Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 36 (2009) 2659–2669
Expert Systems with Applications
www.elsevier.com/locate/eswa

The design of a JADE-based autonomous workflow management system for collaborative SoC design

Charles V. Trappey [a], Amy J.C. Trappey [b,e,*], Ching-Jen Huang [c], C.C. Ku [d]

[a] Department of Management Science, National Chiao Tung University, Hsinchu 300, Taiwan, ROC
[b] Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu 300, Taiwan, ROC
[c] Department of Industrial Engineering and Management, National Chin-Yi University of Technology, Taichung 411, Taiwan, ROC
[d] Mechanical Industrial Research Institute, Industrial Technology Research Institute, Hsinchu 300, Taiwan, ROC
[e] Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei 106, Taiwan, ROC

Abstract

Given a fast changing electronics goods marketplace, designers of integrated circuits and components need to be more competitive, efficient, flexible, and use collaborative workflow to reduce time-to-market and a project's life cycle. In recent years, agent-based workflow management systems (WfMS) have been widely used to monitor and control business design processes. In this paper, intelligent agents are applied to the collaborative system-on-chip (SoC) design environment. The proposed JADE-based autonomous workflow management system (JAWMS) uses a workflow coordination mechanism and an agent integration mechanism to enable the analysis, management and interaction of automated design processes. The workflow coordination mechanism uses five domain specific agents to perform the workflow enactment services and a generic agent to control the system flow logic. The system kernel of JAWMS follows the specifications of the workflow reference model provided by the workflow management coalition (WfMC). The agent integration mechanism supports an agent to interact with other JADE-based agent platforms and to coordinate and monitor workflow coordination messages. All agents are written in the Java language using the JADE platform and work together to perform flexible, adaptive and dynamic design tasks in an autonomous and collaborative way. JAWMS facilitates SoC design and team interaction in a collaborative but distributed product development environment.

Keywords: Workflow management; Collaborative design; Intelligent agent technology; JADE

1. Introduction

In the past decade, workflow management systems (WfMSs) have been distinguished due to their significance and their impact on organizations (Cardoso, Sheth, Miller, Arnold, & Kochut, 2004). Workflow management is a promising technology that automates the business processes to improve the speed, efficiency and manageability of an organization's teamwork (Wang, Wang, & Xu, 2005). In order to reduce time-to-market and a project's life cycle, many organizations propose workflow management system as a mechanism to facilitate teamwork in a collaborative product development environment (Huang, Huang, & Mak, 2000; Huang, Trappey, & Yao, 2003). However, due to the lack of flexible and integrated mechanisms to deal with data exchange and application integration between heterogeneous systems, traditional workflow management is often too inflexible to meet the complex, dynamic situations in distributed, heterogeneous platforms. Recently, the use of agents for collaborative design has been demonstrated (Trappey, Trappey, Hou, & Chen, 2004). Researchers apply agent technology in the workflow system to achieve the benefits of autonomy, reactivity, pro-activeness, and mobility (Huang, Trappey, & Yao, 2006; Kuo, 2004; Xia & Li, 1999). There remain several issues with the use

* Corresponding author. Address: Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu 300, Taiwan, ROC. Tel.: +886 3 5742651; fax: +886 3 5722204.
E-mail address: trappey@ie.nthu.edu.tw (A.J.C. Trappey).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.01.064

of agent-based systems. The problem of coordination and communication between agents and agent-based platforms is often the central focus of the research (Xia & Li, 1999).

In this paper, an autonomous workflow management framework based on multi-agent technology for collaborative system-on-chip (SoC) design is proposed. The proposed framework, named as JADE-based autonomous workflow management system (JAWMS), follows the specifications of the workflow reference model defined by the workflow management coalition (WfMC, 1999). In order to enhance the cooperation and coordination between agents, workflow ontology is created and a standardized workflow metadata model is implemented. JAWMS use a workflow coordination mechanism and an agent integration mechanism, in which a society of intelligent agents work autonomously and collaboratively to perform design tasks. The workflow coordination mechanism uses five domain specific agents to perform the workflow enactment services and a generic agent to control the system flow logic. The agent integration mechanism supports an agent to interact with other agent platforms and to coordinate with workflow coordination mechanism. Further, the java agent development framework (JADE) (JADE, 2007) is used as the agent platform for linking the heterogeneous SoC design systems in a distributed environment.

The contributions of this paper are twofold. First, we adopt multi-agent technology to develop an autonomous workflow management system for collaborative system-on-chip (SoC) design. The proposed framework suits the heterogeneous environment of electronic component design firms. Second, the workflow ontology is enhanced and clarifies the relationship between workflow components. The goals of the proposed system include ease of construction, interoperability, and information transparency.

The remainder of this paper is organized as follows. Section 2 reviews the related literature in the areas of workflow management systems, agent technology, and JADE platforms. Section 3 describes the collaborative SoC design process and environment and provides a workflow diagram for SoC design. Section 4 illustrates our autonomous workflow system framework, the workflow ontology, and the agent communication model. Based on this framework, a prototype of an autonomous workflow management system for collaborative system-on-chip (SoC) design is developed. The last section summarizes the contributions and provides suggestions for future work.

2. Literature review

2.1. Workflow management systems

With the rapid development of Internet and information technologies, many organizations are faced with managing the workflow for collaborative design on heterogeneous platforms linked to a distributed computing infrastructure. A workflow is an ongoing process distributed among multi-participants that transfer information or tasks according to some previous defined rules or sequences (Huang et al., 2003). The workflow management coalition defines a workflow management system (WfMS) as a software application that supports the specification and execution of a workflow (WfMC, 1999). The system contains a set of tools and interfaces that provide support for the necessary services of workflow management, process definition, administrative and monitoring tasks, workflow client applications and other invoked applications (Georgakopoulos, Hornick, & Sheth, 1995; Hollingsworth, 1995). ARTech Consultores (2002) defines four types of workflows, i.e., administrative workflows, production workflows, collaborative workflows and ad hoc workflows. The main differences between these types are the workflow structure, repetitiveness, predictability, complexity and rules of flow logic (Cardoso et al., 2004). Kuo and Cheng (1999) identify three Rs, i.e., Roles, Routes and Rules, which a WfMS must provide for automating well-defined and structured workflow. However, the logical workflow, considering the combination of internal and external flows on heterogeneous platform in a distributed environment, is still a very challenging research area (Wang, Shen, & Hao, 2006).

2.2. Agent technology and agent-based workflow management systems

An agent is a software entity that can autonomously perform routine tasks with a degree of intelligence (Boudriga & Obaidat, 2004). Nwana (1996) divides agents into seven types, i.e., collaborative agents, interface agents, mobile agents, information/Internet agents, reactive agents, hybrid agents and smart agents. Goal driven agents typically possess three key characteristics, i.e., autonomy, cooperation and learning (Etzioni & Weld, 1995; Liang & Huang, 2002; Nwana, 1996). They are capable of acting autonomously, cooperatively, and collectively. The purposes of applying agent technology to develop applications are intended to exchange flexibility and reduce risks of concurrent, distributed and collaborative information processing, particularly for concurrent and distributed product design and manufacturing.

Using agents to perform WfMS functions is an emerging field where researchers are exploring the ability of agents to improve process integration, interoperability, reusability and adaptability (Huang et al., 2006; Trappey, Trappey, & Lin, 2006). More and more researchers believe that agent technology can provide coordinative architectures for integrating multiple heterogeneous workflow system (Kuo, 2007). An agent-based WfMS is a set of software agents that manage and coordinate the flow of work defined by a business process (Odgers, Shepherdson, & Thompson, 1999). The agent-based approach facilitates design process knowledge reuse while supporting distributed dynamic process management. From the system point of view, most WfMSs are multi-agent systems and can be classified as agent-enhanced workflow management and agent-based workflow management (Yan, Maamar, & Shen, 2001). For most multi-agent systems, a coordination mechanism is not considered which makes the system more difficult to design and integrate with other WfMS. Agents must cooperate, communicate and negotiate with other agents to coordinate and control distributed workflow tasks. In order for agents to communicate effectively and enhance interoperability, there is a need for mutually understandable and standard semantic constructs.

Over the past few years, many agent-based WfMS issues have been developed and these efforts are summarized in Table 1. Based on the literature survey, research on designing and developing multi-agent systems focuses on the interaction between agents (Chen & Chen, 2007). Topics frequently discussed in this area of research include agent action, the relationship between agents, multi-agent system architecture and the environment in which the multi-agent systems exist; interactions among agents within the multi-agent system, and agent adaptation ability. Coordination, negotiation, cooperation are three common interactions among agents in multi-agent systems.

As shown in Table 1, Wang et al. (2006) apply web service technology to develop an agent-based workflow prototype for inter-enterprise collaboration. Although the AADA-based prototype provides a basic framework, it may be too simple to satisfy complex industrial scenarios such as SoC design. Sometimes, when considering security, a well-protected and tight-controlled WfMS may be more suitable for real cases like SoC design process. Further, an easy to understand and standard workflow ontology will improve WfMS definition on heterogeneous platforms. Finally, from the viewpoint of system development, the applications should be developed under a free, well-known agent platform to shorten the project development lifecycle. In order to overcome these concerns, this paper develops a concentrated, agent-based WfMS with a well-defined workflow ontology for SoC design management and also uses a free platform, JADE, for development.

2.3. The agent platform

The most primitive form of basic agent coordination is communication. Various types of agent communication languages (ACL), such as the knowledge query and management language (KQML) and its variants, have been made available by Richard Baeza–Yates, Bertheir Ribeiro–Neto, and Tim Finin (Boudriga & Obaidat, 2004). KQML was the first ACL to have substantial application in industry. The Foundation for Intelligent Physical Agents (FIPA) is an international, non-profit association that is dedicated to promoting the industry of intelligent agents by openly developing specifications supporting interoperability among agents and agent-based applications. FIPA defined an agent platform reference architecture based on KQML protocol for agent-based applications (FIPA, 2002). The proposed agent platform is composed of a message transport system (MTS) used for communicating with other agents or agent platforms, an agent management system (AMS) used for managing the agent life cycles such as starting and stopping, and a directory facilitator (DF) used for recording the services provided by an agent.

TILab proposes a software development framework, named JADE (JADE, 2007), aimed at developing multi-agent systems and applications. This software framework uses ACL specifications proposed by FIPA and provides a set of graphical tools that supports the debugging and deployment phases. JADE is middle-ware useful for developing agent-based applications in distributed environments and includes an FIPS-compliant agent platform and a package to develop Java agents (Bellifemine, Caire, Trucco, Rimassa, & Mungenast, 2007). JADE is written in Java language and supports two types of agent containers, the

Table 1
Agent-based WfMS research

Authors	Agent pattern	Contributions
O'Brien and Wiegand (1998)	Agent-based	Combine recent developments in autonomous agent technology and distributed computing platforms to develop agent-based process management system
Oliveira, Fischer, and Stepankova (1999)	Agent-enhanced	Apply agent technology in WfMS to reduce business collaborative cost and solve the coordination problems between WfMS
Huang et al. (2000)	Agent-based	Integrate multi-agent technology to WfMS to facilitate teamwork in collaborative product design
Blake (2001)	Agent-enhanced	Convert user-defined UML process to workflow by using component-based agent architecture as a middleware.
Xu and Wang (2002)	Agent-based	Address intelligent multi-agents to monitor the dynamic nature of transaction processes in B2B e-commerce workflow
Zhuge (2003)	Agent-based	Define an agent-based mechanism to model, control and manage the cognitive flow process to improve the problem solving ability of a team
Wang et al. (2005)	Agent-based	Propose a novel workflow monitoring approach based on intelligent agents to perform flexible monitoring tasks in an autonomous and collaborative way in securities trading
Madhusudan (2005)	Agent-based	Embed autonomous agents in a workflow-based distributed systems infrastructure to support design activities in an industrial context with a case study
Huang et al. (2006)	Agent-based	Develop a prototype of agent-based intelligent workflow system for product design collaboration in a distributed network environment
Wang et al. (2006)	Agent-based	Develop an agent-based workflow model to serve inter-enterprise collaboration by applying web service-based technology for messaging, service description, and workflow enactment

main container and the normal container. The JADE main container consists of a message transport system, an agent message system, and a directory facility. Each JADE agent registers itself using the remote method invocation (RMI) provided by the JADE main container. Therefore, an agent can find any other agent through the DF and send messages to other agents asking for services. This mechanism simplifies the implementation of multi-agent-based workflow systems in distributed environment. Furthermore, JADE supports different agent communication protocols to deliver messages depending upon the environment of the receiving agent (Park & Sugumaran, 2005).

3. Process analysis of collaborative SoC design

Process analysis is used to reduce the complexity in the problem domain and identify the key processes and members involved. The challenge arising when using process analysis is to model the entire process and transform it into an accurate workflow. There are two types of problems. First, the accuracy problem of the world representation: translating the real process should be reasonable, use adequate specification, and be reactive. Second, there is a representation and reasoning problem, i.e., how to choose the processes' symbolic representation. This research uses UML modeling techniques to model the process logic.

Industrial integrated circuit electronic component design requires product collaborative design with a number of people with complementary skills working as a team to realize the design concept and to finish the design project. All participants contribute their different domain expertise at various stages in order to build a single electronic circuit that integrates multiple functionalities. Usually, the design company purchases some of the design know-how or intellectual property (IP) from IP providers. Since IP is sourced globally, the design managers must also coordinate and safeguard technology exchanges between engineers and vendors over the Internet. These scenarios make the design processes more complex and require substantial communications and data exchanges over the Internet.

Fig. 1 shows an example of the collaborative IC design processes for a system-on-chip (SoC) integrated circuit. The life cycle of SoC collaborative design processes includes a series of stages, such as architecture definition, co-design, co-simulation, and layout and post simulation. Each stage requires collaborative partners to fulfill tasks and to ensure the project can be completed on time. In the beginning of design process, the project manager derives a preliminary agreement with his design team or customers on product specifications. The SoC design team consists of a project manager, an IP provider/engineer, an analog IC designer, a digital IC designer, and a layout engineer. After the preliminary design concept is generated, the SoC design process is started.

In the stage of architecture definition, the project manager first defines the specifications of a SoC. And then defines the architecture and block function of the chip system based on the design specifications. In the stage of co-design, there requires three participants' involvement, named IP provider, analog IC designer, and digital IC designer respectively, and forms three sub-processes. Analog IC designers firstly use cadence composer to do transistor level design and draw circuit schematic. Afterward, analog IC designers apply HSPICE software to simulate the device output. Digital IC designers use hardware description language, e.g., Verilog, to do register transfer level design (RTL design). After programming, digital IC designers perform Verilog simulation to generate waveform to check if the waveform is in accordance with the rules. If rules are all satisfied, digital IC designers precede circuit synthesis to compose a transistor level circuit. In this stage, the IC design company may need some helps from IP providers in some critical design know-how. IP providers do SIP authorizing and check if the specifications of manufacturing processes are in accordance with the proposed circuit.

In the stage of co-simulation, IP provider, analog IC designers, and digital IC designers co-verify the combined circuit of their output from the previous stage satisfies the design requirements via simulation runs. If the circuit design process passes the co-simulation test, the process will go to the stage of layout and post simulation. In the stage of layout and post simulation, the layout engineer integrates circuits combined from digital layout (from the "generate layout" step using Apollo software), analog layout (from the "draw layout" step using Cadence composer software), and SIP layout (from the "layout of SIP" step provided by IP providers). Then, the layout engineer verifies the final integrated layout. There are three main steps in the verification, i.e., the design rule check, the layout versus schematic and the layout parasitic extraction. After the verifications are accomplished, the layout engineer conducts the post simulation to check the parasitic effects. If the simulation results are acceptable, the design process goes to the tape out phase, and, afterward, the manufacturing process can start.

After analyzing the industrial IC design process, the process flow is translated into a workflow diagram to indicate the sequences of collaborative tasks and the corresponding roles of team members. Because the workflow is decentralized across heterogeneous environments, the system should be able to integrate with other workflows and applications to execute tasks. This scenario is well suited for software agents that can provide flexible and autonomous solutions for workflow management and business process coordination. In the following sections, the JADE-based autonomous workflow management system (JAWMS) framework for collaborative IC design is described.

4. The JAWMS framework

In this section, the proposed autonomous workflow management system (JAWMS) framework is derived and the functionality and internal structure of the agents and workflow components are defined. As shown in Fig. 2, a

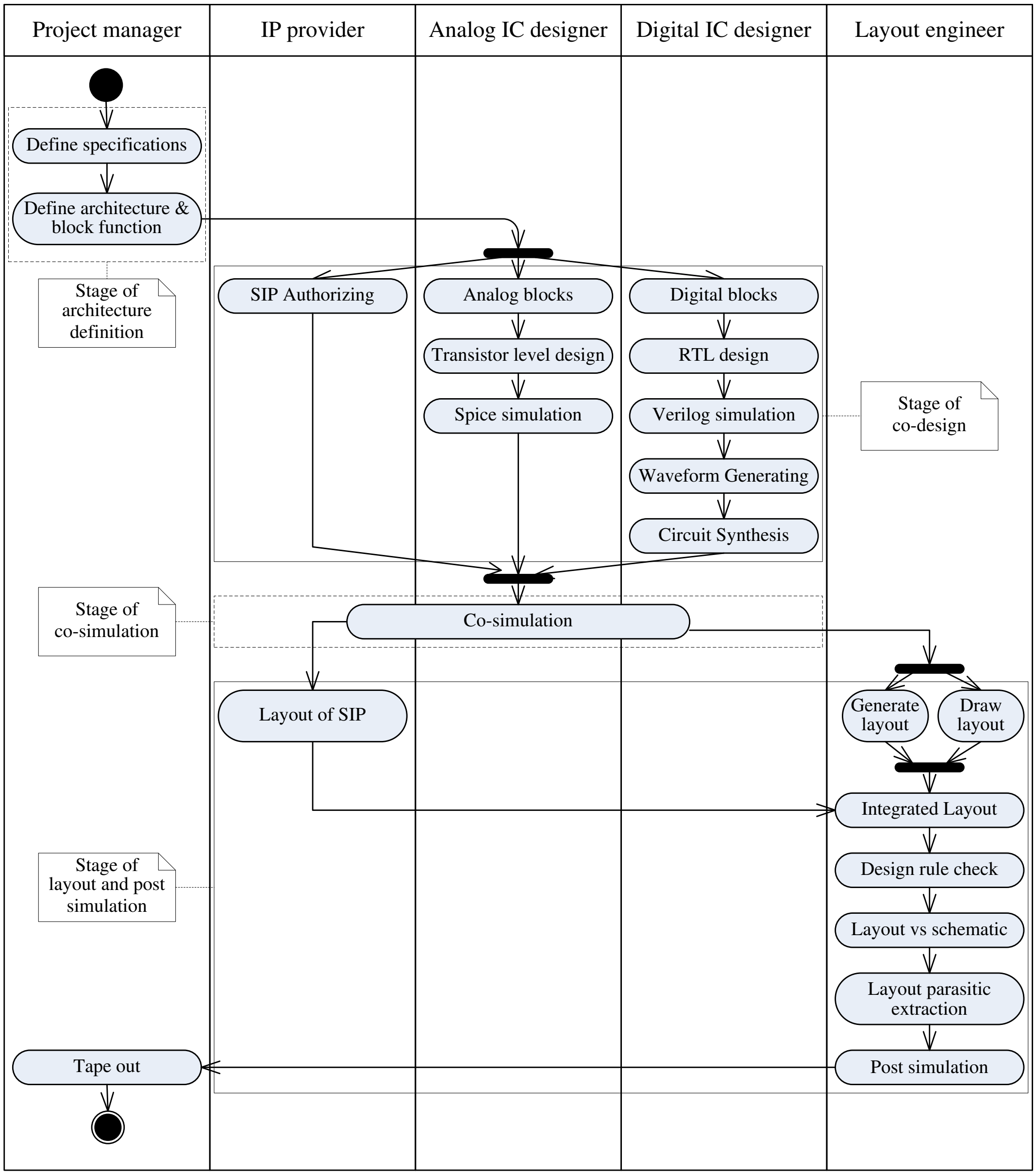

Fig. 1. Collaborative SoC design process.

multi-agent framework is used to provide intelligent support for collaborative design. The framework includes a workflow enactment mechanism (WfEM) and an agent integration mechanism (AIM). The WfEM handles the workflow execution and follows the specifications of the workflow reference model. The WfEM includes a workflow agent group with five agents including the workflow execution agent (WfEA), workflow maintenance agent (WfMA), workflow monitoring and control agent (WfMCA), project performance agent (PPA), and system maintenance agent (SMA) respectively, to communicate with the AIM server. The AIM server works as a JADE main container with a single interaction agent to monitor the messages transferred among agents and to communicate with other agent-based workflow system platforms. There are a total of seven types of agents in the JAWMS system.

4.1. Agent communication model

Agents are capable of acting autonomously, cooperatively with an agent's behavior depending on defined goals, beliefs, and actions. The behavior can be modeled to respond to certain situations that the agent is or changes that have taken place in the environment. As shown in the proposed framework, each JAWMS agent uses its own flow logic and can be viewed as a functional module to accomplish the predefined goals. The agents use a common shared communication protocol, named as agent communication language (ACL) proposed by Foundation for Intelligent Physical Agents (FIPA), among them to understand each other freely and interact with other agents according to their capabilities and functional hierarchy. The seven agents are defined as follows.

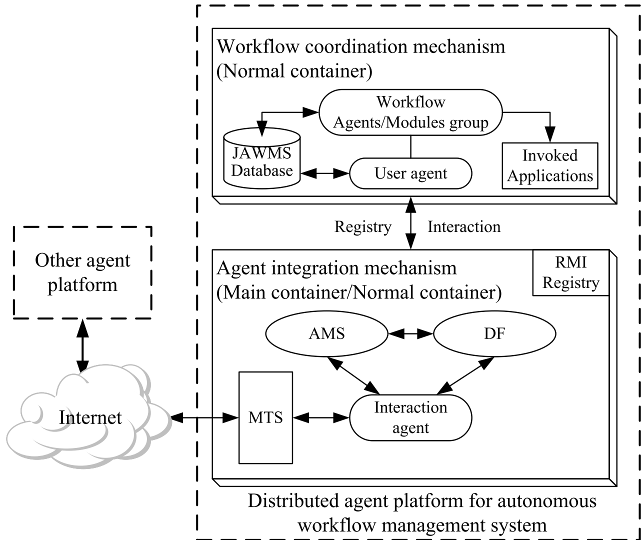

Fig. 2. The JADE-based autonomous workflow management system (JAWMS) framework.

The user agent (UA) is an interface agent that manages the user's access authority, unfinished jobs, and communication with other agents. It acts as an effective bridge between users and workflow agent group. In our JAWMS, the UA is the core agent of JAWMS and help users to complete tasks. When users login, the system presents different interfaces based on role and authority, such as collaborative designer interface, workflow manager interface, system administrator interface, and project performance manager interface. The UA wakes up the corresponding workflow agents according to the user's authority and listens to ACL messages from other agents, as shown in Fig. 3. For example, when a user with an authority type of workflow manager, UA firstly send ACL message with content (INFORM, language: English, content: "Starting") to WfEA, WfMCA, and WfMA separately to wake up them to wait for user's choice if the agents are not started. Agents will start and then response an ACL message with content (CONFIRM, language: English, content: "Standby"). The UA then send an ACL message with content (AGREE, language: English, content: "Agree") to finish the "waking up agents" process. The UA also controls and monitors the whole workflow logic.

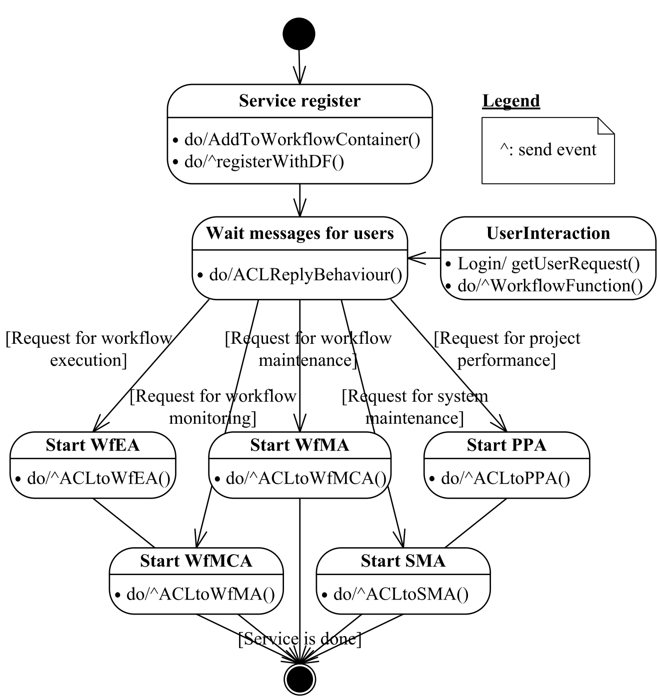

Fig. 3. State diagram activating the UA according to user's authority.

The workflow execution agent (WfEA) manages task execution. The users defined in JAWMS can also ask the WfEA to provide predefined services. When a collaborative designer signs on to the system, the UA wakes up the WfEA by sending a ACL message with content (INFORM, language: English, content: "Working") and then sends the user's account data via a series of ACL handshaking messages. The interactive process of sending ACL messages between the UA and the WfEA is shown in Fig. 4. After the WfEA is started, WfEA firstly searches the JAWMS database to find all unfinished tasks of the user and highlight the delayed tasks as an alarm signal. If the user selects a task from the work list for execution, he will see the attached collaborative working file. After double click the working file, the invoked application will start up for product design. When the user finished the task, the system will ask the user to sign the task. If the user signed, the task is accomplished and WfEA immediately send an ACL message with content (INFORM, language: English, content: "Job_Finished") to notify UA the task is finished and, in the meantime, the next task of the workflow is marked as "Active".

The workflow maintenance agent (WfMA) is responsible for workflow element definition, workflow template creation, and workflow instance generation. It can add, modify, delete, and query any workflow template and the associating tasks. The whole process of workflow definition consists of three main phases, such as workflow elements

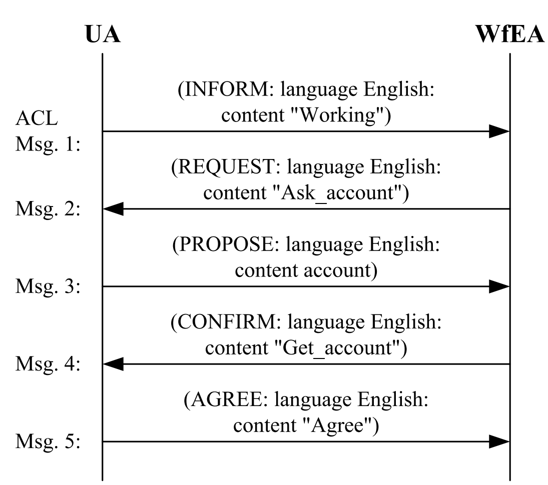

Fig. 4. Handshaking between UA and WfEA.

definition, workflow template creation, and workflow instance generation. In the workflow elements definition phase, workflow manager firstly define message elements for notes in task execution and document elements for references about design specifications separately. Afterwards, workflow manager defines task contexts and adds message and/or document elements as its attached information. In the workflow template creation phase, the workflow manager defines workflow information and combines the tasks with transition rules to form a workflow template. In the workflow instance generation phase, workflow manager can easily selects a workflow template from workflow template list and create a workflow instance for execution. Fig. 5 show simplified ACL message interactions between agents in workflow setup and run-time mode.

The workflow monitoring and control agent (WfMCA) tracks the real time status of workflow progress by querying the workflow instances. WfMCA activates workflow monitoring and workflow control. In workflow monitoring module, the workflow manager can check workflow status from unfinished workflow list and select a workflow to watch the progress of specified workflow. In workflow control module, workflow manager can reset a workflow status as Suspend, Resume, and Abort. Fig. 6 shows the relationship of agent's function to WfMC's workflow reference model.

The project performance agent (PPA) provides functions for periodically evaluating projects including user performance analysis and overall project performance evaluation. The system maintenance agent (SMA) provides user management and function-authority management. For the user management function, SMA can add, modify, and delete a user. Under its function-authority management function, SMA will define corresponding functions depending on the user's role. Fig. 7 shows the use case diagram for JAWMS. The JAWMS has five agent modules for

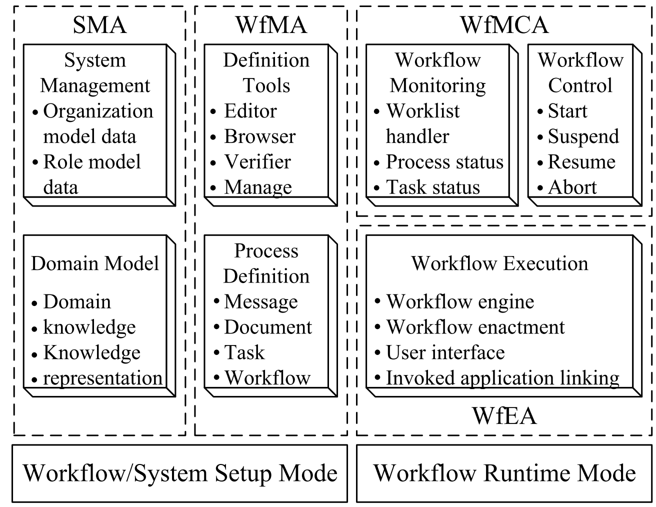

Fig. 6. Relationship of agent's function to the workflow reference model.

workflow management and security and defines four roles for users.

The system administrator has the highest authority and is permitted to perform all system functions. A workflow manager can start the WfMA and the WfMCA to define and modify a workflow and the corresponding task details. A project performance manager can wake up the PPA to analyze the performance of a single user or the whole project. Finally, collaborative designers (end users) are allowed to perform the functions supported by WfEA, e.g., modify personal data, show work lists, and execute designated task.

The interaction agent (IA), which resides on the AIM server, is used to monitor ACL messages sent between the workflow agent groups on the WfEM server. The IA also acts as an expandable interface to other agent platforms for application integration.

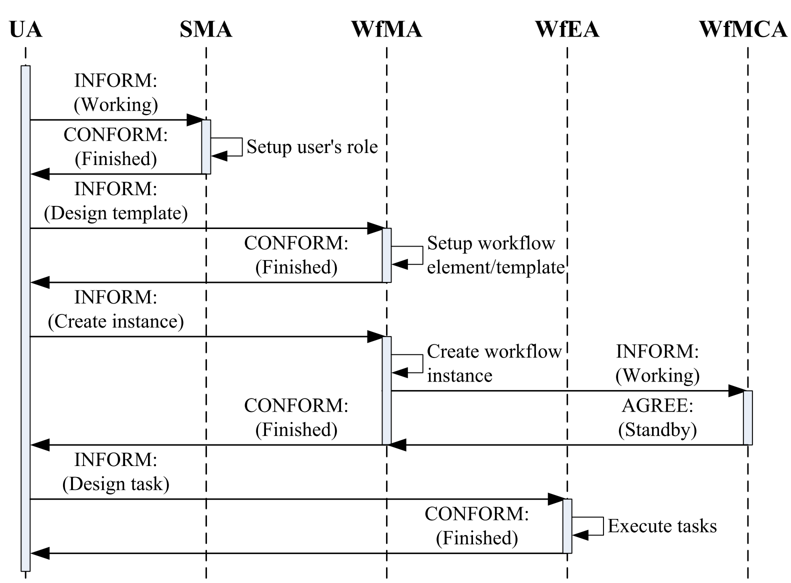

Fig. 5. A simplified ACL messages between agents during workflow execution.

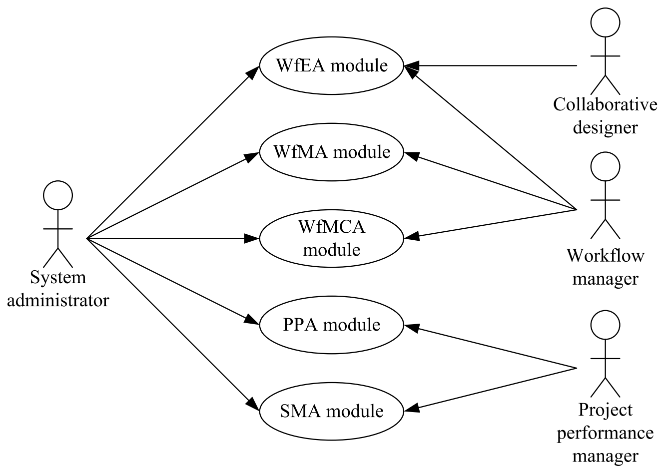

Fig. 7. Use case diagram of JAWMS.

4.2. Workflow ontology definition

The workflow ontology represents relationships between workflows, resources and actors. It is similar to a description of workflow model that depicts a group of processes with interdependence. Fig. 8 shows the workflow ontology defined in JAWMS. The objects in this ontology are the workflow components including the workflow template, workflow instance, tasks, task instances, messages, and documents. The relationship between workflow components includes roles, transition rules, dispatching rules, invoked applications, client applications, and project performance analysis. The objects in the proposed JAWMS workflow model are defined as follows:

- An *actor* is a person or machine that performs a task by fulfilling a defined service; it's a type of resource.
- A *role* abstracts a set of actor type into a logical group of activities. An actor playing a particular role carries out a set of activities/tasks.
- A *workflow* is a collection of partial or total set of tasks organized to accomplish some business processes. It is usually bound to particular resources that fulfill the process.

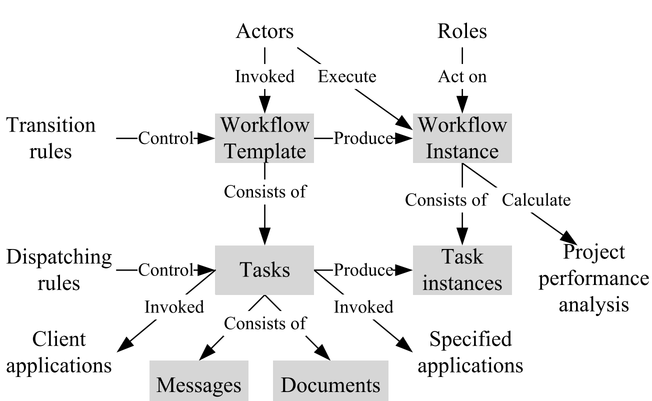

Fig. 8. Workflow ontology in JAWMS.

- A *workflow template* is a logical definition of workflow. It is used to generate workflow instance.
- A *workflow instance* is a duplicate of workflow template for execution.
- A *task* is an atomic work item that is a part of a workflow. Task is bound to particular resources that fulfill the activity.
- A *task instance* is a duplicate of specific task, defined within a specific workflow instance.
- A *message* or a *document* is a type of resource for reference during the execution of tasks in workflow.
- *Transition rules* are used to define the relationship between tasks in order to combine a set of tasks into a workflow. Transition rules in JAWMS include and-join route, or-join route, and-split route, or-split route, conditional route, and deadline route.
- *Dispatching rules* define who has an authority on performing a specified task.
- *Specified applications* or *invoked application* is specified software used to finish a domain task.

The terms, *role*, *task*, *workflow*, *workflow template*, *message*, *document*, and *dispatching rule*s, are used in specification of conditions/relationships that are directly related to the workflow process. However, the *actor*, *workflow instance*, *task instance*, *transition rules,* and *specified applications* or *invoked application* terms specify implementation-oriented information. In JAWMS, messages and documents are the basic elements used for delivering information when the workflow is executing. A workflow template consists of multi cascaded tasks that are defined by applying transition rules. When a new collaborative IC design project starts, the workflow manager creates a workflow instance from a selected workflow template.

The workflow ontology describes the information relevant to a workflow, such as pre and post conditions for the operation of a task, the data structure, the control parameters, and the invoked applications and operational modes. This information is relayed by relevant agents. The workflow metadata model is defined as follows.

Message = {Msg#, Msg, replyMsg, ...}
Document = {Doc#, Abstract, Doc, ...}
Task = {Task#, User, Msg#, Doc#, ClientApps, InvokedApps,
DispatchingRules, ...}
TaskInstance = { TaskIns#, Task#, ...}
WorkflowTemplate = {WfTemp#, Task#, TransitionRules, ...}
WorkflowInstance = {WfIns#, User, WfTemp#, ...}

4.3. System implementation

This section provides details of how the system is implemented using the JAWMS architecture. The prototype has been implemented using JADE from CSELT, Turin, Italy.

The JAWMS system is implemented on two personal computers, one for AIM server (acting as the JADE main container) and the other for WfEM server (acting as the normal container). Both computers are running the Microsoft Windows XP platform with Java Development Kit, JDK1.5, and TILab JADE version 3.4. The database system is the Microsoft SQL server 2000 installed on the WfEM server. The agent system is developed using the JADE framework version 3.4 and edited by JCreator Pro version 3.5. JAWMS is started with a DOS batch file.

From the deployment point of view, each partner participating in the collaborative IC design process installs the client system provided by JAWMS. Once the client system is online, the client computer communicates with the AIM

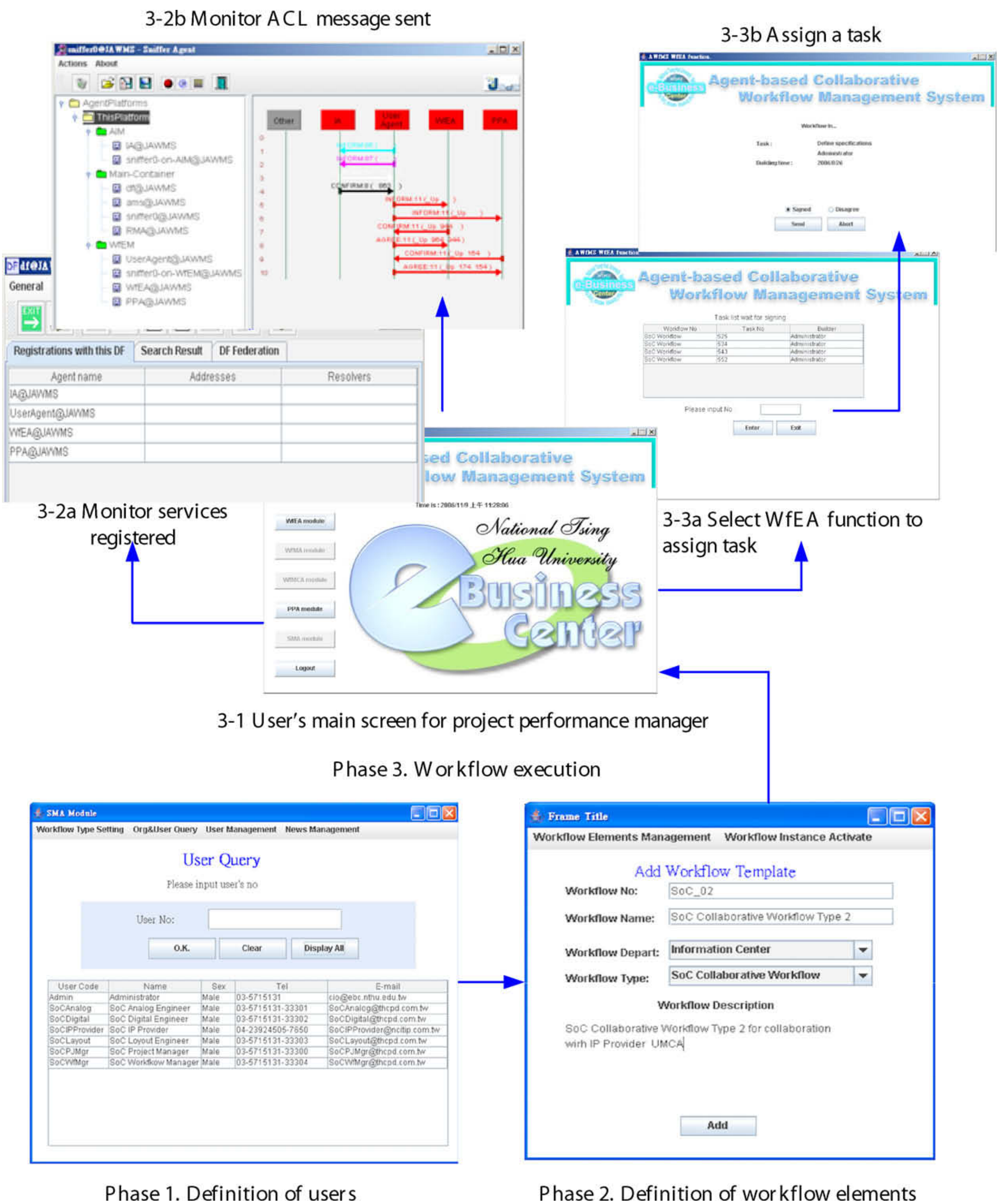

Fig. 9. Examples of JAWMS functions and interface.

server in the primary company to complete the registry process. Users of the client computer can acquire the execution work list through the WfEM server. That is, any collaborative participant can join over the Internet by installing the client system of JAWMS at their location.

Fig. 9 shows an example of the JAWMS functions and user interface. The operation cycle of JAWMS includes three main phases. During Phase 1, the system administrator provides the definition of users and their corresponding authorities. During Phase 2, the workflow manager provides definitions of message elements, document elements, task elements, and workflow templates. Finally, during Phase 3, the workflow manager creates a workflow instance from workflow template and submits it for execution. When a user login the JAWMS, the corresponding agents will register itself to JADE's Directory Facilitator if the agents are not ready, as shown at 3-2a in Fig. 9. The system administrator can also use Sniffer supported by JADE to investigate the ACL messages transferring between agents, as shown at 3-2b in Fig. 9. If a collaborative designer would like to perform a task, he/she can activate the WfEA to select a task from a workflow list, as shown at 3-3a in Fig. 9. When collaborative designer finished the selected task and made a sign, as shown at 3-3b in Fig. 9, the workflow engine will mark the next task as active for execution. The workflow continues task by task until completion.

5. Conclusion

Collaborative IC design is a complex workflow. One of the challenges for workflow system development is to provide better interoperability and transparency for heterogeneous platforms. In this research, we propose a JADE-based autonomous workflow management system (JAWMS) for collaborative IC product design. Using the JADE agent framework to establish communications in distributed environments, JAWMS successfully increases interoperability and information transparency in distributed environments.

The major contribution in this paper is the development of a novel system architecture for multi-agent WfMS applied to collaborative IC design. In addition, the proposed system applies workflow metadata ontology as the data modeling standard for JAWMS. In summary, the system provides the following advantages as a collaborative design workflow management tool.

- The prototype facilitates design cooperation and teamwork communication in a collaborative product development environment.
- The platform increases interoperability in cross-platform, distributed environments.
- The use of a workflow definition template based on workflow metadata ontology increases the flexibility of the system.

Acknowledgements

This research is partially supported by the National Science Council and the Industrial Technology Research Institute Research Grants.

References

ARTech Consultores. (2002). GXflow: Workflow and UML general view. <http://www.genexus.com/> Cited 20.06.05.

Bellifemine, F., Caire, G., Trucco, T., Rimassa, G., & Mungenast, R. (2007). JADE administrator's/programmer's guide. <http://jade.tilab.com/doc/> Cited 30.06.07.

Blake, M. B. (2001). Agent-based workflow configuration and management of on-line services. In *Proceedings of the International Conference on Electronic Commerce Research* (ICECR-4), 567–588.

Boudriga, N., & Obaidat, M. (2004). Intelligent agents on the web: A review. *Computing in Science and Engineering*, 35–42.

Cardoso, J., Sheth, A., Miller, J., Arnold, J., & Kochut, K. (2004). Quality of service for workflows and web service processes. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 1*, 281–308.

Chen, R.-S., & Chen, D.-K. (2007). Apply ontology and agent technology to construct virtual observatory. *Expert Systems with Applications*. doi:10.1016/j.eswa.2007.02.018.

Etzioni, O., & Weld, D. S. (1995). Intelligent agents on the Internet: Fact, fiction, and forecast. *IEEE Expert, 10*(4), 44–49.

FIPA. (2002). FIPA ACL message structure specification/FIPA ontology service specification. <http://www.fipa.org/repository/aclspecs.html> Cited 30.05.07.

Georgakopoulos, D., Hornick, M., & Sheth, A. (1995). An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Database, 3*, 119–153.

Hollingsworth, D. (1995). *The workflow reference model*. Workflow Management Coalition.

Huang, G. Q., Huang, J., & Mak, K. L. (2000). Agent-based workflow management in collaborative product development on the internet. *Computer-Aided Design, 32*, 133–144.

Huang, C. J., Trappey, A. J. C., & Yao, Y. H. (2003). Design and development of a WfMC-based modularized network collaborative design system. *Journal of Chinese Institute of Industrial Engineers, 20*(4), 422–434.

Huang, C. J., Trappey, A. J. C., & Yao, Y. H. (2006). Developing an agent-based workflow management system for collaborative product design. *Industrial Management and Data Systems, 106*(5), 680–699.

JADE. (2007). <http://jade.tilab.com/> Cited 30-05-07.

Kuo, J. Y. (2004). A document-driven agent-based approach for business process management. *Information and Software Technology, 46*, 373–382.

Kuo, M. H. (2007). An intelligent agent-based collaborative information security framework. *Expert Systems with Applications, 32*(2), 585–598.

Kuo, G. S., & Cheng, H. Y. (1999). On workflow and agent technologies on the internet/intranet. In *Global telecommunications conference* (Vol. 3, pp. 1937–1941).

Liang, W. Y., & Huang, C. C. (2002). The agent-based collaboration information system of product development. *International Journal of Information Management, 22*, 211–224.

Madhusudan, T. (2005). An agent-based approach for coordinating product design workflows. *Computers in Industry, 56*, 235–259.

Nwana, H. S. (1996). Software agents: An overview. *The Knowledge Engineering Review, 11*(2), 205–244.

O'Brien, P. D., & Wiegand, M. E. (1998). Agent based process management: Applying intelligent agents to workflow. *The Knowledge Engineering Review, 13*, 161–174.

Odgers, B. R., Shepherdson, J. W., & Thompson, S. G. (1999). Distributed workflow co-ordination by proactive software agents. In *Proceedings of intelligent workflow and process management, IJCAI-99 Workshop.*

Oliveira, E., Fischer, K., & Stepankova, O. (1999). Multi-agent systems: Which research for which applications. *Robotics and Autonomous Systems, 27*, 91–106.

Park, S., & Sugumaran, V. (2005). Designing multi-agent systems: A framework and application. *Expert Systems with Applications, 28*(2), 259–271.

Trappey, A. J. C., Trappey, C. V., Hou, J. L., & Chen, B. J. G. (2004). Mobile agent technology and application for online global logistic services. *Industrial Management and Data System, 104*(2), 169–183.

Trappey, A. J. C., Trappey, C. V., & Lin, F. T. L. (2006). Automated silicon intellectual property trade using mobile agent technology. *Robotics and CIM, 22*, 189–202.

Wang, S., Shen, W., & Hao, Q. (2006). An agent-based web service workflow model for inter-enterprise collaboration. *Expert Systems with Applications, 31*(4), 787–799.

Wang, M., Wang, H., & Xu, D. (2005). The design of intelligent workflow monitoring with agent technology. *Knowledge-Based Systems, 18*, 257–266.

WfMC. (1999). Workflow management coalition terminology and glossary, document number WfMC-tc-1011, document status – issue 3.0.

Xia, M., & Li, H. (1999). Cooperative software agents for workflow management systems. APCC/OECC '99. In *Fifth asia-pacific conference on communications and fourth optoelectronics and communications conference* (Vol. 2, pp. 1063–1067).

Xu, D., & Wang, H. (2002). Multi-agent collaboration for B2B workflow monitoring. *Knowledge-Based Systems, 15*, 485–491.

Yan, Y., Maamar, Z., & Shen, W. (2001). Integration of workflow and agent technology for business process management. In *The sixth international conference on computer supported cooperative work in design* (pp. 420–426).

Zhuge, H. (2003). Workflow- and agent-based cognitive flow management for distributed team cooperation. *Information and Management, 40*, 419–429.