Expert Systems with Applications 37 (2010) 393–400

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An agent-based workflow system for enterprise based on FIPA-OS framework

Yao Jung Yang [a], Tien-Wen Sung [a,b,*], Chuni Wu [a], Hsiang-Yang Chen [a]

[a] Department of Information Management, Hsing-Kuo University of Management, No. 89, Yuying Street, Tainan 709, Taiwan, ROC
[b] Department of Computer Science and Engineering, National Sun Yat-Sen University, No. 70, Lienhai Road, Kaohsiung 80424, Taiwan, ROC

A R T I C L E   I N F O

Keywords:
Multi-agent system
FIPA-OS
Workflow management

A B S T R A C T

The paper describes what we have learned from implementing a multi-agent architecture used to support enterprise notions and principles for intelligent systems design. In the last couple of years, agent-based management systems have been widely used to monitor and control business design processes. In this paper, multi-agent architecture is proposed and applied to the workflow management system in ASE[1] (Advance Semiconductor Electronic) Inc., which is the world's well-known largest provider of independent semiconductor for manufacturing services in assembly and test. The proposed FIPA-OS autonomous workflow management system uses a workflow co-ordination mechanism and an agent integration mechanism to enable the routine daily jobs error handle. Our vision for extending the FIPA-OS architectural elements to cover the development and implementation of generic web-Centric collaborative applications concludes the paper.

1. Introduction

Workflow management systems had a significant impact on organizations in the past decade (Cardoso, Sheth, Miller, Arnold, & Kochut, 2004), and are a promising technology that automates business processes to improve the efficiency and manageability of an enterprise's daily teamwork (Wang, Wang, & Xu, 2005). How to rapidly and cost-effectively satisfy the various dynamic demands of the market adapt to an unpredictable business environment and increasing rates of new production methods are the major challenges for manufacturing enterprises. Achieving the combination of internal and external flows on heterogeneous platforms in a distributed environment is an important research area (Wang, Shen, & Hao, 2006). ASE Inc. is the world's largest provider of independent semiconductor manufacturing services in assembly and test. As a global leader geared towards meeting the industry's ever growing needs for faster, smaller and higher performance chips, ASE Group develops and offers a wide portfolio of technology and solutions, including IC test program design, front-end engineering testing, wafer probe, wafer bump, substrate design and supply, wafer level packaging, flip chip, system-in-package, final testing and design manufacturing services through Universal Scientific Industrial Co. Ltd. In this paper, a distributed multi-agent architecture is proposed and applied to the workflow management system in ASE. The remainder of this paper is organized as follows: in Section 2, we review the related literature in the areas of agent technology, workflow management and the FIPA-OS platform. Section 3 illustrates our autonomous workflow management system and agent communication model using the FIPA-OS framework. In Section 4, we describe what we have learned from implementing a multi-agent application used to support enterprise workflow management system. Finally, we summarize the contributions of this paper and provide suggestions for future work.

2. Related work

2.1. Agent-based workflow management and agent technology

Fox (1981) sees an organizational structure for a distributed system as the collection of processes (i.e. agents), communication paths and a control regime that coordinates the whole. An agent is a software entity that can autonomously perform routine tasks with a degree of intelligence (Boudriga & Obaidat, 2004). In the last decade, many agent-based workflow management system and agent technology have been developed and proposed. For example, Chen and Chen's (2008) work on designing and developing a multi-agent system focuses on the interaction between agents. Mihaela, Norrie, Kremer, and Shen (2000) and Wang et al. (2006) apply web service technology to develop an agent-based workflow prototype for inter-enterprise collaboration. Although their prototype provides a basic framework, it lacks the flexibility and scalability to satisfy complex industrial processes. Charles, Trappey, Huang, and Ku (2009) apply a JADE-based autonomous workflow management

* Corresponding author. Address: Department of Information Management, Hsing-Kuo University of Management, No. 89, Yuying Street, Tainan 709, Taiwan. Tel.: +886 6 2870105; fax: +886 6 2870093.
E-mail addresses: danny.yang@mail.hku.edu.tw (Y.J. Yang), kevin@mail.hku.edu.tw (T.-W. Sung).

[1] http://www.aseglobal.com.

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.05.049

system for collaborative SoC design, and Madhusudan (2005) applies embedded autonomous agents in a workflow-based distributed systems infrastructure to support design activities in an industrial context with a case study. Huang, Trappey, and Yao (2006) develop a prototype of a agent-based intelligent workflow system for product design collaboration in a distributed network environment. Wang et al. (2006) design a multi-agent system and application for inter-enterprise collaboration, while Karacapilidis, Lazanas, Megalokonomos, and Moraïtis (2006) propose an intelligent web-based multi-agent system for transportation services. Liang and Huang (2002) proposes an agent-based system of collaborative information and a solution procedure for designing with modules to develop modular products to meet a customer's requirements. Lee and Kwok (2000) proposes an architecture, which consists of user agents, information management agent and a fuzzy model manager to apply in a marketing decision process. From the viewpoint of system development, applications such as these should be developed under a free, well-known agent platform to shorten the project development lifecycle, and scalability also is an important concern due to the increased size of the related organizations. In order to address these concerns, this paper develops a distributed multi-agent workflow system with a well-defined ontology for handling daily jobs in ASE Inc., and also uses a free platform, FIPA-OS, for development.

2.2. The agent platform

There are three important agent standardization efforts which are attempting to support interoperability between agents on different types of platform: KQML (Knowledge Query Meta Language) (Finin, Labrou, & Mayfield, 1997) community, OMG's[2] (Object Management Group), MASIF (Mobile Agent System Interoperability Facility) and FIPA[3] (Foundation for Intelligent Agent). KQML and FIPA both define interaction in terms of an Agent Communication Language (ACL), whereas MASIF defines interaction in terms of Remote Procedure Calls (RPC) or Remote Method Invocation (RMI). In contrast to the traditional RPC-based paradigm, the ACL as defined by FIPA is an attempt at a universal message-oriented communication language which provides a flexible approach for communication between software entities.

KQML is not true de facto standard in the sense that there is no consensus in the community on a single set of specifications, and it has not yet been ratified by common agreement between members of an organization and with some standing in the community. As a result, variations of KQML exist, and different agent systems which speak different dialects may not be able to interoperate fully.

The OMG's MASIF differs from both KQML and FIPA in that it does not support or standardize interoperability between non-mobile agents on different agent platforms. Further, MASIF restricts the interoperability of agents to these developed on CORBA platforms whereas the focus of FIPA is to directly support the interoperability of agents deployed on agent frameworks which can support heterogeneous transport.

FIPA is a non-profit standards organization established in 1996 and registered in Geneva Switzerland. The FIPA agent standards (O'Brien & Nicol, 1998) aim to bring the commercial world a step closer to true software components, and the benefits of this will include increased re-use together with ease of upgrade. FIPA allows for focused collaboration between both industrial and academic organizations in addressing the key challenges facing commercial agent developers as they turn agent technology into products. The FIPA agent reference model (Fig. 1) currently provides the normative framework within which FIPA agents exist and operate. The Directory Facilitator (DF), Agent Management System (AMS), Agent Communication Channel (ACC) and Internal Platform Message Transport (IPMT) form what are termed the provides white-page service and lifecycle management services for agents and the ACC supports inter-agent communication interoperability, both within and across different platforms. The IPMT provides a reliable and orderly message forwarding service for agents on a particular platform (O'Brien & Nicol, 1998). However, the FIPA ACL (Agent Communication Language) focuses on an internal agent's mental agency of beliefs, desires and intentions, and closure is not enforced (i.e. agents are not compelled to answer) – elements that may hinder multi-agent co-ordination (Charlton, Cattoni, Potrich, & Mamdani, 2000). The multi-agent system (MAS) field is currently a very active research area, and there are now several MAS platforms which report their support for the FIPA agent standards, includeing JADE (Bellifemine, Rimassa, & Poggi, 1999), Grasshopper (Breugst, Hagen, & Magedanz, 1998) and ZEUS (Nwana, Ndumu, Lee, & Collis, 1999). Of these only FIPA-OS and ZEUS are freely available under a general public license and are released with their source-code under an open-source license.

2.3. FIPA-OS agent framework

FIPA-OS is designed to support the FIPA agent standards. The FIPA reference model discussed earlier defines the core components of the FIPA-OS distribution. The FIPA-OS agent framework provides an agent software framework and brings interoperability within and across agent based applications. It employs intelligent agents and the implementations are based on Java and other Internet technologies (XML, SMTP, Jini, Active Objects), elements which have already become regular ''ingredients" in agent applications. The interoperability involves relationships between agents, between agents and platforms, and between implementations of agent services. The FIPA-OS agent framework shown in Fig. 2, proposes the concept of an agent platform (AP) which is built on top of a distributed computing environment, thus integrating the client/server paradigm with agent technology. The AP consists of three basic services: the Agent Management System (AMS), the Directory Facilitator (DF) and the Agent Communication Channel (ACC). When agents are registered with the AMS for a platform they are considered as part of that platform. The AMS is responsible for the management of operations on the agent platform as well as for the management of the agents themselves. The DF (an agent itself) works in a yellow pages manner and supports the localization of agents and the services they provide in the area of a domain or the whole environment. In order for the agents to be registered with a DF, they have to send a registration request to that service. The ACC uses information provided by AMS to enable agent communication between agents on a platform and between platforms by offering a message forwarding service. The high level of interoperability between platforms is made possible by placing a mandatory ACC agent on each agent system belonging to a FIPA compliant environment. FIPA-OS initially focuses on providing abstractions and interfaces (APIs) for developers who wish to extend, enhance and integrate an agent platform with existing software API and instead provides an integrated development environment to allow developers to configure the agent platform.

FIPA-OS is a component-oriented toolkit for constructing FIPA compliant agents using mandatory components, components with switchable implementations, and optional components. Fig. 3 highlights the available components and there relationships with each other. The Database Factory, Parser Factory and CCL (Choice Communication Language) components are optional and do not have an explicit relationship with the other components in the tool-kit.

[2] http://www.omg.org.
[3] http://www.fipa.org.