

# An Optimal and Complete Algorithm for Automatic Web Service Composition

*Pablo Rodriguez-Mier, University of Santiago de Compostela, Spain*

*Manuel Mucientes, University of Santiago de Compostela, Spain*

*Juan C. Vidal, University of Santiago de Compostela, Spain*

*Manuel Lama, University of Santiago de Compostela, Spain*

---

## ABSTRACT

*The ability of web services to build and integrate loosely-coupled systems has attracted a great deal of attention from researchers in the field of the automatic web service composition. The combination of different web services to build complex systems can be carried out using different control structures to coordinate the execution flow and, therefore, finding the optimal combination of web services represents a non-trivial search effort. Furthermore, the time restrictions together with the growing number of available services complicate further the composition problem. In this paper the authors present an optimal and complete algorithm which finds all valid compositions from the point of view of the semantic input-output message structure matching. Given a request, a service dependency graph which represents a suboptimal solution is dynamically generated. Then, the solution is improved using a backward heuristic search based on the A\* algorithm which finds all the possible solutions with different number of services and runpath. Moreover, in order to improve the scalability of our approach, a set of dynamic optimization techniques have been included. The proposal has been validated using eight different repositories from the Web Service Challenge 2008, obtaining all optimal solutions with minimal overhead.*

*Keywords:* A\* Algorithm (A Star Algorithm), Dependency Graph, Semantic Input-Output Matching, Web Service Composition, Web Services

---

## 1. INTRODUCTION

Nowadays, Service-Oriented Architectures (SOA) (Papazoglou & Georgakopoulos, 2003) are gaining importance because of the ability to build interoperable services that can be shared

over a network within multiple platforms. Thus, companies are starting to apply this principle to their business, allowing them to remain cost effective, flexible and competitive. Applications in SOA are built based on services consumed by clients that are not concerned with the underlying implementation. Specifically, web

DOI: 10.4018/jwsr.2012040101

services are the preferred standard-based way to realize SOA.

Web Services are self-contained modular applications described by a collection of operations that are network-accessible through standardized web protocols, and whose features are defined using a standard XML-based language (Alonso, Casati, Kuno, & Machiraju, 2004). One of the advantages of web services is to enable greater and easier integration and interoperability among systems and applications through web service composition. This advantage allows web services to be combined by connecting their inputs and outputs to create larger services (composite services) whose execution is orchestrated by a set of control structures defined in composition languages like WS-BPEL (Weerawarana, Curbera, Leymann, Storey, & Ferguson, 2005; Rouached, Fdhila, & Godart, 2010). Thus, the goal of web service composition is to construct new services from existing web services in order to satisfy a request (basically a set of provided inputs and a set of wanted outputs by the client) which cannot be solved by a single web service. The matching between inputs and outputs can either be done syntactically, using the information described in WSDL (Christensen, Curbera, Meredith, & Weerawarana, 2001), or semantically, using semantic markup languages like OWL-S (Burstein et al., 2004) or WSMO (de Bruijn et al., 2005).

The automatic composition problem may seem trivial problem when there are a limited number of services in a single-service architecture. However, the problem increases in complexity when the goal is to obtain optimal compositions over large web service repositories using different control structures to manage the composition flow. In fact, the web service composition problem can be reduced to the boolean satisfiability problem, i.e., the problem is NP-complete and therefore it cannot be solved in polynomial time (Lee & Kumara, 2005).

Research in this field has grown rapidly in recent years. Some approaches (Hoffmann, Bertoli, & Pistore, 2007; Sirin, Parsia, Wu, Hender, & Nau, 2004; Klusch, Gerber, & Schmidt, 2005; Pistore, Barbon, Bertoli, Shaparau, &

Traverso, 2004; Xu, Chen, & Reiff-Marganiec, 2011) treat the service composition as an artificial intelligence (AI) planning problem, where a sequence of actions lead from a initial state (inputs and preconditions) to a goal state (required outputs). These techniques work well when the repository size is relatively small and the number of constraints is high. However, most of these proposals have some drawbacks: high complexity, high computational cost and inability to maximize the parallel execution of web services.

Other approaches (Aversano & Taneja, 2006; Ghafarian & Kahani, 2009; Rodriguez-Mier, Mucientes, Lama, & Couto, 2010) scale better than other techniques when the interactions among services and the number of constraints is huge. Despite being scalable, these techniques do not guarantee to obtain the optimal solution, and also are extremely slow and memory intensive. The most recent approaches (On & Larson, 2005; Kona, Bansal, Blake, & Gupta, 2008; Yan, Xu, & Gu, 2008; Wu, Li, Wu, & Yin, 2011; Weise, Bleul, Kirchhoff, & Geihs, 2008; Shiaa, Fladmark, & Thiell, 2008; Hennig & Balke, 2010; Hashemian & Mavadat, 2006; Jiang, Zhang, Huang, Chen, Hu, & Liu, 2010), consider the problem as a graph/tree search problem, where a search algorithm is applied over a sub-optimal graph in order to find a optimal (or near-optimal) solution. These proposals are simpler than the AI planners due, in part, to the use of a smaller number of constraints during the search. However, most of these approaches rely on very complex dependency graphs that have not been optimized to reduce data redundancy. Therefore, the scalability of these algorithms may also be adversely affected when the interaction among services and data is huge due to the redundancy of the repository.

This paper addresses the problem of the web service composition as a graph search problem from the point of view of the semantic input-output message structure matching, i.e., we do not take into consideration the non-functional properties (NFPs). The novelties of our proposal are:

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

[www.igi-global.com/article/optimal-complete-algorithm-automatic-web/70387?camid=4v1](http://www.igi-global.com/article/optimal-complete-algorithm-automatic-web/70387?camid=4v1)

This title is available in InfoSci-Journals, InfoSci-Journal Disciplines Computer Science, Security, and Information Technology, InfoSci-Digital Marketing, E-Business, and E-Services eJournal Collection, InfoSci-Networking, Mobile Applications, and Web Technologies eJournal Collection, InfoSci-Journal Disciplines Business, Administration, and Management, InfoSci-Select. Recommend this product to your librarian:

[www.igi-global.com/e-resources/library-recommendation/?id=2](http://www.igi-global.com/e-resources/library-recommendation/?id=2)

## Related Content

---

A Model-based Approach for Diagnosing Fault in Web Service Processes  
Yuhong Yan, Philippe Dague, Yannick Pencolé and Marie-Odile Cordier (2012).  
*Innovations, Standards and Practices of Web Services: Emerging Research Topics*  
(pp. 92-116).

[www.igi-global.com/chapter/model-based-approach-diagnosing-fault/59920?camid=4v1a](http://www.igi-global.com/chapter/model-based-approach-diagnosing-fault/59920?camid=4v1a)

An Access-Control Framework for WS-BPEL

Frederica Paci, Elisa Bertino and Jason Crampton (2008). *International Journal of Web Services Research* (pp. 20-43).

[www.igi-global.com/article/access-control-framework-bpel/3122?camid=4v1a](http://www.igi-global.com/article/access-control-framework-bpel/3122?camid=4v1a)

## F-DRARE: A Framework for Deterministic Runtime Adaptation of Cyber Physical Systems

Fahad Bin Tariq and Sandeep Korrapati (2013). *Adaptive Web Services for Modular and Reusable Software Development: Tactics and Solutions* (pp. 263-276).

[www.igi-global.com/chapter/drare-framework-deterministic-runtime-adaptation/69478?camid=4v1a](http://www.igi-global.com/chapter/drare-framework-deterministic-runtime-adaptation/69478?camid=4v1a)

## Big Data Analytics in Health Care

Keerthi Suneetha (2019). *Web Services: Concepts, Methodologies, Tools, and Applications* (pp. 377-386).

[www.igi-global.com/chapter/big-data-analytics-in-health-care/217840?camid=4v1a](http://www.igi-global.com/chapter/big-data-analytics-in-health-care/217840?camid=4v1a)