(WA \cup RA, E)$  and (2) scheduling of operations/workflows for each agent involved in  $SCM(WA \cup RA, E)$  to meet the order requirements, including product demand and due date. The aforementioned problem calls for the development of a problem-solving mechanism to determine whether there exist  $WA, RA$ , where  $WA \subseteq \mathbf{WA}$  and  $RA \subseteq \mathbf{WR}$ , and associated schedules  $S_a$  for each  $a \in WA \cup RA$  such that the order requirements can be met. To solve this problem, a divide-and-conquer approach that combines distributed computation capability of MAS, formal models of agents’ workflows/activities and optimization theory is adopted. We develop a solution methodology to solve the dynamic supply chain configuration and collaborative workflow scheduling problem based on interaction of agents and application of optimization technique to individual agents. To facilitate negotiation between order agents, workflow agents and resource agents, formal models for workflow agents and resource agents are proposed in the next section.

### A model-based collaborative scheduling approach

Our approach to workflow scheduling relies on interaction between order agents, manager agent, workflow agents, resource agents and collaborative scheduling agents. Figure 4 shows the connection between an order agent, workflow agents, resource agents and collaborative scheduling agents, where a collaborative scheduling agents consists of two procedures: automated scheduling problem formulation based on network flow model construction and a subgradient-based algorithm for solving scheduling problem. In Fig. 4, Order Agent 1 places an order to Workflow Agent 1, which invokes Collaborative Scheduling Agent 1. Note that in the process of scheduling, Collaborative Scheduling Agent 1 interacts with Resource Agent 1 and Resource Agent 2. Workflow Agent 1 then requests Workflow Agent 2, which is at the upstream of Workflow Agent 1, to invoke Collaborative Scheduling Agent 2 to schedule operations. Collaborative Scheduling Agent 2 then interacts with Resource Agent 3 and Resource Agent 4. Interactions among different types of agents are based on a negotiation mechanism that extends the well-known CNP by Smith (1980). CNP relies on an infrastructure for individual agents to publish and discover their services and communicate with each other based on the ACL language defined by the FIPA international standard

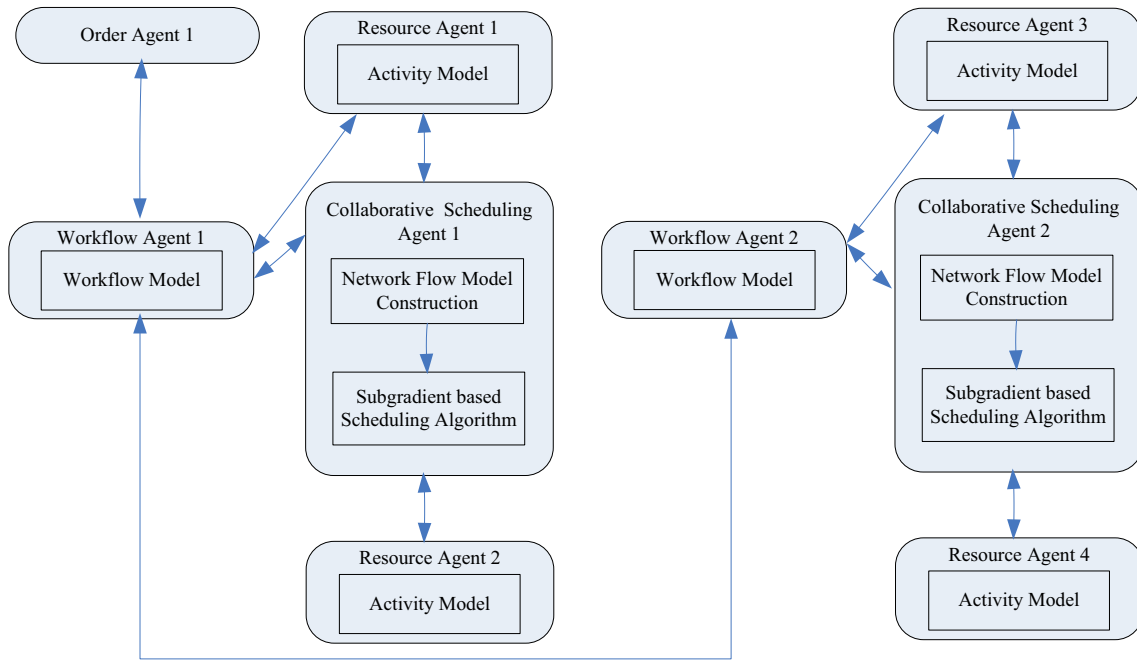


Fig. 4 Architecture for scheduling collaborative workflows

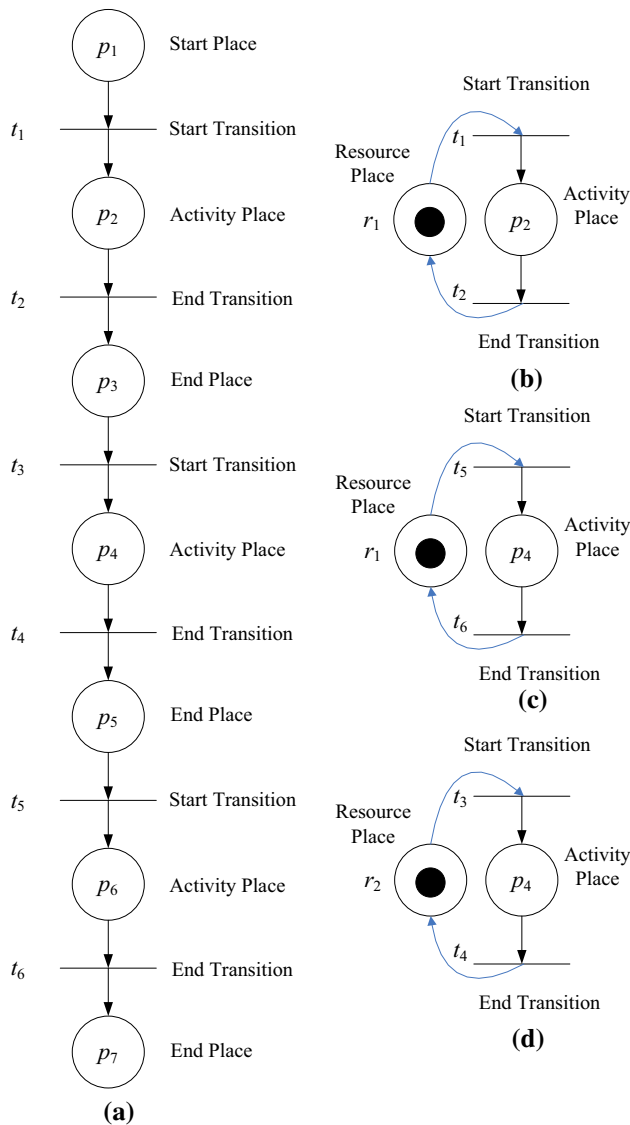
for agent discovering other related agents. To realize the proposed idea, a platform that supports the development of multi-agent systems, publishing/discovery of agent services is required. Java Agent Development Environment (JADE) is a multi-agent platform that fulfills the aforementioned requirements. Therefore, we develop a system based on JADE to realize our methodology.

Petri nets are a powerful tool to model workflows and activities in a supply chain. To model the workflow agents and resource agents, a brief introduction to Petri net can be found in the paper by Murata (1989). A timed Petri net (TPN)  $G$  is a five-tuple  $G = (P, T, F, m_0, \mu)$ , where  $P$  is a finite set of places,  $T$  is a finite set of transitions,  $F \subseteq (P \times T) \cup (T \times P)$  is the flow relation,  $m_0 : P \rightarrow Z^{|P|}$  is the initial marking of the TPN with  $Z$  as the set of nonnegative integers and  $\mu : T \rightarrow R^+$  is a mapping that specifies the firing time for each transition performed by RA. The marking of  $G$  is a vector  $m \in Z^{|P|}$  that indicates the number of tokens in each place and is state of the system. In TPN,  $\cdot t$  denotes the set of input places of transition  $t$  and  $t \cdot$  denotes the set of output places of transition  $t$ . A transition  $t$  is enabled and can be fired under a marking  $m$  if and only if  $m(P) \geq F(p, t) \forall p \in \cdot t$ . Firing a transition once removes one token from each of its input places and adds one token to each of its output places. To model the activity of a workflow agent in Petri net, we use a place to represent a state in the workflow while a transition to represent an event or operation that brings the workflow from one state to another.

**Definition 3.1** The workflow of a workflow agent  $w_n$  is an acyclic timed marked graph ATMG  $W_n = (P_n, T_n, F_n, m_{n0}, \mu_n)$ . As each transition represents a distinct operation in a task,  $T_j \cap T_k = \Phi$  for  $j \neq k$ . Individual workflow must satisfy certain timing constraints so that overall collaborative workflow can meet the timing requirements. The timing constraints for a workflow agent are determined by the timing constraints imposed on downstream workflow agent.

A workflow cannot be performed without using the required resources. In each step of the workflow, specific resource requirements must be met to start its operation. Each operation in a workflow consumes a number of different types of resources. An activity is a sequence of operations to be performed by certain type of resources. A cycle indicates that the resource activity includes resource allocation and de-allocation. Each resource has an idle state and each resource activity starts and ends with an idle state. The Petri net model for the  $k$ th activity of resource agent  $a_r$  is described by a Petri net  $A_r^k$  that starts and ends with the resource idle state place  $p_r$  as follows.

**Definition 3.2** Petri net  $A_r^k = (P_r^k, T_r^k, F_r^k, m_{r0}^k, \mu_r^k)$  denotes the activity model of the  $k$ th activity of resource agent  $a_r$ , where  $a_r \in RA$ . The initial marking  $m_r^k$  is determined based on the number of resources allocated to the  $k$ th activity. There is no common transition between  $A_r^k$  and  $A_r^{k'}$  for  $k \neq k'$ . Note that  $\mu_r^k$  only specifies the firing time for each transition in  $A_r^k$ . Figure 5a shows the model of a workflow



**Fig. 5** a An example of Petri net model for a workflow agent  $w_n$ , b Petri net models for three activities of two resource agents  $a_{r_1}$  and  $a_{r_2}$

agent. Figure 5b–d illustrates the resource activity models associated with the workflow in Fig. 5a.

To formulate the scheduling problem for a collaborative scheduling agent, we first obtain the parameters from the corresponding timed Petri net models. In our system, each order agent places only one order. To formulate the problem, we define the following notations.

Notations:

- $O$  The number of order agents, i.e.,  $O = |OA|$ .
- $N$  The number of workflow agents, i.e.,  $N = |WA|$ .
- $K_n$  The number of different resource activities involved in  $W_n$ .
- $k$  The index of the  $k$ th resource activity,  $A_r^k$ , in  $W_n$ ;  $k \in \{1, 2, \dots, K_n\}$ .

- $T$  The total number of time periods.
- $t$  A time period index;  $t \in \{1, 2, 3, \dots, T\}$ .
- $\mathfrak{R}$  The set of all resources in the system.
- $C_{rt}$  The capacity of resource  $r$  at time period  $t$ , where  $C_{rt} = m_{r0}^k(r)$ .
- $r_k$  Resource agent that performs the  $k$ th resource activity,  $A_r^k$ , in  $W_n$ .
- $\mu_r^k(t_s^k)$  The firing time for starting transition  $t_s^k$  of the  $k$ th activity,  $A_r^k$ .
- $\mu_r^k(t_e^k)$  The firing time for ending transition  $t_e^k$  of the  $k$ th activity,  $A_r^k$ .
- $\pi_{nk}$  The processing time  $\pi_{nk}$  of the  $k$ th resource activity  $A_r^k$  in  $W_n$ .
- $D_n$  The quantity of products demand for order  $o$ .
- $d_n$  The due date of order  $o$  that is placed to workflow agent  $w_n$ .
- $S_{nkt}$  The input buffer constraint of the  $k$ th resource activity of  $W_n$  at time period  $t \in \{1, 2, 3, \dots, T\}$ .
- $u_{onkt}$  The number of parts of order  $o$  loaded onto the corresponding resource  $r_k$  for processing the  $k$ th resource activity in  $W_n$  during time period  $t$ , where  $u_{onkt} \geq 0$  and  $u_{onkt} \in Z^+$  is the set of nonnegative integers.
- $z_{ot}$  The number of parts of order  $o$  in workflow  $W_n$  finished during time period  $t$ .
- $x_{onkt}$  Be the number of parts of order  $o$  at the input buffer of the  $k$ th resource activity in  $W_n$  at the beginning of period  $t$ , where  $x_{onkt} \geq 0$  and  $x_{onkt} \in Z^+$  is the set of nonnegative integers.

Note that the due date  $d_n$  for workflow agent  $w_n$  is set by its downstream workflow agents in the negotiation processes.

Given  $W_n, A_r^k, \pi_{nk}, D_n$  and  $d_n$ , where  $n \in \{1, 2, 3, \dots, N\}$  and  $k \in \{1, 2, 3, \dots, K_n\}$ , the problem for scheduling the parts in  $W_n$  requested by the orders is formulated as follows.

We now define the earliness/lateness penalty coefficient  $\theta_{ont}$  for each product of workflow  $W_n$  completed at time  $t$  as

$$\theta_{ont} = \begin{cases} 0 & \text{if } t = d_n \\ \psi(t - d_n) & \text{otherwise} \end{cases}, \text{ where } \psi(t - d_n) > 0 \forall t \neq d_n > 0$$

The scheduling problem for workflow  $W_n$  is to find an allocation of resource capacities over the scheduling horizon that minimizes the total production costs while satisfying all production constraints. Mathematically, it is formulated as

$$\begin{aligned}
 (OP_n) \quad & \min \sum_{o=1}^O \sum_{n=1}^N \sum_{t=1}^T (\theta_{ont} z_{ont}) \\
 \text{s.t.} \quad & \\
 \sum_{o=1}^O \sum_{n=1}^N \sum_{\tau=t-\pi_{nk}+1}^t u_{onk\tau} & \leq C_{rkt} \quad \forall k \in \{1, 2, \dots, K_n\}, \forall t \\
 & (3.1)
 \end{aligned}$$

$$z_{ot} = u_{onK_n(t-\pi_{nk_n})}, \quad \forall o, \forall t \quad (3.2)$$

$$x_{on11} = D_n \quad (3.3)$$

$$x_{on1(t+1)} = x_{on1t} - u_{on1t}$$

$$\begin{aligned}
 x_{onk(t+1)} &= x_{onkt} - u_{onkt} + u_{on(k-1)(t-\pi_{n(k-1)})} \\
 \forall k \in \{2, 3, \dots, K_n\} & \\
 & (3.4)
 \end{aligned}$$

$$x_{on(K_n+1)(t+1)} = x_{on(K_n+1)t} + u_{onK_n(t-\pi_{nk})} \quad (3.5)$$

$$x_{onkt} \leq S_{nkt} \quad \forall k \in \{1, 2, \dots, K_n\}, \quad \forall t \quad (3.6)$$

Note that constraints (3.1) are the capacity constraints, constraints (3.2) state the number of parts of order  $o$  in workflow  $W_n$  finished during time period  $t$  and constraints (3.3)–(3.5) are the flow balance equations. Constraints (3.6) state the buffer level of intermediate parts.

### Subgradient method for scheduling collaborative workflows

In this paper, we combine optimization theory with multi-agent system architecture to allocate resources and perform the operations of workflows. We adopt a divide-and-conquer approach to perform optimization locally by each workflow agent involved and determine whether the temporal constraint can be satisfied based on the solutions of individual workflow agents. Optimization is achieved by each workflow agent that applies the Lagrangian relaxation technique to develop a solution algorithm for workflow scheduling. To apply optimization scheme, an optimization problem is formulated based on transformation of the corresponding time Petri net model. The structure of the scheduling problem faced by each workflow agent can be represented by a minimum cost flow (MCF) problem. The Lagrangian relaxation technique provides a systematic way to determine the cost of each arc in MCF.

In problem  $OP_n$ , we observe that the coupling among production flows of different product types is caused by contention for resources. Based on this observation, we apply Lagrangian relaxation to relax resource capacity constraints (1) and form the Lagrangian function as

$$\begin{aligned}
 L(\lambda) = & \min \sum_{o=1}^O \sum_{n=1}^N \sum_{t=1}^T \theta_{ont} z_{ont} \\
 & + \sum_{k=1}^{K_n} \sum_{t=1}^T \lambda_{onkt} \left( \sum_{o=1}^O \sum_{n=1}^N \sum_{\tau=t-\pi_{nk}+1}^t u_{onk\tau} - C_{rkt} \right)
 \end{aligned}$$

s.t. constraints (3.2), (3.3), (3.4), (3.5) and (3.6), where  $\lambda_{onkt}$  is the associated Lagrange multipliers that must be nonnegative.

We define the optimization problem for type  $n$  workflow as follows:

$$\begin{aligned}
 MCF_{on}(v_{on}, \lambda_{on}) \\
 \equiv \min \left[ \sum_{t=1}^T \left\{ \theta_{ont} z_{ont} + \sum_{r=1}^{K_n} \lambda_{okt} \sum_{n=1}^N \sum_{\tau=t-\pi_{nk}+1}^t u_{onk\tau} \right\} \right]
 \end{aligned}$$

s.t. constraints (3.2), (3.3), (3.4), (3.5) and (3.6). Note that the flow balance equations described by constraints (3.2), (3.3), (3.4), (3.5) and (3.6) can be represented by a network flow model. Please refer to the work by Hsieh and Lin (2014b).

Subgradient-based algorithm to find a solution

Step 0: Initialize parameters

$$l \leftarrow 1, \lambda^l \leftarrow \lambda_0, L(\lambda^{l-1}) \leftarrow 0$$

Step 1: Compute  $u^l$  by solving  $L(\lambda^l)$

Step 2: Calculate  $\frac{L(\lambda^l) - L(\lambda^{l-1})}{L(\lambda^l)}$

$$\text{If } \left| \frac{L(\lambda^l) - L(\lambda^{l-1})}{L(\lambda^l)} \right| > \epsilon$$

$$\alpha^l = \frac{\beta[L(\lambda^*) - L(\lambda^l)]}{\sum_{k,t} (g_{onkt}^l)^2}$$

$$g_{onkt}^l = \sum_{o=1}^O \sum_{\tau=t-\pi_{nk}+1}^t u_{onk\tau} - C_{or_kt} \text{ for each } k \in \{1, \dots, K_n\}, t \in \{1, \dots, T\}$$

$$\lambda_{onkt}^{l+1} = \begin{cases} \lambda_{onkt}^l + \alpha^l g_{onkt}^l & \text{if } \lambda_{onkt}^l + \alpha^l g_{onkt}^l \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$l \leftarrow l + 1$$

Go to Step 1

Else

$$u^* \leftarrow u^l$$

If (3.1) is satisfied

$$\bar{u} \leftarrow u^*$$

Else

Iteratively apply heuristic rules to adjust  $u^*$  to obtain  $\bar{u}$

End If

End If

Our approach to finding a solution of  $\max_{\lambda \geq 0} L(\lambda)$  is based on an iterative scheme for adjusting Lagrangian



multipliers according to the solutions of MCF subproblems.

Let  $l$  be the iteration index. Let  $v^l$  denote the optimal solution to MCF subproblems for given Lagrange multipliers  $\lambda^l$  at iteration  $l$ . We define the subgradients of  $L(\lambda)$  with respect to Lagrangian multipliers  $\lambda^l$  as follows:

$$g_{onkt}^l = \sum_{o=1}^O \sum_{n=1}^N \sum_{\tau=t-\pi_{nk}+1}^t u_{onk\tau} - C_{rkt}, \quad \forall k = 1, \dots, K_n, \\ \forall t = 1, \dots, T$$

The subgradient method proposed by Polyak (1969) is adopted to update  $\lambda$  as follows:

$$\lambda_{onkt}^{l+1} = \begin{cases} \lambda_{onkt}^l + \alpha^l g_{onkt}^l & \text{if } \lambda_{onkt}^l + \alpha^l g_{onkt}^l \geq 0, \\ 0 & \text{otherwise} \end{cases}$$

where  $\alpha^l = \frac{\beta[\bar{L}(\lambda^*) - L(\lambda^l)]}{\sum_{k,t} (g_{onkt}^l)^2}$  and  $\bar{L}$  is an estimate of the optimal dual cost and  $0 < \beta < 2$ .

A heuristic algorithm to adjust a dual solution

Step 1: Check Constraint (3.1)

For each  $k \in \{1, \dots, K_n\}, t \in \{1, \dots, T\}$

$$\text{If } g_{onkt} = \sum_{o=1}^O \sum_{\tau=t-\pi_{nk}+1}^t u_{onk\tau} - C_{or,t} > 0$$

$$o' \leftarrow o$$

$$n' \leftarrow n$$

$$k' \leftarrow k$$

$$t' \leftarrow t$$

Go to Step 2

End If

End For

Step 2: Check the excessive flow from the network

$$\text{If } g_{o'n'k't'} < D_n$$

$$\delta \leftarrow g_{o'n'k't'}$$

Else

$$\delta \leftarrow D_{n'}$$

End If

remove  $\delta$  from the network  $Net_{o'n'}$

Go to Step 3

Step 3: Reroute the excessive flow to the network

Update the capacity of arc  $o'n'k't'$  of  $Net_{o'n'}$

Reroute the flow,  $\delta$ , by following the minimal cost route in the network  $Net_{o'n'}$

Go to Step 4

the optimal solution to the underlying problem. Thus, the solution generated may not satisfy the complementary slackness conditions. In case the solution is not feasible, we must develop a heuristic algorithm to find a feasible solution. In our system, we implement a simple heuristic algorithm that removes the excessive flows from the arcs with capacity violation by setting the arc capacity to zero and reroutes the excessive flows to other part of the network based on MCF algorithm.

### Agent interaction model for collaborative scheduling

Interactions among resource agents, workflow agents, order agents and collaborative scheduling agents are through a mechanism that extends the well-known contract net pro-

Iterative application of the subgradient algorithm will converge to an optimal dual solution  $(u^*, \lambda^*)$ . It should be emphasized that Lagrangian relaxation does not guarantee

protocol originally proposed by Smith (1980) by taking into account the dependency between workflows in supply chains. In contract net protocol, there are two roles an agent

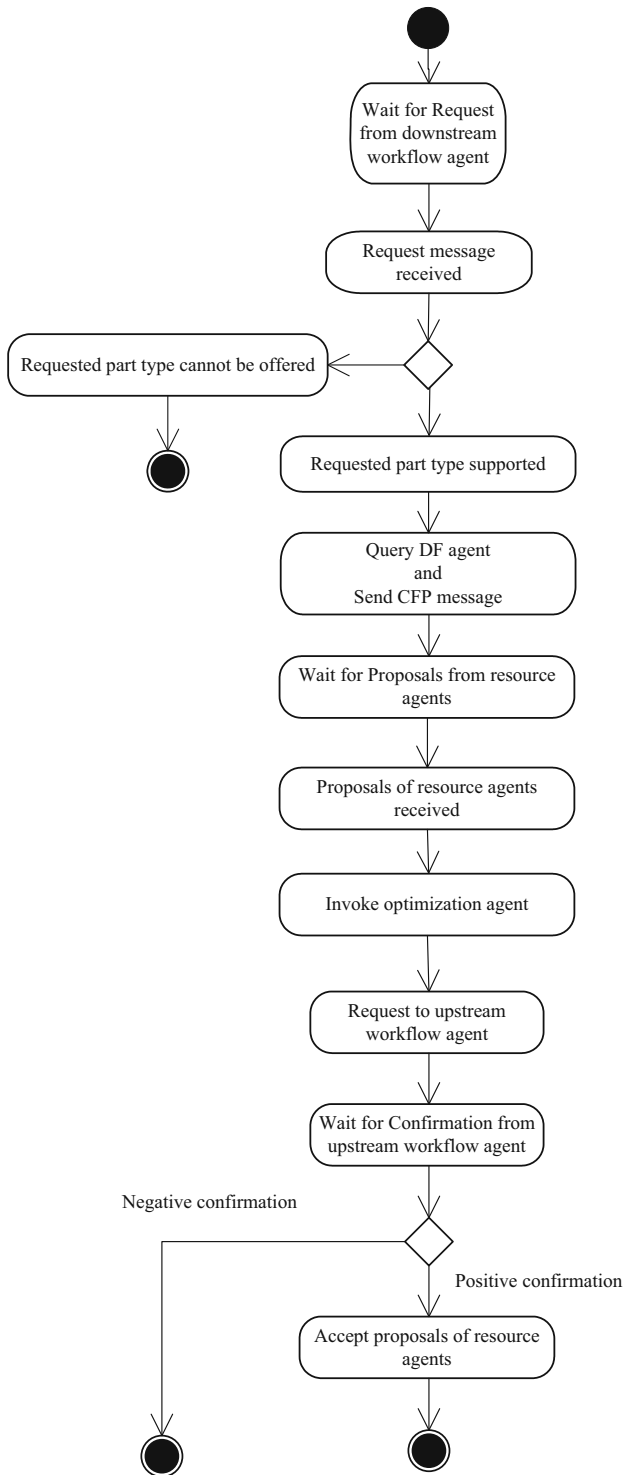


Fig. 6 Flowchart of a workflow agent

can play: manager or bidder. Four stages are involved to establish a contract between a manager and one or more bidders: (1) call for proposals (CFP): The manager announces a task to all potential bidders. The announcement contains the description of the task. (2) Submission of proposals: On receiving the tender announcement, bidders

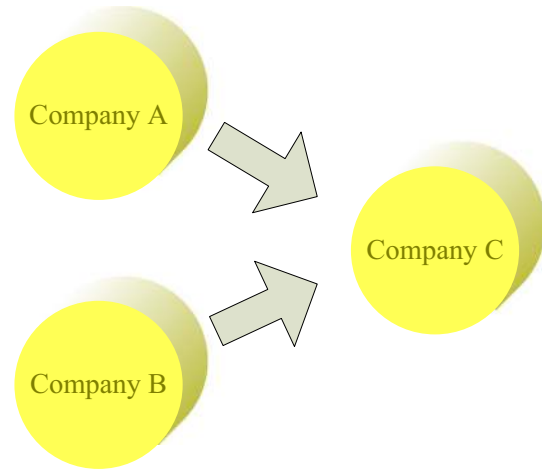


Fig. 7 Three companies, A, B and C, in a supply chain

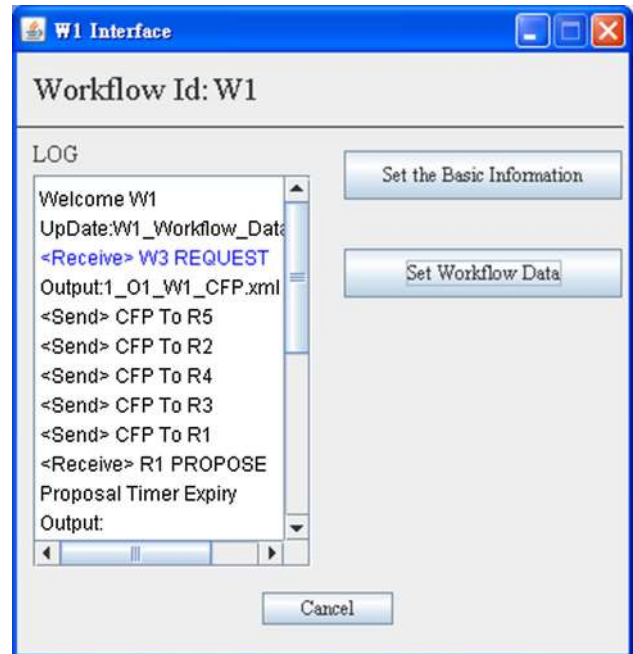


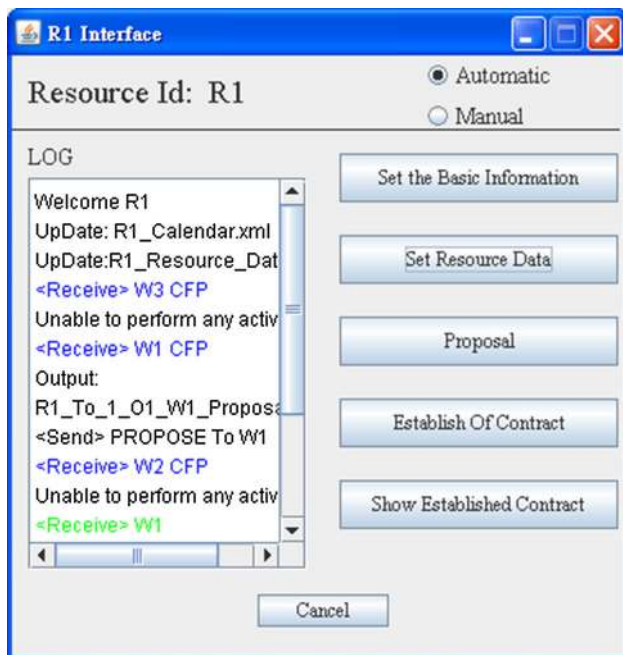
Fig. 8 GUI for a workflow agent

capable of performing the task draw up proposals and submit them to the manager. (3) Awarding of contract: On receiving and evaluating the submitted proposals, the manager awards the contract to the best bidder. (4) Establishment of contract: The awarded bidder may either commit itself to carry out the task or refuse to accept the contract by sending messages to the manager. For the latter case, the manager will reevaluate the bids and award the contract(s) to another bidder(s).

Each workflow may rely on some type of products from other workflows and may produce some other type of products. Each workflow agent has an internal process

**Table 1** Firing time of transitions

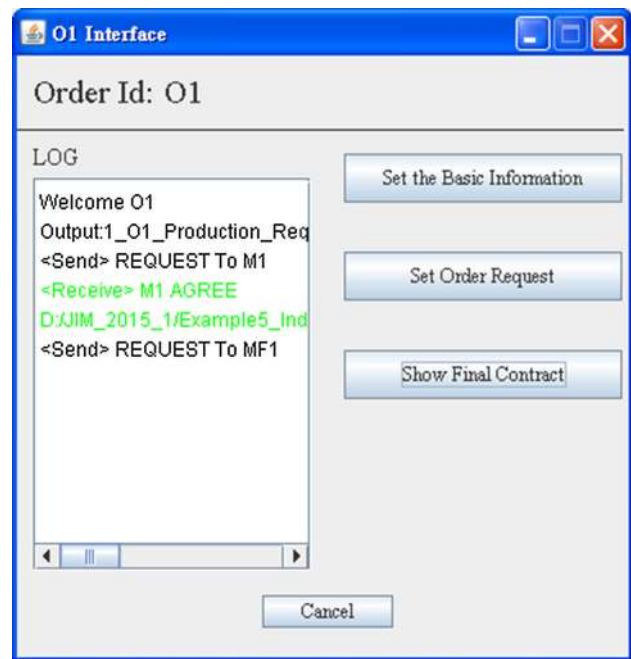
Transition	Firing time
$t_1$	$\mu_{r_1}^1(t_1) = 2$
$t_2$	$\mu_{r_1}^1(t_2) = 3$
$t_3$	$\mu_{r_2}^1(t_3) = 2$
$t_4$	$\mu_{r_2}^1(t_4) = 1$
$t_5$	$\mu_{r_3}^1(t_5) = 2$
$t_6$	$\mu_{r_3}^1(t_6) = 2$
$t_7$	$\mu_{r_4}^1(t_7) = 1$
$t_8$	$\mu_{r_4}^1(t_8) = 1$
$t_9$	$\mu_{r_5}^1(t_9) = 1$
$t_{10}$	$\mu_{r_5}^1(t_{10}) = 2$



**Fig. 9** GUI for a resource agent

flow, the required input types and output types. This leads to dependency between workflows. Due to the dependency between workflows, the original contract net protocol must be extended to be applied to solve the collaborative scheduling problem in supply chains.

Figure 6 shows the flowchart of a workflow agent. Initially, a workflow agent waits for a request from an order agent or another workflow agent at its downstream. It will discover the potential resource agents only if the requested part type is supported. Otherwise, it will not respond to the request. If the part type is supported, the workflow agent will query the Directory Facilitator (DF) agent, which provides directory services in JADE platform, to find the potential resource agents and apply the contract net protocol to determine the best proposals. The workflow agent



**Fig. 10** GUI for an order agent

then invokes the collaborative scheduling agent to schedule the workflow. Based on the schedule, the workflow agent then requests its upstream workflow agents to schedule their workflows and then waits for the confirmation message from them. If it receives a negative confirmation indicating that there does not exist a feasible schedule, the negotiation process is aborted. Otherwise, it will accept the proposals of resource agents.

We have implemented a software system based on the methodology proposed in this paper. Each agent has a graphical user interface (GUI) and a software module to interact with other agents in the system. The requirements of an order agent are specified by a GUI and are represented and stored in XML format. In addition, a workflow agent consists of a proper GUI to specify its properties and represent its workflow model. The workflow model is described by a timed Petri net model. The properties of a workflow agent are described by an XML file. The activities of the workflow to be performed by resource agents are also represented by timed Petri net models. The capabilities of a resource agent are defined by a GUI and are also described by an XML file. The order requirements are the inputs of order agents and are represented in XML in our system. The inputs of workflow agents are the workflow Petri net models represented in PNML, and the inputs of resource agents are the activity Petri net models represented in PNML.

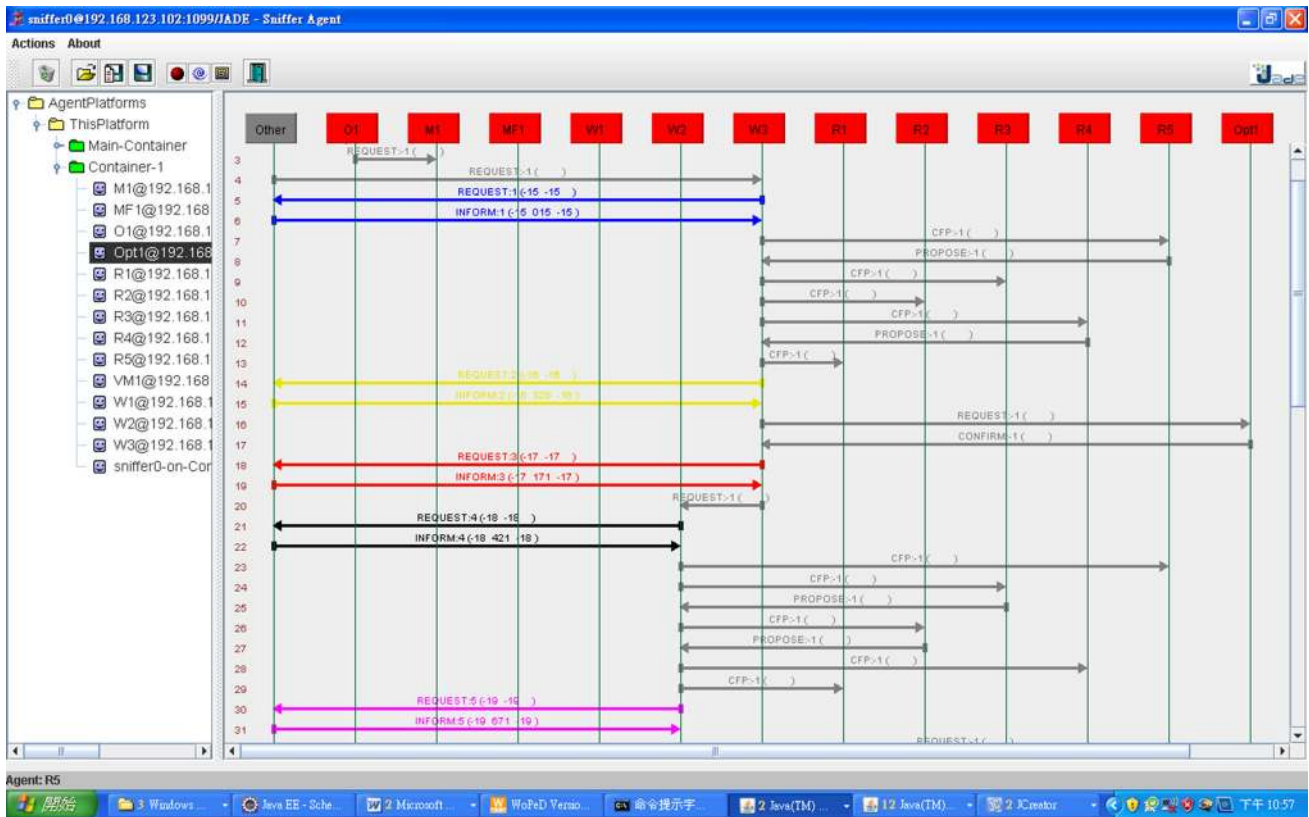


Fig. 11 Interaction between agents

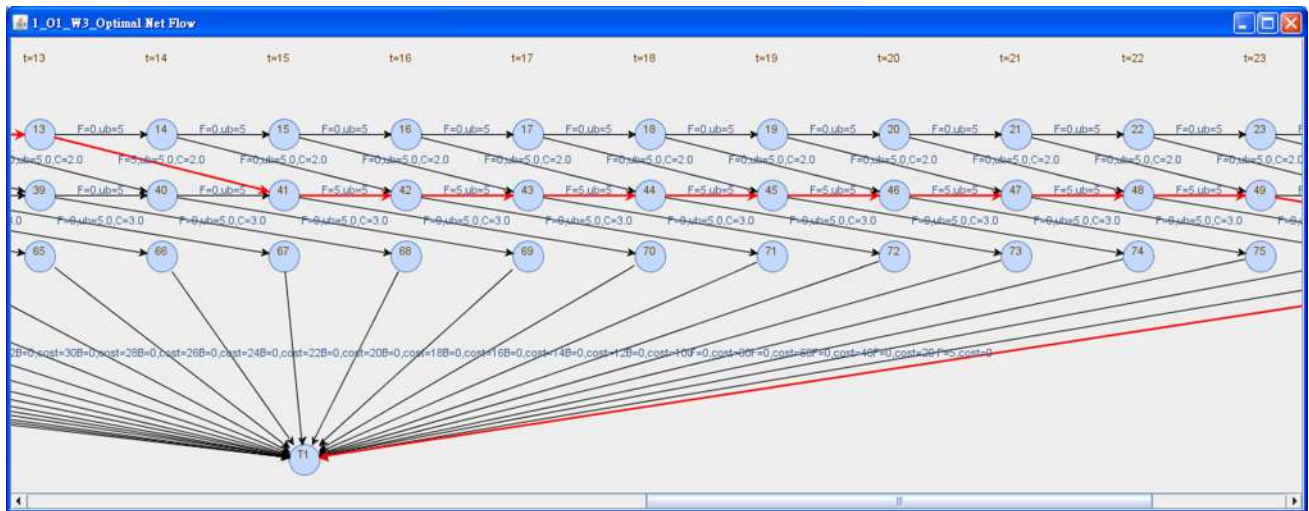


Fig. 12 Network model of workflow agent  $W_3$

**Numerical results**

Based on the algorithms proposed in the previous section, we verify our method by examples. We first use a small example to illustrate the functions of the software developed in this paper. We then present the results for several examples by applying our software.

*Example 1* Consider three companies, A, B and C, which may cooperate to form a supply chain, as shown in Fig. 7. Company A produces type 1 parts, whereas Company B produces type 2 parts. Company C depends on type 1 parts from Company A and type 2 parts from Company B to produce the products (type 3 parts). Suppose Company C receives an order. The requirements of the order are to

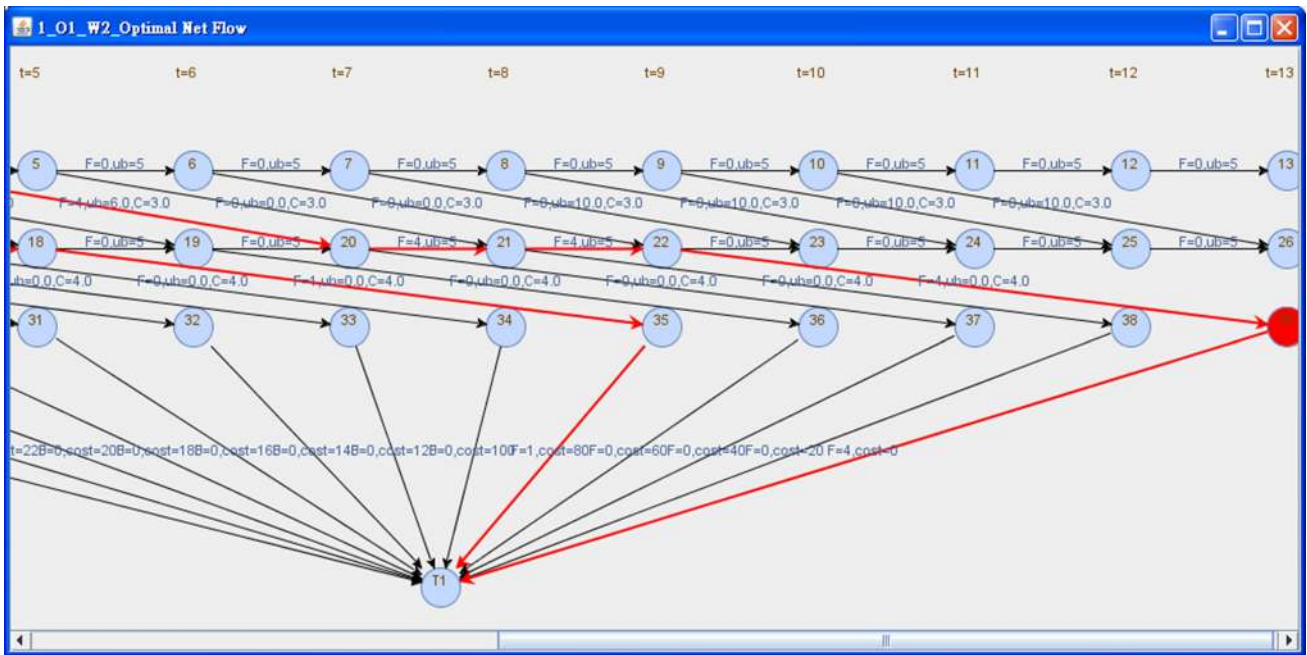


Fig. 13 Network model of workflow agent  $W_2$

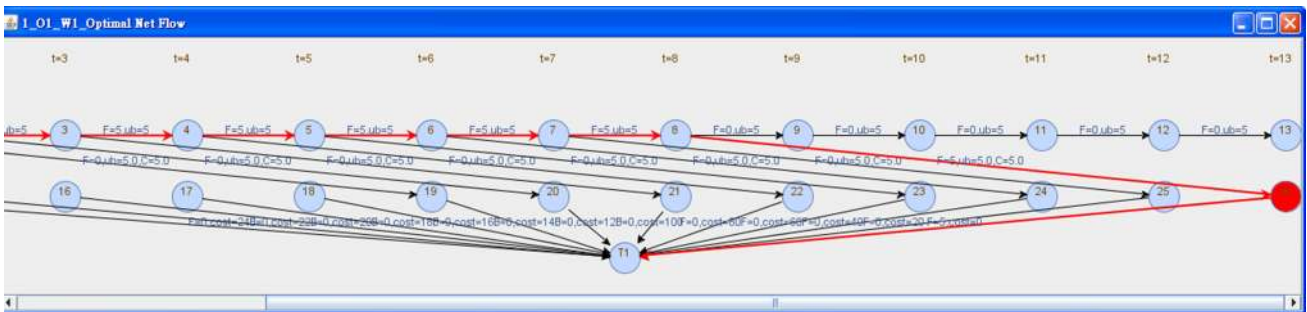


Fig. 14 Network model of workflow agent  $W_1$



Fig. 15 Calendar of resource agent  $R_1$

Fig. 16 Calendar of resource agent  $R_2$

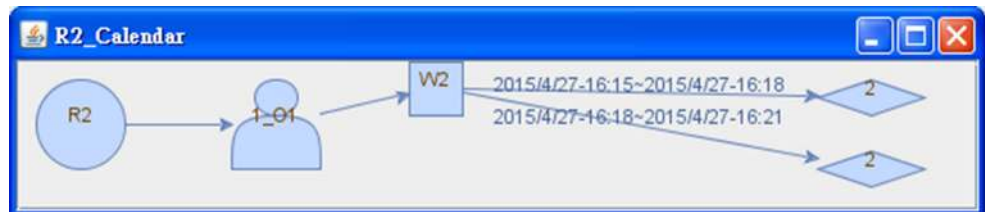


Fig. 17 Calendar of resource agent R3

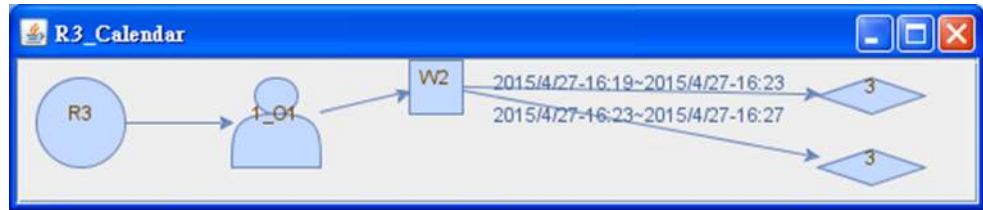


Fig. 18 Calendar of resource agent R4

Fig. 19 Calendar of resource agent R5

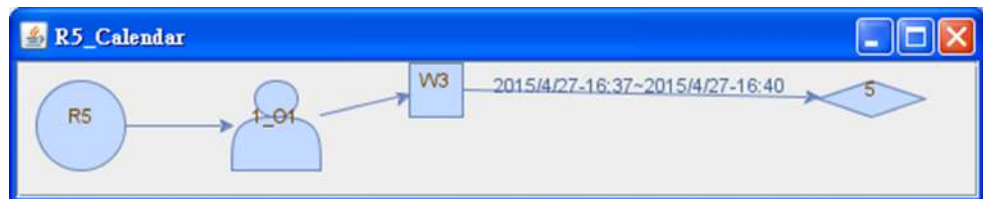
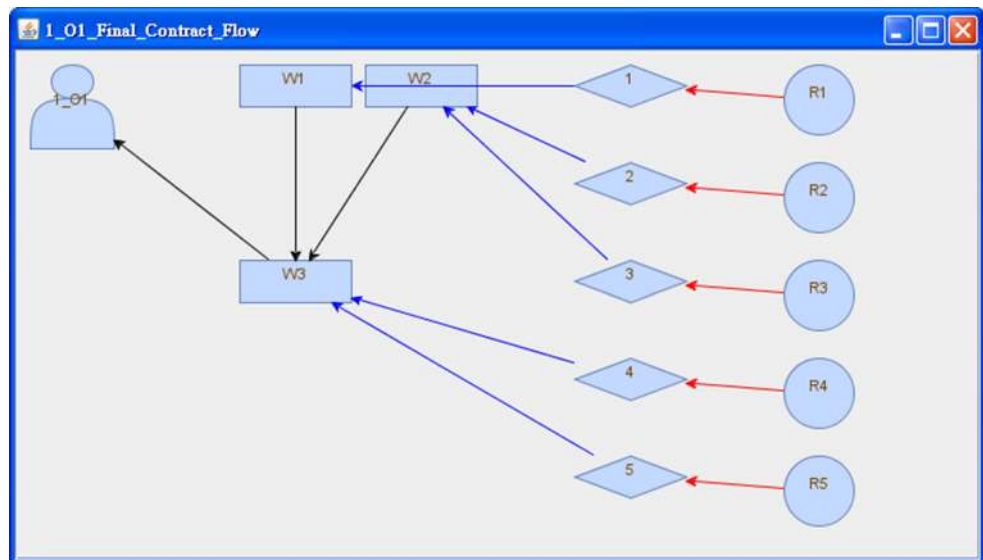


Fig. 20 Contracts established between agents for Example 1



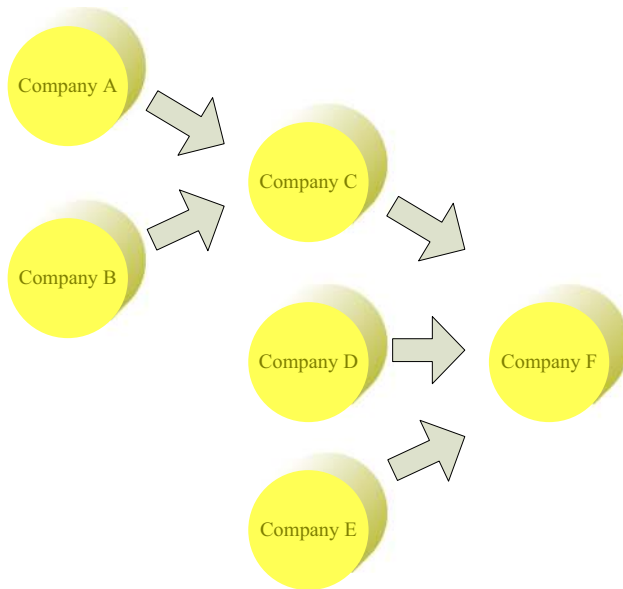
produce five units of type 3 parts by the time PM 16:40, April 26, 2015 ( $d_1$ ). By applying our software, Company A, Company B and Company C must first define their workflows and resources. The GUI for defining the properties of a workflow agent is shown in Fig. 8. “Appendix A.1” shows the PNML models for workflow agents  $W_1$ ,  $W_2$  and

$W_3$ . “Appendix A.2” shows the PNML models for activities  $A_{r1}^k$ ,  $A_{r2}^k$  and  $A_{r3}^k$  of resource agents. The firing time of transitions is listed in Table 1.

Figure 9 shows the GUI for a resource agent. Company C first defines an order agent. Figure 10 illustrates the graphical user interface (GUI) for an order agent. An order

**Table 2** Schedules for resource agents

Resource	Start time	End time	Quantity	Activity
R1	2015/4/27–16:22	2015/4/27–16:27	5	1
R2	2015/4/27–16:15	2015/4/27–16:18	1	2
R2	2015/4/27–16:18	2015/4/27–16:21	4	2
R3	2015/4/27–16:19	2015/4/27–16:23	1	3
R3	2015/4/27–16:23	2015/4/27–16:27	4	3
R4	2015/4/27–16:27	2015/4/27–16:29	5	4
R5	2015/4/27–16:37	2015/4/27–16:40	5	5



**Fig. 21** Six companies in a supply chain

**Table 3** Firing time of transitions

Transition	Firing time
$t_1$	$\mu_{r_1}^1(t_1) = 115$
$t_2$	$\mu_{r_1}^1(t_2) = 5$
$t_3$	$\mu_{r_2}^1(t_3) = 80$
$t_4$	$\mu_{r_2}^1(t_4) = 10$
$t_5$	$\mu_{r_3}^1(t_5) = 70$
$t_6$	$\mu_{r_3}^1(t_6) = 2$
$t_7$	$\mu_{r_4}^1(t_7) = 45$
$t_8$	$\mu_{r_4}^1(t_8) = 15$
$t_9$	$\mu_{r_5}^1(t_9) = 85$
$t_{10}$	$\mu_{r_5}^1(t_{10}) = 5$
$t_{11}$	$\mu_{r_6}^1(t_{11}) = 30$
$t_{12}$	$\mu_{r_6}^1(t_{12}) = 30$
$t_{13}$	$\mu_{r_7}^1(t_{13}) = 110$
$t_{14}$	$\mu_{r_7}^1(t_{14}) = 10$

is specified by a due date, a product type, quantity and the penalty cost (earliness penalty cost and lateness penalty cost) to achieve just-in-time production. For this example, the earliness/lateness penalty coefficient  $\theta_{11t}$  is as follows:

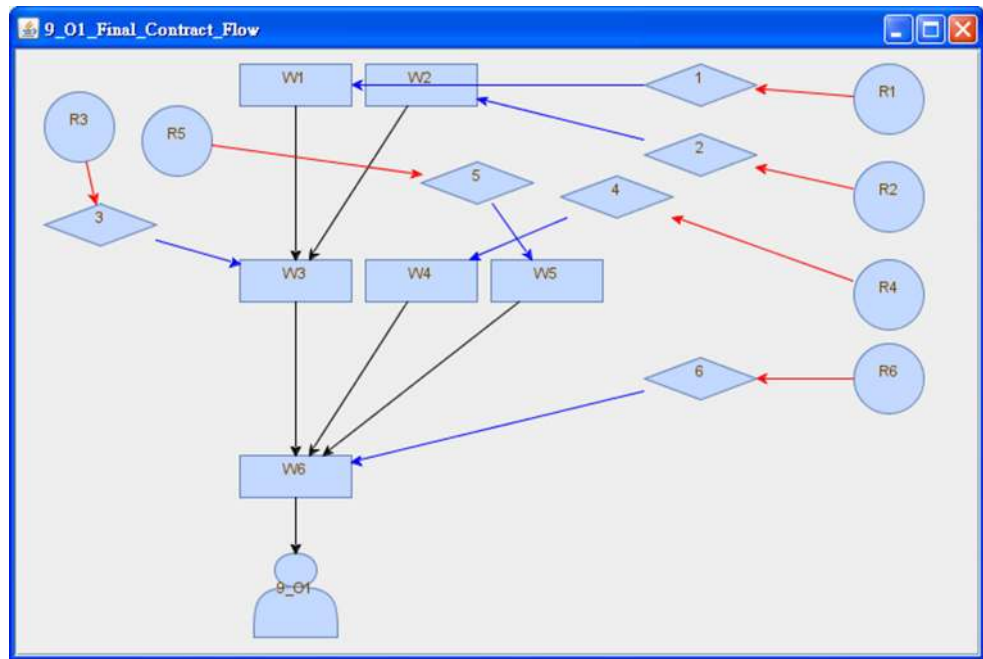
$$\theta_{11t} = \begin{cases} 0 & \text{if } t = d_1 \\ 40 & \text{if } t > d_1 \\ 20 & \text{otherwise} \end{cases}$$

For this example, three workflow agents ( $W_1, W_2, W_3$ ), five resource agents ( $R_1, R_2, R_3, R_4, R_5$ ) and one order agent ( $O_1$ ) need to be defined and created.

Handling an order relies on the collaboration of a number of agents. Our software solves the workflow scheduling problem based on interaction of different types of agents in the system. Figure 11 shows interactions between agents. Order agent  $O_1$  issues a request to the potential workflow agents. As workflow agent  $W_3$  can produce type 3 parts, it will send a CFP to the potential resource agents. As resource agent  $R_4$  and resource agent  $R_5$  can perform the operations, they submit proposals to workflow agent  $W_3$ . Note that as resource agent  $R_1$ , resource agent  $R_2$  and resource agent  $R_3$  cannot perform the operations, they will not submit any proposal to workflow agent  $W_3$ . Once the proposal has been received by workflow agent  $W_3$ , it will request a collaborative scheduling agent  $Opt1$  to optimize the schedule. Figure 12 shows the network model constructed by the collaborative scheduling agent for  $W_3$  in the process of optimization. The collaborative scheduling agent will send a message to confirm feasibility of the solution. On receiving the message, workflow agent  $W_3$  will request its potential upstream workflow agents,  $W_1$  and  $W_2$ , to optimize their schedules. Workflow agents  $W_1$  and  $W_2$  will send CFP to the potential resource agents, wait for the proposals from the resource agents  $R_1, R_2$  and  $R_3$  and invoke collaborative scheduling agents to optimize the schedules based on the proposals. Figures 13 and 14 show the network models constructed by the collaborative scheduling agent for  $W_2$  and  $W_1$  in the process of optimization. For this example, allocation of resource agents to perform the operations in the associated workflow for the order agent is depicted in Figs. 15, 16, 17, 18 and 19, respectively. For this example, the order due date can be met. The contracts established between agents are shown in Fig. 20. The schedules for each resource agent are shown in Table 2.

**Example 2** Consider six companies, A, B, C, D, E and F, which may cooperate to form a supply chain, as shown in Fig. 21. Company A produces type 1 parts, whereas Company B produces type 2 parts. Company D produces type 4 parts and Company E produces type 5 parts. Company C depends on type 1 parts from Company A and type 2 parts from Company B to produce the products (type 3

**Fig. 22** Contracts established between agents for Example 2



parts). Company F depends on type 3 parts from Company C, type 4 parts from Company D and type 5 parts from Company E to produce the products (type 6 parts).

Suppose Company F receives an order. The requirements of the order are to produce 150 units of type 6 parts by the time AM 12:00, April 23, 2017 ( $d_1$ ). By applying our software, Company A, Company B and Company C must first define their workflows and resources. “Appendix B.1” shows the PNML models for workflow agents  $W_1, W_2, W_3, W_4, W_5$  and  $W_6$ . “Appendix B.2” shows the PNML models for activities  $A_{r1}^k, A_{r2}^k, A_{r3}^k, A_{r4}^k, A_{r5}^k, A_{r6}^k$  and  $A_{r7}^k$ , of resource agents. The firing time of transitions is listed in Table 3.

For this example, the earliness/lateness penalty coefficient  $\theta_{11t}$  is as follows:

$$\theta_{11t} = \begin{cases} 0 & \text{if } t = d_1 \\ 40 & \text{if } t > d_1 \\ 20 & \text{otherwise} \end{cases}$$

For this example, six workflow agents ( $W_1, W_2, W_3, W_4, W_5$  and  $W_6$ ), seven resource agents ( $R_1, R_2, R_3, R_4, R_5, R_6, R_7$ ) and one order agent ( $O_1$ ) need to be defined and created.

The output of our collaborative workflow management system includes the contracts established between agents for handling orders, the schedules for executing each workflow and the schedules for performing the operations of workflows by resource agents. Figure 22 shows the contracts established between agents for handling Order 1. It indicates that resource agents  $R_1$  through  $R_6$  take part in the operations of workflows  $W_1$  through  $W_6$  required for

Order 1. Our system also shows the assignment of resources to process workflows and orders. The schedules for each resource agent are shown in Table 4.

### Scalability analysis and verification by examples

Response time is an important performance index in supply chain management. To be applicable in supply chains, response time should be acceptable as the number of partners in the network grows. In the remainder of this paper, we will present the analysis and numerical results to show that our method is scalable in terms of response time as the size of supply chain network grows. Response time is the total amount of time it takes to respond to a request for service. The response time for solving the problem defined in this paper consists of two parts: computation time and transmission time. With the widespread adoption of broadband network, transmission time is much less than computation time for solving the problem defined in this paper. To evaluate the scalability of our proposed methodology in terms of response time, we first analyze and compare the computational complexity of our proposed method and that of an industrial centralized optimizer. We then conduct experiments based on the developed software system.

We focus on assembly supply chain networks. In this type of networks, the response time can be measured based on the concept of “depth” of supply chains. In a typical assembly supply chain network, let’s call a node without

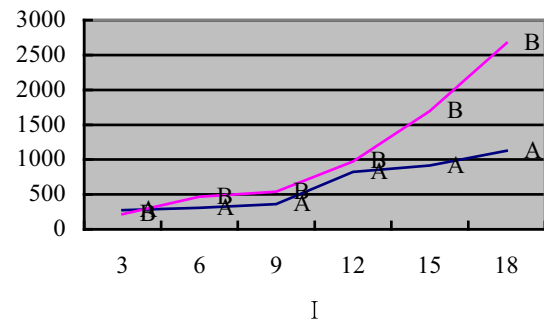


**Table 4** Schedules for resource agents

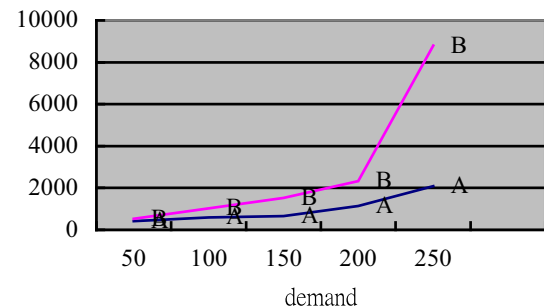
Resource	Start time	End time	Quantity	Activity
R1	2017/4/22-14:00	2017/4/22-16:00	25	1
R1	2017/4/22-16:00	2017/4/22-18:00	25	1
R1	2017/4/22-18:00	2017/4/22-20:00	25	1
R1	2017/4/22-20:00	2017/4/22-22:00	25	1
R1	2017/4/22-22:00	2017/4/23-00:00	25	1
R1	2017/4/23-00:00	2017/4/23-20:00	25	1
R2	2017/4/22-14:00	2017/4/22-15:30	10	2
R2	2017/4/22-15:30	2017/4/22-17:00	20	2
R2	2017/4/22-17:00	2017/4/22-18:30	20	2
R2	2017/4/22-18:30	2017/4/22-20:00	20	2
R2	2017/4/22-20:00	2017/4/22-21:30	20	2
R2	2017/4/22-21:30	2017/4/22-23:00	20	2
R2	2017/4/22-23:00	2017/4/23-00:30	20	2
R2	2017/4/23-00:30	2017/4/23-02:00	20	2
R3	2017/4/23-02:00	2017/4/23-03:12	30	3
R3	2017/4/23-03:12	2017/4/23-04:24	30	3
R3	2017/4/23-04:24	2017/4/23-05:36	30	3
R3	2017/4/23-05:36	2017/4/23-06:48	30	3
R3	2017/4/23-06:48	2017/4/23-08:00	30	3
R4	2017/4/23-03:00	2017/4/23-04:00	30	4
R4	2017/4/23-04:00	2017/4/23-05:00	30	4
R4	2017/4/23-05:00	2017/4/23-06:00	30	4
R4	2017/4/23-06:00	2017/4/23-07:00	30	4
R4	2017/4/23-07:00	2017/4/23-08:00	30	4
R5	2017/4/22-20:00	2017/4/22-21:30	10	5
R5	2017/4/22-21:30	2017/4/22-23:00	20	5
R5	2017/4/22-23:00	2017/4/23-00:30	20	5

**Table 4** (continued)

Resource	Start time	End time	Quantity	Activity
R5	2017/4/23-00:30	2017/4/23-02:00	20	5
R5	2017/4/23-02:00	2017/4/23-03:30	20	5
R5	2017/4/23-03:30	2017/4/23-05:00	20	5
R5	2017/4/23-05:00	2017/4/23-06:30	20	5
R5	2017/4/23-06:30	2017/4/23-08:00	20	5
R6	2017/4/23-08:00	2017/4/23-09:00	30	6
R6	2017/4/23-08:00	2017/4/23-09:00	30	6
R6	2017/4/23-08:00	2017/4/23-09:00	30	6
R6	2017/4/23-09:00	2017/4/23-10:00	40	6
R6	2017/4/23-10:00	2017/4/23-11:00	40	6
R6	2017/4/23-11:00	2017/4/23-12:00	40	6



**Fig. 23** Response time with respect to the depth of supply chain (in second). A: MAS architecture, B: centralized architecture



**Fig. 24** Response time with respect to demand (in second). A: MAS architecture, B: centralized architecture

any upstream node a leaf node and a node without any downstream node a final node. Note that there is one and only one final node in an assembly supply chain network. For each directed path that starts with a leaf node and ends with a final node, the number of nodes in the directed path is called the depth of the path. The response time of a supply chain network is the longest response time of all directed paths that start with a leaf node and end with the final node.

We compare the response time of supply chains based on the distributed MAS architecture used in this paper and a centralized architecture as follows. To compute  $L(\lambda)$  for a workflow agent  $w_n$  with given  $\lambda$ , it is necessary to solve the minimum cost flow problem. The computational complexity to solve a minimum cost flow problem with  $n$  nodes and flow of  $f$  is  $O(n^2f)$ . As the number of nodes in the network associated with  $L(\lambda)$  is proportional to  $K_nT$  and the flow is  $D_n$ , the computational complexity is  $O(K_n^2T^2D_n)$ . Note that the Lagrange multipliers are updated as follows:

$$g_{onkt}^l = \sum_{o=1}^O \sum_{\tau=t-\pi_{nr}+1}^t u_{onk\tau}^l - C_{or_k t} \quad \text{for each}$$

$$k \in \{1, \dots, K_n\}, \quad t \in \{1, \dots, T\}$$

$$\lambda_{onkt}^{l+1} = \begin{cases} \lambda_{onkt}^l + \alpha^l g_{onkt}^l & \text{if } \lambda_{onkt}^l + \alpha^l g_{onkt}^l \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

As the number of Lagrange multipliers is proportional to  $K_n$  and  $T$ , the computation time involved in updating  $\lambda$  will increase approximately with  $O(K_nT)$ . Therefore, the overall computational complexity is  $O(K_n^2T^2D_n)$ . This indicates the computational complexity of our algorithm is polynomial with respect to problem size. Suppose the depth is  $V$ . In a distributed MAS architecture, a supply chain with depth  $V$  has at most  $V$  node in each directed path from a leaf node to the final node. Therefore, the overall response time will be  $O(\sum_n K_n^2T^2D_n)$ . Let  $K = \max_n K_n$ . Then, the overall response time will be bounded by  $O(VK^2T^2D_n)$ .

Let's analyze the overall response time for a centralized computing architecture as follows. If we solve the scheduling problem for  $V$  echelon supply chain based on a centralized computing architecture, the Petri net models of all  $V$  workflow agents will be merged first. As the number of nodes in the network associated with  $L(\lambda)$  is proportional to  $(\sum_n K_n)T$  and the flow is  $D_n$ , the computational complexity to compute  $L(\lambda)$  is bounded by  $O(V^2K^2T^2D_n)$ . As the number of Lagrange multipliers is proportional to  $(\sum_n K_n)T$ , the computation time involved in updating  $\lambda$  will increase approximately with  $O((\sum_n K_n)T)$  and is bounded by  $O(VKT)$ . Therefore, the overall response time for a centralized architecture will be bounded by  $O(V^2K^2T^2D_n)$ .

To verify the analysis above, we conduct experiments by increasing the depth of a supply chain network and comparing the response time. All the experiments are conducted for multiple echelon supply chains. Figure 23 shows and compares the response time obtained based on MAS and a centralized problem solver, CPLEX (CPLEX integer programming solver 2015), as the depth of supply chains grows. As expected, the response time of our agent-based approach is significantly less than that of the centralized CPLEX problem solver as the depth of supply chains grows. Figure 24 shows and compares the response time with respect to demand in supply chains based on MAS and the CPLEX centralized problem solver. It also indicates that MAS architecture is much more efficient than the centralized architecture.

## Conclusions

Management of collaborative workflows in supply chains is an important issue. In supply chains, the workflows of a company depend on those of its upstream partners and have influence on those of its downstream partners. Such dependency complicates the workflow management problem in supply chains. The workflow scheduling problem in supply chains is a dynamic and challenging one. Scheduling workflows in supply chains relies on the development of a methodology that is flexible, reconfigurable and scalable. In this paper, we propose a reconfigurable, flexible and scalable architecture for scheduling workflows in supply chains based on MAS, workflow specification language and optimization theories. Our proposed methodology achieves reconfigurability and flexibility by using Petri net as the workflow specification language, specifying all the messages of CNP in XML and adopting a FIPA compliant multi-agent platform that supports ACL, contract net protocol (CNP) and publication/discovery infrastructure. Our approach attains scalability by developing algorithms based on distributed computing architecture to solve collaborative workflow scheduling problem. To take into account the dependency among the different workflow scheduling subproblems, a multi-level contract net protocol is applied in this paper to facilitate negotiation of different companies. A divide-and-conquer approach is adopted to take advantage of the distributed computation provided by MAS to optimize the workflow schedules. The original workflow scheduling problem is decomposed into a number of agents' subproblems that can be solved efficiently based on the collaboration of agents. A prototype system has been implemented based on our proposed methodology. As Petri net is adopted as a process specification language in our scheduling system, our approach formulates the scheduling problem based on the Petri net models and the order requirements dynamically. The cost and

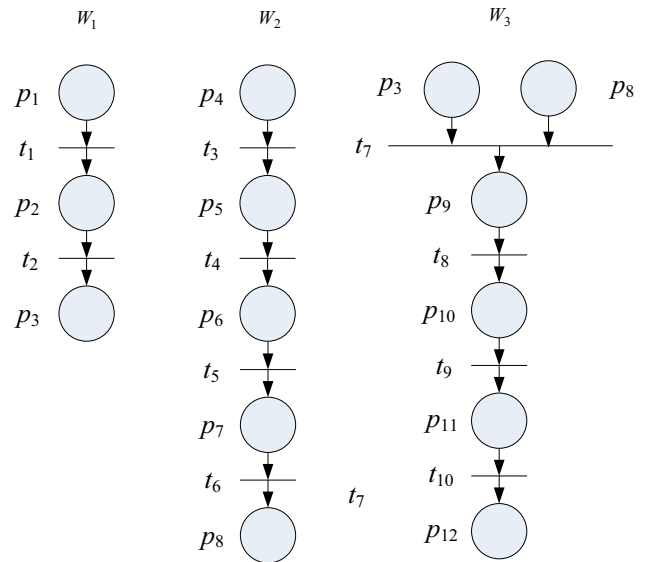


time involved in the development of scheduling software can be significantly reduced. In addition to the advantage of reduction in development cost and time, we also illustrate the effectiveness and analyze scalability of our approach by examples. To study scalability of our approach, we analyze response time with respect to the depth and the demand of supply chains. Both our analysis and numerical results indicate that our approach is much more efficient than an industrial centralized problem solver as the depth and the demand of supply chains grow. Our approach relies on the service discovery function which is provided by many multi-agent platforms. It relies on construction of workflow models and activity models for individual partners represented by agents in the supply chains. Currently, the class of workflow models used in the proposed approach is acyclic timed marked graph. Extension of workflow models to more general classes of Petri nets is one of our current research directions. As the ability to cope with external and internal disruptions and disturbances gains more and more importance, another research direction relevant to this paper is to study mechanisms to deal with uncertainties in supply chains.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

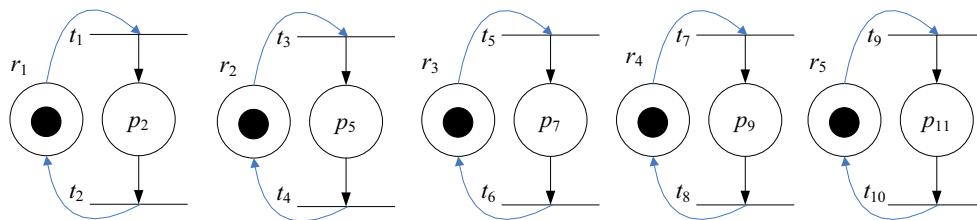
### Appendix A.1

PNML models for  $W_1, W_2$  and  $W_3$ .



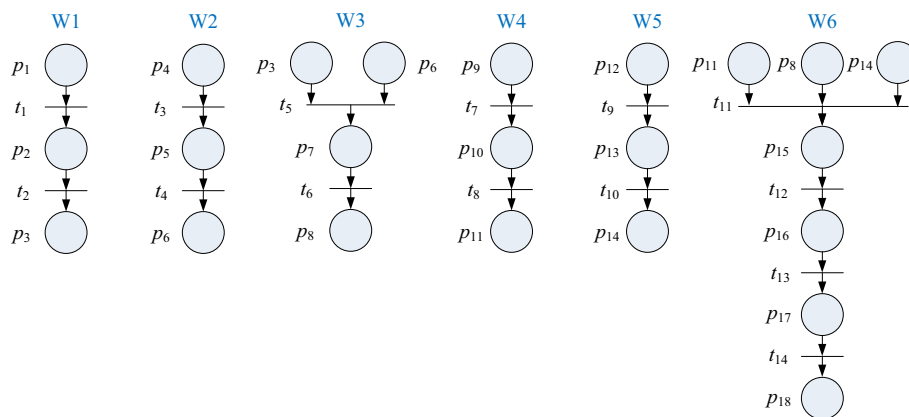
### Appendix A.2

PNML models for  $A_{r1}^k, A_{r2}^k$  and  $A_{r3}^k$ .



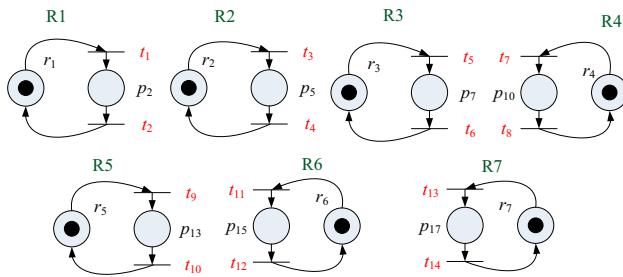
### Appendix B.1

Petri net models for  $W_1, W_2, W_3, W_4, W_5$  and  $W_6$ .



## Appendix B.2

PNML models for  $A_{r1}^k$ ,  $A_{r2}^k$ ,  $A_{r3}^k$ ,  $A_{r4}^k$ ,  $A_{r5}^k$ ,  $A_{r6}^k$  and  $A_{r7}^k$ .



## References

- Afsarmanesh H, Analide C (2009) Virtual enterprises: methods and approaches for coalition formation. *Int J Prod Res* 47(17):4655–4659
- Billington J, Christensen Søren, van Hee Kees, Kindler Ekkart, Kummer Olaf, Petrucci Laure, Post Reinier, Stehno Christian, Weber Michael (2003) The Petri net markup language: concepts, technology, and tools. *Lect Notes Comput Sci* 2679:483–505
- Camarinha-Matos LM, Cardoso T (1999) Selection of partners for a virtual enterprise. In: Camarinha-Matos LM, Afsarmanesh H (eds) *Pro-VE'99 book, infrastructure for virtual enterprises: networking industrial enterprises*. Kluwer Academic Publishers, Boston, pp 259–278
- Camarinha-Matos LM, Pantoja-Lima C (2001) Cooperation coordination in virtual enterprises. *J Intell Manuf* 12(2):133–150
- Camarinha-Matos LM, Oliveira A, Sesana M, Galeano N, Demsar D, Baldo F, Jarimo T (2009) A framework for computer-assisted creation of dynamic virtual organisations. *Int J Prod Res* 47(17):4661–4690
- Durfee EH, Lesser VR, Corkill DD (1989) Trends in cooperative distributed problem solving. *IEEE Trans Knowl Data Eng* 1(1):63–83
- Ferber J (1999) *Multi-agent systems: an introduction to distributed artificial intelligence*. Addison Wesley, Reading, MA
- Grefen P, Mehandjiev N, Kouvas G, Weichhart G, Eshuis R (2009) Dynamic business network process management in instant virtual enterprises. *Comput Ind* 60(2):86–103
- Hoffner Y, Field S, Grefen P, Ludwig H (2001) Contract-driven creation and operation of virtual enterprises. *Comput Netw* 37(2):111–136
- Hsieh FS, Chiang CY (2011) Collaborative composition of processes in holonic manufacturing systems. *Comput Ind* 62(1):51–64
- Hsieh FS, Lin JB (2012) Virtual enterprises partner selection based on reverse auction. *Int J Adv Manuf Technol* 62(5–8):847–859
- Hsieh FS, Lin JB (2014a) Context-aware workflow management for virtual enterprises based on coordination of agents. *J Intell Manuf* 25(3):393–412
- Hsieh FS, Lin JB (2014b) A dynamic scheme for scheduling complex tasks in manufacturing systems based on collaboration of agents. *Appl Intell* 41(2):366–382
- Hugos M (2018) Chapter 1: Key concepts of supply chain management. In: Hugos M (ed) *Essentials of supply chain management*, 4th edn. Wiley, Hoboken, pp 1–41
- Ivanov D, Dolgui A, Sokolov B (2016) Robust dynamic schedule coordination control in the supply chain. *Comput Ind Eng* 94:18–31
- Jagdev H, Vasiliu L, Browne J, Zaremba M (2008) A semantic web service environment for B2B and B2C auction applications within extended and virtual enterprises. *Comput Ind* 59(8):786–797
- Kovács G, Kot S (2017) Economic and social effects of novel supply chain concepts and virtual enterprises. *J Int Stud* 10:237–254
- Lambert DM, Croxton KL, Garcia-Dastugue SJ, Knemeyer M, Rogers DS (2006) *Supply chain management processes, partnerships, performance*, 2nd edn. Hartley Press Inc., Jacksonville
- Liu X, Chung TP (2017) An outsourcing-scheduling problem in a two-stage supply chain via improved immunoglobulin-based artificial immune system. *Comput Ind Eng* 113:819–830
- McFarlane DC, Bussmann S (2000) Developments in holonic production planning and control. *Int J Prod Plan Control* 11(6):522–536
- Mentzer JT, Dewitt W, Keebler JS, Min S, Nix NW, Smith CD, Zacharia ZG (2001) Defining supply chain management. *J Bus Logist* 22(2):1–26
- Moktadir MA, Ali SM, Rajesh R, Paul SK (2018a) Modeling the interrelationships among barriers to sustainable supply chain management in leather industry. *J Clean Prod* 181:631–651
- Moktadir MA, Rahman T, Rahman MH, Ali SM, Paul SK (2018b) Drivers to sustainable manufacturing practices and circular economy: a perspective of leather industries in Bangladesh. *J Clean Prod* 174:1366–1380
- Monostori J (2018) Supply chains robustness: challenges and opportunities. *Proc CIRP* 67:110–115
- Murata T (1989) Petri nets: properties, analysis and applications. *Proc IEEE* 77(4):541–580
- Neligwa T, Fletcher M (2003) An HMS operational model. In: Deen SM (ed) *Agent-based manufacturing: advances in the holonic approach*. Springer, Berlin, pp 163–191
- Nilsson NJ (1998) *Artificial intelligence: a new synthesis*. Morgan Kaufmann Publishers Inc, San Francisco
- OASIS (2009) Web services business process execution language version 2.0. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>. Accessed 1 Apr 2018
- IBM ILOG CPLEX Optimizer (2015) <https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>. Accessed 1 Apr 2018
- Parunak HVD (1987) Manufacturing experiences with the contract net. In: Huhns M (ed) *Distributed artificial intelligence*. Pitman, London, pp 285–310
- Petrie C, Bussler C (2003) Service agents and virtual enterprises: a survey. *Internet Comput* 17(4):68–78
- Polyak BT (1969) Minimization of unsmooth functionals. *USSR Comput Math Math Phys* 9:14–29
- Putnik G, Sluga A, ElMaraghy H, Teti R, Koren Y, Tolio T, Hon B (2013) Scalability in manufacturing systems design and operation: state-of-the-art and future developments roadmap. *CIRP Ann Manuf Technol* 62(2):751–774
- Ramos C (1996) A holonic approach for task scheduling in manufacturing systems. In: *Proceedings of the 1996 IEEE international conference on robotics and automation*, pp 2511–2516
- Samdantsoodol A, Cang S, Yu H, Eardley A, Buyantsogt A (2017) Predicting the relationships between virtual enterprises and agility in supply chains. *Expert Syst Appl* 84:58–73
- Smith RG (1980) The contract net protocol: high-level communication and control in a distributed problem solver. *IEEE Trans Comput* 29(12):1104–1113
- Soares AL, Azevedo AL, De Sousa JP (2000) Distributed planning and control systems for the virtual enterprise: organizational



- requirements and development life-cycle. *J Intell Manuf* 11(3):253–270
- van der Aalst WMP (1998) The application of Petri nets to workflow management. *J Circuit Syst Comput* 8(1):21–66
- van der Aalst WMP, Kumar A (2001) A reference model for team-enabled workflow management systems. *Data Knowl Eng* 38(3):335–363
- Wang Dongsheng, Nagalingam Sev V, Lin Grier CI (2007) Development of an agent-based Virtual CIM architecture for small to medium manufacturers. *Robot Comput Integr Manuf* 23(1):1–16
- Weber M, Kindler E (2002) The Petri net markup language. [http://www2.informatik.hu-berlin.de/top/pnml/download/about/PNML\\_LNCS.pdf](http://www2.informatik.hu-berlin.de/top/pnml/download/about/PNML_LNCS.pdf). Accessed 1 Apr 2018
- Weske M, van der Aalst WMP, Verbeek HMW (2004) Advances in business process management. *Data Knowl Eng* 50(1):1–8
- Workflow Management Coalition (1999) The workflow management coalition specifications: terminology and glossary. <http://www.wfmc.org>. Accessed 1 Apr 2018

