ABSTRACT

This thesis presents our work concerning slicing of SOA-based software and its application to testing. A technique for static slicing of SOA-based software based on SoaML service interface diagram is devised. In this technique, a model using SoaML service interface diagram is designed. Then, an intermediate representation, called service interface dependency graph (SIDG) is created from the model. The SIDG identifies service call dependency and composite dependency from SoaML model. Giving slicing criterion as an input, our proposed algorithm SSSIM traverses the SIDG and identifies the affected service interface nodes.

Next, our intermediate representation SIDG is extended to be able to compute dynamic slices based on SoaML sequence diagram. In our technique, first each message in sequence diagram is mapped with corresponding web service input and output messages. This mapping is static. After that, an intermediate representation of SoaML sequence diagram is constructed which we have named service-oriented software dependence graph (SOSDG). SOSDG is an intermediate representation that needs to be stored and traversed to get a dynamic slice as and when web service gets executed. The SOSDG identifies data, control, intra-service and inter-service dependencies from SoaML sequence diagram and the corresponding web service execution. For a given slicing criterion, our proposed algorithm MBGDS computes global dynamic slice from SOSDG and identifies the affected messages from respective service. The novelty of our work lies in the computation of global dynamic slice based on SOSDG and its dependencies induced within and across organizations.

To support testing of SOA-based software at development phase, an extension to WSDL is proposed for carrying out black-box testing. In this context, program slicing artifact, i.e. dependence graph is applied for testing SOA-based software. In this approach, a hierarchical structure is imposed on web service description language (WSDL) document. The extension to WSDL is carried out by introducing web service dependence graph (WSDG) and dependencies, both data and control in XML schema definition (XSD) document. This WSDG identifies both data and control dependency from WSDG.
Next, techniques to test SOA-based software using business process modeling notation (BPMN) diagram and SoaML service interface diagram are presented. To test it using BPMN diagram, first BPMN diagram is converted into its control flow graph (CFG) using our proposed algorithm. Then depth first search (DFS) method is used to generate test paths. Finally, test cases are executed on the generated test paths. Our second technique of testing uses SoaML service interface diagram to model SOA-based software. Next, XML schema and its instance are generated using the visual paradigm enterprise tool. At last, test cases are executed based on schema constraints.

Finally, for all the above techniques or approaches, prototype tools are developed to experimentally verify the correctness and preciseness of our proposed algorithms or techniques. The computed slices and generated test cases for several models or web services are found to be correct using both, manual procedure and our prototype tools.