ABSTRACT

The term “Web services” has been used very often nowadays. According to W3C, "A Web service is a software system identified by a URI (Berners-Lee, Fielding, and Masinter ,1998), whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols "(Austin, Barbir, Ferris, and Garg, 2004). Currently, an increasing number of companies and organizations implement their applications over Internet. Thus, the ability to select and integrate inter-organizational and heterogeneous services on the Web efficiently and effectively at runtime is an important step towards the development of the Web service applications. Recent researches study how to specify (in a formal and expressive enough language), compose (automatically), discover and ensure the correctness of Web services. In this chapter, a significant portion of the work has been dedicated to scenarios aimed at both automating Web service discovery and composition functionality. As W3C defined (Booth et al.), “Discovery is the act of locating a machine-processable description of a Web servicerelated resource that may have been previously unknown and that meets certain functional criteria”. Indeed, Web service discovery is the process of finding a suitable Web service for a given task. When no atomic Web service can fulfill the user’s requirements, there should be a possibility to combine existing services together in order to satisfy the request requirement. This trend has inaugurated a considerable number of research efforts on the Web service composition (WSC) both in academia and industry. Most of current approaches related to WSC applied following techniques: HTN (Sirin, Parsia, and Hendler, 2005) , Golog (McIlraith and Son, 2002), classic AI planning (Rao et al.,2006), Rule-based planning (Medjahed, Bouguettaya, and Elmagarmid, 2003), model checking (Kuter et al., 2005) , theorem proving(Rao, Kungas, and Matskin, 2004), etc. Some approaches need too much human effort; some overlook the problem of discovery. Overcoming both discovery and composition of services is the key to automatic generation of executable process. In this chapter, an architecture, called AIMO, to do both Web service discovery, based on Web Service Modeling Ontology (WSMO) (WSMO working group), and WSC based on AIplanning is proposed. First, the AIMO architecture and all its components are described. Then, in order to automatically generate the control flow of the planning process of AIMO, an extension of HTN-DL formalism described in (Sirin, 2006), that combines Hierarchical Task Networks and Description Logics (Baader, Calvanese, McGuinness, Nardi, and Patel-Schneider, 2003) is proposed. The proposed extension involves the capability of Web service discovery using WSMO. On the other hand, the AIMO approach can solve the problem of WSC for WSMO using AIplanning (i.e., HTN-DL). This extension enhances the performance of HTN-DL and correctness verifiability of WSMO. Moreover, a translator to provide interaction between WSMO and AI-planning is proposed. AIMO continues to support loose coupling paradigm of SOA by separating the Web service discovery from the WSC.