Available online at www.sciencedirect.com

SCIENCE DIRECT®

Decision Support Systems 40 (2005) 31–50

Decision Support Systems

www.elsevier.com/locate/dsw

Agent-oriented compositional approaches to services-based cross-organizational workflow

M. Brian Blake[a,*,1], Hassan Gomaa[b]

[a] Georgetown University, Room 234, Reiss Science Building, 37th and O Street, NW, Washington, DC 20057, USA
[b] George Mason University, 4400 University Drive, Mail Stop 4A4, Fairfax, VA 22030-4444, USA

Available online 20 May 2004

Abstract

With the sophistication and maturity of distributed component-based services and semantic web services, the idea of specification-driven service composition is becoming a reality. One such approach is workflow composition of services that span multiple, distributed web-accessible locations. Given the dynamic nature of this domain, the adaptation of software agents represents a possible solution for the composition and enactment of cross-organizational services. This paper details design aspects of an architecture that would support this evolvable service-based workflow composition. The internal coordination and control aspects of such an architecture is addressed. These agent developmental processes are aligned with industry-standard software engineering processes.

Keywords: Agent architectures; Workflow modeling; Coordination; UML; Web services

1. Introduction

Online businesses are beginning to adopt a developmental paradigm where high-level component-based services and semantic web services [37] are becoming sufficiently modular and autonomous to be capable of fulfilling the requirements of other businesses. We use the term, *services-based cross-organizational workflow* (SCW)[5], to describe the workflow interaction that occurs when one business incorporates the services of another within its own processes (also described as *business-to-business* (B2B)). This term is sometimes associated with the idea of a third-party organization that composes the services of multiple businesses, which is also described as *virtual enterprise* [15].

In general, the major problems in this domain relate to the *dynamic and distributed nature of the Internet environment*. In this environment, business processes and the underlying services are constantly removed and updated. It is a major problem to create systems that operate with respect to these dynamic conditions. A second major issue is related to the distribution of services. Since services are distributed across physical and geographical boundaries, any solution architecture must support an equivalent degree of distribution.

* Corresponding author.
E-mail addresses: blakeb@cs.georgetown.edu (M.B. Blake), hgomaa@gmu.edu (H. Gomaa).
[1] For identification purposes only, this author also has an affiliation with The MITRE Corporation, Center for Advanced Aviation System Development, M/S N420, 7515 Colshire Drive, McLean, VA, 22102-7508, USA.

0167-9236/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2004.04.003

Although there are several related projects that define solutions to the problems of cross-organizational workflow (which will be described in more detail in Section 2.3), a distinguishing innovation of this work is the use of the autonomy of agent technologies. In applying agent technologies to the problems related to dynamism and distribution, an agent architecture and design is introduced which places emphasis on the specific roles, responsibilities, and actions of the individual agents. A further contribution of this work is the specification and programming of the control mechanisms internal to the agents. A unique feature introduced here is the integration of this specification approach with current, industry-standard software engineering processes and methodologies. Considering the dynamism of the Internet environment, these specification-driven approaches are essential for the dynamic reconfiguration that results from distributed service and process changes.

The paper proceeds in Section 2 with an overview and motivation of the cross-organizational workflow domain with respect to the integration of distributed services. In Section 3, the *Workflow Automation through Agent-based Reflective Processes* (WARP) is introduced to support the SCW domain. In Section 4, there is a discussion of the integration of independent services and, in Section 5, the agent-based modeling approach to support the community of services in the SCW domain. Section 6 contains details of the general agent interaction protocols that support this environment. Finally, Section 7 discusses the WARP prototype and its performance.

2. The SCW environment and web services

The SCW approaches are a natural extension of the related areas of component-based software engineering, in particular component composition. In traditional component composition research [19], components, such as CORBA, COM+, J2EE, Enterprise Java Beans, and .NET, and their interfaces are modeled using formal (text and visual) languages. Consequently, these specifications are evaluated and deployed to support automated component composition. Currently, these components can be specified with web service technologies to facilitate large-scale electronic market interoperability.

2.1. Web service technologies

Components, which are specified with web services technologies, have the capability of being discovered and accessed from distributed locations. These web services are specified with the Web Services Description Language (WSDL) [37] and can be invoked using the Simple Object Access Protocol (SOAP) [31]. Traditional WSDL supports a syntactical form of specifying web services, while the term, *semantic web services* refers to an extended Web of machine-readable information and automated services. The DARPA Agent Markup Language for Services (DAML-S) and the Web Ontology Language (OWL-S) [26] both provide an ontology of services to allow automated support for components such as agents to locate, select, employ, compose, and monitor Web-based services. Distributed registries, such as the Universal Description, Discovery, and Integration (UDDI) [34,35] architectures, advertise the specifications of distributed services universally. In addition, other Extensible Markup Language (XML)-based languages, such as the Web Services Flow Language (WSFL), Business Process Execution for Web Services (BPEL4WS), and the Business Process Modeling Language (BPML), specify the process-based composition of these services as described in Refs. [7,25,32,40]. However, these process languages are specified with a text-based approach that tends to conflict with the visual design notations commonly accepted for software development, in particular the use of the Unified Modeling Language (UML) [6,17]. A main goal of this work, which will be discussed in detail in Section 3, is to support service-based composition using accepted *visual* developmental approaches coupled with agent-based protocols.

2.2. A sample SCW environment

The SCW environment described in this paper incorporates the interoperability of general web services. In Fig. 1, an example is given of a SCW environment for multiple travel-related businesses.

The initiating business is the travel agency company. The Travel Agency has internal services for managing customers' accounts and credit card numbers. However, the travel agency uses other third-party vendors to realize the hotel reservation and car rental