



Queensland University of Technology
Brisbane Australia

This is the author's version of a work that was submitted/accepted for publication in the following source:

Trappey, Amy, Hsiao, David W., & Ma, Lin (2011) Maintenance chain integration using Petri-net enabled multiagent system modeling and implementation approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part C : Applications and Reviews*, 41(3), pp. 306-315.

This file was downloaded from: <http://eprints.qut.edu.au/42335/>

© Copyright 2010 IEEE

Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Notice: *Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source:*

<http://dx.doi.org/10.1109/TSMCC.2010.2057425>

Maintenance Chain Integration Using Petri-Net Enabled Multi-Agent System Modeling and Implementation Approach

Amy J.C. Trappey^{1,2,*}, David W. Hsiao¹, and Lin Ma³

¹Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Taiwan
trappey@ie.nthu.edu.tw

²Department of Industrial Engineering and Management, National Taipei University of Technology, Taiwan
trappey@ntut.edu.tw

³School of Engineering Systems, CRC for Infrastructure and Engineering Asset Management, Queensland University of Technology, Australia
l.ma@qut.edu.au

Abstract

Engineering asset management (EAM) is a broad discipline and the EAM functions and processes are characterized by its distributed nature. However, engineering asset nowadays mostly relies on self-maintained experiential rule-bases and periodic maintenance, which is lacking a collaborative engineering approach. This research proposes a collaborative environment integrated by a service center with domain expertise such as diagnosis, prognosis and asset operations. The collaborative maintenance chain combines asset operation sites, service center (i.e., maintenance operation coordinator), system provider, first tier collaborators, and maintenance part suppliers. Meanwhile, to realize the automation of communication and negotiation among organizations, multi-agent system (MAS) technique is applied to enhance the entire service level. During the MAS design processes, this research combines Prometheus MAS modeling approach with Petri-net modeling methodology and unified modeling language (UML) to visualize and rationalize the design processes of MAS. The major contributions of this research include developing a Petri-net enabled Prometheus

* Please send all correspondence to Professor Amy Trappey, Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei (10608), Taiwan, E-mail: trappey@ntut.edu.tw. Tel: +886-2-2771-2171 Ext. 4541; Fax: +886-2-2776-3996

MAS modeling methodology and constructing a collaborative agent-based maintenance chain framework for integrated engineering asset management.

Keywords: Engineering Asset Management, Maintenance Chain Integration, Multi-Agent System, Petri-Net, Unified Modeling Language.

1. Introduction

Integrated Engineering Asset Management (IEAM) is a continuous process covering the whole life cycle of an asset containing conceptual design, construction/manufacture, operational use, maintenance, rehabilitation and/or disposal [5]. The purpose of IEAM is to make the best use of assets considering all possible situations in the entire life cycle and combine with the enterprise resources to maximize the total enterprise benefits. Engineering assets contain public facilities, manufacturing machineries and service facilities which are high-valued and require efforts from multiple organizations to construct, install and maintain. For IEAM, extending the asset operational time is always one of the key issues. To extend operational life of an engineering asset, capabilities and resources, such as monitoring techniques, data transformation and transmission, database technology, diagnosis and prognosis expertise, maintenance manpower allocation and maintenance part preparation, must be considered [8] [11] [13] [19]. However, these capabilities and resources are often distributed among different organizations. Therefore, the ability to integrate and harmonize the decision process is critical for maintenance efficiency.

To date, significant research achievements have been made in asset symptom diagnosis and asset health prognosis based on the asset condition monitoring data for preventive maintenance actions. In these two research fields, many complex reasoning methods, such as artificial neural network, fuzzy logic, expert system, and statistical methodologies, have been developed [21] [30] [32]. However, these researches only provide independent systems of

diagnosis or prognosis and are loosely related to real asset operations. To connect the condition-monitored asset with diagnosis and prognosis systems, MAS is considered as the most appropriate owing to its characteristics of being autonomous, communicative, goal-oriented, proactive, rational, learning and active. These features enable distributed systems to coordinate one another to generate decisions for maintenance or renewal schedules [16]. However, current agent-based maintenance studies only focus on the supports of diagnosis and prognosis. These studies do not help the integration among maintenance demanders and suppliers to increase the entire maintenance chain efficiency.

In the proposed maintenance chain, maintenance demanders operate monitored engineering assets and share transformed data with maintenance coordinator. The maintenance coordinator is devoted to engineering asset diagnosis and prognosis to provide scientific references for maintenance and integrates maintenance information as the basis of future design improvement. Moreover, the maintenance coordinator takes charge of integrating and coordinating maintenance resources to provide timely and reliable maintenance services. Meanwhile, maintenance providers co-operate one another to offer sufficient cross-enterprise human resources and maintenance parts according to schedule to accomplish maintenance jobs. Furthermore, in order to remove the constraints from physical boundaries and offer collaborative environment for the integrated maintenance chain, agent technology is applied to smooth the communication and negotiation among organizations.

In the following sections, relevant research literature is reviewed. The current practice with case description is presented to formulate on-going challenges. Then, the architecture of agent-based maintenance chain integration with modified MAS modeling methodology is proposed. Finally, the comparison of MAS modeling methodologies and the evaluation of proposed maintenance chain integration are made to demonstrate the benefits brought from this research.

2. Literature Review

This literature review aims to discover the research development trends of engineering asset management, multi-agent technology and applications, as well as the MAS modeling methodology.

2.1. Engineering asset management

The development trend of engineering asset management are summarized by Judd et al. [18], McArthur et al. [23], and Han and Yang [13]. Before 1950, the prevailing engineering asset management action was shutdown-driven maintenance, which can potentially cause human injuries and financial losses. Moreover, engineering assets operation data were not carefully recorded to provide engineers with sufficient and reliable information for repairing. From year 1950 to 1975, routine maintenance was applied to ensure sustaining operation of engineering assets. During this period, engineers started to collect the operational data and failure data of the assets. After 1975, periodic maintenance of engineering assets based on historical data was proposed. However, periodic maintenance is not able to deal with emergent issues, which potentially lead to unpredictable accidents and losses. From 1998, enterprises applied continuous condition monitoring techniques to record real-time asset condition in order to realize the concept of predictive maintenance. Thus, asset maintenance providers can have a comprehensive view of asset health based on tracking operating parameters. With the prognosis analysis based on operating data, the accurate predictive maintenance or renewal schedules can be achieved [23].

With the improvement of information technology and condition monitoring techniques, it becomes affordable for enterprises to record operating data and detect health condition about an asset. The data generation rate is highly accelerated due to database technique advancement. Therefore, experts are no longer able to interpret all the recorded data and

propose predictive maintenance schedule manually.

In the condition monitoring field, condition monitoring systems for individual components are considered as distributed systems. Each system contains several detected parameters, such as vibration, temperature, pressure and voltage [15]. Though these systems can provide more information, they still lack the ability to integrate. These legacy systems may be written in different programming languages and the information offered by them are usually not identically formatted. Therefore, MAS technology based on JAVA and combined with web service technique is proposed to solve this problem. In the MAS, condition monitoring agent (CMA) takes charge of monitoring the corresponding component and formats the generated data into pre-defined format. Afterwards, maintenance engineers can make use of the extracted and formatted data as input values to their pre-developed diagnostic models to generate diagnostic results [7] [22].

2.2. Agent technology and engineering asset management

Now that engineering assets provide more functions than ever, more uncertainties exist within every stages of engineering asset management processes. Among them, how to extend the operational life of assets is always one of the most concerned issues. Therefore, how to effectively and efficiently arrange routine, emergency and predictive/preventive maintenance based on existing knowledge and asset condition data becomes a critical topic. According to field research in the United States, maintenance cost occupies 15% to 40% of the entire manufacturing cost, which contributes to 200 billion dollars every year [19]. Consequently, effective predictive maintenance may contribute to a large amount of cost saving. Bangemann et al. [1] and Han and Yang [13] pointed out that MAS can help integrate the required sub-systems, e.g., condition monitoring, prognosis and diagnosis expertise, maintenance schedule coordination, crew and materials allocation for predictive maintenance. MAS can also resolve conflict during the planning and coordinating process [2].

Initial engineering asset maintenance framework focuses on real-time condition monitoring techniques to provide data for rule-based diagnosis to arrange timely maintenance schedules. Bretthauer et al. [3] proposed an integrated maintenance scheduling (IMS) system, which separates scheduling management into two sections, i.e., device-specific level and system-specific level. Fu et al. [11] stated that traditional maintenance concepts and broadly-applied maintenance scheduling systems are no longer suitable to deal with the ever-changing engineering environment. They propose an artificial neural network (ANN) prediction model as the system kernel.

The linkage between sub-systems dictates the flexibility of maintenance decision-making. Li et al. [19] proposed an agent-based maintenance framework linking monitoring agent (MA), diagnostic agent (DA), prognostic agent (PA) and maintenance decision-making agent (MDMA). In specific domain, Davis [8] proposed a multiple-agent decision support system (MADSS) for water supply infrastructure rehabilitation and development. From the literature, the most recent research on agent-based engineering asset management is mainly focused on applying agent technology to integrate required functions and resources within enterprise in order to increase maintenance efficiency. However, enterprises nowadays start to focus on their core competences and outsource maintenance works to their collaborators. Therefore, the functions and resources required for predictive/preventive maintenance, including monitoring techniques, data transformation and transmission techniques, database technology, diagnosis and prognosis expertise, maintenance crew allocation and maintenance part preparation, are often distributed among different organizations. As a result, this research develops a formal MAS methodology to form a collaborative environment for the proposed integrated maintenance chain.

2.3. Multi-agent system modeling methodology

A number of MAS methodologies and applications have been developed in recent years

[16] [20] [26] [29]. This section will discuss current multi-agent system modeling methodologies, including Gaia, Role Oriented Analysis and Design for Multi-Agent Programming (ROADMAP), Tropos, Multiagent Systems Engineering (MaSE), Process for Agent Societies Specification and Implementation (PASSI), and Prometheus approaches. This discussion intends to crystallize their characteristics. Based on these characteristics, this research integrates their advantages to develop a modified hybrid MAS modeling methodology for developing the integrated maintenance chain agent system.

The Gaia methodology [31] [33] is based on a well-founded organizational metaphor and exploits in a clean and rational way a suitable set of organizational abstractions. The Gaia methodology contains some major phases, i.e., collection of requirements, analysis phase, architectural design phase, detailed design phase, and implementation phase.

ROADMAP [17] extends the Gaia methodology for a complex open system. To improve Gaia methodology, ROADMAP points out the directions for further improvement : (1) support for requirements gathering; (2) explicit models to describe the domain knowledge and the execution environment; (3) levels of abstraction during the analysis phase, to allow iterative decomposition of the system; (4) explicit models and representation of social aspects and individual agent characteristics, from the analysis phase to final implementation; and (5) runtime reflection, modeling mechanisms to reason and change the social aspects and the individual agent characteristics at runtime.

The Tropos methodology [4] [12] is based on the idea of using requirements modeling concepts to build a model of the system-to-be within its operational environment. The methodology of Tropos contains five major steps, including early requirements, late requirements, architectural design, detailed design and implementation.

The MaSE methodology [9] [10] was designed to be used to analyze, design and

implement MAS by proceeding in an orderly fashion through the development lifecycle. MaSE has been automated via an analysis and design environment called agentTool which supports MaSE and helps guide the system designer through a series of models, from high-level goal definition to automatic verification, semi-automated design generation, and finally to code generation. There are seven steps in MaSE methodology, including capturing goals, applying use cases, refining roles, creating agent classes, constructing conversations, assembling agents and system deployment.

The PASSI approach [6] [14] is a step-by-step requirement-to-code methodology for designing and developing multi-agent societies, integrating design models and concepts from both object-oriented software engineering and artificial intelligence approaches using the UML notation. The models and phases of PASSI encompass representation of system requirements, social viewpoint, solution architecture, code production and reuse, and deployment configuration supporting mobility of agents. Detailed phases contain system requirement model, agent society model, agent implementation model, code model and deployment model.

The Prometheus method [24] [25] aims to provide a practical introduction to building intelligent agent systems. Prometheus methodology includes an introduction to the notion of agents, a description of the concepts and a software engineering methodology covering specification, analysis, design and implementation of agent systems.

These six methodologies provide formal guidelines for designing MASs. The procedures of using these methodologies can be summarized into three steps.

(1) Requirement analysis: Goal-oriented, behavior-oriented and organization-oriented approaches are the three common ways to determine the system requirement.

(2) Initial role identification: To fulfill the system requirement, the roles (agents) are required

to help human users for more convenience.

(3) Conversation design: To ensure the agent interactions can help to achieve the set goals, the detailed conversation designs among agents are required.

The above six MAS modeling methodologies provide good references for engineers to design MAS using systematic approaches. The agent identification should be consistent with the business processes. For this requirement, Tropos design process provides the best MAS design guidelines based on business processes. Furthermore, in determining the system requirement with the agent roles, some methodologies apply waterfall methodology and some in iterative mode. Waterfall methodology is relatively straightforward but it is less flexible in adding new components into the models once they are set up. However, iterative method can be used in dealing with this problem. With the discussion of scenarios, the system requirements are refined iteratively till no space for further improvement. For this requirement, Prometheus provides the best iterative way of determining the system requirements. Since iterative requirement specification and business process oriented agent role design are both important, we should consider both simultaneously while designing MAS. So far, no individual methodology considers these two aspects simultaneously. Therefore, this research will combine Prometheus MAS designing processes with Petri-net [27] based business process descriptions in order to deal with both concerns.

3. Problems Identification: A Case Study

A case study is used to demonstrate the proposed new business model and the adoption of MAS architecture. The case study focuses on the maintenance chain integration of power transformers, especially the large-scale step-up transformer in the power transmission and distribution network. Figure 1 show the image and drawing of a large-scale transformer. According to field research and interviews with participants in the energy sector, the current

process in periodic maintenance and repair of engineering asset is shown in Figure 2. The trend of transformer maintenance chain is evolving in recent years. In the past, the transformer system provider takes charge of the equipment production, installation and after-sales maintenance. With the growth of market share, the system provider can no longer supply all human resources and maintenance parts in house. Consequently, the system provider focuses on research and development of transformers and outsources maintenance tasks to first-tier collaborators. This new practice causes some concerns in the maintenance chain deficiency due to the lack of predictive maintenance procedure, inefficient maintenance resource allocation, ill-management of spare parts, and disjointed multiple systems. Details of these four concerns are to be addressed in the new MAS collaborative maintenance chain model and its supporting information platform developed in this research.

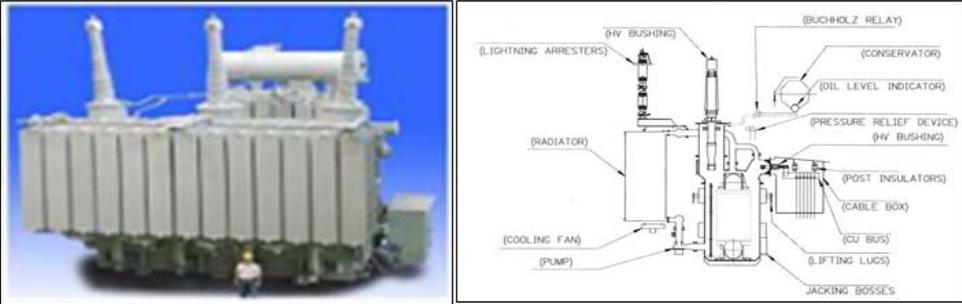


Figure 1. The structure drawing of a large-scale voltage transformer

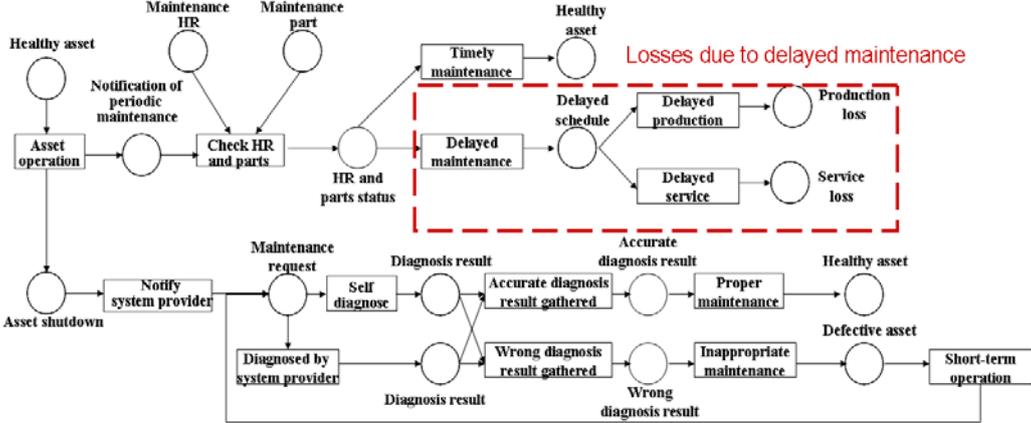


Figure 2. As-is process model

4. Agent-based Maintenance Chain Integration and System Design

This research proposes a new agent-based collaborative maintenance chain which is integrated by a service center with prognosis and diagnosis as well as operational expertise (as shown in Figure 3).

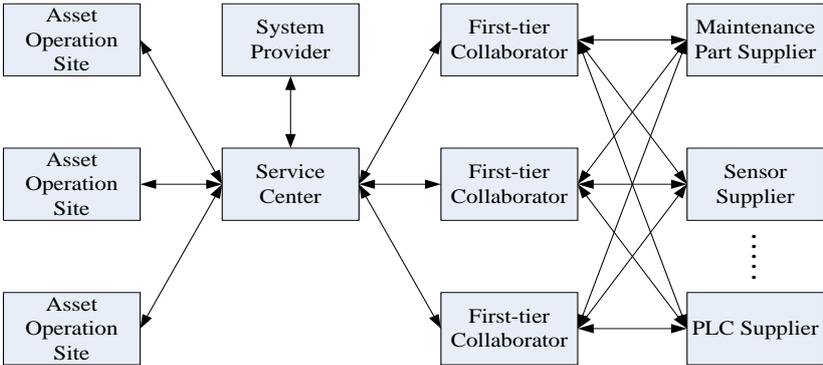


Figure 3. The proposed agent-based maintenance chain

In this chain, maintenance demanders operate monitored engineering assets and share their condition data with maintenance coordinator. The maintenance coordinator is devoted to the researches of engineering asset diagnosis and prognosis to provide good references for maintenance and integrates maintenance information as the basis of future design improvement. The maintenance coordinator takes charge of integrating and coordinating maintenance resources to provide timely and reliable maintenance services. Meanwhile, maintenance providers cooperate with one another to offer sufficient cross-enterprise human resources and maintenance parts for the scheduled maintenance jobs. In order to minimize the constraints from physical boundaries and offer collaborative environment for the integrated maintenance chain, agent technology is applied to enable the communication and negotiation among organizations. The following sections provide detailed design processes for the integrated maintenance chain MAS.

4.1. Requirement analysis

In the requirement analysis phase, the expected goals and how to achieve these goals are

summarized (Table 1). Further, the use case diagram (Figure 4) is defined to describe the initial relationship between human users and the system. In this diagram, there are six kinds of system participants, namely maintenance provider, maintenance part supplier, condition monitor, production/service manager, diagnosis expert and prognosis expert. Moreover, there are seven function modules containing maintenance schedule coordination, maintenance dispatching, maintenance part inventory, condition monitoring, production/service scheduling, production/service dispatching, diagnosis module and prognosis module.

Table 1. The goals of the proposed to-be maintenance chain

Goals	How to achieve these goals (sub-goals)
Data/signal extraction	<ol style="list-style-type: none"> 1. Condition monitoring system 2. Data/signal transformation
Accurate diagnosis	<ol style="list-style-type: none"> 1. Collect knowledge from diagnosis experts 2. Integrate expert knowledge into diagnosis knowledge base 3. Provide diagnosis results
Reliable prognosis	<ol style="list-style-type: none"> 1. Collect knowledge from prognosis experts 2. Integrate expert knowledge into prognosis knowledge base 3. Provide prognosis results
Timely and reliable maintenance	<ol style="list-style-type: none"> 1. Arrange maintenance provider 2. Provide expected maintenance time 3. Provide expected maintenance start time 4. Have enough maintenance resources
Personalized interface	<ol style="list-style-type: none"> 1. Provide personalized work list
Inventory management	<ol style="list-style-type: none"> 1. Notification of procurement

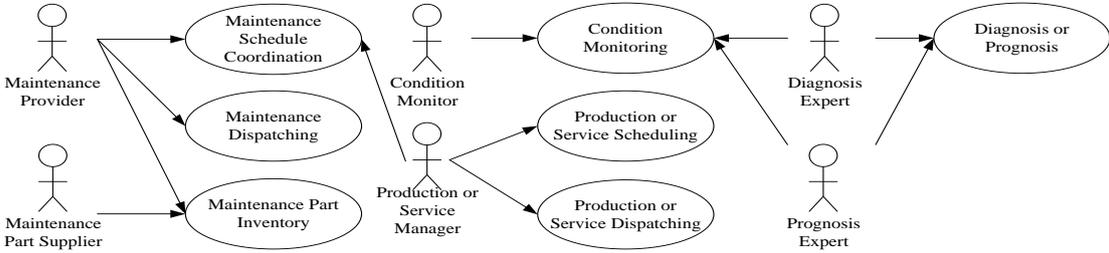


Figure 4. The use case diagram of the agent-based integrated maintenance chain

4.2. Architectural design

This phase is to determine the system architecture based on the system requirement analysis. Firstly, agents are defined based on the use case diagram. The goals and authorities of different agents are defined to mark their values to human users. After that, scenarios are applied to iteratively check the completeness of agent type design. Meanwhile, the agent relationships of different scenarios are generated. The data and agents from different organizations can be summarized to form the system framework.

4.2.1. Agent identification

After defining the expected functions and users of the system, the agent types required in the system can be identified. The agent definition process is combined with agent relationship definition (i.e., scenario identification) in the next section to complete the design of agent types. Table 2 shows the agent types and their descriptions.

Table 2. Agent types and corresponding descriptions

Organization	Agent type	Description
Asset operation site	Monitoring agent (MA)	<ol style="list-style-type: none"> 1. Real-time monitoring the parameters of engineering assets 2. Notify asset agent abnormal signals 3. Share information with service center
	Asset agent (AA)	<ol style="list-style-type: none"> 1. Interface agent of engineering asset 2. Receive notification from MA 3. Proceed initial diagnosis 4. Send diagnosis request 5. Send prognosis request 6. Receive diagnosis/prognosis results 7. Provide information to service system agent for maintenance schedule coordination 8. Send proposed maintenance schedule from service center to PSA and MSA
	Production or service agent (PSA)	<ol style="list-style-type: none"> 1. Arrange production or service schedule
	Dispatching service agent (DSA)	<ol style="list-style-type: none"> 1. Allocate human resource based on schedule from PSA
	Maintenance scheduling agent (MSA)	<ol style="list-style-type: none"> 1. Arrange maintenance schedule

Service center	Service system agent (SSA)	<ol style="list-style-type: none"> 1. Interface agent of service center 2. interface agent of the integrated maintenance chain 3. Receive the diagnosis and prognosis request from engineering asset site 4. Coordinate maintenance schedule between maintenance demanders and suppliers 5. Send diagnosis request to diagnosis agent 6. Send prognosis request to prognosis agent
	Diagnosis agent (DA)	<ol style="list-style-type: none"> 1. Receive diagnosis request from service system agent 2. Proceed diagnosis 3. Communicate with MA if more information required 4. Send diagnosis result to service system agent
	Prognosis agent (PA)	<ol style="list-style-type: none"> 1. Receive prognosis request from service system agent 2. Proceed prognosis 3. Communicate with MA if more information required 4. Send prognosis result to service system agent
First-tier collaborator	Maintenance decision support agent (MDSA)	<ol style="list-style-type: none"> 1. Interface agent of system provider 2. Provide information to service system agent for maintenance schedule coordination 3. Send request to MPSA, HRA and MPA 4. Receive proposed maintenance schedule 5. Send proposed maintenance schedule to MPSA, HRA and MPA
	Maintenance provider scheduling agent (MPSA)	<ol style="list-style-type: none"> 1. Arrange maintenance schedule 2. Response the request from MDSA
	Human resource agent (HRA)	<ol style="list-style-type: none"> 1. Allocate human resource 2. Response the request from MDSA
	Maintenance part agent (MPA)	<ol style="list-style-type: none"> 1. Check inventory level 2. Response the request from MDSA 3. Send replenishment request to Supplier interface agent
Maintenance part supplier	Supplier interface agent (SIA)	<ol style="list-style-type: none"> 1. Interface agent of supplier
	Inventory agent (IA)	<ol style="list-style-type: none"> 1. Check inventory level 2. Response the request from SIA 3. Send request to PLA
	Production line agent (PLA)	<ol style="list-style-type: none"> 1. Check production status to response time for replenishment

4.2.2. Agent relationship

This section applies different scenarios to check the completeness of agent type design. The agent relationship and the agent definition are checked iteratively to ensure the proposed performance of system. The information generated also contributes to the update of system requirement analysis.

- Condition monitoring: Monitoring agent (MA) takes charge of continuously monitoring the parameters of the engineering assets.
- Request for diagnosis or prognosis: Upon receiving abnormal signals, MA automatically notifies AA for initial inspection. If detailed diagnosis or prognosis is required, AA sends requests to SSA.
- Diagnosis: If diagnosis is requested to understand the potential symptoms of an engineering asset, SSA passes the request to DA for detailed diagnosis. During the diagnosis processes, more information may be required. Consequently, DA and MA would communicate each other to obtain enough information for more accurate diagnosis. After diagnosis, the results are delivered to SSA and PA.
- Prognosis: Similar to diagnosis processes, the prognosis agent is connected to SSA, MA and DA to generate reliable prognosis results to determine the asset health for preventive maintenance.
- Maintenance schedule coordination: After the generation of diagnosis results or prognosis results, SSA sends the report to AA. If AA sends the request of maintenance, SSA starts to coordinate suitable maintenance schedule for both maintenance demander and supplier. In determining the suitable maintenance schedule, SSA may turn to AA from maintenance demander and MDSA from maintenance supplier for more information request. After the schedule is determined, SSA sends related information to both sides and updates their maintenance schedules.

- Maintenance schedule update: After receiving the maintenance schedule, MDSA updates the information for MPSA for the arrangement of maintenance jobs.
- Resource allocation: Based on the coordinated maintenance schedule, MPSA starts to prepare and allocate resources required by each maintenance job. The resources mainly contain human resources and maintenance parts.
- Inventory management: MDSA cooperates with SIA for replenishment and request of time for delivery. With their cooperation, MDSA can have better view about the logistics information for determining the maintenance schedules.

With the agent relationship formally defined, the system architecture containing four organizations can be determined (Figure 5). In the maintenance demander site, there are six data vaults involved, containing condition monitoring, production/service schedule, finance, human resource, maintenance schedule, and material inventory. In the service center site, there are two kinds of databases, i.e., diagnosis rules and prognosis rules. In the maintenance provider site, maintenance schedule, human resource and maintenance part database are involved. In the supplier site, inventory and production schedule are considered.

4.3. Detailed design

This section designs detailed blueprints of agent conversations for the agent system implementation. From integrating individual agent relationship diagrams, the overall agent relationship is generated. In addition, the initial conversation protocols based on the agent communication language are defined to generate the agent overview (Figure 6).

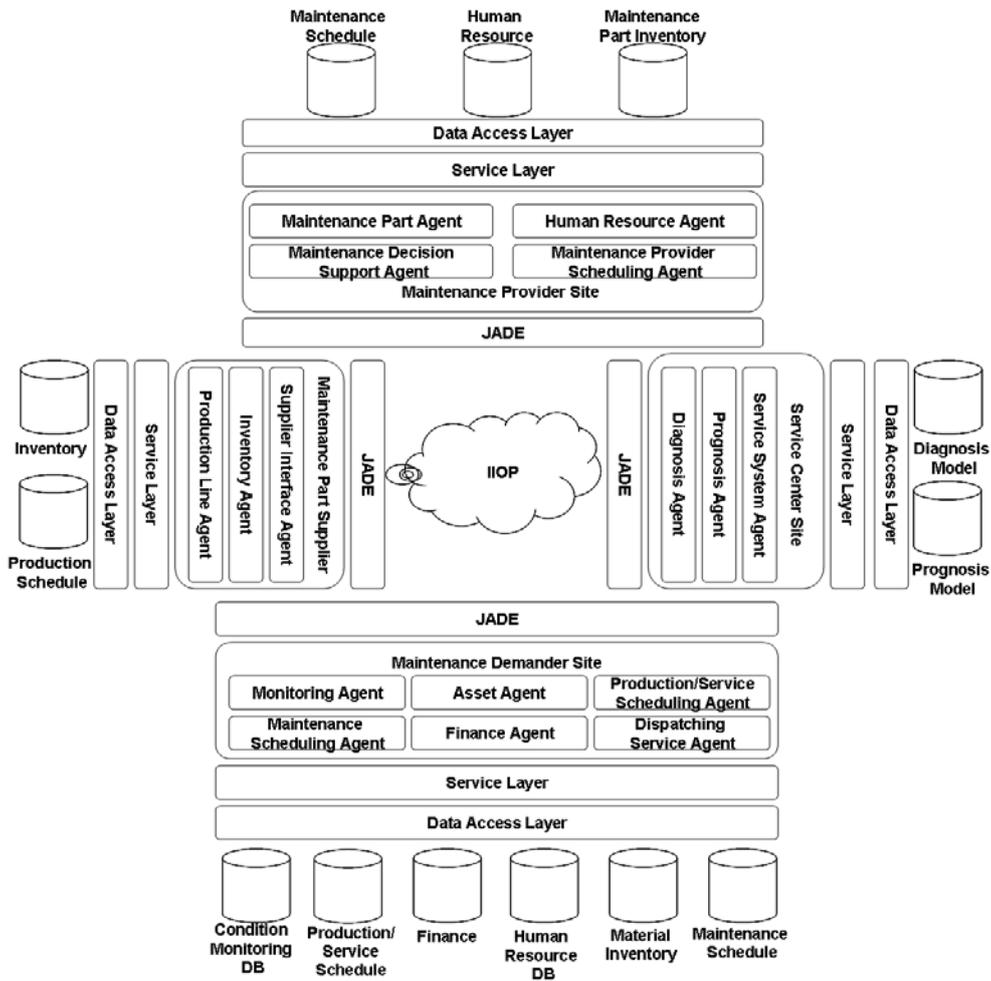


Figure 5. The system architecture of to-be maintenance chain

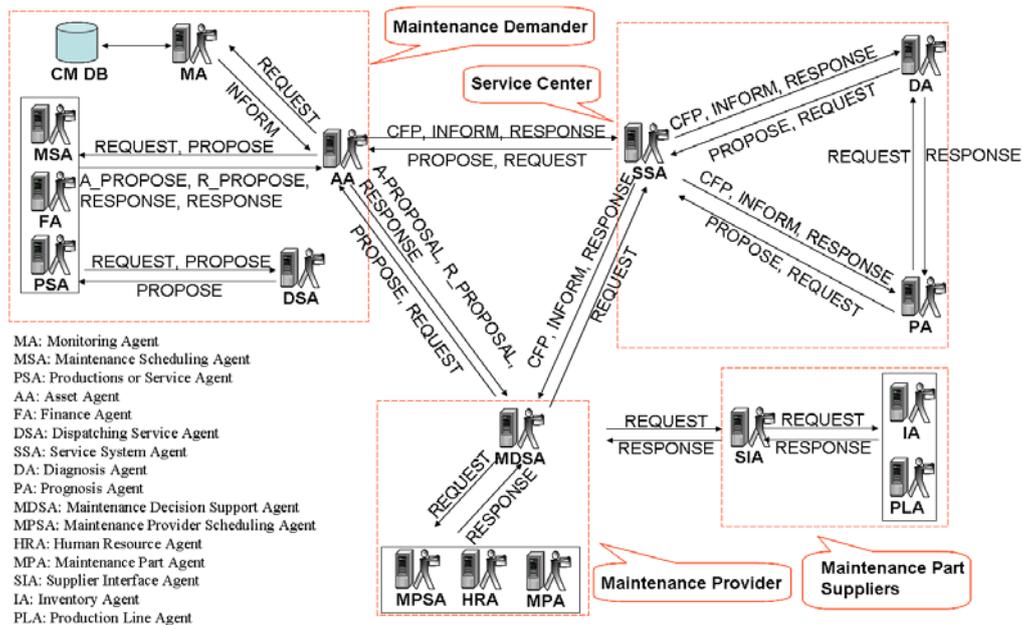


Figure 6. Agent overview diagram of the new maintenance chain

Different scenarios require different agents' involvement and interaction. The behavior diagrams for different processes show the agent interaction. After that, the UML sequence diagram is also defined for each process detail. During the MAS implementation, there are six major processes for the transformer maintenance chain integration, i.e., diagnosis, prognosis, maintenance schedule coordination, maintenance schedule update, resources allocation, and inventory management. Due to page limitation, we only demonstrate the detailed designs of “prognosis” and “maintenance schedule coordination” processes.

4.3.1. Prognosis

In prognosis scenario, MA continuously monitors the parameters of engineering assets, and notifies AA the trending of certain parameters which signify potential failure in the near future. After receiving the data, AA processes initial inspection to determine if maintenance is required. If it is, the request is sent to SSA. The PA is then asked for asset health prediction to provide references for further maintenance decision. Detailed agent communication logic is shown in Table 3. Figure 7 visualizes the description of the operation processes of prognosis. Figure 8 is the UML sequence diagram of the “prognosis” process.

Table 3. The agent interaction logic for “prognosis” scenario

Step	Description
1.	Monitoring agent (MA) continuously monitors engineering assets and saves the recording data into database.
2.	Once abnormal trending is detected, MA notifies asset agent (AA).
3.	AA processes initial prognosis. (1) If the trending dose not mean any potential failure in the near future, go back to step 1. (2) If self maintained rule base can process the prognosis, go to step 4. (3) Else, go to step 5.
4.	AA generates prognosis results.
5.	AA sends prognosis request to service center, and service center agent (SSA) passes the request to prognosis agent (PA).
6.	PA processes initial data identification.

	(1) If current data are capable of diagnosis, go to step 7. (2) Else, go to step 8.
7.	PA inputs the data into expert system to generate prognosis reports.
8.	PA communicates with MA to gather sufficient data for the expert system to generate prognosis reports.

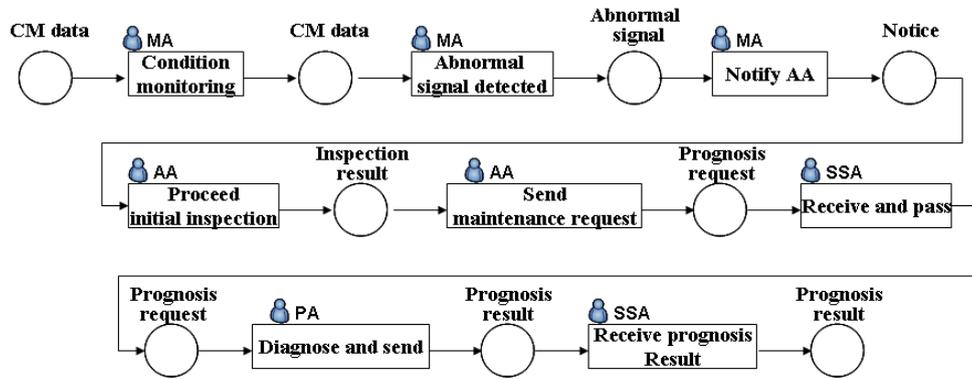


Figure 7. Petri-net based behavior diagram of prognosis scenario

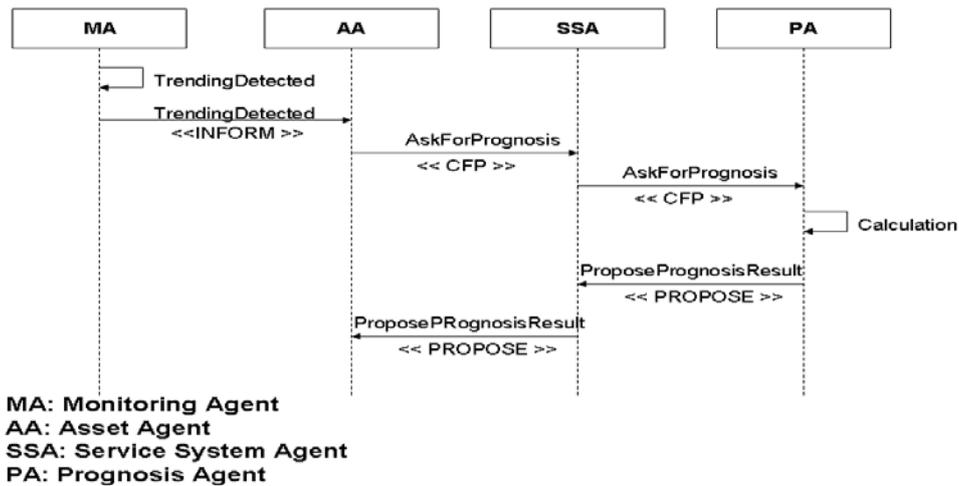


Figure 8. UML sequence diagram of prognosis scenario

4.3.2. Maintenance schedule coordination

In maintenance schedule coordination scenario, SSA arranges preliminary maintenance schedule for both sides while maintenance request is received. Then, the maintenance schedule is sent to maintenance suppliers and maintenance demanders to examine if the proposed schedule is suitable. It might be an iterative activity to determine the final

maintenance schedule. Detailed agent communication logic of the schedule coordination is shown in Table 4. Figure 9 visualizes the description of its operation processes and Figure 10 illustrates its UML sequence diagram.

Table 4. The agent interaction logic for “maintenance schedule coordination” scenario

Step	Description
1.	SSA receives maintenance request from AA.
2.	SSA starts to determine and re-allocate maintenance schedule. (1) If routine maintenance, first-in first-out policy is applied. (2) If emergent maintenance, the priority of maintenance determines the maintenance sequences. (3) If predictive maintenance, combining routine maintenance with predictive maintenance policy is applied.
3.	SSA sends the arranged maintenance schedule to both AA at maintenance demander site and Maintenance decision support agent (MDSA) at maintenance supplier site.
4.	AA and MDSA replies if they accept the arranged maintenance schedule. (1) If both sites accept, go to step 5. (2) If either site does not accept, the expected maintenance time should be provided and go to step 6.
5.	SSA sends final confirmation to both AA and MDSA.
6.	SSA reschedules the maintenance schedule based on the provided expected maintenance time, and go to step 3.

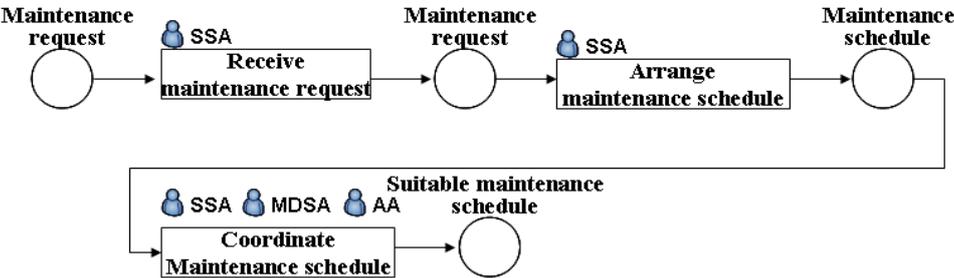


Figure 9. Petri-net based behavior diagram of maintenance schedule coordination scenario

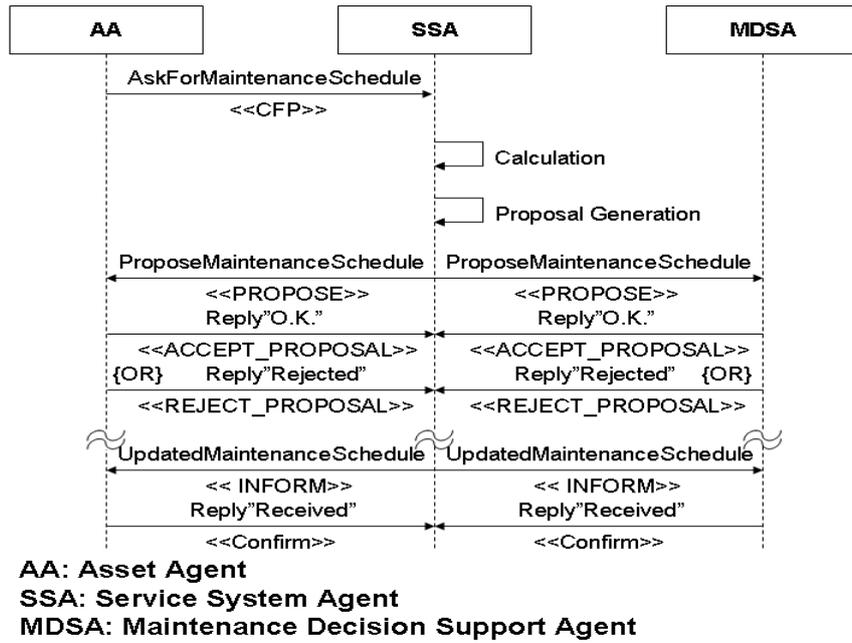


Figure 10. UML sequence diagram of maintenance schedule coordination scenario

5. Comparison and Evaluation Using Simulation

With detailed requirement analysis, agent relationship analysis and agent conversation design, the branches and leaves of the collaborative maintenance chain are carefully depicted. Afterward, this research applies the agent development named JADE, which follows the Foundation for Intelligent Physical Agents (FIPA) specifications and provides Graphical User Interface (GUI), to enable the development and debugging of the agent-based information system.

In order to demonstrate the methodology developed from this research, comparison of the original and proposed modified MAS modeling methodology is made through a simulation study based on the maintenance chain for power transformer case. The simulations runs of current and proposed to-be maintenance chain are also conducted to manifest the advantages of the new agent-based maintenance chain integration. Table 5 summarizes the differences between Prometheus and the proposed MAS modeling methodology in seven areas, i.e., current practice analysis, system requirement gathering, business process oriented,

development framework, operation process description, system modeling language, and verification.

Table 5. Comparison of Prometheus modeling methodology and the proposed modeling methodology

	Prometheus Method	This Research
Current practice analysis	Not applicable	Petri-net business process description is applied to detect bottlenecks for improvement
System requirement gathering	Scenarios are applied to iteratively check the completeness of system requirements	Scenarios are applied to iteratively check the completeness of system requirements
Business process oriented	Prometheus is more goal-oriented than behavior- or process-oriented	Process oriented agent design based on business processes
Development framework	Java Agent Compiler and Kernel (JACK) is the suggested development platform	No limitation about development platform
Operation process description	Prometheus does not describe the operation process since it is not business process oriented	Petri-net is applied to assist visualization of the business operation processes
System modeling language	Unified modeling language (UML)	UML
Verification	Prometheus does not have verification mechanism for to-be system	This research applies the business process descriptions based on Petri-net simulation

This research applied iterative requirement gathering via visualized description to bridge the domain expertise of managers and system engineers. In order to keep the flexibility of the new information system, this research provides MAS design guidelines without limiting the agent system development framework. In addition, with petri-net process descriptions, simulation runs are more easily made to verify the quantitative differences between current and the new systems.

After system implementation, this research applied Income Suite [28] based on petri-net

process description to perform quantitative simulation runs to compare current maintenance chain and the new agent-based maintenance chain integration. Before the simulation runs, current practice and new practice models are depicted to show the differences. In the current practice, the functions and resources required for maintenance are not well integrated, which potentially leads to production loss and service loss. Moreover, the ability to realize predictive maintenance is not sufficient. Therefore, the percentage of shutdown driven maintenance is increased and eventually causes even more severe losses and injuries. In the new practice, functions and resources required for periodic maintenance are collaboratively determined in advance to ensure sufficient and timely provision for maintenance jobs. Moreover, the concept of preventive maintenance is realized via the expertise of diagnosis and prognosis and the integration of maintenance chain. Furthermore, predictive maintenances reduce the occurrence rate of shutdown driven maintenance, and eventually reduce the losses and injuries caused by emergent engineering asset shutdowns.

The simulation run scenarios for the power transformer case are shown in Table , and the major difference is adding service center in the new model. After the simulation runs, the results are summarized in Table. It can be seen that the days spent in handling the emergent electricity recovery is 2.58 days shortened which leads to US\$ 72,000 effective savings. In the current practice, different maintenance providers have their own maintenance schedules. Therefore, if a high level maintenance request arises, it is difficult to timely adjust and prepare required maintenance human resources and maintenance parts. However, in the new model, multiple maintenance providers are integrated to accomplish the maintenance demanders' maintenance requests. Via human resources integration, time for human resources and maintenance part preparation are shortened which leads to more efficient maintenance jobs. Consequently, 2.58 days are saved. Moreover, due to cross-enterprise cooperation, the utilization rate of human resources is increased by 24.3%. In the current practice, maintenance

provider has narrower view about cooperative partners' human resources. Therefore, the extra human resources of cooperative partners are not efficiently applied to help maintenance providers do the maintenance jobs. However, in the new practice, the human resource is better utilized since it can be allocated more efficiently due to enterprise cooperation.

For maintenance part preparation, current practice mainly depends on individual's estimation to prepare maintenance parts which leads to overstock or low service level. However, in the new practice, the service center provides a forum in advance to collaboratively discuss the chain requirement to determine the maintenance part preparation level. Based on the integrated maintenance part preparation level, the production schedule, the safety stock level and lead time can be determined. According to the simulation results, the inventory cost within one year reveals a saving of US\$ 3.6 million.

Table 6. The simulation runs of current and new models

	As-Is	To-Be
Maintenance demander	3	3
Transformer number	60	60
Service center	N/A	1
Maintenance supplier	4	4
Maintenance part supplier	3	3

Table 7. Simulation result comparison

	Current	New	Difference
Emergent electricity recovery (1 transformer)	6.13 days	3.55 days	2.58 days decreased
Loss due to emergent shutdown (1 transformer)	US\$ 170,000	US\$ 98,000	US\$ 72,000 decreased
Engineer utilization rate (1 year period)	51.00 %	75.30 %	24.3 % increased
Annual inventory cost 3 maintenance providers/year	US\$ 13 million	US\$10 million	US\$ 3 million saved

6. Conclusion

The purpose of this research is to provide a comprehensive collaborative maintenance chain architecture and realize the architecture via multi-agent techniques. Detailed agent relationship and agent communication models are described as the basic guidelines of further implementation. During the MAS design phase, this research employs Prometheus MAS modeling processes and combines the design process with Petri-net process and UML modeling approaches for detailed design. With the assistance of Petri-net behavior diagram and UML diagram, the MAS design are visualized and rationalized in a formal procedure. A simulation study is also developed based on the modeling of maintenance chain of power transformer system. The MAS architecture and design, proposed in this research, are developed using the transformers maintenance chain as reference. Nonetheless, the framework of the maintenance chain MAS is generic for adoptions to other industries. In order to modify the MAS for other industries, the formal MAS methodology, combining Prometheus, Petri-Net and UML, can be applied to adjust the MAS design and implementation.

There are four advantages of the proposed maintenance chain integration. First, the tedious maintenance coordination tasks are outsourced to first-tier collaborators so as to help enterprises keep their focuses on their core operations. Second, the service center acts as the coordinator of maintenance suppliers and demanders, which contributes to the chain efficiency and customer satisfaction. Third, the service center provides a forum for maintenance chain participants to communicate their requirements and determine the subsequent schedules and actions. Finally, the agents contribute to consistent communication and coordination among enterprises, which enables better capability for dealing emergent events and reduces physical boundary constraints.

Acknowledgement

This research is partially supported by the National Science Council research grant.

References

- [1] Bangemann, T., Rebeuf, X., Reboul, D., Schulze, A., Szymanski, J., Thomesse, J.P., Thron, M. and Zerhouni, N., 2006, "PROTEUS—Creating distributed maintenance systems through an integration platform," *Computers in Industry*, Vol. 57, Issue 6, pp. 539-551.
- [2] Barber, K.S.; Liu, T.H.; Ramaswamy, S, "Conflict detection during plan integration for multi-agent systems," *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 2001: 31(4), p. 616-628.
- [3] Bretthauer, G., Gamaleja, T., Handschin, E., Neumann, U. and Hoffmann, W., 1998, "Integrated maintenance scheduling system for electrical energy system," *IEEE Transactions on Power Delivery*, Vol. 13, No. 2, April, pp.655-660.
- [4] Castro, J., Kolp, M., and Mylopoulos, J., 2002, "Towards requirements-driven information systems engineering: The Tropos project," *Information Systems*, 27(6), pp. 365-389.
- [5] CIEAM (CRC for Integrated Engineering Asset Management), <http://www.cieam.com/>, Accessed 15/2/2009.
- [6] Cossentino, M., 2005, "From Requirements to Code with the PASSI Methodology," In: Henderson-Sellers, B. and Giorgini, P. (eds.), *Agent-Oriented Methodologies*. Idea Group Publishing.
- [7] Davidson, E.M., McArthur, S.D.J., McDonald, J.R., Cumming, T. and Watt, I., 2006, "Applying multi-agent system technology in practice: automated management and analysis of SCADA and digital fault recorder data," *IEEE Transactions on Power Systems*, Vol. 21, Issue2, pp. 559 – 567.

- [8] Davis, D.N., 2000, "Agent-based decision-support framework for water supply infrastructure rehabilitation and development," Computers, Environment and Urban Systems, Vol. 24, pp. 173-190.
- [9] DeLoach, S. and Kumar, M., 2005, "Multi-Agent Systems Engineering: An Overview and Case Study," In: Henderson-Sellers, B. and Giorgini, P. (eds.), *Agent-Oriented Methodologies*. Idea Group Publishing.
- [10] DeLoach, S.A., Wood, M.F. and Sparkman, C.H., 2001, "Multiagent systems engineering," *The International Journal of Software Engineering and Knowledge Engineering*, 11(3), pp.231-258.
- [11] Fu, C., Ye, L., Liu, Y., Yu, R., Iung, B., Cheng, Y. and Zeng, Y. 2004, "Predictive Maintenance in Intelligent Control Maintenance Management System for Hydroelectric Generating Unit," *IEEE transactions on energy conversion*, Vol. 19, No. 1, March, pp.1-8.
- [12] Giorgini, P., Kolp, M., Mylopoulos, J. and Castro, J., 2005, "Tropos: A Requirement-Driven Methodology for Agent-Oriented Software," In: Henderson-Sellers, B. and Giorgini, P. (eds.), *Agent-Oriented Methodologies*. Idea Group Publishing.
- [13] Han, T. and Yang, B.S., 2006, "Development of an e-maintenance system integrating advanced techniques," *Computers in Industry*, Vol. 57, Issue 6, pp. 569-580.
- [14] Henderson-Sellers, B., Debenham, J., Tran, Q.N.N., Cossentino, M. and Low, G., 2006, "Identification of Reusable Method Fragments from the PASSI Agent-Oriented Methodology," In: *Agent Oriented Information Systems III. Lecture Notes in Computer Science*, vol. 3529. Springer-Verlag GmbH. pp. 95-110.
- [15] Hossack, J.A., Menal, J., McArthur, S.D.J. and McDonald, J.R., 2003, "A multiagent architecture for protection engineering diagnostic assistance," *IEEE Transactions on Power Systems*, Vol. 18, Issue 2, pp. 639 – 647.
- [16] Huang, C.J., Trappey, A.J.C. and Yao, Y.H., 2006, "Developing an agent-based workflow

- management system for collaborative product design,” *Industrial Management and Data System*, Vol. 106, No. 5, pp. 680-699.
- [17] Juan, T. and Sterling, L., “The ROADMAP Meta-model for Intelligent Adaptive Multi-Agent Systems in Open Environments,” *Proceedings of the Fourth International Workshop on Agent Oriented Software Engineering*, at AAMAS'03, Melbourne, Australia, July 2003.
- [18] Judd, M.D., McArthur, S.D.J., McDonald, J.R. and Farish, O., 2002, “Intelligent condition monitoring and asset management. Partial discharge monitoring for power transformers,” *Power Engineering Journal*, Vol. 16, Issue 6, pp. 297- 304.
- [19] Li, Y., Chun, L. and Ching, A.N.Y., 2005, “An Agent-based Platform for Web-enabled Equipment Predictive Maintenance,” *Proceedings of the 2005 IEEE/WIC/ACM international Conference on Intelligent Agent Technology (IAT'05)*, pp.132-135.
- [20] Lin, Fu-ren; and Pai, Yu-Hua , “Using multi-agent simulation and learning to design new business processes,” *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 2000: 30(3), pp. 380-384
- [21] Majidian, A. and Saidi, M.H., 2007, “Comparison of fuzzy logic and neural network in life prediction of boiler tubes,” *International Journal of Fatigue*, Vol. 29, pp. 489-498.
- [22] McArthur, S.D.J., Booth, C.D., McDonald, J.R. and McFadyen, I.T., 2005, “An agent-based anomaly detection architecture for condition monitoring,” *IEEE Transactions on Power Systems*, Vol. 20, Issue 4, pp. 1675 – 1682.
- [23] McArthur, S.D.J., Strachan, S.M. and Jahn, G., 2004, “The design of a multi-agent transformer condition monitoring system,” *IEEE Transactions on Power Systems*, Vol. 19, Issue 4, pp. 1845 – 1852.
- [24] Padgham, L. and Winikoff, M., 2004, *Developing intelligent agent systems: A practical guide*. New York: John Wiley & Sons.
- [25] Padgham, L. and Winikoff, M., 2005, “Prometheus: A Practical Agent-Oriented

- Methodology,” In: Henderson-Sellers, B. and Giorgini, P. (eds.), *Agent-Oriented Methodologies*. Idea Group Publishing.
- [26] Pechoucek, M.; Rehak, M.; Charvat, P.; Vlcek, T.; Kolar, M, “Agent-Based Approach to Mass-Oriented Production Planning: Case Study,” *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 2007: 37(3), pp. 386-395
- [27] Petri-net, Wiki, http://en.wikipedia.org/wiki/Petri_net, Accessed 5/2/2009.
- [28] Promatis, “Income Suite: Modeling, simulation and visualization of business processes,” <http://www.promatis.com/cbusiness/products/income/index.htm>, Accessed 11/1/2009.
- [29] Santa-Eulalia, L.A., Frayret, J.-M., D’Amours, S., “Essay on conceptual modeling, analysis and illustration of agent-based simulations for distributed supply chain planning,” *INFOR*, 2008: 46(2), pp 97-116.
- [30] Sun, Y., Ma, L., Mathew, J., Wang, Y. and Zhang, S., 2006, “Mechanical systems hazard estimation using condition monitoring,” *Mechanical Systems and Signal Processing*, Vol. 20, Issue 5, pp. 1189-1201.
- [31] Wooldridge, M., Jennings, N. and Kinny, D., 2000, “The Gaia Methodology for Agent-Oriented Analysis and Design,” *Journal of Autonomous Agents and Multi-Agent Systems*, Vol. 3, pp. 285-312.
- [32] Yao, Y.H., Lin, G.Y.P. and Trappey, A.J.C., 2005, “Using knowledge-based intelligent reasoning to support dynamic equipment diagnosis and maintenance,” *International Journal of Enterprise Information Systems*, Vol. 2, No. 1, pp. 17-29.
- [33] Zambonelli, F., Jennings, N.R. and Wooldridge, M., 2003, “Developing multi-agent systems: The Gaia methodology,” *ACM Transactions on Software Engineering and Methodology*, 12(3), pp. 417-470.