ALPACAS: A Language for Parametric Assessment of Critical Architecture Safety (Artifact)

Maxime Buyse ✉
Uber Elevate, Paris, France

Rémi Delmas ✉
Uber Elevate, Paris, France

Youssef Hamadi ✉
Uber Elevate, Paris, France

—— Abstract ——

This artifact contains a virtual machine allowing to use ALPACAS, a domain-specific language and algorithms aimed at architecture modeling and safety assessment for critical systems. ALPACAS allows to study the effects of random and systematic faults on complex critical systems and their reliability. The underlying semantic framework of the language is Stochastic Guarded Transition Systems, for which ALPACAS provides a feature-rich declarative modeling language and algorithms for symbolic analysis and Monte-Carlo simulation, allowing to compute safety indicators such as minimal cutsets and reliability. Built as a domain-specific language deeply embedded in Scala 3, ALPACAS offers generic modeling capabilities and type-safety unparalleled in other existing safety assessment frameworks. This improved expressive power allows to address complex system modeling tasks, such as formalizing the architectural design space of a critical function, and exploring it to identify the most reliable variant. The features and algorithms of ALPACAS are illustrated on a case study of a thrust allocation and power dispatch system for an electric vertical takeoff and landing aircraft.

1 Scope

This artifact contains a working installation of the ALPACAS software library, along with main programs allowing to reproduce the illustrative and experimental results presented in the corresponding research paper.

2 Content

The artifact includes a Linux Virtual Machine in OVA format, which can be run using virtualbox v6.x (available at `https://www.virtualbox.org` for windows and macOS and in the default repositories for Ubuntu). The virtual machine contains a folder with the source code of ALPACAS and example projects which can be browsed using Visual Studio Code (already installed in the OVA). Using the Scala Build Tool (already installed in the OVA), one can run the programs reproducing all experimental results presented in the paper.

3   Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). The latest version of the ALPACAS source code can be found online at `https://gitlab.com/maximebuyse/alpacas`.

4   Tested platforms

Virtualbox 6.x running under Windows 10, MacOS 10.15+, Ubuntu Linux 20.04.

5   License

The artifact is available under the following Uber Research OSS licence.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by the text below.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under this License.

This License governs use of the accompanying Work, and your use of the Work constitutes acceptance of this License.

You may use this Work for any non-commercial purpose, subject to the restrictions in this License. Some purposes which can be non-commercial are teaching, academic research, and personal experimentation. You may also distribute this Work with books or other teaching materials, or publish the Work on websites, that are intended to teach the use of the Work.

You may not use or distribute this Work, or any derivative works, outputs, or results from the Work, in any form for commercial purposes. Non-exhaustive examples of commercial purposes would be running business operations, licensing, leasing, or selling the Work, or distributing the Work for use with commercial products.

You may modify this Work and distribute the modified Work for non-commercial purposes, however, you may not grant rights to the Work or derivative works that are broader than or in conflict with those provided by this License. For example, you may not distribute modifications of the Work under terms that would permit commercial use, or under terms that purport to require the Work or derivative works to be sublicensed to others.

In return, we require that you agree:

1. Not to remove any copyright or other notices from the Work.
2. That if you distribute the Work in Source or Object form, you will include a verbatim copy of this License.

3. That if you distribute derivative works of the Work in Source form, you do so only under a license that includes all of the provisions of this License and is not in conflict with this License, and if you distribute derivative works of the Work solely in Object form you do so only under a license that complies with this License.

4. That if you have modified the Work or created derivative works from the Work, and distribute such modifications or derivative works, you will cause the modified files to carry prominent notices so that recipients know that they are not receiving the original Work. Such notices must state: (i) that you have changed the Work; and (ii) the date of any changes.

5. If you publicly use the Work or any output or result of the Work, you will provide a notice with such use that provides any person who uses, views, accesses, interacts with, or is otherwise exposed to the Work (i) with information of the nature of the Work, (ii) with a link to the Work, and (iii) a notice that the Work is available under this License.

6. THAT THE WORK COMES "AS IS", WITH NO WARRANTIES. THIS MEANS NO EX-PRESS, IMPLIED OR STATUTORY WARRANTY, INCLUDING WITHOUT LIMITATION, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR ANY WARRANTY OF TITLE OR NON-INFRINGEMENT. ALSO, YOU MUST PASS THIS DISCLAIMER ON WHENEVER YOU DISTRIBUTE THE WORK OR DERIVATIVE WORKS.

7. THAT NEITHER UBER TECHNOLOGIES, INC. NOR ANY OF ITS AFFILIATES, SUPPLI-ERS, SUCCESSORS, NOR ASSIGNS WILL BE LIABLE FOR ANY DAMAGES RELATED TO THE WORK OR THIS LICENSE, INCLUDING DIRECT, INDIRECT, SPECIAL, CON-SEQUENTIAL OR INCIDENTAL DAMAGES, TO THE MAXIMUM EXTENT THE LAW PERMITS, NO MATTER WHAT LEGAL THEORY IT IS BASED ON. ALSO, YOU MUST PASS THIS LIMITATION OF LIABILITY ON WHENEVER YOU DISTRIBUTE THE WORK OR DERIVATIVE WORKS.

8. That if you sue anyone over patents that you think may apply to the Work or anyone's use of the Work, your license to the Work ends automatically.

9. That your rights under the License end automatically if you breach it in any way.

10. Uber Technologies, Inc. reserves all rights not expressly granted to you in this License

6 MD5 sum of the artifact

MD5 (AlpacasReproPackageECOOP2021.ova) = 03c104c7fe22a72e8d51eea7cbdcfa09

7 Size of the artifact

AlpacasReproPackageECOOP2021.ova: 2,258,633,216 bytes (2.27 GB on disk)

A Usage instructions

VirtualBox can be downloaded from this URL:
   https://www.virtualbox.org/wiki/Downloads

A.1 Launching visual studio code

Inside the VM, open a terminal and change directory to the root project directory, and lauch the scala build tool at the root of the project.

```
cd ~/Desktop/alpacas-master
sbt
```

Inside the sbt console, type the following command to open vscode on the scala project

```
launchIDE
```

You can now browse the source code of the tool in `src/main/scala/alpacas`.
You can now browse the test suites of the tool in `src/test/scala/alpacas`.
You can now browse the code examples used to illustrate the paper `src/test/scala/tutorialpacas`.

A.2   Browsing the code examples

Open the following files in vscode:
- `src/test/scala/tutorialpacas/RunningExample.scala` to view the code of the running example presented in Listing 1 of the paper
- `src/test/scala/tutorialpacas/FlattenedGTS.scala` to view the code producing the flat SGTS encoding of the running example presented in Example 2 of the paper
- `src/test/scala/tutorialpacas/DesignSpaceExploration.scala` to view the code of the design-space exploration case study presented in section 7 of the paper.
- `src/test/scala/tutorialpacas/ExplorationExample.scala` to view the code of Listing 13, Section 7 of the paper

A.3   Running the code examples

Inside sbt, the following commands allow to reproduce results presented in the paper:
- `test:runMain tutorialpacas.dse.DesignSpaceExploration` reproduces the results presented in Table 3, Section 7
- `test:runMain tutorialpacas.dse.ExplorationExample` runs the code of Listing 13, Section 7
- `test:runMain tutorialpacas.runningexample.FlattenedExample` reproduces the flat SGTS compilation of the running example presented in Section 5, Example 2
- `test:runMain tutorialpacas.runningexample.RunningExample` runs the running example and reproduces the results presented in Section 6, Table 1, Table 2 of the paper.
- `test` runs the whole collection of ScalaTest test suites of alpacas.

A.4   Running an interactive step simulation

The sbt command

```
test:runMain tutorialpacas.runningexample.InteractiveSimulation
```

launches an interactive step simulation of the running example, where you can manually select events to fire from a list and see the state updates it triggers. You can also backtrack the simulation by entering option -1.