



Fault-Based Combinatorial Testing of Web Services



Bellanov Apilli
bsapilli@ncsu.edu

Lydia Richardson
lr00018@georgiasouthern.edu

Cory Alexander
crayboy711@yahoo.com

Dr. Kera Bell-Watkins, Advisor
kzbell@georgiasouthern.edu

Background

- Internet houses diverse applications (i.e., banking, networking, etc.), commonly implemented as web services
- We propose fault-based combinatorial testing and compare its fault-detection capability to current web service testing techniques.

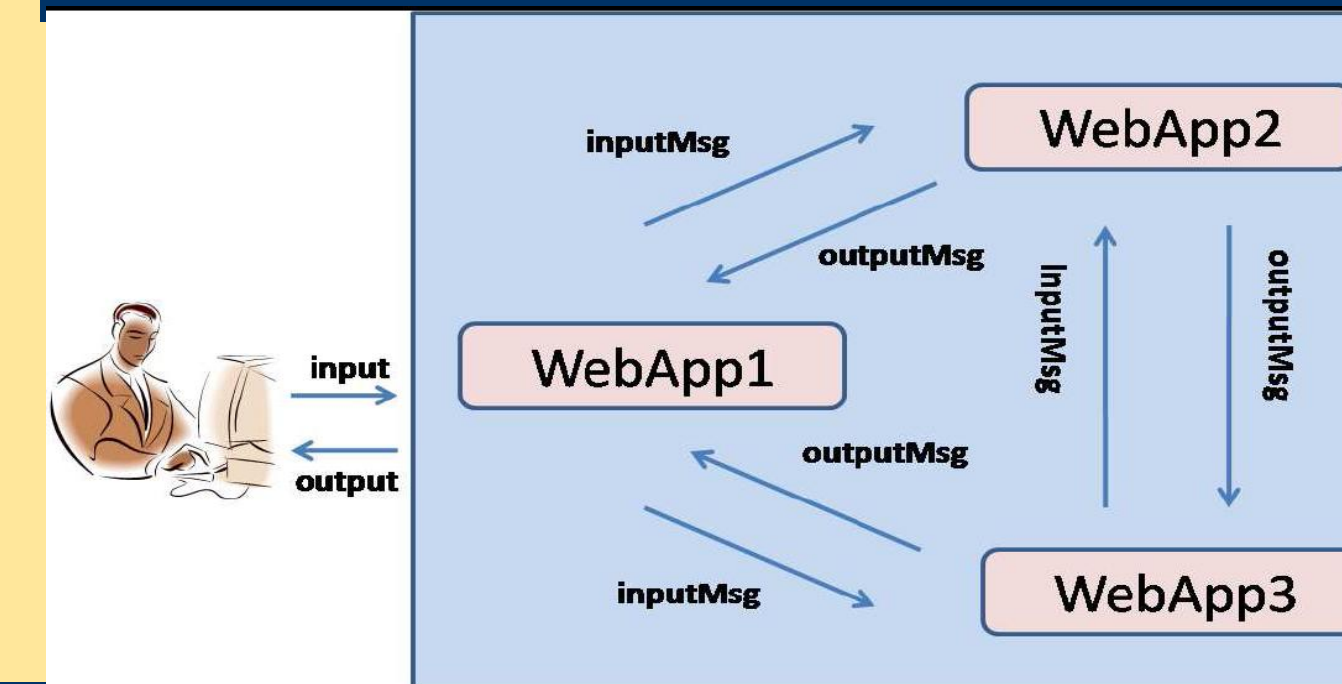
Problem

- Web services can be very complex in structure
 - Difficulty in quality assurance

Proposed Solution:

- Knowledge-based web service testing
 - Using known information in strategic ways to test software

Fault-Based Combinatorial Test Generation



- Create problem by injecting fault into web service
 - Falsify conditional statement in source code, creating a faulty web service
- Generate test inputs that will be executed by the faulty code using combinatorial algorithm

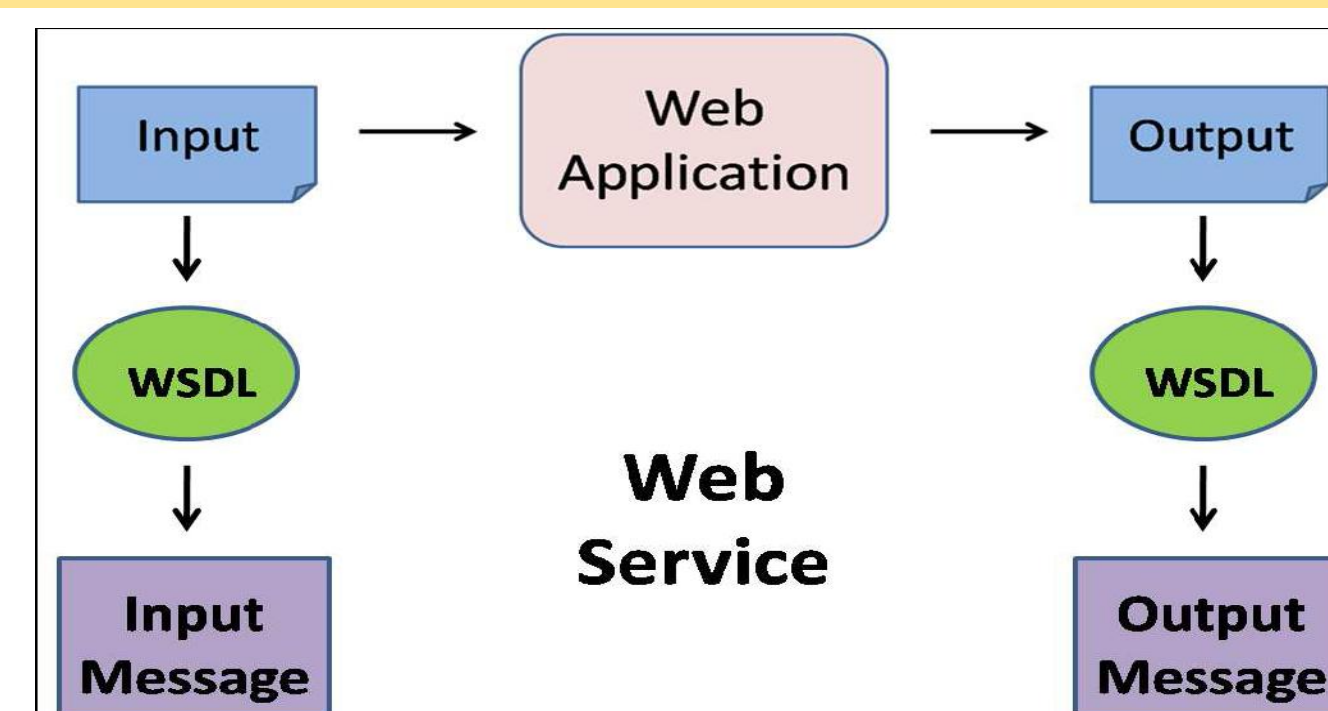
Client accessing with a web service example

Web Services & Applications



Web Application Example

- Web services defined as server component in client-server relationship
- Client-server relationship could be described using web based mail services (ex. Yahoo)
 - Client(user) communicates with server(Yahoo)
- Web application is accessed via web browser over network
 - Commonly implemented in languages such as HTML & Java
 - Software components of web services
 - Receive input from client & produce output



Input & Output Messages Example

Web services built on SOAP (Simple Object Access Protocol)

- SOAP is communication protocol that allows transfer of data in XML over the Internet
- SOAP allows different applications on different operating systems with different languages to communicate
- Inputs & outputs of each application w/in a web service are wrapped through the SOAP protocol into input and output messages

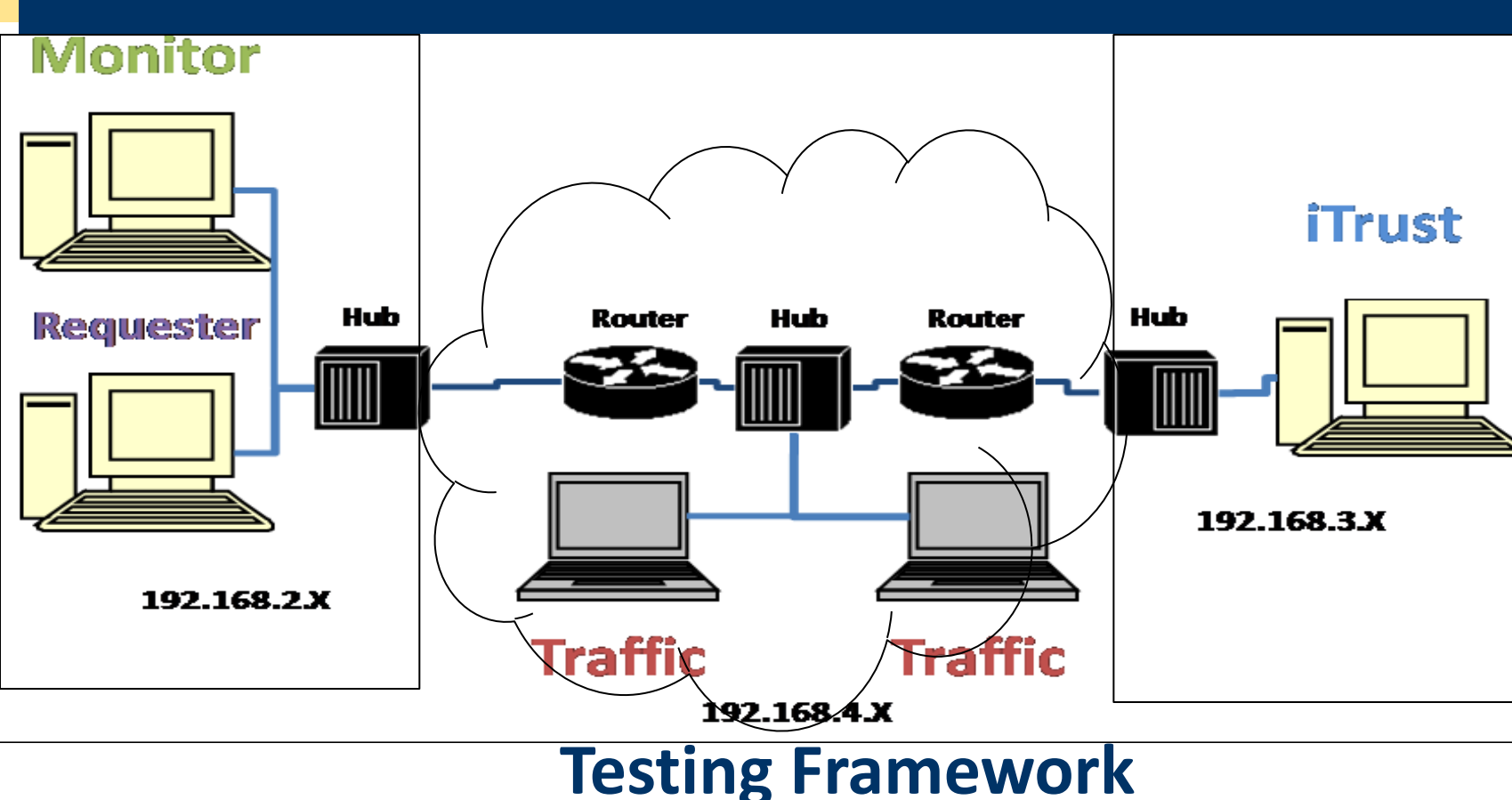
Anticipated Contributions

- Web services can grow & be very complex, making it difficult to assure quality
 - Web service testing is required
- Combinatorial testing techniques proven to be efficient in testing software
- Combining fault-based & combinatorial testing techniques, assessing & evaluating web services may be better

References

- [1] M. Gudgin, M. Hadley, N. Mendelsohn, J. Moreau, H. Nielsen, A. Karmarkar, and Y. Lafon. Soap version 1.2. <http://www.w3.org/TR/soap12-part1/>, 2007.
- [2] D. Kuhn, D. Wallace, and A. G. Jr. Software fault interactions and implications for software testing. *IEEE Transactions on Software Engineering*, 30:418–421, 2004.
- [3] C. Mao. Performing combinatorial testing on web service-based software. In *IEEE International Conference on Computer Science and Software Engineering*, pages 755–758, Nanchang, China, 2008. IEEE Computer Society.
- [4] J. Offutt and W. Xu. Generating test cases for web services using data perturbation. *SIGSOFT Softw. Eng. Notes*, 29(5):1–10, 2004.
- [5] V. Pretre, F. Bouquet, and C. Lang. Automating uml models merge for web services testing. In *iiWAS '08: Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, pages 55–62, New York, NY, USA, 2008. ACM.
- [6] N. C. S. U. RealSearch Research Group.itrust: Role-based healthcare v7.0.1n. <http://agile.csc.ncsu.edu/iTrust/wiki/doku.php>, 2008.
- [7] O. U. specification TC. Uddi version 3.0.2. http://uddi.org/pubs/uddi_v3.htm, 2005.
- [8] W. Vogels. Web services are not distributed objects. *Web services are not distributed objects*, 7:59–66, 2003.

Web Service Emulation: iTrust



- iTrust is a medical application that allows patients to keep up with their medical history and records
- Through SOAP, WSDL, & UDDI specifications, iTrust is wrapped
 - Enables iTrust to emulate a web service
- Testing framework is a network where we emulate the Internet
 - In center hub, Traffic machines generate random network traffic
 - Requester is client that will be accessing iTrust
 - Monitor observes & collects information on traffic coming to and from Requester
 - iTrust machine is location where iTrust is deployed as web service