Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 83 (2016) 480 – 487

Procedia Computer Science

The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016)

An Application Suite for Service Enabled Workflow

Mostafijur Rahman[a], Wendy MacCaull[a]

[a]*St. Francis Xavier University, Antigonish, Nova Scotia, Canada*

Abstract

Recently our research group introduced the notion of Service Enabled Workflow (SEW) with the integration of Semantic Web Service and Workflow. SEW considers workflow as a collection of tasks with specific control flow where tasks are carried out as services. In this paper, we present a Service Oriented Architecture (SOA)-based integrated application suite to design and develop ontology-based SEWs. Quality-aware semantic web services for tasks in a workflow, more closely aligned to the needs of the consumer through the use of ontology matching, are discovered, selected, composed and executed automatically by a smart phone-based software agent. We evaluate the effectiveness of the application suite using a case study where we introduced service enabled functionality for tasks in a simplified workflow.

Peer-review under responsibility of the Conference Program Chairs

Keywords: Semantic Web Service Discovery; Service Composition; Workflow; Service Enabled Workflow; Quality of Service; OWL-S

1. Introduction

Service Enabled Workflow (SEW)[2] is a relatively new concept in the area of Semantic Web (SW)-based research. The literature shows that the underlying concepts are about 10 years old arising from Business Process Model and Notation, Business Process Execution Language, Business Process Execution Language for Web Services and Web Service composition. While SEW has a lot of potential, it still requires a great deal of maturity and support of tools to become an industry standard. In this paper, we present a SOA-based application suite called MOSEW that provides functionalities to design and develop ontology-based SEWs. With this application suite one can graphically define workflow task specifications using ontology instances and execute the workflows automatically through the consumer agent, developed on a mobile platform where QoS-aware semantic web services are discovered, selected, composed and executed automatically for the tasks in a workflow to complete the overall execution.

A workflow is a collection of interconnected tasks with a specific control flow. In SEW, tasks are carried out as services. A service is a self-contained unit of functionality that is placed in a location or is performed to provide a repeatable and consistent set of outcomes to systems. Workflow tasks can be classified into service-configurable tasks

* Corresponding author. Tel.: +1-902-872-1113 ; fax: +1-902-867-2329.
E-mail address: x2013ici@stfx.ca

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.04.212

which have specifications representing the action needed to be carried out and service-non-configurable tasks which have only task execution rules. Typically, a task specification (TS) is described by several elements including a name, a list of input and output parameters and a task execution rule. To extend the specifications and include the semantic aspects into it, we followed Grossmann et al.[6] work to describe TS using domain specific ontology instances.

Web Services (WSs) are platform independent computational elements that can be described, published, discovered and programmed using XML for the purpose of developing massively distributed applications[15]. SOA is an approach to build distributed systems that deliver application functionality as services which are language and platform independent. It consists of three main building blocks, namely, a Service Repository, a Service Consumer and a Service Provider. Semantic Web Service (SWS) is a standalone piece of functionality that is self-descriptive, machine-readable and can be automatically discovered and executed from the web[16]. WSs does not support automated service discovery and execution from web while SWS does. Consumers can utilize this advantage to accomplish their desired tasks automatically by using SWS based discovery, selection and execution approach. With an increasing number of WSs providing similar functionalities, QoS properties become an important criterion for the selection of the most suited WS. In general, QoS properties are divided into two sub-categories: Measurable (response time, execution price, throughput and reliability etc.) and Non-measurable (reputation and security etc). In our research, we worked with measurable QoS properties.

Over the last few years, smart phones have become very common and convenient devices with high usability. Advances in Information and Communication Technology have influenced service providers to improve their infrastructures so that consumers can consume their services through the smart phone driven application or agent. The agent can act as a mediator to discover services automatically in a dynamic workflow execution environment. Nowadays, an increasing number of companies and organizations implement their core business functionalities as services and publish them over the Internet to provide access to others. Thus, the ability to efficiently and effectively select and integrate cross-organizational and heterogeneous services on the Web at runtime is an important step towards the development of WS-based applications. In particular, if no single WS can satisfy the functionality required by the user, the services need to be combined or composed to achieve the desired goals of the consumer[12]. For example, setting up an appointment requires discovery and selection of both patient and physician. These are provided by two different services: Discover Patient and Discover Physician. In this example, the Discover Patient service would have to be invoked first, because this will select the patient. Only after the invocation of the Discover Patient service is completed and the patient is selected, will the Discover Physician service be invoked. So, the services need to be combined or composed to achieve the desired goals of the consumer.

The rest of the paper is organized as follows: Section 2 describes a Motivating Example, and Section 3 presents our approach for QoS-aware SWS discovery. Section 4 provides a brief description on the MOSEW architecture. In Section 5, we present experimental results. Section 6 provides a brief description on related works and in Section 7, the conclusion and future works are given. While we use the example described in Section 2 to evaluate the effectiveness of the application suite, our work is designed to be domain independent and is not restricted to only this example. We kept the ability to integrate any domain specific ontology into our application suite to manage any complex scenarios and to execute related workflows automatically to achieve their desired goals.

2. Motivating Example

We[19] have been working with the clinical guidelines related process models for last few years. The example we describe here is based on the simplified Hospice Palliative Care[20] process model. Suppose Emma performs most of the tasks involved in this process model and the use-cases are as follows: 1. Patient referrals are received by the Palliative-Care program. 2. Appointments are set with the physicians based on their availability for each of the patients. 3. A Physician consults with the patient. 4. A Physician decides if the patient is appropriate for the program or not. 5. If the patient is not appropriate she is refused with an explanation. 6. Otherwise the patient is registered into the program. 7. A team is formed with formal and informal caregivers for each of the registered patients. 8. The teams continue to provide care to each of the patients until they are released.

In this process model, depending on the eligibility status of each patient, a part of the process model is executed. Consider a scenario, where Emma can use SWSs to perform all these tasks. It would be difficult for her to execute all these tasks by consulting the SWSs manually because: