Benefits of Semantics on Web Service Composition from a Complex Network Perspective

Chantal Cherifi\textsuperscript{1,2}, Vincent Labatut\textsuperscript{1}, and Jean-François Santucci\textsuperscript{2}

\textsuperscript{1} Galatasaray University, Computer Science Department, Ortaköy Istanbul, Turkey
\textsuperscript{2} University of Corsica, UMR CNRS, SPE Laboratory, France
vlabatut@gsu.edu.tr

Abstract. The number of publicly available Web services (WS) is continuously growing, and in parallel, we are witnessing a rapid development in semantic-related web technologies. The intersection of the semantic web and WS allows the development of semantic WS. In this work, we adopt a complex network perspective to perform a comparative analysis of the syntactic and semantic approaches used to describe WS. From a collection of publicly available WS descriptions, we extract syntactic and semantic WS interaction networks. We take advantage of tools from the complex network field to analyze them and determine their properties. We show that WS interaction networks exhibit some of the typical characteristics observed in real-world networks, such as short average distance between nodes and community structure. By comparing syntactic and semantic networks through their properties, we show the introduction of semantics in WS descriptions should improve the composition process.

Keywords: Web Services, Service Composition, Complex Networks, Interaction Networks, Semantic Web.

1 Introduction

A Web Service (WS) is an autonomous software component which can be published, discovered and invoked for remote use. For this purpose, its characteristics must be made publicly available, under the form of a so-called service description. This file comparable to interfaces defined in the context of object-oriented programming lists the operations implemented by the service. Currently, production WS use syntactic descriptions expressed with the WS description language (WSDL), which is a W3C (World Wide Web Consortium) recommendation. Such descriptions basically contain the names of the operations and their parameters names and data types, plus some lower level information regarding the network access to the service. WS were initially designed to interact with each other, in order to provide a composition of WS able to offer higher level functionalities [1]. Current production discovery mechanisms support only keyword-based search in WS registries and no form of inference nor flexible match can be performed [2]. More advanced research (non-production)
approaches rely on comparing structured data such as parameters types and names, or analyzing unstructured textual comments [3-5]. This is generally not enough to distinguish WS in terms of functionality, and consequently makes it difficult, or even impossible, to use these methods to automate WS composition. Indeed, syntactically discovered WS must be manually validated to ensure they implement the desired behavior, leading to static, a priori compositions.

To solve this limitation, the WS research community introduced semantics in WS descriptions, through the use of new semantic description languages. Different formats exist, among which we can distinguish purely semantic descriptions (OWL-S, a W3C recommendation), from annotated WSDL descriptions (WSDL-S and SAWSDL). Although those languages allow to associate ontological concepts with various elements of the description, the research community has been focusing only on the concepts qualifying the operations inputs and outputs. Retrieving semantic information is far more costly than collecting syntactic descriptions, even when considering only parameters. The latter can be performed quickly and completely automatically. The former is a long task, requiring human intervention to label each parameter with the proper concept. Annotation tools exist to help, but they are clearly not mature yet, and often defined for specific collections or languages [6, 7]. Maybe for these reasons, no semantic annotation language emerged as an industry standard. Although they appeared more than five years ago, all production WS still rely on WSDL. Even at a research level, very few publicly available significant collections of semantically annotated WS exist, making it very difficult to test new algorithms.

This situation leads to one question: is describing WS semantically worth the cost? To our knowledge, no one did ever compare the information underlying syntactic and semantic WS descriptions. In this work, we try to tackle this problem from the service composition perspective, through the use of complex networks. We consider a set of WS as a broad interaction space, in which WS are related if they can be chained up in a composition process. We model this space by building so-called interaction networks, based on syntactic and semantic descriptions of a given WS collection. We assume that the information conveyed by the two different kinds of descriptions appears in the corresponding interaction networks. We then compare the syntactic and semantic descriptive approaches through the networks topological properties. Our main contributions are the formal definition of three types of semantic networks, an extended investigation of the WS networks topology and the comparison of syntactic and semantic networks. In section 2, we present complex networks and their main topological properties. Section 3 introduces interaction networks and explains how they can be extracted from WS descriptions. Section 4 is dedicated to the presentation and discussion of our experimental results, i.e. the obtained networks, their topological properties and how they compare. Finally, in section 5, we emphasize the original points of our work, discuss its limitations and their possible solutions.

2 Complex Networks Properties

Complex networks are a specific class of graphs, characterized by a huge number of nodes and non trivial topological properties. Used in many different fields to model real-world systems [8], they have been intensively studied both theoretically and practically [9]. Because of their complexity, specific tools are necessary to analyze