

# Takuan: A Tool for WS-BPEL Composition Testing Using Dynamic Invariant Generation

Manuel Palomo-Duarte, Antonio García-Domínguez, Inmaculada Medina-Bulo,  
Alejandro Alvarez-Ayllón, and Javier Santacruz

Department of Computer Languages and Systems. Universidad de Cádiz, Escuela  
Superior de Ingeniería, C/Chile 1, CP 11003 Cádiz, Spain  
{manuel.palomo,antonio.garciadominguez,inmaculada.medina@uca.es,  
alejandro.alvarez,javier.santacruz}@uca.es

**Abstract.** WS-BPEL eases programming in the large by composing web services, but poses new challenges to classical white-box testing techniques. These have to be updated to take context into account and cope with its specific instructions for web service management. Takuan is an open-source system that dynamically generates invariants reflecting the internal logic of a WS-BPEL composition. After several improvements and the development of a graphical interface, we consider Takuan to be a mature tool that can help find both bugs in the WS-BPEL composition and missing test cases in the test suite.

**Keywords:** Web services, service composition, WS-BPEL, white-box testing, dynamic invariant generation.

## 1 Introduction

The OASIS WS-BPEL 2.0 standard allows the user to develop advanced web services (WS) by composing others. However, it presents a challenge [1] for traditional white-box testing techniques, firstly due to its dependency on context, and secondly because of the inclusion of WS-specific instructions not found in most programming languages (like those for fault and compensation handling).

Automatic dynamic invariant generation [2] has proved to be a successful technique to assist in white-box testing of programs written in imperative languages. Let us note that, throughout this work, the term *dynamic invariant* (or *likely invariant*) is considered, as in most related works, in its broadest sense: a property that a program holds for a specified test suite.

Takuan [3] is an open-source system that dynamically generates invariants reflecting the internal logic of a WS-BPEL composition. After several improvements [4,5] Takuan can be easily used from a graphical interface to support WS-BPEL composition testing, helping find bugs in the composition code and improve the test suite.

## 2 Dynamic Invariant Generation for WS-BPEL

Dynamic invariant generation process is not based on a formal analysis of the composition, but on information collected from several executions. This way, if

we use a good test suite as input, all of the features of the environment and the complex internal logic of the composition (loops, fault handling, etc.) will be reflected in the execution logs, and the generator will infer true invariants. On the other hand, if we use an incomplete or biased test suite, we will get false invariants. These are due to an incomplete test suite that needs additional test cases which will falsify those invariants.

There are different ways to use dynamically generated invariants to help program testing [2]. We will focus on two of them, as shown in figure 1: debugging a program and improving a test suite.

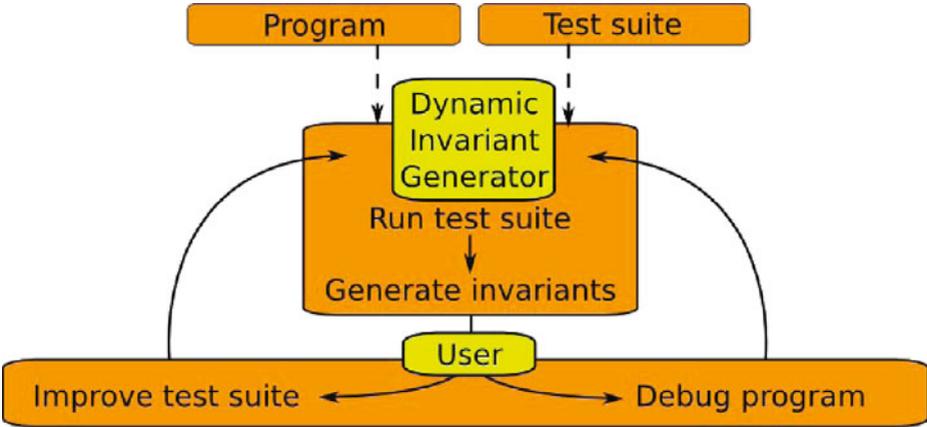


Fig. 1. Dynamic invariant generation feedback loop

The usual dynamic invariant generation workflow starts with the user providing an initial test suite and a program. After running the program against the test suite, unexpected invariants may be obtained. In that case, the user needs to check if that difference was caused by a bug in the code. After fixing the bug, the user could run the new (corrected) program against the same test suite and check if those invariants are not inferred.

If the user considers the program to be correct, the fault may lie on the test suite, which is lacking test cases that disprove those invariants. By appropriately extending the test case, they would not be inferred again in the next run. This cycle can be run as many times as needed, fixing bugs in the program code and improving the test suite until the expected invariants are obtained.

### 3 Takuan

Takuan integrates our own code with several well-tested open-source systems [6]: the BPELUnit unit testing library, the ActiveBPEL engine and the Daikon dynamic invariant generator. All of them have been modified to create a dynamic invariant generation workflow that includes several optimizations to reduce running time and improve the invariants for the WS-BPEL language [4,5].

Takuan is available for free download under the terms of the GNU/GPL license. In its home site [7] we provide both source code for automatic script-guided compilation and a ready-to-use virtual machine.

Takuan can be invoked directly from a shell or through a graphical plugin for the NetBeans IDE. It can connect with a Takuan instance, both locally and remotely through its IP address. Figure 2 shows the plugin running.

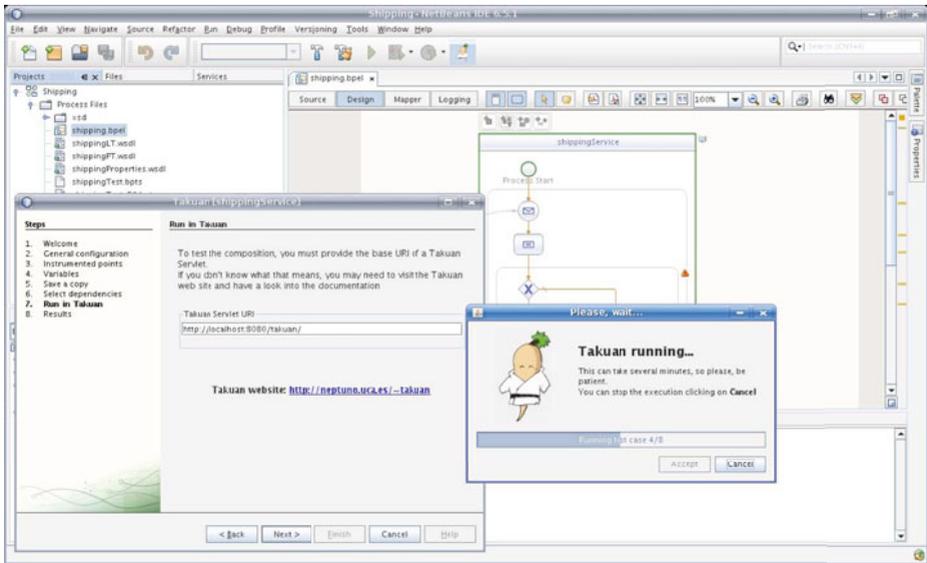


Fig. 2. Takuan NetBeans Plugin running

By default, Takuan logs every variable value before and after every sequence instruction in the composition to check for invariants. The user can choose which variables to consider for invariants and in which instructions.

Additionally, Takuan can replace some of the external services with *mock-ups*: dummy services which will act according to user instructions. It is only recommended when not all external services are available for testing, or to define *what-if* scenarios under certain predefined WS behavior. In any of these cases, the behavior provided by each mockup in each test case (a SOAP answer or a certain fault) will be part of the test suite specification. The user will be responsible for providing suitable values for them.

## 4 Conclusions and Future Work

Takuan is an open-source dynamic invariant generator for WS-BPEL compositions. It takes a WS-BPEL 2.0 process definition with its dependencies and a test suite, and infers a list of invariants. In this paper we have shown two ways to use Takuan in a feedback loop: highlighting bugs in the composition code and helping to improve a test suite.

We are currently developing a new graphical user interface, named IdiginBPEL, that will ease massive systematic Takuan use [7]. Our future work will be to extend Takuan to infer non-functional properties that can be included in composition specification, so it can be used as automatically as possible inside a Service-Oriented-Architecture-specific developing methodology [8].

## Acknowledgments

This paper has been funded by the Department of Education and Science (Spain) and FEDER funds under the National Program for Research, Development and Innovation. Project SOAQSim (TIN2007-67843-C06-04).

## References

1. Bucchiarone, A., Melgratti, H., Severoni, F.: Testing service composition. In: Proceedings of the 8th Argentine Symposium on Software Engineering, ASSE '07 (2007)
2. Ernst, M.D., Cockrell, J., Griswold, W.G., Notkin, D.: Dynamically discovering likely program invariants to support program evolution. *IEEE Transactions on Software Engineering* 27(2), 99–123 (2001)
3. Palomo-Duarte, M., García-Domínguez, A., Medina-Bulo, I.: Takuan: A dynamic invariant generation system for WS-BPEL compositions. In: ECOWS '08: Proceedings of the 2008 Sixth European Conference on Web Services, Washington, DC, USA, pp. 63–72. IEEE Computer Society, Los Alamitos (2008)
4. Palomo-Duarte, M., García-Domínguez, A., Medina-Bulo, I.: Improving Takuan to analyze a meta-search engine WS-BPEL composition. In: SOSE '08: Proceedings of the 2008 IEEE International Symposium on Service-Oriented System Engineering, Washington, DC, USA, pp. 109–114. IEEE Computer Society Press, Los Alamitos (2008)
5. Palomo-Duarte, M., García-Domínguez, A., Medina-Bulo, I.: Enhancing WS-BPEL dynamic invariant generation using XML Schema and XPath information. In: Gaedke, M., Grossniklaus, M., Díaz, O. (eds.) ICWE '09. LNCS, vol. 5648, pp. 469–472. Springer, Heidelberg (2009)
6. Palomo-Duarte, M., García-Domínguez, A., Medina-Bulo, I.: An architecture for dynamic invariant generation in WS-BPEL web service compositions. In: Proceedings of ICE-B 2008 - International Conference on e-Business, Porto, Portugal. INSTICC Press (July 2008)
7. SPI&FM Group: Official Takuan home site, <http://neptuno.uca.es/~takuan>
8. García Domínguez, A., Medina Bulo, I., Marcos Bárcena, M.: Hacia la integración de técnicas de pruebas en metodologías dirigidas por modelos para SOA. In: Actas de las V Jornadas Científico-Técnicas en Servicios Web y SOA, Madrid (October 2009)