Multi-Agent Based Peer-to-Peer Workflow Management System

A. Aldeeb, K. Crockett and M. J. Stanton

Department of Computing and Mathematics,
Manchester Metropolitan University, Manchester, M1 5GD, UK
{a.aldeeb, k.crockett, m.stanton}@mmu.ac.uk

Abstract. Current workflow management systems are under dynamic evolution and tend to move to distributed business processes. Software agents and Peer-to-Peer (P2P) technology are being recognized as a new approach to decentralized workflow management systems. Plugging Multi-agents into P2P workflow management systems can support run-time functions, workflow instance monitoring, adaptability and exception handling. In this paper, a novel Multi-agent based P2P architecture for a workflow management system is presented. The proposed system can be applied for an inter-organizational workflow scenario to form a virtual organization. The adoption of multi-agent within P2P network facilitates the capability of workflow actors to share, collaborate, and communicate in their own private web.

Keywords: Workflow, Workflow Management System, Multi-agents, Peer-to-Peer, Exception Handling, Virtual Organization

1 Introduction

Workflow is the computerized facilitation or automation of a business process in whole or part [22]. A workflow management system (WFMS) is a system that completely defines, manages and executes “workflows” through the execution of software whose order of execution is driven by a computer representation of the workflow logic [22]. Conventional workflow management systems are based on centralized client/server architecture. This requires a centralized database to store the workflow process definition and a centralized workflow engine to manage activities such as coordination and monitoring process execution [3, 4]. The main disadvantages of any such architecture are the potential bottleneck that can arise during process execution, and that the central database can become single point of fault.

Workflow processes are moving from long-lasting, well-defined, centralized business processes to dynamically changing, distributed business processes with many variants [3, 4, 13, 16]. Workflow applications are inherently distributed. They involve people, resources and tools that may be distributed over a wide geographic area [16]. Recent research has shown an increased interest in P2P based workflow systems [1, 2, 4, 8] to decentralize workflow systems. P2P is a way to develop a
distributed applications at different nodes (Peers) to share resources and these nodes have symmetric roles either server or client. The P2P WFMSs are proposed to avoid the bottleneck and the central point of fault caused by centralized client/server workflow systems. P2P based workflow can also be used to improve scalability, system openness and support incompletely specified processes [4]. In P2P based WFMS peers join “virtual communities” according to their capabilities and discover each other using the services provided by an open P2P network. The coordination between peers is performed by exchange of notification messages.

In recent years, the integration of workflow and agent technology has attracted the attention of many researchers to support distributed business process in a dynamic and unpredictable environment [5, 6, 7, 14, 17, 18, 19, and 20]. Agents are persistent active entities that have the properties of autonomy, reactivity, and pro-activity and can perceive, reason, and communicate with other agents [10]. In addition, agents have the capability to dynamically form social structures through which they share commitments to the common goal of workflow enactment by forming a collective entity called Multi-agent systems [6]. In order to form and participate in multi-agent systems, they must be able to compromise on their autonomy level so they can coordinate with others [10]. Multi-agent can help in conducting the run-time function of WFMS, monitoring, adaptability and exception handling [19, 20].

The Multi-agent paradigm can be superimposed on the P2P architecture, where agents can reside in different peers and perform workflow tasks utilizing their decision support capabilities, the collective behavior, and the interaction protocols of the P2P system. Furthermore, P2P infrastructure supports dynamic service construction, modification and movement, and allows a dynamic agent to participate in multiple applications and dynamically form partnerships [12]. The contribution of this research can be summarized by the following statement: P2P system + Multi-agent = Adaptive Workflow Engine. The adoption of multi-agent within P2P network facilitates the capability of workflow actors to share, collaborate, and communicate in their own private web. The rest of this paper is organized as follow: section 2 describes the proposed Multi-agent P2P Architecture for WFMS. Section 3 outlines the adaptability and exception handling mechanism in the systems while section 4 presents a virtual organization application for the system. A case study and prototype implementation is presented in section 5 and section 6 concludes and states the future work.

2 Multi-Agent P2P Architecture for WFMS

A high level architecture of a Multi-agent based P2P WFMS is shown in Figure 1. This system is upgraded from previously designed P2P WFMS [1, 2] and identifies the major components and interfaces based on the Workflow Reference Model which released by the Workflow Management Coalition in October 1994 [22]. The new multi-agent version of the systems is proposed to take the run-time function and exception handling and workflow monitoring out of the P2P system level to multi-agent level. The build-time function is conducted at P2P level which includes workflow process modeling, storing process definitions and distributing the process to workflow agents deployed to the peers. The run-time function (Multi-agent level)
includes workflow process instantiation, task coordination and exception handling procedures. As shown in figure 1, the P2P network provides services that include advertisement services, group services, peer services, pipe services, and discovery services. In addition, it facilitates a user interface with human workflow participant. The workflow peers can be classified into three types based on their capabilities: Workflow Peer (WFP), Workflow Definition Peer (WFDP) and Exception Handling Peer (EHP). The WFP can reside on any machine on the P2P network enabling direct communication with other workflow peers to enact the workflow process. Each WFP is associated with a workflow participant and each uses a Local Agent (LA) to perform a part of the workflow. Once the task is completed, the LA informs its successor (Remote Agent – RA) at another node and the next task of the process may be executed. Process co-ordination is achieved by the exchange of messages between P2P agents. The second type of peers is WFDP which facilitates the design and the storage of the whole workflow schema at build-time. The workflow process is partitioned to separate tasks according to the roles of the workflow participants and the organizational structure. Then, the WFDP creates and deploys P2P agents loaded with their tasks to the corresponding peers. When a workflow agent is deployed, it registers its symbolic name and address at its destination node and keeps address book for other agents. A workflow process definition resulting from an evolutionary change is called a “version” of the workflow process. A workflow process definition resulting from an ad-hoc change which affects the running workflow instances is called variant of the workflow process [16]. P2P agents are able to learn and help in the workflow process evolution and play a key role in versioning mechanism. The third type of peers is the EHP which has the capability to deal with various types of exceptions. The exception handling procedure is discussed in the next section.

Fig. 1. Multi-Agent Based P2P WFMS
3 The Adaptability in Multi-Agent Based P2P WFMS

Two workflow exceptions can be identified in the proposed Multi-agent based P2P WFMS: local workflow exception and global workflow exception. Local workflow exception affects the task of local agent of one node. The local agent can handle this exception by applying one of two possible self-recovery policies; forward recovery or backward recovery. Forward recovery policy is based on correcting and isolating the effect of the exception and returning the workflow task to a normal state so the normal operation can be continued. In contrast, backward recovery policy is based on restoring the workflow task to a consistent state that occurred before the appearance of the exception. If the local workflow exception cannot be handled within the affected local agent, it can propagate to the other agents in the other nodes leading to a global workflow exception. These types of exceptions will, of course, affect more than one node and a coordinating node is required to deal with this exception. In this research, the coordinating node which has the capability to deal with the exceptions is the exception handling peer (EHP). This peer uses a mobile agent to transfer to other nodes to captures exceptions, characterizes the exceptions and applies a recovery policy if these exceptions cannot be handled locally at those nodes. Using the mobile agent leads to the reduction of the amount of the communication between nodes as the interaction will take place locally at the exception raising node [20]. In addition, The EHP will acquire some knowledge from previous exceptions and build an exception handling knowledge base. The Mobile Exception Handling Agent will be updated with this knowledge so it can respond to new similar exceptions rapidly by using exceptions knowledge learnt before. Using the EHP can guarantee the separation of business logic from exception handling logic. Mixing business logic and exception handling logic makes it difficult to keep track of both, complicating the verification of processes, as well as later modification [9].

In distributed systems, backward recovery of one process in a group of communicating processes will often require other processes in the group to be rolled back because of the interdependencies caused by message communication. The result is a cascade of rollbacks called the ‘domino effect’ [11]. To avoid the domino effect in the proposed system a conversation scheme is used. The conversation between P2P agents is formed by a group of P2P agents affected by an exception, and a workflow agent in the P2P agents conversation can only communicate with workflow peers that are in the same conversation. This can prevent the error propagation and limits the domino effect. The P2P agents’ conversation represents an atomic action consisting of interactions in a group of agents. After the effect of the exception is contained and resolved, the P2P agents’ conversation will be terminated and the workflow agents will return to the normal mode of the system.

4 P2P Agents Virtual Organization

In an Inter-Organizational workflow, the organizations involved have different legal and organizational systems, different security aspects, and heterogeneous hardware, operating systems and workflow applications. Inter-Organizational workflow requires
flexible, on-the-fly alignment of business partners; in other words, adaptive workflow capabilities [6]. In inter-organizational WFMS, no central control exists and the partners are autonomous workflow actors, who can join and leave a virtual organization (VO) at any time. This scenario makes the proposed multi-agent based P2P WFMS a potential workflow solution. The organizations which involved in the inter-organizational workflow will be represented by workflow peers with workflow agent engines and these peers will form a Multi-agent based P2P VO. A public workflow model is agreed between different organizations which collaborate as peers, while keeping their internal private workflow within their boundaries.

Figure 2 shows an overview of the proposed Multi-agent based P2P inter-organizational WFMS working environment where three organizations customer, supplier and manufacturer are involved in managing of a workflow process. This includes; workflow advertisement, workflow interconnection, and workflow cooperation. Each organization acts as a workflow peer (WFP) in this process. The three workflow peers discover each other by the advertisement service provided by the P2P network infrastructure and the process enactment is conducted by multi-agent workflow engines. There are three phases of interactions between workflow peers.

The first phase is workflow peers identification where workflow peers for customer, supplier, and manufacturer publish their services and join different groups of a specific product or service e.g. (customers will form a group of customer of specific product). Each WFP needs to find a desired partner. The second phase is the multi-agent interconnection; starting with electronic negotiation and connection whereby a multi-agent based P2P virtual organization is formed. The third phase consists of multi-agent cooperation, instantiating a workflow cases and coordination of their tasks.
To better illustrate how the proposed Multi-agent based P2P WFMS works, an example of a motor insurance claim process is presented. The process consists of nine tasks as shown in Figure 3. These tasks are distributed over the workflow agent associated with nodes and their workflow participants, based on the roles of the participants themselves and structure of the organization. To examine the system, both build-time and run-time functions are implemented. Build-time function implementation includes:

1. Modeling the process using Petri Net (PN) notation (Figure 3). PN is chosen because it facilitates formal semantics despite the graphical nature and state-based instead of event-based [15]. Multi-agent can be easily modeled using PN notation, each TRANSITION in the PN diagram represents a workflow agent task while the PLACE corresponding to the pre-condition for that task. The dependency between any two nodes is represented by ARC. TOKEN is corresponding to the workflow instance.

2. Transferring the Petri net model to a workflow schema stored in database. The database design includes creating several tables, e.g., a Workflow table to hold the workflow definition and its versions, PLACE table holds the details of each PLACE, TRANSITION table holds details of each TRANSITION and ARC table handles details for each ARC connecting different agents’ tasks and their pre-conditions within the workflow process.

3. Partitioning the stored process into tasks according to the roles of the workflow participants and the organizational structure of the insurance company.

4. Creation of Workflow Agents based on workflow partitions.

5. Deploying the agents to the relevant Workflow Peers using the P2P network.

After the build-time functions are completed, the system can carry out its run-time functions. These will include: first, instantiating a workflow case and coordination of
Multi-Agent Based Peer-to-Peer Workflow Management System

Fig. 4. Exception Handling Peer User Interface.

tasks between agents. Second, ad-hoc changes will be made to the running instances to examine the exception handling procedures. Exception handling can be achieved by backward recovery when the case attributes and all valid conditions are returned to their original values prior to the start of the activity where the exception occurs.

Multi-agent conversation will be examined in exception handling process either automatically or by user intervention. A snapshot of the exception handling peer user interface is shown in Fig. 4. Initial prototyping of the system has been carried out using Java coded software agents to provide an efficient workflow engine for the runtime function of the system. The P2P network environment of this prototype is based on Sun Microsystems JXTA [23]. XPDL (XML Process Definition Language) is used for process definition as it offers portability between different process design tools [21].

6 Conclusions and Future Work

This paper has proposed a Multi-agent based P2P Workflow Management System. P2P systems and multi-agents cannot replace each other for workflow application, however, they plug into each other to take advantage of both. The P2P network will be the frame where agents can work as distributed workflow engines, monitor workflow instance and react to unforeseen events. The feasibility of this work is being examined by implementing proof-of-concept prototype. Further work is required to validate and verify the ideas, the architectures and the techniques presented in this paper. This will include the completion of the underlying Multi-agent based P2P workflow architecture and their applications to a number of workflow problems and a verification and validation of the system, consisting of both research and experimental analysis.
References

23. Sun Microsystems JXTA, https://jxta.dev.java.net/