Chapter IV

Dynamically Adaptable
Web Services Based on
the Simple Object
Access Protocol

Kevin Curran, University of Ulster, UK

Brendan Gallagher, University of Ulster, UK

Abstract

Dynamic protocol stacks enable a developer to select a particular protocol profile at
bind time where each protocol profile is built from a rich library of protocol modules
including UDP, packet loss detection, data encryption, TCP, Multicast among others.
Communicating objects can be represented as object graphs that together realise
the required behaviour built upon the IP service offered by the host computer. All
protocols down to device driver level can be implemented at the user level, provid-
ing the maximum potential for configurability. The simple object access protocol
(SOAP) is a lightweight remote procedure calling (RPC) protocol for the exchange
of structured data in a decentralized environment. SOAP enables programs to run
and interoperate with other SOAP applications (called Web services) in a distributed environment. The SOAP protocol is based on extensible markup language (XML) and hypertext transmission protocol (HTTP), which, it is claimed, makes it a language and platform neutral vehicle for RPC over the Internet and through firewalls. This chapter describes a SOAP Web service deployed which enables clients to download protocol stack components as simple MIME attachments.

Introduction

Mobile devices, such as PDAs and smart phones, can access the Internet and they must also use networking protocols similar to those used by the average desktop computer. These networking protocols, called a protocol stack, are layered such as file transfer protocol (FTP), hypertext transmission protocol (HTTP) and transmission control protocol (TCP). However, the protocols underlying the Internet were not designed for the latest cellular type networks with their low bandwidth, high error losses and roaming users. Thus, many ‘fixes’ have arisen to solve the problem of efficient data delivery to mobile resource constrained devices (Saber, 2003). Mobility requires adaptability, meaning that systems must be location-aware and situation-aware, taking advantage of this information in order to dynamically reconfigure in a distributed fashion (Katz, 1994; Solon, 2003; Matthur, 2003). However, situations, in which a user moves an end-device and uses information services, can be challenging. In these situations the placement of different co-operating parts is a research challenge. The heterogeneity is not only static but also dynamic as software capabilities, resource availability and resource requirements may change over time. The support system of a nomadic user must distribute, in an appropriate way, the current session among the end-user system, network elements and application servers. In addition, when the execution environment changes in an essential and persistent way, it may be beneficial to reconfigure the co-operating parts.

A protocol stack consists of a linear list of protocol objects, which between them can support a range of quality of service such as reliable delivery, virtual synchrony, or encrypted communication. Our framework, Webber, provides the inter-layer services necessary for supporting new communication protocols. Webber consists of a set of Java classes for representing uniform resource locators, protocol stacks, the framework API and posting objects. Dynamically composable protocol stacks overcome the limitations imposed by generic protocol stacks allowing optimisation for particular traffic. Some uses that dynamic protocols may be used for include using the best-fit protocol for a particular network and application behaviour, so that performance can always be optimal; upgrading network protocols at run-time without having to restart applications and increasing security at run-time.
Dynamic Workflow Composition: Using Markov Decision Processes
[www.igi-global.com/article/dynamic-workflow-composition/3052?camid=4v1a](www.igi-global.com/article/dynamic-workflow-composition/3052?camid=4v1a)