

# Handling SQL Databases in Automated System Test Generation

ANDREA ARCURI, Kristiania University College, Oslo, Norway

JUAN P. GALEOTTI, Depto. de Computación, FCEyN-UBA, and ICC, CONICET-UBA, Argentina

Automated system test generation for web/enterprise systems requires either a sequence of actions on a GUI (e.g., clicking on HTML links and form buttons) or direct HTTP calls when dealing with web services (e.g., REST and SOAP). When doing *white-box* testing of such systems, their code can be analyzed, and the same type of heuristics (e.g., the *branch distance*) used in search-based unit testing can be employed to improve performance. However, web/enterprise systems do often interact with a database. To obtain higher coverage and find new faults, the state of the databases needs to be taken into account when generating white-box tests. In this work, we present a novel heuristic to enhance search-based software testing of web/enterprise systems, which takes into account the state of the accessed databases. Furthermore, we enable the generation of SQL data directly from the test cases. This is useful when it is too difficult or time consuming to generate the right sequence of events to put the database in the right state. Also, it is useful when dealing with databases that are “read-only” for the system under test, and the actual data are generated by other services. We implemented our technique as an extension of EvOMASTER, where system tests are generated in the JUnit format. Experiments on six RESTful APIs (five open-source and one industrial) show that our novel techniques improve coverage significantly (up to +16.5%), finding seven new faults in those systems.

CCS Concepts: • **Software and its engineering** → **Software verification and validation**; **Search-based software engineering**;

Additional Key Words and Phrases: SQL, database, SBST, automated test generation, system testing, REST, web service

## ACM Reference format:

Andrea Arcuri and Juan P. Galeotti. 2020. Handling SQL Databases in Automated System Test Generation. *ACM Trans. Softw. Eng. Methodol.* 29, 4, Article 22 (July 2020), 31 pages.

<https://doi.org/10.1145/3391533>

## 1 INTRODUCTION

Web and enterprise applications are very popular in industry. They can be very complex, which makes their automated *system testing* quite difficult. Hence, it is common that automated approaches only deal with *black-box* testing. Crawlers like Crawljax [43] are an example of this.

This work is funded by the Research Council of Norway (project on Evolutionary Enterprise Testing, grant agreement No 274385), and partially by UBACYT-2018 20020170200249BA, PICT-2015-2741.

Authors' addresses: A. Arcuri, Kristiania University College, Department of Technology, PB 1190 Sentrum, 0107, Norway; email: andrea.arcuri@kristiania.no; J. P. Galeotti, Depto. de Computación, FCEyN-UBA, and ICC, CONICET-UBA. Adolfo Alsina 2401 Florida, Buenos Aires, 1602, Argentina; email: jgaleotti@dc.uba.ar.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2020 Association for Computing Machinery.

1049-331X/2020/07-ART22 \$15.00

<https://doi.org/10.1145/3391533>