IntroductionC:intro

Problem Statement Service-oriented computing (SOC) is a novel computing paradigm that employs services as fundamental elements to achieve the agile development of cost-efficient and integrable enterprise applications in heterogeneous environments papazoglou2003service, papazoglou2006p. One of the primary purposes of SOC is to overcome conflicts due to diverse platforms and programming languages to make integrable and seamless communication among those existing or newly built independent services. Service Oriented Architecture (SOA) could abstractly realise service-oriented paradigm of computing. This accomplishment has been contributing to reuse of software components, from the concept of functions to units and from units to services during the evolution of development in SOA booth2004web, overdick2007resource. One of the most typical implementation of SOA is web service, which designated as "modular, self-describing, self-contained applications that are available on the Internet" curbera2001web. Several standards play a significant role in registering, enquiring and grounding web services across the web, such as UDDI curbera2002unraveling, WSDL lausen2007semantic and SOAP fensel2011semantic. Web service composition aims to loosely couple a set of web services to provide a value-added composite service that accommodates complex functional and non-functional requirements of service users.

Two most notable challenges for web service composition are ensuring interoperability of services and achieving Quality of Service (QoS) optimisation fensel2011semantic. Interoperability of web services presents challenge in syntactic and semantic dimensions. The syntactic dimension is covered by the XML-based technologies yu2008deploying, such as the previously discussed $WSDL$ and $SOAP$. In this dimension, most of service compositions are merely based on the matching of input-output parameters. The semantic dimension enables a better collaboration through ontology-based semantics o2005review, in which many standards have been established in this dimension. E.g., OWL-S martin2004owl, Web Service Modeling Ontology (WSMO) lausen2005w3c, SAWSDL kopecky2007sawsdl, Semantic Web Services Ontology (SWSO) petrie2016web. This dimension bring around some other services' resources that could effect the execution of web services and their composition (i.e., precondition and postcondition). On the whole, these two challenges give birth to Semantic web services composition and QoS-aware service composition. Semantic web services composition is distinguished from the tranditional service composition (i.e., only syntactic dimension presented in web services). The resources of semantic web services are described semantically to enable a better interoperability for chaining web services. Another challenge is related to QoS optimisation. E.g., minimum cost and maximum reliability. This problem gives birth to QoS-aware service composition that aims to find composition solutions with optimised QoS.

Further more, the environment of service composition is changing in the real world, rather than static. E.g, QoS values of services being composed of are fluctuating over time, or service chosen at the planning stage may not avaible to be invoked at the runtime. Most of importance is static web service composition supports the environment change badlly because of outdated composition solutions. Therefore, Dynamic web service composition become a very demanding research field with a growing interest for providing solutions that adapt to the changing environment. Additionly, in context of semantic web service composition, semantics of web services can make the problem of dynamic web service composition more complicated for the changing ontology.

Different approaches have been proposed to solve those composition problems discussed above and they can be classified into two main categories: semi-automated web service composition and fully automated web service composition. The first composition problem requires human beings to manually create abstract workflows. Generally, researchers assume the pre-defined abstract workflow is given and provided by the users. The optimisation problem in this approach turns to selecting the concrete services with the best possible quality to each slot of the given workflow. Due to a tremendous growth in industries and enterprise applications, the number of web services has increased dramatically and unprecedentedly. The process of conducting abstract workflows manually is fraught with difficulties. Therefore, fully automation of the composition process was introduced in web service composition for less human intervention, less time consumption, and high productivity. The advantages of fully automated approach is that an abstract workflow is not not provided, but it is established while service are selected.

Generating composition plans automatically in discovering and selecting suitable web services is a NP-hard problem moghaddam2014service, which means the composition solution is not likely to be found with reasonable computation times in a large searching space. Artificial Intelligence (AI) planning-based approaches, Evolutionary Computation (EC) techniques and hybrid techniques are introduced to handle this problem. AI planning problem is solved in making a plan, from initial states to a set of actions to desired goal states-composite web services, where services are considered as actions triggered by one state (i.e., inputs) and resulted in another state (i.e., outputs). In the second approach, heuristics have to be employed to generate near-optimal solutions, where a variety of EC techniques have been used in this context. E.g., Genetic Algorithms (GA), Genetic Programming (GP) and Particle Swarm Optimisation (PSO). EC-based techniques have been effectively proposed to solve QoS-aware web service composition problems with different designed data structures for representation. Recently, different solution representation utilised in different EC-based method have been investigated in QoS-aware web service composition problems, since they could significantly impact on the performance while performing fully automated service composition. In the third approach, a hybrid of AI planning-based approaches and EC-based approaches are proposed to fulfil the correctness in constructing workflows with users' constraints, while the quality of composition solutions are also optimised according to users' requirement.

The overall goal of this thesis is to propose a comprehensive-quality aware automated web service composition. This comprehensive quality aims to jointly optimise semantic matchmaking quality and QoS. Meanwhile, this new approach also tackles serveral service composition problems, such as semantic web service compostion, multi-objective optimisation, dynamic web service composition.

Motivations The motivations of automated semantic web service composition lies in the requirements from three key aspects that simultaneously account for. enumerate*

Quality of service composition: previous studies have suggested many approaches to optimise QoS, which refers to the non-functional quality of service composition. However, the importance of the functional quality (i.e, semantic matchmaking quality) are not recognised in these works.

Composition constructs:

dynamic service composition: