JOURNAL OF INNOVATION IN DIGITAL ECOSYSTEMS 3 (2016) 197–207

HOSTED BY

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/jides

QoS-Aware approach to monitor violations of SLAs in the IoT

Mohammed Alodib

Qassim University, Saudi Arabia

ARTICLE INFO

Article history:
Received 15 October 2016
Received in revised form
29 October 2016
Accepted 30 October 2016
Published online 13 December 2016

Keywords:
Internet of Things
Web services
Service oriented Architecture
Model Driven Architecture
Service Level Agreement
Quality of Service

ABSTRACT

The Internet of Things (IoT) is an ecosystem comprising interrelated wireless devices. Web services developed over Service oriented Architectures (SoA) are among the most promising solution to facilitate the communication of things in the IoT. Web services interact with each other, irrespective of features such as operating system, or programming language. One of the main challenges facing such a platform is the declaration of SLAs, and the monitoring of violations. This is because the IoT allows users to build large, distributed, and complex applications. Therefore, it critical to develop a method to facilitate the supervision and management of SLAs. The method proposed in this paper aims to automate the generation of a QoS-Aware service, providing real-time monitoring. The algorithm used to produce the service is inspired by the diagnosability theory of Discrete Event System (DES). In this approach, the SLAs defined by users are automatically mapped into a UML QoS model. The SoA composite application for each site is then mapped into a Petri net. The UML QoS model is then transferred and combined with the Petri net model. An algorithm to compute the QoS-Aware service is then applied to the Petri net model, and the service produced finally incorporated into the system.

1. Introduction

The ability to understand the factors that determine customers' satisfaction with the services provided is increasingly significant. This is considered especially challenging in case of emergent Service oriented Architecture (SoA) [1,2], which offers solutions to develop ubiquitous and distributed enterprise systems. SoA provides an efficient communication platform for devices involved in the Internet of Things (IoT) [3,4]. Interactions between things can be facilitated using Web services, which provide promising mechanisms for developing composite applications [5]. A Web service can also be accessed irrespective of the programming language, and operating system used [6].

The development of Web services over the IoT requires use of composite Software as a Service (SaaS). Services can be discovered for execution using Universal Description Discovery and Integration (UDDI), which is an XML-based protocol introduced to facilitate interactions between the service provider and users. In this context, the service provider publishes meta-information relevant to the Web services located within its domain. Users send requests to the target services using UDDI.

Peer review under responsibility of Qassim University.
E-mail address: alodib@qu.edu.sa.

http://dx.doi.org/10.1016/j.jides.2016.10.010

Interactive SoA applications consider both functional, and non-functional requirements of the Web service's composition. The functional requirements describe the expected functioning of SoA services, and the UDDI support details the functional requirement. Non-functional requirements relate to the Quality of Service (QoS), (e.g. response, time, availability, through cost, price reliability, sustainability, interoperability, accuracy, and Service Level Agreements (SLAs)). SLAs are contracts perceived to exist between the service provide and users. Quality of Service (QoS) requirements are declared by imposing binding Service Level Agreements (SLAs).

The principal challenge facing the IoT ecosystem is the dynamic definition of SLAs. This is because of the absence of standardized mechanisms to represent these requirements. Moreover, the nature of the IoT, which allows the development of large, distributed and complex applications, complicates the challenge; each IoT site can only access the execution interfaces of those services deployed in different locations; i.e. it is impossible to manage the implementation of these services. In addition, the high level of the interoperability of Web services also complicates the case; i.e. applications developed using different programming languages are generally deployed in different locations. These challenges affect the process ti monitor the SLAs violation and take remedial actions to avoid customer dissatisfaction or financial penalties. Therefore, it is crucial to develop a method that facilitates the SLAs declaration process, and to provide a real-time monitoring of violations.

SLAs violations can be due to failures affecting underlying services. Another more subtle source of violation can be erroneous interactions between services resulting in undesirable scenarios. This type of violation relates to the underlying business process that governs interactions in services. In addition, receipt of different information from sensors can affect normal execution. A violation caused by a breakdown in each service is often dealt by exception handling, whereas detection of failure caused by the erroneous execution of a business process often requires the provision of additional information to monitor interactions between the services. Any type of breakdown has consequences in terms of customer satisfaction, and may result in a violation of a Service Level Agreement (SLA), resulting in fines.

This paper proposes a model-based method to expedite the integration of SLAs, providing a real-time monitoring. In this context, the SLAs defined for each site are mapped automatically to the UML QoS model. For each location, the composite application, which is an assembly of web services, is mapped into a Petri net. Then, the SLA model is transferred into a Petri net model, which is then combined with the site's Petri net. It captures actions that can be carried out when an SLA is violated. An algorithm is proposed here, to generate a real-time QoS-Aware monitoring service. This service is based on obtaining a detailed knowledge of the system; this involves proposing encoded information that relates to the state of the system. In this model, the part of the Petri net, which captures the normal behavior of the system (generated without SLAs modeling), is considered observable. Whereas, the part responsible for the actions executed as consequence of violations of SLAs is considered unobservable. The algorithm proposed here to compute the QoS-Aware service was inspired by diagnosis theory of Discrete Event System [7].

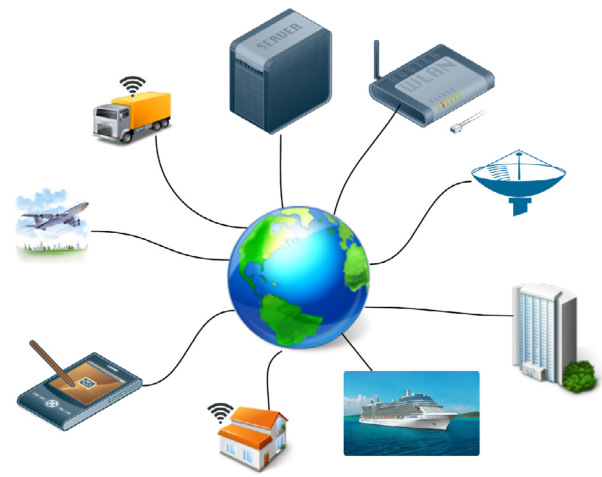

Fig. 1 – Internet of Things.

The paper is organized as follows. Section 2.1 presents a brief review of the Internet of Things (IoT). Section 2.2 presents fundamental of Petri net. In Section 3 a running example, based on a logistics company is described. In Section 4, an outline of the approach, and the associated solution are discussed.

2. Preliminaries

2.1. *Internet of Things (IoT)*

The Internet of Things (IoT) is an increasing growth domain, consisting of large, distributed, and complex wireless devices. This involves different things or objects, such as Radio-Frequency IDentification (RFID) tags, sensors, mobile phones, and radars [3,4]. Such things interact with each other, using efficient communications channels, as depicted in Fig. 1. The IoT plays a key role in, and has a high impact on our daily lives. It can form a large and distributed system with multiple different components.

Service oriented Architecture (SoA) over the cloud is used in the IoT to develop composite distributed applications [8]. Interactions between services are represented here, with the assistance of Business Process Execution Language (BPEL) [2]. BPEL is used to express complex behavior, such as sequential, parallel, iterative and conditional interactions. BPEL also incorporates model elements for specifying Reply, Receive, Invoke and Terminate [2,9], which are used in business processes.

2.2. *Petri net*

A Petri net [10,11] is a mathematical method used to model and analyze concurrent enterprises systems, comprising synchronous and asynchronous activities. This modeling language supports a graphical representation of the of behavior