**Capacity-Driven Web Services: Concepts, Definitions, Issues, and Solutions**

Samir Tata, TELECOM SudParis and CNRS UMR Samovar, France
Zakaria Maamar, Zayed University, UAE
Djamel Belaïd, TELECOM SudParis and CNRS UMR Samovar, France
Khouloud Boukadi, SFAX, Tunisia

**ABSTRACT**

This paper presents the concepts, definitions, issues, and solutions that revolve around the adoption of capacity-driven Web services. Because of the intrinsic characteristics of these Web services compared to regular, mono-capacity Web services, they are examined in a different way and across four steps denoted by description, discovery, composition, and enactment. Implemented as operations to execute at run-time, the capacities that empower a Web service are selected with respect to requirements put on this Web service such as data quality and network bandwidth. In addition, this paper reports on first the experiments that were conducted to demonstrate the feasibility of capacity-driven Web services, and also the research opportunities that will be pursued in the future.

Keywords: Adoption, Capacity, Composition, Requirement, Web Service

**INTRODUCTION**

It is widely known that Web services standards/specifications (e.g., Web Services Definition Language (WSDL) and Universal Description Discovery and Integration (UDDI)) are not keeping the pace with the increasing number of challenges that dynamic environments (e.g., ubiquitous) and modern enterprises pose on Web services (Ardissono, Goy, & Petrone, 2003; Langdon, 2003; Papazoglou, Traverso, Dustdar, & Leymann 2007). Sudden drop in network bandwidth, limited power of mobile platforms, and ad-hoc joint ventures are examples of these challenges. As a result, these standards/specifications are regularly subject to enhancements and extensions as the literature indicates (Mostéfaoui & Younas, 2007; Luo, Montrose, Kim, Khashnobish, & Kang, 2006; Beek, Gnisi, Mazzanti, & Moiso, 2006). Since Web services are here to fulfill the promise of developing loosely-coupled, cross-enterprise business applications, empowering them with other mechanisms on top of these extensions and enhancements would be highly appreciated by those who advocate for Web services’ benefits. In this paper we denote these mechanisms by capacities.

We embrace capacities to address the particular issue of Web services limited-ability (i.e.,}

DOI: 10.4018/jssoe.2010100105
in terms of willingness and responsiveness) to cater to different levels of service offers. Chatterjee et al. mention that “most Web services platforms are based on a best-effort model, which treats all requests uniformly, without any type of service differentiation or prioritization” (Chatterjee, Chaudhari, Das, Dias, & Erradi, 2005). Today’s Web services are designed and deployed without taking into account if the requests they receive originate from regular or new customers (Tao & Yang 2007) or from users who are at work or on the roads (Vukovic & Robinson, 2004). Some users are strict with the minimum network throughput to use while others are flexible with the freshness of the data to receive. Web services simply ignore the requirements and constraints of the environments in which users reside!

The objective of this paper is to discuss the design and development of capacity-driven Web services across the steps of description, discovery, composition, and enactment1. In the description step we show how to structure a capacity. In the discovery step we show how to look for relevant Web services based on their capacities and the environment’s requirements and constraints. In the composition step we show how the discovered Web services take part in composition scenarios. Last but not least, in the enactment step we show how to invoke capacities of Web services that are now engaged in these scenarios.

Our contributions are as follows: (1) provide a structured way to define and load capacities into Web services, (2) develop mechanisms to select and activate a capacity out of several ones in a Web service, and (3) extend current standards/specifications if necessary to support the characteristics of capacity-driven Web services. The rest of this paper is organized as follows: we provide a running example while discussing the foundations upon which the capacity-driven Web Services are built. Then we present how to describe, discover, compose, and enact capacity-driven Web Services. Next we review the implementation work that was conducted. Then we report on related work, followed by a conclusion and discussions of future work.

RUNNING EXAMPLE

Our running example concerns a real-state office that runs and sells different types of properties such as villas and flats. The office is equipped with a set of PCs, while the staff in charge of conducting the visits is equipped with various heterogeneous handheld devices. The staff is usually in contact with customers as per the following description. The customers contact the office to request an estimate, purchase, or rent a property. The office can also contact customers as per their initial requests. “Get the map of properties in the vicinity” is among the services that the real-state staffs use in their day-to-day business. This service returns all the properties that the office manages and is within a walking distance from a staff. This distance and other criteria like price range and number of bedrooms are set by the staff. “Get the map of properties in the vicinity” is treated as a composite Web service that relies on internal and external Web services as per the following description:

1. “LocateAgent” returns the current position of a staff in terms of latitude and longitude.
2. “Locate Properties” retrieves the list of properties (identifier, location, etc.) that are in the vicinity of the staff (at a maximum distance from her) and whose characteristics match her criteria.
3. “GenerateMap” produces a navigation map that shows some properties in response to the staff’s request.
4. “Display” displays the map on the device of the staff.

It is worth mentioning that “LocateAgent” and “GenerateMap” services are context aware. Their performance depends on the characteristics of the staff’s device: network interfaces type (GSM, WiFi, GPS, etc.), quality of the signal strength of the connexion, screen size, etc.
User-Centered Design of Mobile Geo-Applications

Validation and Verification of Software Systems Using Virtual Reality and Coloured Petri Nets