

Chapter 2.6

A Logic Programming Based Framework for Intelligent Web Service Composition

Enrico Pontelli

New Mexico State University, USA

Tran Cao Son

New Mexico State University, USA

Chitta Baral

Arizona State University, USA

ABSTRACT

This chapter presents a comprehensive logic programming framework designed to support intelligent composition of Web services. The underlying model relies on the modeling of Web services as actions, each described by a logic programming theory. This view allows the use of logic-based planning to address the Web service composition problem, taking advantage of the fact that logic-based planning enables the elegant introduction of a number of extensions and generalizations (e.g., dealing with incomplete knowledge and preferences). The theory describing each Web service is encoded as a logic programming module, and different semantics are allowed within different modules, thus better reflect-

ing the practical use of different service description formalisms and ontologies.

INTRODUCTION

One of the main goals of the Semantic Web initiative (Berners-Lee, Hendler, & Lassila, 2001) is to extend the existing Web technology to support the development of intelligent agents, which can *automatically* and *unambiguously* process the information available in millions of Web pages. This led to numerous works on *Web services* and *Web service composition*. The primary goal of Web service composition is to determine an appropriate sequence of Web services to accomplish a user goal. The majority of the existing proposals dealing with Web service composition build on the principle of

DOI: 10.4018/978-1-60566-042-4.ch009

viewing Web services as *actions*, thus, representing the Web service composition problem as a *planning problem* that can be addressed using existing *planning* techniques. A second popular approach to Web service composition relies on techniques developed in the area of workflow development. The survey by Rao and Su (2004) provides a good overview of various proposals for Web service compositions.

McIlraith and Son (2002) propose to use *GOLOG* (Levesque, Reiter, Lesperance, Lin, & Scherl, 1997), a logic programming based language, for Web service composition. In such a proposal, each Web service is translated to a primitive action. *GOLOG* provides control-flow constructs, such as **if-then-else**, **while-do**, **sequence** (denoted by ‘;’), **procedure**, and **test** (denoted by ‘?’), which can be used to combine the primitive actions into programs. The resulting programs can be provided to a *GOLOG* interpreter for finding the sequence of Web services that need to be executed to achieve the goal of the user. Alternatively, the program can be given to an execution monitoring module, for direct execution. Sufficient conditions for a successful execution of a program are also provided. This direction of work has been adopted by Au, Kuter, and Nau (2005), Kuter, Sirin, Nau, Parsia, and Hendler (2005), and Wu, Parsia, Sirin, Hendler, and Nau (2003), where *SHOP2*, a hierarchical planning system, is used as the underlying system for automatic Web service composition. The latter work also addressed an important aspect of Web service composition, namely, the problem of incompleteness of information in Web service composition, by adding to the planning algorithm a module for gathering information during the planning process.

Viewing Web service composition as high-level planning is not only natural, but also advantageous for different reasons:

- AI planning has made remarkable progress in the last 10 years, and several robust and

scalable planning systems have been developed and are available for use, such as *FF* (Hoffmann & Nebel, 2001), *SHOP* (Nau, Cao, Lotem, & Muñoz-Avila, 1999), SAT-based planners (Kautz & Selman, 1996), and logic programming based planners (Lifschitz, 1999). All these planners can be used as the backbone in the development of systems for Web service composition with an architecture similar to the one described by McIlraith and Son (2002).

- AI planning allows the Semantic Web research community to focus on the development of Web service representation and reasoning languages and tools for translating Web service representation into a planning language.

Indeed, this view of Web service composition has been embraced by many researchers, and a number of tools have been proposed, for example, translators to map Web services encoded using DAML-S or OWL-S to PDDL (PDDL Technical Committee, 1998), a well-known planning language used by many planning systems.

While the use of planning in Web service composition is advantageous, there are a number of issues that need to be addressed before this approach can be widely applied. The first problem, which can be termed as the *service selection problem*, derives from the huge number of available services that can be used to achieve the same goal. The second problem lies in the lack of information in service composition, which imposes additional requirements on the planning system, such as the ability to plan with incomplete knowledge and to handle knowledge producing actions. The third problem centers on the *quality* of the composed service, which can be translated into the problem of planning with preferences, an area of research that has only recently attracted interest from the planning community. The fourth problem derives from the fact that each Web service is often encoded within an ontology or a knowledge base, whose

22 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/logic-programming-based-framework-intelligent/37641

Related Content

A Novel Approach to Construct Semantic Mashup using Patterns

Khayra Bencherif, Djamel Amar Bensaberand Mimoun Malki (2017). *International Journal of Information Technology and Web Engineering* (pp. 19-36).

www.irma-international.org/article/a-novel-approach-to-construct-semantic-mashup-using-patterns/170369

Predicting Temporal Exceptions in Concurrent Workflows

Iok-Fai Leong, Yain-Whar Siand Robert P. Biuk-Aghai (2012). *Models for Capitalizing on Web Engineering Advancements: Trends and Discoveries* (pp. 196-218).

www.irma-international.org/chapter/predicting-temporal-exceptions-concurrent-workflows/61907

The Influence of National Culture on Customer Service Experience: Case of China

Ying Ying Liao, Ebrahim Soltaniand Wei-Yuan Wang (2016). *Web-Based Services: Concepts, Methodologies, Tools, and Applications* (pp. 1388-1412).

www.irma-international.org/chapter/the-influence-of-national-culture-on-customer-service-experience/140857

Digital Keepsakes: Older Adults and the Extended Use of ICTs and Digital Artifacts

Declan Tuite (2015). *Artificial Intelligence Technologies and the Evolution of Web 3.0* (pp. 144-166).

www.irma-international.org/chapter/digital-keepsakes/127287

An Efficient Spatiotemporal Approach for Moving Object Detection in Dynamic Scenes

Min Liu, Yang Liu, Cong Liu, Juan Wangand Minghu Wu (2017). *International Journal of Information Technology and Web Engineering* (pp. 62-73).

www.irma-international.org/article/an-efficient-spatiotemporal-approach-for-moving-object-detection-in-dynamic-scenes/182265