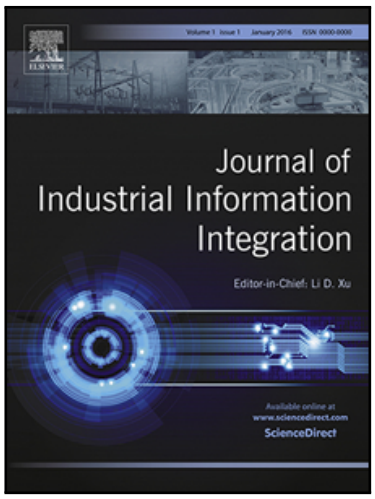

The Reviews and Analysis of the State-of-the-Art Service Workflow Specification Languages

Wattana Viriyasitavat , Andrew Martin

PII: S2452-414X(17)30035-3
DOI: 10.1016/j.jii.2017.07.002
Reference: JII 44

To appear in: *Journal of Industrial Information Integration*

Received date: 27 June 2017
Revised date: 19 July 2017
Accepted date: 24 July 2017

Please cite this article as: Wattana Viriyasitavat , Andrew Martin , The Reviews and Analysis of the State-of-the-Art Service Workflow Specification Languages, *Journal of Industrial Information Integration* (2017), doi: 10.1016/j.jii.2017.07.002

The Reviews and Analysis of the State-of-the-Art Service Workflow Specification Languages

Wattana Viriyasitavat[1], Andrew Martin[2]

[1]*Business Information Technology Division, Department of Statistics, Chulalongkorn University, Pathumwan, Bangkok, 10330, Thailand,*
[2]*Department of Computer Science, University of Oxford, Oxford, OX1 3 PR United Kingdom*

Abstract

In the globalized market and business environment, enterprises strive to cope with rapid market changes and competition. To sustain competitiveness, enterprises need to continually adapt their business processes to accommodate changes promptly. This view includes the integration of external services with their internal services for required capabilities to catch new business opportunities. A number of services with similar service functionalities are available, posing a great challenge in composing optimized business processes in a timely manner. This requires effective methodologies and tools in selecting services and composing them as service workflows for specified goals. Service workflow technology facilitates these by providing methodologies to support business process modeling and reengineering to optimize and automate processes according to workflow requirement specifications. This paper provides reviews and analysis of the state-of-the-art service workflow specification languages, including (1) Self-Adaptive Configuration based on HMS (2) BAM, (3) CTR, (4) SOLOIST, and (5) SWSpec. These languages are evaluated against the fundamental principles for general service workflow specification languages.

Keywords: Service; Workflow; Specification Language; Business Processes; Logic; CTL;

1. Introduction

The Internet and Service Oriented Computing (SOC) technologies have revolutionized the ways of businesses management. For enterprises, the boundaries with business environments become ambiguous; enterprises are able to utilize geographically distributed services to implement solutions for their specific goals (Bi et al 2017, Cochran et al 2016, 2017, Wang et al. 2016). Enabled by SOC technology, the paradigms for business collaboration have been evolved from data-driven to process-driven modes (Lin et al 2009, Zhang et al. 2014, Moser et al 2012, Yen et al 2008, Huemer et al 2008). In this context, service workflow is an important tool to model and formulate the coordination and composition of services (Viriyasitavat et al 2012, 2014a, 2014b). The workflow has been widely used to model large-scale scientific and engineering applications in various domains such as earth science, astronomy, physics, healthcare, telecommunications, military, bioinformatics, administration at universities, smart grid, and digital libraries (Anderson et al 2003, Cardoso et al 2004, Deelman et al 2016, Espinoza et al 2013).

Nowadays, the business processes are facing important challenges as they encounters frequently changes from customers, global competition and technological advances. Service workflows are effective methodologies to deal with these challenges (Viriyasitavat and Martin 2012). They rely on services as basic building blocks to construct dynamic, self-adaptable, cost-effective, and optimized solutions to complex business processes, according to certain requirements. Functionalities from inside or outside an organization are viewed as services, which can be orchestrated by service workflows. They have a number of advantages over conventional methods such as easiness of computer representation, reusability, failure managements, adaptability, and suitability of parallel computing for fast solutions (Viriyasitavat 2016). Moreover, service workflows reduce the need to develop new components each time a new business process arises. Once an