Research Statement

Michael Ruth

mruth@roosevelt.edu

My research focuses on the intersection of Web services and software engineering, specifically on software design patterns, message exchange patterns, and software testing for Web services.

Research Accomplishments

Message Exchange Patterns

The processing time for an average call to some services was on the order of hours, in which time the calling applications would have to remain connected. Since this was clearly not optimal and message passing is an integral part of Web service functionality, the available message exchange patterns, such as request/response, were examined for a means of supporting callback. Web service Description Language (WSDL), which defines the interfaces and the message exchange patterns that those interfaces can use, does not directly support callback. Additionally, network security, specifically firewalls, was an additional concern because in most enterprises the calling application is behind a firewall and may not be reachable from a service outside of the firewall.

Clearinghouse Web Service

In order to combat the network security barrier, a framework was developed [1] using the notion of a clearinghouse. The clearinghouse was developed as a set of Web services which are accessible outside the firewall, which can then relay the response messages back to the original calling application using a variety of means, including sockets. Additionally, code generation tools were developed so that developers wishing to use the approach and framework could generate client-side agents which make the calls and asynchronously wait for the response from the clearinghouse.

Single-Request/Multiple-Response Messaging Pattern

In developing the approach, a close relative to callback, namely the single-request/multiple-response messaging pattern was considered. This pattern is a key ingredient in notification schemes, publish/subscribe schemes, and in batch processing. Instead of matchmaking the request and response, the clearinghouse matches several responses to the original request [2]. This framework was developed and has seen use in a variety of applications, including a forensics image matching database application.

Resumable Clients

Resumable clients are those clients which call a service asynchronously, shut down, and then resume sometime in the future without a loss in progress. Some examples of resumable clients are mobile applications or a client application which only needs the data being requested. Some mapping services fit into the latter category, because the map would be downloaded, displayed, and then the user would quit the application. The frameworks were modified to support resumable clients using both the callback [3] and the single-request/multiple-response message patterns [4].

Regression Testing Web Services

Regression testing is an economical means of verifying that a modified system is not adversely affected by those modifications. A key element in regression testing is regression test selection (RTS), which aims to reduce the cost of performing regression testing by removing test cases. Safe RTS techniques add an additional constraint by asserting that no modification revealing tests will be left unselected. Since safe RTS techniques involve white-box testing, they cannot be directly applied to Web services in an end-to-end manner, which is important due to the nature of Web services possibly composed of other Web services.

Safe RTS Technique for Java Web Services

A framework was developed for applying a safe RTS technique to Java-based Web services in an end-to-end manner [5, 6]. The framework transformed composite and simple services developed in Java into standalone monolithic code so that dynamic analysis of the code could take place. Although the framework was not interoperable, it did allow the use of the safe RTS technique developed by Rothermel and Harrold [7], which was based on Java-based control-flow graphs (CFG). Another limitation of the framework was that it was also limited to those services which were developed by the same service provider, since it required all the requisite code for all the services in the interaction, which was a severe limitation of the approach.

Safe RTS Technique for Web Services

After developing an approach for Java Web services, the approach was broadened to more general Web services [8]. This approach was the first safe RTS technique to be applied to end-to-end Web services. This approach also uses CFGs since they are ideal for use in Web service environments for a number of reasons. First, CFGs can be generated from programs written in any language, or extracted from designs at any granularity. Thus, they can be used as a common representation mechanism among Web services which could be written in any language on any platform. Second, since CFGs are special cases of finite-state machines, they can be composed into global CFGs. These two characteristics of CFGs are essential for supporting both the interoperability and composition of Web services.

In the approach, composite services have to build their CFGs, their test cases, and mappings of their test cases to their CFGs using the CFGs, test cases, and mappings of test cases to CFGs of the services they call. A limitation of this approach is that it requires a great deal of work on the part of the developers of the services in terms of updating and propagating the CFGs in order to perform the RTS and RT processes.

Automating RTS and RT for Web Services

After developing the approach, an approach was developed to automate the regression testing and RTS process through monitoring, exchanging, and updating CFGs between related parties using a decentralized event notification scheme [9]. The automation of the RTS framework is carried out by a set of distributed agents, one for each service and application, which interact together to perform the regression test selection and regression testing processes.

Concurrency in RTS Frameworks for Web Services

The automation of the approach provides an entirely new set of challenges. More specifically, the handling of concurrent modifications becomes critical [10]. There are a number of challenges inherent in concurrency in such a system, but the major hurdle was test consistency, which involves ensuring that each test case has a consistent view of the system under test. The developed approach ensures that once the system reaches a stable-state, the last set of test cases are guaranteed to be consistent, which was called eventual test consistency.

Empirical Analysis of RTS Techniques for Web Services

Additionally, in order to validate the approach, a group of five Web service-based systems were developed as a benchmark to perform an empirical analysis of the approach [11]. The systems were developed because there were no preexisting empirical studies of RTS techniques for Web services. Each of the five systems has roots in the literature and was instrumented with CFGs, test cases, and coverage information. After which, the technique was performed and compared the cost of the technique to the “select-all” technique, which simply runs all tests with no selection step. The study demonstrated that the approach is both feasible and can be effective in reducing the cost of performing regression testing for the systems developed.

Current Research

I am currently researching the information sharing aspect of RTS. In order to perform end-to-end RTS, some information must be shared between the participants in an interaction. Service providers are unlikely to share implementation details due to intellectual property concerns. In the approach, the information which must be shared was carefully considered to ensure maximum participation through decentralized control and information hiding. The information which must be shared was minimized into requiring only the CFGs of each operation the service provides, test cases which cover the CFG, and coverage information mapping the test cases to the CFGs. In addition, each participant maintains control over the granularity of the CFGs they provide which can vary from very detailed (statement) to very abstract (operation), depending on the needs of the service provider. Also, the source code, which is contained in the CFGs, is shielded from the testers, which allows the testers to identify what part of the system was modified, without knowing exactly how it was modified. In summary, I wish to compare our approach to other RTS frameworks for Web services in terms of their information sharing. I expect to develop this idea further in the next few months, and be looking for a conference venue shortly after. In addition, I have been working on a journal article detailing my work on safe RTS for Web services, which I am expecting to have completed later this year, and am looking for a venue for that now.

Future Research Plans

Benchmark for Comparing RTS techniques for Web Services

The performance and selectivity of RTS technique in traditional application is normally determined using a set of standard programs which can be used to test the technique. Since this standard does not exist for Web services, its development is of specific interest to me because it would address some standardization issues in relation to how well these RTS techniques perform. This particular avenue implies the creation and implementation of a standardized benchmark which is representative of Web services as a whole. The systems developed in my empirical analysis need some adjustments in terms of their representativeness of Web services. There needs to be greater variety of interactions between the involved services. The development of a standard benchmark would provide a means of comparing and contrasting RTS techniques for Web services in a unique and open way. Additionally, this would open a new avenue for research, namely actually comparing two techniques for Web services developed for performance.

Concurrency and Fault Locatability

In my earlier work, a framework was developed which provides eventual test consistency, but did not provide an alternative. This is another problem I am very interested in because in some ways, it left some of the concurrency concerns unsolved. Specifically, I want to focus on solving fault locatability, which provides the testers with the location of the modification which produced the fault, should a fault occur, regardless of where the fault occurred within the system. Solving fault locatability is possible, but it does require synchronization of some modifications, which can be very restrictive. Synchronization may imply that only one modification may occur in a system at any given time and no new modification can occur until the last modification was completely tested. However restrictive, it may prove to be invaluable for those enterprises in which that type of synchronization is necessary and the restrictions are acceptable.

Data Flow-Based RTS Techniques

In addition to regression testing using CFG-based approaches, I would like to focus on data-flow based techniques. This is a direct extension of my earlier approach since data-flow techniques use CFGs as a means to analyze software for determining the lifecycle of variables which flow in and out of the services. Extending this approach would require decentralizing control of the technique so that it can be performed at various parts of an interaction. Another possible caveat of extending the approach to data-flow analysis is that it may require the sharing of too much information, which would prevent service providers from participating, but I believe this caveat can be overcome.

Student Involvement

Student involvement in research at the undergraduate, as well as the graduate, level is very important to me. Students can contribute to my research by contributing their thoughts and ideas which may complement or even extend my own. A small group of undergraduates assisted with the empirical analysis portion of my dissertation and they offered some very interesting ideas on how to improve the project itself [9]. Their ideas provided me with much needed insight on how to improve the benchmark to become more representative of Web services as a whole. They certainly had an impact on my scholarly work and on my attitude towards undergraduate involvement. There are parts of my research plan which can be segmented into small projects for use in the classroom, which would allow the students to put into practice many of the concepts and skills they are concurrently learning in the classroom, or as semester long project for independent study.

References

[1] Ruth, M., Lin, F, and Tu, S., "*A Client-Side Framework Enabling Callbacks from Web Services*", Proceedings of the 3rd European Conference on Web Services (ECOWS'05), Vaxjo, Sweden, pp. 105-116, Nov. 2005

[2] Ruth, M., Lin, F, and Tu, S., "*Adapting Single-Request/Multiple-Response Messaging to Web Services*", Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC'05), Edinburgh , UK , pp. 287-292 , July 2005

[3] Ruth, M., Lin, F., and Tu, S., "*A Framework for Web Services with Callbacks for Resumable Clients*", Proceedings of the 1st International Conference on Next Generation Web Services Practices (NWeSP'05), Seoul, South Korea, pp. 412-417, Aug. 2005.

[4] Ruth, M., Lin, F., and Tu, S., *"A Client-side Framework for Resumable Applications to Utilize Web Services With SRMR Callbacks"*. International Journal of Web Service Practices, 2005, Vol. 1, Issue 1 pp. 73-88

[5] Lin, F., Ruth, M., and Tu, S., "*Applying Safe Regression Test Selection Techniques to Java Web Services*", Proceedings of the 2rd International Conference on Next Generation Web Services Practices (NWeSP'06), Seoul, South Korea, pp. 133-140, Aug. 2006.

[6] Ruth, M., Lin, F., and Tu, S., "*Applying Safe Regression Test Selection Techniques to Java Web Services*". International Journal of Web Service Practices, 2006, Vol. 2, Issue 1-2, pp. 1-10

[7] Rothermel, G., and Harrold, M. J., "A Safe, Efficient Regression Test Selection Technique", ACM Transactions on Software Engineering Methodology, vol. 6, no. 2, pp. 173-210, Apr. 1997.

[8] Ruth, M., and Tu, S., "*Applying Safe Regression Test Selection to Web Services*", Proceedings of the 2nd International Conference on Internet and Web Applications and Services (ICIW'07), pp. 278-89, Mauritius, May 2007

[9] Ruth, M., Oh, S., Loup, A., Horton, B., Gallet, O., Mata, M., Tu, S., "*Towards Automatic Regression Test Selection for Web Services*", International Workshop on Testing Emerging Software Technology 2007 (TEST '07), Proceedings of International Conference on Computer and Software Applications 2007 (COMPSAC '07), Beijing, China, pp. 729-736, July 2007

[10] Ruth, M., Tu, S., "*Empirical Studies of a Decentralized Regression Test Selection Framework for Web Services*", Proceedings of the 2008 Workshop on Testing, Analysis, and Verification of Web Services and Applications, (TAV-WEB '08), pp. 8-14, Seattle, Washington, Jul. 2008

[11] Ruth, M., "*Concurrency in a Decentralized Automatic Regression Test Selection Framework for Web Services*", Proceedings of the Mardi Gras Conference, (SIGAPP MG'08), Art. 7, Baton Rouge, LA, Feb. 2008