Agent-Based Integration of Web Services with Workflow Management Systems

Bastin Tony Roy Savarimuthu, Maryam Purvis, Martin Purvis, Stephen Cranefield
Department of Information Science
University of Otago
P O Box 56, Dunedin, New Zealand
tonyr@infoscience.otago.ac.nz

ABSTRACT
Rapid changes in the business environment call for more flexible and adaptive workflow systems. Researchers have proposed that Workflow Management Systems (WfMSs) comprising multiple agents can provide these capabilities. We have developed a multi-agent based workflow system, JBees, which supports distributed process models and the adaptability of executing processes. Modern workflow systems should also have the flexibility to integrate available Web Services as they are updated. In this paper we discuss how our agent-based architecture can be used to bind and access Web Services in the context of executing a workflow process model.

Categories and Subject Descriptors
I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence – Multiagent systems; H.4.1 [Information Systems Applications]: Office Automation – Workflow management;

General Terms
Design

Keywords
Web Services, Multi-agent systems, Workflow systems

1. INTRODUCTION
Workflow management systems (WfMSs) [5] are widely used to manage business processes due to their known benefits such as automation, coordination and collaboration between entities. Still, the existing, commercially available workflow management systems do not offer sufficient flexibility for distributed organizations participating in the global market. Existing systems have rigid, centralized architectures that do not operate across multiple platforms [6]. Improvements can be made by employing a distributed network of autonomous software agents that can adapt to changing circumstances.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 2005 ACM 1-59593-094-9/05/0007 ...$5.00.

The work of Fleurke et al. [3, 4] deals with the framework of a distributed network of autonomous software agents that can adapt to the changing circumstances in a workflow management system. The business processes undergo changes over time to accommodate a changing environment such as the availability of Web Services.

Business processes should be able to take advantage of Web Services that are available in an intranet as well as in the Internet, such as stock monitoring Web Services. A workflow system that models these business processes should have mechanisms to integrate and use these Web Services. In this paper we describe an enhanced framework that provides mechanisms to create agents that are capable of accessing various Web Services.

2. INTEGRATING WEB SERVICES
Web Services are software components available on the Internet, which provide certain services that may be of general interest, such as weather monitoring services, currency converters, etc. Large fractions of Web Services are used within companies and protected within their own firewalls. These Web Services can be accessed for day-to-day business transactions. Examples of these Web Services include banking services and air ticket booking. Our workflow process modeller can integrate Web Services with the existing workflow system. In our framework [4] a process agent executes the process model. Each resource in the system has its own resource agent, which is capable of performing certain tasks.

In our design we have wrapped Web Service as an agent. A specialized agent called Web Service Agent (WSA) is created, and it can be used to query various operations exposed by a Web Service. Thus a resource agent in the WfMS is specialized into a Web Service Agent. When the process agent executes a process model, a job token object is created. This job token consists of a map that stores various attributes, including details such as the URI of the WSDL file, and the operation to be invoked on the Web Service. The workflow manager will create these attributes in the job token. The attributes encoded in the job token in the form of name-value pairs are:

- Attribute 1: (URIName, URI)
- Attribute 2: (transition name, operation name)

The process agent assigns a task to a resource agent. It passes the job token to the resource agent. The resource agent accesses the
URI from the job token and reads the WSDL document and uses
wsdl2java from the Axis toolkit [1] to create the necessary stubs.
It also generates necessary code for handling requests from other
agents to access appropriate operations exposed by the Web
Services and also the code for sending responses to other agents.
Figure 1 shows the code that is generated during the creation of a
Web Service Agent. The resource agent sends a message to
the process agent about the creation of the Web Service Agent, which
is capable of handling requests for operations defined in WSDL.
The process agent then assigns the task of creating a
Web Service Agent, by handing over the job token to the
WSA. The Web Service Agent matches the task name from the
map and retrieves the operation to be performed. The Web
Service Agent connects to the Web Service, retrieves the result
and sends it to the process agent. In our framework, FIPA ACL
[2] is used for agent communication. In the current system the
Petri net model provides an abstract view of the process, and the
process agent has the built-in intelligence to map the transitions to
the agents that are capable of performing the tasks and thereby
sends messages to appropriate agents. Flexibilities offered by
the Web Service Agent include the storage of the history of
successes/failures of Web Services (which can be used for
performance analysis) and matchmaking to find Web Services
available on the Internet that offer services specified in WSDL.

Figure 1. Code generation for a Web Service Agent (WSA)

3. CONCLUSIONS
Our enhanced agent-based architecture facilitates the easy
integration of Web Services with the workflow system. A Web
Service Agent will be able to connect to any Web Service
dynamically. In the future, we intend to integrate workflow
ontologies and domain-specific ontologies into our system and
harness the power of the Semantic Web when using Web
Services. We are currently extending our architecture to
accommodate a process model that executes composite Web
Services. A full version of our paper can be found in [7].

4. REFERENCES
system - an approach based on Petri nets and agents. Master’s thesis, Department of Computer Science, University of
Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands,
2004.
adaptive and distributed framework for workflow systems. In
Proceeding of the Workshop on Collaboration Agents:
Autonomous Agents for Collaborative Environments
Organisational Workflow Coordinated by Software Agents.
In CEUR Workshop Proceedings No 17. Cross
Organisational Workflow Management and Coordination,
USA, 1998.
[7] Savarimuthu, BTR., Purvis, M.A., Purvis, M.K., and
Cranefield, S., “Agent-Based Integration of Web Services
with Workflow Management Systems”, Information Science
Discussion Paper Series, Number 2005/05, ISSN 1172-6024,
University of Otago, Dunedin, New Zealand (2005).