Chapter XI
Semantic Web Service Discovery:
Methods, Algorithms, and Tools

Vassileios Tsetsos
University of Athens, Greece

Christos Anagnostopoulos
University of Athens, Greece

Stathes Hadjiefthymiades
University of Athens, Greece

ABSTRACT

This chapter surveys existing approaches to Semantic Web service discovery. Such semantic discovery will probably substitute existing keyword-based solutions in the near future, in order to overcome the limitations of the latter. First, the architectural components along with potential deployment scenarios are discussed. Subsequently, a wide range of algorithms and tools that have been proposed for the realization of Semantic Web service discovery are presented. Moreover, key challenges and open issues, not addressed by current systems, are identified. The purpose of this chapter is to update the reader on the current progress in this area of the distributed systems domain and to provide the required background knowledge and stimuli for further research and experimentation in semantics-based service discovery.

INTRODUCTION

Right after the Web infrastructure had matured enough, both academia and industry have recognized the necessity of enabling interactions between processes over the Web. The nature of the Web itself dictated that such enabling infrastructure should be loosely-coupled, based on open interfaces and, mostly, asynchronous communication. The outcome of the academic and industrial research in this field was the Web Services infrastructure, which has been standard-
Semantic Web Service Discovery

ized by W3C (World Wide Web Consortium). Web Services (WS) can be defined as “programmatic interfaces for applications (i.e., business logic), available over the WWW infrastructure and developed with XML technologies.”

The lifecycle of WSs includes the following main steps (see Figure 1):

- **Advertisement**: The process of publishing the description and endpoints of a WS in a service registry.
- **Discovery**: The process of locating all WSs that match the requestor’s functional requirements.
- **Selection**: The process of selecting the most suitable WS out of the discovered ones, usually based on application-dependent metrics (e.g., QoS).
- **Composition**: The process of integrating selected WSs into a complex process.
- **Invocation**: The process of invoking a single WS or complex process, by providing it with all the necessary inputs for its execution.

In practice, service discovery and selection are not always discrete processes, since the discovery algorithm may also rank the discovered services according to their relevance to the user inquiry. Out of all five steps, WS discovery is regarded the most important one, or, at least, it has attracted the major attention of the research community and industry. The main reason is that one cannot use a service unless she is aware of its existence or discovers it. Furthermore, the deviation of the provided service functionality from the requestor’s desired functionality heavily depends on the ability to locate the best available services.

At this point, we would like to discriminate the concept “Web service discovery” from the more generic concept “service discovery.” The first refers only to Web services while the latter can be applied to other types of services, too (e.g., printing services). Although both concepts are enabled by similar architectures, they use different protocols and technologies, and they apply to different application domains. Some indicative service discovery frameworks are Service Lo-