A Service Workflow Management Framework Based on Peer-to-Peer and Agent Technologies

Jun Yan1,2, Yun Yang2, Ryszard Kowalczyk1, Xuan T. Nguyen1

1Centre for Intelligent Agents and Multi-Agent Systems
2Centre for Internet Computing and E-Commerce
Faculty of Information and Communication Technologies
Swinburne University of Technology
P.O. Box 218, Hawthorn, Melbourne, Australia 3122
{yan,yyang,rkowalczyk,xnguyen}@ict.swin.edu.au

Abstract

Service workflow management which consolidates multiple services to achieve business goals has become a critical issue in service-oriented computing. Due to the unique feature of service workflows such as full distribution and high dynamicity, the contemporary workflow technology has exhibited inefficiency, vulnerability, inflexibility and lack of adaptability. Based upon the authors’ previous work on peer-to-peer based workflow management and seamless integration of agents and Web services, this paper innovatively presents SwinDeW-A which is a decentralised service workflow management framework based on peer-to-peer and agent technologies. Based on this framework, this paper also discusses how autonomous service level agreement negotiation, decentralised process enactment, and autonomous process adaptation can be achieved. The proposed approach takes advantages of the peer-to-peer computing technology and the agent technology to provide more efficient, reliable, flexible and adaptive service workflow management.

1. Introduction

Today’s enterprises face more challenges than ever before in developing business applications and offering business values [5]. The global competition, dynamic customer requirements and ever-changing marketplaces require an enterprise to be able to reduce the cost and cycle time, support agile business models and integrate distributed resources within and across organisational boundaries. In light of this situation, business organisations are moving towards an emerging paradigm, known as Service-Oriented Computing (SOC), which utilises services as fundamental elements for developing distributed applications [13]. SOC has the potential to result in reduced system complexity, lower development and maintenance costs, shorter time to the market, new revenue streams, enhanced flexibility, and improved operational efficiency [16]. The visible promise of SOC is a world where distributed business applications, or services, are composed with little effort into a network of services that are loosely coupled to create dynamic business processes and agile applications. To achieve this, the workflow technology has been exploited to consolidate multiple distributed, even heterogenous services.

A workflow process in the context of SOC is known as a service workflow which consists of a set of interrelated services as activities. Like traditional workflow processes, the enactment of these services needs to be well coordinated as planned by the workflow enactment engine in order to achieve the goals. However, due to the unique features of the service workflow such as full distribution and high dynamicity, today’s workflow technologies used in traditional domains are inappropriate for service workflow management, as they have exhibited inefficiency, vulnerability, inflexibility and lack of adaptability which have been witnessed as the major obstacles to wide adoption in the real world [20]. The existing service workflow technologies, such as BPEL4WS (http://www-128.ibm.com/developerworks/library/ws-bpel/), adopt the client-server system architecture which is popularly used in traditional workflow management approaches. As a result, enactment of a service workflow is based on a centralised workflow engine (e.g., on the side of the service requestor) to intermediate at each step of the service sequence. As pointed out by many researchers (e.g. [10]), such a centralised model is inappropriate to practically meet the needs of service-oriented applications, as it cannot reflect the inherently
decentralised nature of service workflows. The centralised workflow engine in this case often becomes the performance bottleneck causing delays in data transmission. At the same time, the service workflow seriously suffers from single-source failure, as engine malfunction would bring the whole system down. Moreover, today’s workflow management is inflexible in reflecting the dynamic nature of service workflow applications. A significant effort is normally required to manually select and contract service providers for various services in the workflow. In the presence of unexpected situations, the lack of direct interaction between component services in a process prevents individual services proactively deciding on the course of further actions.

To address the above difficulties, this paper presents a novel framework based on peer-to-peer (p2p) and agent technologies for decentralised workflow management in service environments. Built on some of the authors’ previous work, this research takes advantages of the p2p computing technology and the agent technology, aiming at providing efficient, reliable, flexible and adaptive service workflow management. More specifically, major objectives of the proposed approach are to allow for:

1) autonomous agent negotiation for contracting services,

2) direct interaction between component services during workflow enactment in order to achieve decentralised workflow coordination, and

3) autonomous workflow adaptation and distributed problem solving.

The rest of this paper is organised as follows. Major related work is discussed in Section 2. Then, the authors’ previous work which builds the foundation of this research is introduced in Section 3. In Section 4, SwinDeW-A, the novel framework based on p2p and agent technologies for decentralised service workflow management is presented, followed by mechanisms for supporting the framework in Section 5. Finally, Section 6 concludes this paper and outlines the authors’ future work.

2. Related work

Research into process management has been conducted by several communities for many years. Very recently, some research efforts on p2p-based decentralised workflow management, although very limited, have opened brand-new ground in workflows, and process support in general. To name a few, PeCo [7] presents a p2p-based workflow approach composed of peers, core services, applications, and portable plug-ins, that enable a generic system to decentralise workflow management using collaborative technologies and concepts. [9] proposes two concepts, i.e., Web Workflow Peers Directory (WWPD) and Web Workflow Peer (WWP), based on which a p2p architecture can be built for dynamic workflow management. The Matrix project (http://www.npaci.edu/DICE/SRB/matrix) aims at delivering the grid workflow protocols and workflow language descriptions necessary to build a p2p infrastructure for grid workflow management system. However, it is also evident from the literature that research on p2p-based workflow is at a very initial stage. Issues such as autonomous workflow instantiation and decentralised workflow coordination have not been addressed satisfactorily.

Software agents have been recognised as a promising technology for managing workflows [11]. For example, a well known agent-based business process management project (ADEPT: http://www.ecs.soton.ac.uk/%7Enrj/adept/index.html) developed and demonstrated a basic technology including a system architecture and some decision-making mechanisms of negotiating agents for agent-enabled workflow management to coordinate the information delivery from multiple business systems. Integration and benefits of the agent and workflow technologies for business process management are studied in [20]. An agent-based workflow management system architecture for services with the specialised agents responsible for different tasks in workflow management is proposed in [3]. Software agent based approaches have also been proposed for evolving distributed workflow models [19], and more recently for orchestrating Web services [8] and workflow management in Grid computing [6]. However, research on agents in the context of workflow management is immature. It has typically focused on multi-agent system architectures, agent interaction and coordination, and other agent technology centric issues that map to some basic requirements of distributed process management. In fact, most agent-based approaches offer centralised workflow management where the agents are controlled by a central workflow engine or are deployed internally within the engine for specialised management tasks, which lead to inefficiency and vulnerability as pointed out previously. They also typically offer a very limited support for decentralised exception handling and adaptation to changes in distributed workflows which are key issues, especially in service environments. To the best of the authors’ knowledge, there is no work on combining p2p and agent technologies for service workflow management. Moreover, the existing agent-based approaches for service workflows assume tight coupling between agents and the workflow domain tasks or services (e.g. services implemented within agents or coordination capabilities implemented within each service) that is inflexible and usually implies complete re-engineering of the business application services.
3. Background work

The research presented in this paper is carried out on the basis of some previous work of the authors, including:
1) the p2p-based decentralised workflow management system in non-service environments, known as SwinDeW [20], and
2) seamless integration of Web services and agents, known as WS2JADE [13][14].

3.1. SwinDeW (Swinburne Decentralised Workflow)

As traditional client-server based workflow management has encountered more and more challenges, p2p-based workflow has become one of the most strategic directions in workflow research [12]. Based on the philosophy that the p2p computing model reflects workflow’s distributed feature more naturally, the authors have conducted intensive research on p2p-based workflow management and developed a system called SwinDeW to better manage workflows in a decentralised way [20].

SwinDeW has contributed a novel framework for p2p-based decentralised workflow management, which implies the presence of neither a centralised data repository for information storage, nor a centralised workflow engine for workflow coordination. Based on this framework, corresponding mechanisms have also been developed concretely for both build-time and run-time workflow functions, in support of both completely and incompletely specified workflows. More specifically, SwinDeW consists of a set of distributed peers which are software components spreading over the network. Each peer is associated with and acts on behalf of a workflow participant. To carry out workflow functions, peers are capable of interacting with one another in a p2p manner to deliver core workflow services such as the peer management service, the process definition service, the process enactment service, and the monitoring & administration service. SwinDeW proposes a novel philosophy of “know what you should know” for process data storage, which divides a process definition into task partitions and distributes each partition to relevant peers for storage. Equipped with essential process knowledge, peers can communicate directly to create process instances and coordinate the enactment of various tasks.

SwinDeW offers significant advantages such as improved performance, enhanced reliability, increased scalability and flexibility, better user support, and system openness. These advantages have been reflected in the SwinDeW prototype based on Sun Microsystems JXTA (http://www.sun.com/software/jxta/).

3.2. WS2JADE

Web services technology and software agent technology have complementary strengths. The combination of these two technologies could create an environment which enables universal interoperability between heterogenous applications with enhanced intelligent and autonomous capabilities. To integrate Web services and the JADE agent platform [18], a toolkit called WS2JADE has been presented in [13], which enables run-time deployment and control of Web services with JADE agents [14].

WS2JADE allows JADE agents to access and use Web services. It is an attempt to provide a framework in which agents and Web services, with separation of concerns in their implementations, can communicate with each other. WS2JADE provides facilities to deploy and control Web services as agent services at run-time for deployment flexibility and active service discovery. WS2JADE follows the gateway architecture described in [1] to proxy agents’ requests to targeted Web services. Proxy agents in WS2JADE can be dynamically generated at run-time and provide seamless access of JADE agents to the underlying Web services [13][14]. The toolkit’s latest version 1.1 is freely available at http://www.it.swin.edu.au/centres/ciamas.

4. SwinDeW-A framework

Based on the authors’ previous work as introduced in Section 3, this section presents the framework of a decentralised, agent-based service workflow management system using the p2p technology, named SwinDeW-A (Swinburne Decentralised Workflow with Agents).

Figure 1. Framework of SwinDeW-A
To provide efficient, reliable, flexible and adaptive service workflow management, this research combines the p2p technology, the agent technology, and the workflow technology to develop a decentralised and collaborative workflow management environment with the intelligence and adaptation abilities. The proposed approach is in essence to provide a distributed multi-agent middleware in which agents collaborate in a p2p manner to ensure the fulfilment of service workflow functions.

Figure 1 depicts the framework of SwinDeW-A at a very high level. In this approach, the existing SwinDeW, a multi-agent system built upon JADE, and WS2JADE are innovatively integrated. Workflow semantics for decentralised provision of core workflow functions through peer collaboration are defined in SwinDeW. The multi-agent system built upon JADE consists of a collection of distributed software agents that work in conjunction with each other to provide core workflow services defined in SwinDeW. Therefore, each peer in SwinDeW becomes an intelligent agent in SwinDeW-A with the improved ability for collaboration and problem solving. Naturally, each agent in SwinDeW-A is correlated with and acts on behalf of a service consumer or provider. The integration of agents and Web Services is supported by WS2JADE.

As shown in Figure 2, with support of SwinDeW-A, automatic, dynamic and adaptive management of service workflow may be achieved in a decentralised way. The agent associated with the service requestor first negotiates with the agents of service providers autonomously to determine the provision of individual services in the workflow (e.g., who provides what service) dynamically. Then the agents of the service requestor and the selected service providers interact and cooperate with each other to automate different aspects of workflow management for successful execution/completion of the workflow. More specifically, individual service providers no longer need to send control data as well as potentially large amounts of intermediate application data to a single site (e.g., the workflow engine on the side of the service requestor) for coordination. The agent of the service requestor only subscribes monitoring data to keep itself informed about the state of workflow enactment. Control and intermediate application data are transmitted between agents of relevant service providers directly to coordinate the execution of the workflow. Relevant agents evaluate the enactment conditions of the corresponding services and manage their enactment through WS2JADE. By this means, system efficiency and flexibility can be improved significantly. In the case that exceptions occur during workflow enactment, the involved agents collaborate and negotiate with each other to handle the exceptions and recover automatically (e.g., discovering new services available, enquiring and requesting other services, etc.). This may largely enhance system reliability and adaptability.

5. Supporting mechanisms

Based on the framework proposed in Section 4, this section discusses mechanisms to support autonomous service level agreement negotiation, decentralised process enactment and agent-based process adaptation.

5.1. Agent-based service level agreement negotiation

As in traditional workflow applications, an instance of a service workflow represents a single enactment of this
An important requirement is to dynamically select and contract service providers for various service types in the process. At the same time, an agreement known as Service Level Agreement (SLA) needs to be formed between the service requestor and each selected service provider, which regards the guarantees of service provision. Most current service workflow management approaches do not offer such capabilities or assume they are manually performed by the users, which can be very cumbersome or impractical, especially when there are multiple, distributed, and even heterogeneous providers who can offer the same functionality.

In SwinDeW-A, agent’s capabilities of interaction and decision-making enable autonomous negotiation as a means of dynamic provider selection and contracting. In distributed communication, particularly in a heterogeneous environment, agents can use a standard agent communication language (ACL) to exchange information and knowledge in meaningful semantics so the agents can understand the requirements of the process instance and the capabilities of each other. One direct advantage of agent negotiation in this case is autonomous combination of service providers for the functional requirements of service workflow provision. In addition, based on business rules, strategies and utilities, agents can interact and negotiate the non-functional parameters such as time, cost and availability, aiming at collectively fulfilling end-to-end Quality of Service (QoS) of service workflow provision and/or optimising service workflow delivery. The ultimate result of agent negotiation for process instantiation is a set of contracted service providers with established SLAs for process enactment.

By deploying WS2JADE, each agent can create and publish the ontology of the associated Web service. Thus, the agent of the service requestor can discover a list of providers for the same functionality using a service matchmaking algorithm. After that, the agent of the service requestor negotiates with agents of service providers for the provision of services in the workflow. The negotiation for individual services in the workflow needs to be well coordinated aiming at collectively fulfilling end-to-end QoS. To achieve this, the agent of the service requestor contains a compound SLA coordinator agent and a SLA negotiation agent factory. The compound SLA coordinator agent is responsible for mapping end-to-end (process level) QoS to service level QoS and coordinating concurrent negotiation for different services to achieve end-to-end QoS. The SLA negotiation agent factory is responsible for initialising, deploying and activating SLA negotiation agents for difference services in the workflow. Therefore, each SLA negotiation agent is able to negotiate with a number of service providers (their negotiation agents) for the provision of a service. Based on the agreed SLA values, the best combination of service providers is contracted for the provision of the service workflow. The basic mechanisms for automatic agent negotiation and partner selection have been reported in [2][4]. The results demonstrate capabilities of negotiation agents to dynamically resolve conflicts, coordinate distributed tasks, and adapt to changes in the environment in the presence of incomplete and imprecise information without centralised control. Figure 3 illustrates the
scenario of SLA negotiation in SwinDeW-A, as explained above.

Once the service providers are selected and contracted, a Process Context Knowledge Base (PCKB) is created for the agent of each selected service provider. This knowledge base not only contains contractual obligations, i.e., SLA, but also defines the contextual information of the Web service so that the agent can communicate with agents of other services straight away to manage the required interaction between services without the involvement of an extra workflow engine. The “know what you should know” philosophy for agents proposed for this purpose is described in [20]. In brief, this approach decomposes a process into a set of interrelated task partitions. Each task partition represents a service and its position, i.e., interaction and dependency with other services in the process. Then individual task partitions are distributed to corresponding agents of contracted service providers. After each task partition is distributed to the service provider who offers the required service for this process instance, the agent associated with this provider acquires essential knowledge to manage the enactment of the corresponding service in the present process instance. In the simplest case, this agent is capable of exchanging the state of service enactment with other agents, automating the enactment of the associated service, and controlling the interaction (e.g., transfer intermediate application data from this service to other services) between this service and other relevant services.

5.2. Agent-based, decentralised process enactment

Coordination of process enactment within the framework proposed in Section 4 can be addressed in two aspects. One aspect is monitoring of service enactment which involves interaction between the agent of the service requestor and the agents of the service providers. The other aspect is the control of service enactment sequence and the transmission of intermediate application data which involve interaction between the agents of the service providers.

Monitoring of service enactment refers to continuous scanning of the state of service enactment and the actual values of non-functional parameters. In SwinDeW-A, the agent of the service requestor collects up-to-date monitoring data from the agents of the service providers from time to time. By this means, the service requestor is able to know the status of the process enactment and detect the breach of SLA which is established at the process instantiation stage. Regarding workflow control, the SwinDeW mechanisms initially described in [20] can be naturally migrated to the service environment with agent support. As stated in Section 5.1, each agent of a service provider has a local knowledge base describing the process context of the corresponding service. Therefore, each agent can act independently to automate the enactment of its service and collaborate with other agents to schedule the service enactment sequence. More specifically, an agent is able to receive control information and application data from other agents, make decisions on whether its service needs to be enacted and when, invoke its service autonomously though WS2JADE, notify the agents of succeeding services properly, and manage the intermediate application data to be transferred to succeeding services as required. Agents communicate with one another in a p2p fashion, using an ACL to exchange meaningful semantics for the purpose of service enactment scheduling.

![Figure 4. Monitoring of a Web service using WSDM](image)

Note that monitoring of service enactment in SwinDeW-A is based on the recently published Web Services Distributed Management (WSDM) specification [15][16]. WSDM defines how the management of resources can be accessed via Web service protocols—Management Using Web Services (MUWS)—and how Web services can be managed using Web services—Management of Web Services (MOWS). As shown in Figure 4, each service provider who exposes its application as a Web service also publishes a reference to the manageability endpoint of its application. The agent of the service requestor can perform monitoring actions on the application by exchanging messages with the manageability endpoint through WS2JADE.
5.3. Agent-based process adaptation

While exception handling normally requires human intervention, the intelligent problem-solving and social capabilities of agents to interact and collaborate with each other enable SwinDeW-A to have the automatic exception handling ability based on agent negotiation. More specifically, the agent-based process adaptation in SwinDeW-A targets at the following three major exceptions in the service workflow scenario:

1) The enactment of a service fails to achieve the expected result.
2) A contracted service provider is unavailable to provide the required service.
3) An SLA is broken because of QoS violation.

Different types of exceptions can be easily detected by different agents when the service workflow is under enactment. The first two exceptions can be detected by agents associated with corresponding service providers and reported to the agent of the service requestor. The last exception can be detected straight away by the agent of the service requestor which collects monitoring data at run-time. In the case of these exceptions, intelligent agents in SwinDeW-A are able to interact and negotiate automatically on what actions to take in order to bring the service workflow back to normal. ACLs provide mechanisms for agents to enquire other agents, find other agents with the required capabilities, and negotiate for their services. To deal with a service failure or an unavailable provider exception, the agent of the service requestor automatically negotiates with the agents of other available service providers who can provide the required service, and dynamically contracts a new provider for the failed service. Relevant agents of other contracted service providers are updated with their new collaborative agent. To deal with a broken SLA exception, the agent of the service requestor re-negotiates with the agents of the affected service providers to re-establish SLAs and where required, negotiates with and contracts new service providers to replace the contracted ones. The above capability of automatic exception handling makes SwinDeW-A more capable of supporting service workflow’s adaptability.

6. Conclusions and future work

Workflow management which manages complex interactions between multiple, distributed and heterogenous applications plays a critical role in service-oriented computing. However, today’s workflow technology exhibited deficiencies such as inefficiency, vulnerability, inflexibility and lack of adaptability in support workflows in service environments. To address these problems, this paper has introduced the recent trend of shifting traditional workflow management to the service environment aiming at decentralising the service workflow management to improve efficiency and reliability. In addition, this paper has addressed the novel integration of the agent technology with the service workflow management aiming at enhancing flexibility and adaptability. As a result, this paper has presented SwinDeW-A, a peer-to-peer (p2p) and agent based decentralised service workflow management which innovatively and seamlessly integrates the authors’ previous research on p2p-based workflow management, named SwinDeW, with Web services and agents. The corresponding mechanisms for scenarios of autonomous service level agreement negotiation, decentralised process enactment, and process adaptation have also been discussed within this framework.

The SwinDeW-A framework provides a good basis for future research and further development. More specifically, different negotiation strategies will be explored to support service level agreement negotiation. Peer communication protocols and message types defined in SwinDeW will be translated into agent interaction models and ACL messages, respectively. Support for more advanced behavioural models in SwinDeW such as service suspension and resumption will also be investigated.

Acknowledgements

This work is partially supported by the Adaptive Service Agreement and Process Management (ASAPM) in Services Grid project (AU-DEST-CG060081) and the EU FP6 Integrated Project on Adaptive Services Grid (EU-IST-004617). The ASAPM project is proudly supported by the Innovation Access Program - International Science and Technology established under the Australian Government’s innovation statement, Backing Australia's Ability. This work is also partly supported by Swinburne Vice Chancellor’s Strategic Research Initiative Grant 2002-2004, as well as the National Natural Science Foundation of China under grants No.60273026 and No.60273043.

References

Infrastructure for Collaborative Enterprises (WETICE’02), 176-181, Pittsburgh, USA, June 2002


