

User Guide

Baile V&V Prototype

Felipe M. Besson
Pedro M. B. Leal

¹Department of Computer Science
Institute of Mathematics and Statistics
University of São Paulo (USP)

`{besson, pedrombl}@ime.usp.br`

1. Overview

This software prototype is part of HP Baile Project¹, more specifically, this prototype belongs to the research line “Verification & Validation of Choreographies”. More details about this research line and the basis of our software can be found in this technical report [?]. In this user guide, we described in steps how to install and use the software.

2. Prerequisites

The following software elements must be installed and working:

- Java 6 [2]
- SQLite 3 [3]
- Ant [1]

3. Where to download ?

The source code, the scripts and its dependencies are available on:

4. Software directory structure

Our prototype are structure in the following files and directories (just the most important are described bellow):

file/directory	Description
baile.sh	script to start the application
src/	source code
test/	test code
config/	OpenKnowledge configuration files
lcc/	OpenKnowledge interaction model
lib/	third-party dependencies
log/	log files
test-reports/	html test reports

¹Baile Project: <http://ccsl.ime.usp.br/baile/>

5. How to use ?

In this section, we detailed the main steps to interact with the application.

1. Start the script **baile.sh**, and then, the application will start:

```
besson@metropolis:~/BookTripWSCor$ ./baile.sh
Baile V&V - dez/2010

Commands:
start_chore - start the choreography
stop_chore - stop the choreography
run_unit-tests - run the unit tests
run_integration-tests run the integration tests
run_acceptance-tests - run the acceptance tests
run_all-tests - run all tests
exit - stop the choreography if it is running and quit
help - show this message

Contacts:
Felipe Besson <besson@ime.usp.br>
Pedro Leal <pedrombl@ime.usp.br>

baile$
```

Figure 1. Welcome screen

2. In the prompt, start the choreography by typing **start_chore** (as showed in Figure 2)

```
baile$ start_chore
Determining localhost's IP... 192.168.1.100
Stopping currently running instances... Done
Compiling components... Done
Publishing the web services... Done
Starting message trace queue... Done
Launching the Discovery Service... Done
Publishing the Interaction Model... Done
Launching the traveler... Done
Launching the travel agency... Done
Launching the airline... Done
Launching the acquirer... Done
Setting up roles... Done
Choreography started.

baile$
```

Figure 2. Starting the choreography

3. Once all components of choreography (services, roles, OpenKnowledge entities) have been started successfully (as showed in Figure 2), the tests can be execute. The following commands are valid:

Command	Description
run_unit-tests	run the unit tests
run_integration-tests	run the integration tests
run_acceptance-tests	run the acceptance tests
run_all-tests	run all tests

6. How to add more test cases ?

The script compiles every test class of the test folder before any test execution. So, modifications on a existing test class, will be compiled before the next command tests executed.

Any new test class need to be added in its test strategy folder (unit, integration, or acceptance). After the addition, the script will recognised the new class automatically.

7. Known Issues

Our choreography does not support interactions interruptions properly. So, if any test was forced interrupted, in some cases, the choreography must be restarted.

8. Acknowledgements

This research received funding from HP Brasil under the Baile Project.

References

- [1] Apache Ant. Java library and command-line for automated building. Available on: <http://ant.apache.org/>, 2010.
- [2] Oracle. Java se downloads. Available on: <http://www.oracle.com/technetwork/java/javase>, 2010.
- [3] SQLite. Sql database engine. Available on: <http://www.sqlite.org>, 2010.