XML INJECTION MUTATION FOR WEB SERVICES VULNERABILITY TESTING BASED ON SOAP MESSAGES

Bimal Varghese
Simi Stephen

Federal Institute of Science and Technology
Mookkannoor
Angamaly

18 May 2015

Outline

1 INTRODUCTION
- Web Service
- Vulnerability Testing of Web Service

2 LITERATURE SURVEY
- METHODS COMPARISON

3 METHODOLOGY
- SOAP message muatation
- Mutation Operators
- Worload Generation
- Attackload Generation

4 EXPERIMENTAL RESULTS

5 CONCLUSION

INTRODUCTION

Web Service

- Common way of implementing SOA.
- Widely used in the Internet.
- Good Encapsulation and Strong Integration capabilities making it more useful.
- Architecture of Web Service may cause security vulnerabilities.
  - Quality and Reliability of Web Service must be heavily tested.

Vulnerability Testing of Web Service

- Components in WS.
  - WSDL.
  - SOAP.
  - XML.
  - UDDI.

Vulnerability Testing of Web Service

- Components in WS.
  - WSDL.
  - SOAP.
  - XML.
  - UDDI.
- Vulnerability refers to flaws in the service that threaten the security of the Application/computer system.
- Traditional Software testing methods may not work with Webs Services.
  - Heterogeneous nature.
  - Lack of User Interface (UI).

Difficulties in Testing Web Service

Factors that contributing to the difficulty of Web Service Testing

1. Different development and application environments.
2. The characteristics of Web service distribution, discovery, and dynamic bindings.
3. The need for a service interface for Web service design and implementation when applying automatic testing methods and techniques

Shortcommings in Web Service Testing

- The need for significant human intervention in the process.
- Simple performance and access testing have been performed.
- Extensible Markup Language (XML) in web service considered only for data transmission and not of data storage

Literature Survey

[1] Automated Robustness Testing of Web Service

Presents a framework for automatically generating and execution of web service.

Method

1. Code Generator
    1. Generate necessary code to implement a service consumer
2. Test Generation
    1. Generated test class is supplied to a test generation tool such as JCrasher inorder to generate JUnit test.
3. Test Execution
    - Invoke test case and call web service. Collect response from web service.

[2] Exploring Perturbation Based Testing for Web Service

- Extended approach based on XML message perturbation
- Utilized SOAP Perturbation operators and a Web service Testing tool (SMAT-WS)

SOAP Perturbation Operator

SOAP perturbation primitive operators include.

1. *Null (n)*
2. *Incomplete (n)*
3. *Inversion (n)*
4. *ValueInversion (n)*
5. *Mod_Len (n)*
6. *Space (n)*

[3] Efficient Web Services Message Exchange by SOAP Bundling Framework

- A SOAP message bundling framework is Proposed.
- Framework enables bundling of multiple messages into one message.
- Application developers do not have to consider the service granularity for performance reasons.

[4] Generating Test Cases for Web Services Using Data Perturbation

- Existing XML messages are modified based on rules defined on the message grammars, and then used as tests.
- Data perturbation uses 2 methods to test Web services :
  Data value perturbation :
  modifies values according to the data type.
  Interaction perturbation :.
  classifies the communication messages into two categories : RPC communication and data communication

[5] Adaptive Random Testing : the ART of Test Case Diversity

Base Idea

- Many program faults result in failures at contiguous areas of the input domain - failure patterns.
- For detecting failure patterns, ART systematically filters randomly generated candidates.

Contd. . .

Adaptive Random Testing

Principle

- Given a set of previously executed test cases that have not revealed any failures, new test cases located away from these old ones are more likely to reveal failures.

Types of ART methods

- Fixed Size Candidate Set ART (FSCS-ART) :
  Candidate with largest distance from current Test case is considered next.
- Restricted Random Testing (RRT) :
  Create Exclusion zone for current test case.
  Take random selection if it lie outside the zone.

[6] Testing Web Services by XML Perturbation

- Web services uses XML to describe and transmit data.
- XML schema is utilized to generate data formats and test cases.
- Some applications and web services do not validate XML messages against an XML schema, and sometimes no schema exists.

Contd. . .

XML Data Model

- An XML schema can be modeled as a tree.

  XML tree T = (N, D, X, E, n_r), where :

  - N is a finite set of elements and attribute nodes.
  - D is a finite set of built-in and derived data types.
  - X is a finite set of constraints (integrity and representation).
  - E is a finite set of edges.
  - n_r is the root node.

[7] Combinatorial Mutation Approach to Web Service Vulnerability Testing

Combinatorial mutation testing focuses on using combinations of at least two faulty input data parameter to find faults within the software.

- A set of operators that can be combined are presented
- SOAP message is obtained by parsing the WSDL file, and data perturbation techniques are adopted to generate simple initial test data.
- A combinatorial testing method is developed.

Contd. . .

Mutation Operator Design

Two perturbation policies which directly act on the SOAP message are used

- Data Value perturbation :
  modifies values in SOAP messages according to their data types
- Interaction perturbation :
  consider the data values and data relationships

[8] Worst-Input Mutation Approach to Web Services Vulnerability Testing Based on SOAP Messages

Proposed Worst Input Mutation approach based on SOAP message mutation.

Method

- Worst Input Mutation :
  Utilizing characteristics of SOAP message.
- Automatic Test Case Generation :
  Test Case Generation based on Farthest Neighbor (TCFN) algorithm
- A prototype Web Service Vulnerability Testing Tool is implemented

TABLE: Comparison of various Testing Approaches

Sl.No	Paper	Approach	Technology
1	Automated Robustness Testing of Web Service	Automatic code generator and testing framework	Existing java Technology
2	Exploring Perturbation Based Testing for Web Service	XML message perturbation	SOAP Perturbation operators
3	Efficient Web Services Message Exchange by SOAP Bundling Framework	Multiple message bundling	SOAP message bundling framework
4	Generating Test Cases for Web Services Using Data Perturbation	Data perturbation	RPC and Data communication
5	Adaptive Random Testing : the ART of Test Case Diversity	Fixed Size Candidate Set ART (FSCS-ART) Restricted Random Testing (RRT)	Random candidate generation
6	Testing Web Services by XML Perturbation	XML schema perturbation	XML Tree Data Model
7	Combinatorial Mutation Approach to Web Service Vulnerability Testing	Combinations of at least two faulty input data parameter to find faults	SOAP message mutation
8	Worst-Input Mutation Approach to Web Services Vulnerability Testing Based on SOAP Messages	Farthest Neighbor Approach	SOAP message mutation

METHODOLOGY

Current Scenario

- Test case generation :- An effective way of testing the credibility of Softwares.
- Web Service components depend mainly on XML.
- Both WSDL & SOAP component of Web service depend on XML.
- Efficient Test cases for Web Service depends on XML.
- Efficient mutation of XML will be an effective Test case.

SOAP message muatation

- Existing SOAP message mutation drawbacks are.
  - Redundancy of data in the SOAP message.
  - Low efficiency of the mutation operators.
  - One mutant injected at one time

Mutation Operators

- Based on SOAP messages.
- XML model for SOAP message can be considered as an Extended Regular Tree Grammar.
- eRTG (extended RTG) contains 6 tuples $< E; N; DT; P; A; n_s >$
  - E is a finite set of elements ;
  - N is a finite set of non-terminals
  - DT is a finite set of data types int, string, bool, numerical, char, object
  - P is a finite set of production rules
  - n_s is the starting non-terminal
  - A is a 2-tuple $< n; type >$ n - number of parameters and "type" - parameter type.

Cont . . .

Given a set of all element instances N

- Mutation operator r = $f(n_1, n_2, ...n_i)$.
  - f - a function.
  - $(n_1, n_2, ...n_i) \epsilon N$ for i $\geq$ 1 with arbitrary data type.
  - r outputs mutated $(n_1, n_2, ...n_i)$ with the same data type as the input $(n_1, n_2, ...n_i)$

Security rule for testing the vulnerability of Web services.

- VWS = $G(r)$, where r=$f(n_1, n_2, ..., n_i)$ is the mutation operator for the tested Web service
- G(r) the vulnerability that is triggered by r
- $n_i \epsilon N$ the Web service input parameters.

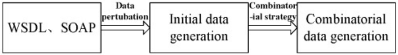


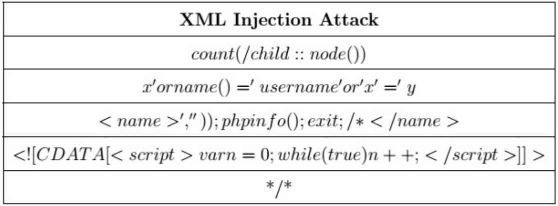
FIGURE: Steps to generate combinatorial data to Web service vulnerability testing

Workload Generation

- Specific workload for each service need to be generated.
- Workload is generated based on the following steps :
  1. Generate test values for each input parameter
  2. Generate test calls for each operations
  3. Execute call for each operation

Attackload Generation

- Based on number and type of SOAP message parameters, input domain is partitioned.
- Corresponding test case is then selected.



XML Injection Attack
$count(/child :: node())$
$x'orname() =' username'or'x' =' y$
$< name >','')); phpinfo(); exit; /* < /name >$
$<![CDATA[< script > varn = 0; while(true)n++; < /script >]] >$
$*/*$

FIGURE: Examples of XML Injection Attack Loads

Specific Combinatorial Methods are

- *Data Value Perturbation :* intended to check the integrity of data validity of the system.
    - The combinatorial test data are generated by calling minimum K factors (K=2) of PS
- *Interaction Perturbation :* To check the integrity of the various interaction mechanism such as data retrieval and communications between the system.
    - Based on the faults generally occurring among the parameters of the adjacent locality principle, $k = \lceil (i + j)/2 \rceil$

EXPERIMENTAL RESULTS

Experimental implementation

Web service vulnerability testing system (WSVTS) is implemented in C#

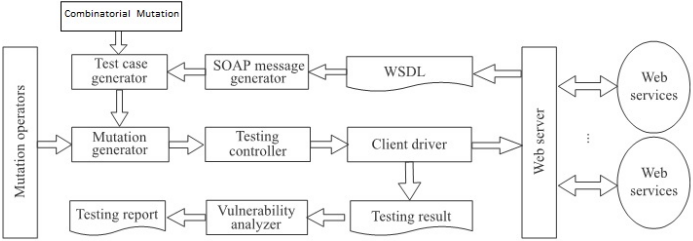


FIGURE: The WSVTS framework.

Four main function modules

1. SOAP message generator
2. SOAP message mutation generator
3. Test case generator
4. Vulnerability analyzer

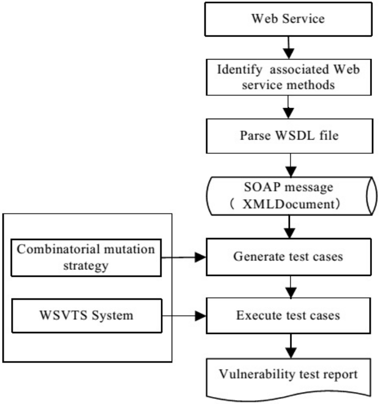
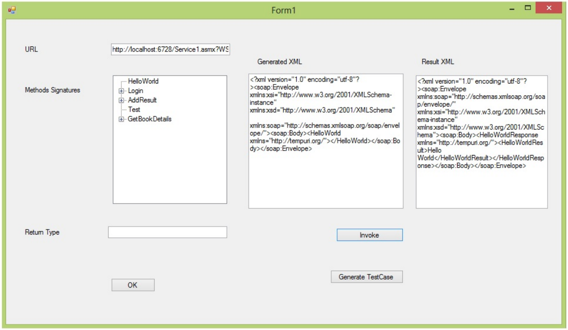

Web Service
Identify associated Web service methods
Parse WSDL file
SOAP message （XMLDocument）
Combinatorial mutation strategy
Generate test cases
WSVTS System
Execute test cases
Vulnerability test report

FIGURE: Flow chart of the Web Service vulnerability testing system



URL http://localhost:6728/Service1.asmx?WS

Methods Signatures
- HelloWorld
- Login
- AddResult
- Test
- GetBookDetails

Return Type

Generated XML

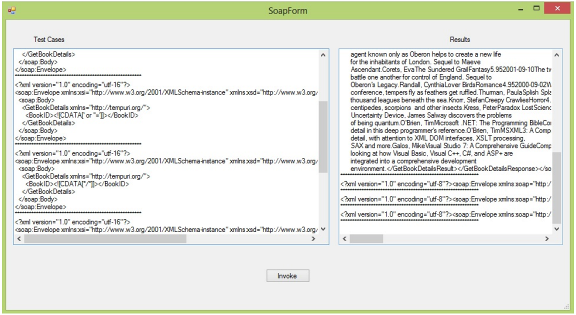
```
<?xml version="1.0" encoding="utf-8"?
><soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:soap="http://schemas.xmlsoap.org/soap/envel
ope/"><soap:Body><HelloWorld
xmlns="http://tempuri.org/"></HelloWorld></soap:Bo
dy></soap:Envelope>
```

Result XML

```
<?xml version="1.0" encoding="utf-8"?
><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soa
p/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSch
ema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSc
hema"><soap:Body><HelloWorldResponse
xmlns="http://tempuri.org/"><HelloWorldRes
ult>Hello
World</HelloWorldResult></HelloWorldResp
onse></soap:Body></soap:Envelope>
```

FIGURE: Simple Test case using WSVTS



```
  </GetBookDetails>
 </soap:Body>
</soap:Envelope>
**************************************************

<?xml version="1.0" encoding="utf-16"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/
 <soap:Body>
  <GetBookDetails xmlns="http://tempuri.org/">
   <BookID><![CDATA[' or ''=']]></BookID>
  </GetBookDetails>
 </soap:Body>
</soap:Envelope>
**************************************************

<?xml version="1.0" encoding="utf-16"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/
 <soap:Body>
  <GetBookDetails xmlns="http://tempuri.org/">
   <BookID><![CDATA[*/*]]></BookID>
  </GetBookDetails>
 </soap:Body>
</soap:Envelope>
**************************************************

<?xml version="1.0" encoding="utf-16"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/
```

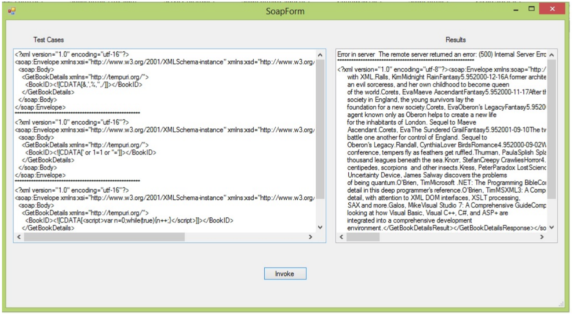
```
    agent known only as Oberon helps to create a new life
    for the inhabitants of London. Sequel to Maeve
    Ascendant.Corets, EvaThe Sundered GrailFantasy5.952001-09-10The tv
    battle one another for control of England. Sequel to
    Oberon's Legacy.Randall, CynthiaLover BirdsRomance4.952000-09-02W
    conference, tempers fly as feathers get ruffled.Thurman, PaulaSplish Spla
    thousand leagues beneath the sea.Knorr, StefanCreepy CrawliesHorror4.
    centipedes, scorpions  and other insects.Kress, PeterParadox LostScienc
    Uncertainty Device, James Salway discovers the problems
    of being quantum.O'Brien, TimMicrosoft .NET: The Programming BibleCo
    detail in this deep programmer's reference.O'Brien, TimMSXML3: A Comp
    detail, with attention to XML DOM interfaces, XSLT processing,
    SAX and more.Galos, MikeVisual Studio 7: A Comprehensive GuideComp
    looking at how Visual Basic, Visual C++, C#, and ASP+ are
    integrated into a comprehensive development
    environment.</GetBookDetailsResult></GetBookDetailsResponse></so
**************************************************

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http:/
**************************************************

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http:/
**************************************************

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http:/
**************************************************
```

FIGURE: Combinatorial Test cases for XML injection 1



```
<?xml version="1.0" encoding="utf-16"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/
  <soap:Body>
    <GetBookDetails xmlns="http://tempuri.org/">
      <BookID><![CDATA[&,',%,",/]]></BookID>
    </GetBookDetails>
  </soap:Body>
</soap:Envelope>
**************************************************
<?xml version="1.0" encoding="utf-16"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/
  <soap:Body>
    <GetBookDetails xmlns="http://tempuri.org/">
      <BookID><![CDATA[' or 1=1 or "="]]></BookID>
    </GetBookDetails>
  </soap:Body>
</soap:Envelope>
**************************************************
<?xml version="1.0" encoding="utf-16"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/
  <soap:Body>
    <GetBookDetails xmlns="http://tempuri.org/">
      <BookID><![CDATA[<script>var n=0;while(true){n++;}</script>]]></BookID>
    </GetBookDetails>
```

```
Error in server  The remote server returned an error: (500) Internal Server Erro
**************************************************

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http:/
     with XML.Ralls, KimMidnight RainFantasy5.952000-12-16A former archite
     an evil sorceress, and her own childhood to become queen
     of the world.Corets, EvaMaeve AscendantFantasy5.952000-11-17After th
     society in England, the young survivors lay the
     foundation for a new society.Corets, EvaOberon's LegacyFantasy5.9520
     agent known only as Oberon helps to create a new life
     for the inhabitants of London. Sequel to Maeve
     Ascendant.Corets, EvaThe Sundered GrailFantasy5.952001-09-10The tv
     battle one another for control of England. Sequel to
     Oberon's Legacy.Randall, CynthiaLover BirdsRomance4.952000-09-02W
     conference, tempers fly as feathers get ruffled.Thurman, PaulaSplish Spla
     thousand leagues beneath the sea.Knorr, StefanCreepy CrawliesHorror4.
     centipedes, scorpions  and other insects.Kress, PeterParadox LostScienc
     Uncertainty Device, James Salway discovers the problems
     of being quantum.O'Brien, TimMicrosoft .NET: The Programming BibleCo
     detail in this deep programmer's reference.O'Brien, TimMSXML3: A Comp
     detail, with attention to XML DOM interfaces, XSLT processing,
     SAX and more.Galos, MikeVisual Studio 7: A Comprehensive GuideComp
     looking at how Visual Basic, Visual C++, C#, and ASP+ are
     integrated into a comprehensive development
     environment.</GetBookDetailsResult></GetBookDetailsResponse></so
```

FIGURE: Combinatorial Test cases for XML injection 2

Comparison with other system

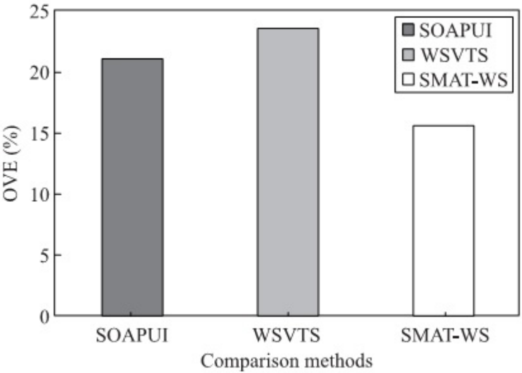


FIGURE: Comparison of the overall efficiency of SOAPUI, WSVTS, SMAT-WS.

CONCLUSION

- Due to specific nature of Web services, traditional software testing techniques cannot be completely adopted to test Web services.
- Test data generation is an important content of testing Web service
- By designing appropriate SOAP message mutation operators, the security of Web services can be tested from the client side
- Vulnerability and faults can be identified from the user's perspective.

REFERRENCES

[1] K. Yue, X.L. Wang, A.Y. Zhou, “Underlying Techniques for Web Services : A Survey”, Journal of Software,Vol. 15, No. 3, 2004, pp.428-442.

[2] T. Takase and K. Tajima, “Efficient web service message exchange by SOAP bounding framework,” in the 11th IEEE International Enterprise Distributed Object Computing, Annapolis, MD, USA, 2007, pp. 63-72.

[3] L. Wu, X. K. Li, and H. Wang, “Research on the reliability testing of web service based on fault injection technology,” Journal of Chinese Computer System, vol. 28, no. 1, pp. 127-131, 2007.

[4] Y. Jiang, G.M. Xin, J.H. Shan, et al. “A Method of Automated Test Data Generation for Web Service”, Journal of Computer, Vol. 28, No. 4, 2005, pp.568-577.

[5] C. A. Sun, G. Wang, B. H. Mu, H. Liu, Z. S. Wang, and T. Y. Chen,“ A metamorphic relation-based approach to testing web services without oracles,” International Journal of Web Services Research, vol. 9, no. 1, pp. 51-73, 2012.

[6] M.P. Papazoglou, “Web Services Principles and Technology”, Pearson Prentice Hall, Holland, 2010, pp. 99-100.

[7] L. F. de Almeida and S. R. Vergilio,“ Exploring perturbation based testing for web services,” presented at the IEEE International Conference on Web Services, Chicago, USA,

[8] L. Novak and A. Zamulin,“ A formal model for XML schema,” in Proceedings of the 21st International Conference on Data Engineering Workshops, Tokyo, Japan, pp. 1283-1293, 2005.

[9] S. Bekrar, C. Bekrar, R. Groz, and L. Mounier,“ Finding software vulnerabilities by smart fuzzing,” in Proceedings of the Fourth IEEE International Conference on Software Testing, Verification and Validation, Berlin, Germany, 2011, pp. 427-430.

[10] J. Offutt and W. Xu,“ Generating test cases for web services using data perturbation,” ACM SIGSOFT Software Engineering Notes, vol. 29, no. 5, pp. 1-10, 2004.

[11] A. C. V. de Melo and P. Silveira,“ Improving data perturbation testing techniques for web services”, Information Science, vol. 181, no. 3, pp. 600-619, 2011.

[12] W. Xu, J. Offutt, J. Luo, “Testing Web Services by XML Perturbation”, Proc. of the 16th IEEE International Symposium on Software Reliability Engineering (ISSRE’05), IEEE Computer Society, 2005, pp.257-266

[13] W.T. Tsai, R. Paul Y. Wang, et a1 , “Extending WSDL to Facilitate Web Services Testing”, Proc. of the 7th IEEE International Symposium on High Assurance Systems Engineering (HASE’02), IEEE Computer Society, 2002, pp.171- 172.

[14] M. Evan, S. Bas, T. Xie, “Automated Testing and Response Analysis of Web Services”, 2007 IEEE International Conference on Web Services (ICWS 2007), IEEE Computer Society, 2007, pp.647-654..

[15] M.S. Harry, S.H. Huan, “WSDLTest-A Tool for Testing Web Services”, Proc. of the 8th IEEE International Symposium on Web Site Evolution (WSE’06), IEEE Computer Society, 2006, pp.14-21.

[16] X.Y. Bai, W.L. Dong, W.T. Tsai, Y. Chen, “WSDL-based Automatic Test Case Generation for Web Services Testing”, Proc. of the 2005 IEEE International Workshop on Service-oriented System Engineering (SOSE’05), IEEE Computer Society, 2005, pp.207-212.

[17] J.F. Chen, Y.S. Lu, X.D. Xiao, “A Fault Injection Model of Component Security Testing”, Journal of Computer Research and Development, Vol. 46, No. 7, 2009, pp.1127-1135.

[18] Tom Henzinger, Ranjit Jhala, Rupak Majumdar, Dirk Beyer,“ Blast (berkeley lazy abstraction software verification tool) model checker.” <http ://embedded.eecs.berkeley.edu/blast/> (last access May 2010).

[19] H. Huang, W. Tsai, R. Paul, Y. Chen, “Automated model checking and testing for composite Web services,” in : ISORC ’05 : Proceedings of the Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC’05), IEEE Computer Society, Washington, DC, USA, 2005, pp. 300–307.

[20] K.Z. Watkins, “Introducing Fault-Based Combinatorial Testing to Web Services”, Proc. of the IEEE SoutheastCon 2010 (SoutheastCon), 2010, pp.131-134.

[21] P. C. Jorgensen., “Software Testing : A Carftsman’s Approach”, Third Edition, CRC Press, USA, 2011, pp.315-324.

[22] SoapUI, SmartBear software, http ://www.soapui.org, 2012.

[23] Mikio Aoyama, Sanjiva Weerawarana, Hiroshi Maruyama, Clemens Szyperski, Kevin Sullivan, and Doug Lea. “Web services engineering : Promises and challenges.” In Proceedings of the 24th International Conference on Software Engineering, Orlando, Florida, May 2002.

[24] G. Canfora, M. Di Penta, “Testing services and service-centric systems : challenges and opportunities,” IT Professional, 2006, pp. 10–17.

[25] G. Canfora, M. Di Penta, “Service-oriented architectures testing : a survey,” in : Andrea De Lucia, Filomena Ferrucci (Eds.), ISSSE, Lecture Notes in Computer Science, vol. 5413, Springer, 2009, pp. 78–105.

THANK YOU