

On Robustness of Service-based Applications

by

Andrew Roczniak

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements
for the Ph.D. degree in
Electrical and Computer Engineering

School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

© Andrew Roczniak, Ottawa, Canada, 2008

Abstract

Service-based applications is the name given to a situation arising when a user or a group of users interact with a set of distributed and autonomous functionalities such that some value is created for those users. For interoperability purposes, this interaction should take place through open and standardized interfaces.

Collaborative applications allow users to participate in achieving a common goal or objective. From a basic collaborative application that allows participants to author a multimedia document, three distinct functionalities were extracted and implemented as distributed and autonomous services. An example on how a group of users might interact with those services using standard set of protocols and off-the-shelf techniques is presented.

From the point of view of a user, a service-based application can only be useful if it provides some indication about its robustness. Creation of such applications therefore must consider the issue of undependability of a required service. A particular source of service failure (service provider refuses a service) or degraded performance (occasional lack of service or degraded quality of service) can be attributed to producers self-interested behavior.

The state of the art in dealing with the effects of rational behavior comprises trust and reputation networks, and methods based on game theory and mechanism design. These are presented and discussed in a broader survey of available strategies, tools and available techniques to analyze and mitigate effects of rational behavior.

The basis for the investigated solution is the observation that part of a service may be redundant with respect to some part of another service. Instead of adding redundancy to the system, our solution exploits the already existing redundancy to improve robustness of these applications.

The interaction with services is modeled as finite-state transducers and a heuristic is proposed to obtain redundancy between any pair of services for which an interaction model exists. Then, an algorithm that uses this inter-service redundancy to recreate the interaction with one service (target) from the other (candidate) is described. This approach allows users to have access to some specific functionality but not receive it from a service which may be deemed problematic. A qualitative reference framework to classify available services into candidate and target services respectively is also discussed.

The computational cost is polynomial with respect to services size, and in practice, the non-redundant functionality and related control need to be implemented locally.

Acknowledgements

Graduate studies are a bit like a journey through an unknown country. It can be difficult at times, but it is ultimately a deeply rewarding and enriching experience. I am indebted to my advisors, Dr. Abdulmotaleb El Saddik and Dr. Pierre Lévy for the opportunity to undertake this journey, for their guidance and their support. They were instrumental in guiding me to my destination while encouraging me to take roads less traveled but more scenic.

Long journeys are usually made after a suitable preparation. For help with this, I am indebted to Dr. Emil Petriu and Dr. George Costache.

I am grateful to my wife, my parents and my sister for their unwavering support and encouragements. I could not have done it without you!

Finally, I'd like to thank my fellow travelers for their encouragements, tips and enlightening conversations: Shawn Ferguson, Darek Baingo, Christian Champenois, Dima Kabrelyan, Max Nozin, Nick Soveiko, Daniel Shteyn, Cyril Wisniewski, Jonathan Layes and Jeff Fiedorowicz, you made my journey that much more interesting!