From FMTV to WATERS: Lessons Learned from the First Verification Challenge at ECRTS (Artifact)

Sebastian Altmeyer ✉
Universität Augsburg, Germany

Étienne André[1] ✉ 🆔
Université Sorbonne Paris Nord, LIPN, CNRS UMR 7030, F-93430 Villetaneuse, France

Silvano Dal Zilio ✉ 🆔
Univ. de Toulouse, INSA, LAAS, F-31400 Toulouse, France

Loïc Fejoz ✉
RealTime-at-Work (RTaW), 615, Rue du Jardin Botanique, F-54600 Villers-lès-Nancy, France

Michael González Harbour ✉
Universidad de Cantabria, Santander, Spain

Susanne Graf ✉ 🆔
Univ. Grenoble Alpes, CNRS, Grenoble INP, VERIMAG, F-38000 Grenoble, France

J. Javier Gutiérrez ✉
Universidad de Cantabria, Santander, Spain

Rafik Henia[1] ✉
Thales Research & Technology, F-91767 Palaiseau, France

Didier Le Botlan 🆔
Univ. de Toulouse, INSA, LAAS, F-31400 Toulouse, France

Giuseppe Lipari ✉ 🆔
Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France

Julio Medina ✉
Universidad de Cantabria, Santander, Spain

Nicolas Navet ✉
University of Luxembourg, Luxembourg

Sophie Quinton[1] ✉
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France

Juan M. Rivas ✉
Universidad de Cantabria, Santander, Spain

Youcheng Sun 🆔
The University of Manchester, UK

— **Abstract** ———

We propose here solutions to the FMTV 2015 challenge of a distributed video processing system using four different formalisms, as well as the description of the challenge itself. This artifact contains several solutions to various subchallenges, and instructions and scripts to reproduce these results smoothly.

---

[1] Corresponding author.

1 Scope

We propose here all necessary models and binaries to reproduce solutions to the FMTV 2015 challenge of a distributed video processing system using four different approaches. The results for each approach can be reproduced automatically.

The artifact package is made of five items, in five subdirectories:

- the description of the challenge itself;
- the solution using the timed model-checker Uppaal [7];
- the solution using the parametric timed model-checker IMITATOR [2];
- the solution using the timed model-checking framework Fiacre/Tina [3];
- the solution using the tool for real-time systems schedulability analysis MAST [6].

The challenge description. We propose here the complete UML model of the FMTV 2015 challenge in the open-source papyrus UML tool (`https://www.eclipse.org/papyrus/`).

Solution using Uppaal. This artifact presents a solution [12] using the Uppaal model-checker for the first part of the challenge. It allows to reproduce the results shown in the global solution table and gives more insight in the chosen approach.

Solution using IMITATOR. This artifact contains all necessary models and binaries to reproduce solutions [13] to challenge 1A using the formalism of parametric timed automata [1] and the IMITATOR toolkit [2].

Solution using Fiacre/Tina. This artifact contains models, scripts and software used to reproduce the Fiacre/Tina solution to questions 1A and 1B [5]. This approach relies on the model-checking toolbox Tina [4], and on models defined using the Fiacre [3] real-time specification language.

Solution using MAST. This artifact encompasses links and content providing the various models, and tools used to produce the results shown by the MAST set of solutions [11] to the challenges. The responses explained in our paper and later collected for comparison reasons in the paper to which this artifact complements are backed by the analyses scenarios pointed here and the response times resulting from their study with the MAST tool set.

2   Content

The artifact package is made of five subdirectories. We give the detailed content of each subdirectory in the following.

The challenge description. The artifact package contains:
- the UML Model of the FMTV 2015 Challenge (`.uml`); and
- the Diagram Interchange (`.di`) file for the UML model.

Solution using Uppaal. The artifact package contains:
- `README.txt`: a file that explains how to download the Uppaal model-checker and how to redo all the verification steps;
- files with the `*.xml` and `*.q` extension containing the Uppaal models and properties checked;
- a bash script `run-all.sh` allowing to run the verification for different choices of parameters allowing to reproduce the result tables in the document mentioned below;
- a document `Uppaal-solution-ECRTS23.pdf` which explains the approach that we have chosen for verifying the challenge with Uppaal, and that contains tables summarizing the results obtained when running the script;
- text files `Table-*.txt` containing the output of running the script file with different arguments.

Solution using IMITATOR. The artifact package contains:
- a `README.md` file containing everything you may need to know to run the artifact;
- IMITATOR 2.7 "Butter Guéméné" compiled for Linux 64 bits (dating from July 2015), and mirrored from GitHub;
- the IMITATOR models compatible with IMITATOR 2.7;
- an extra Python script `parseIMI.py` to parse the IMITATOR 2.7 results, and to display them nicely;
- a basic script `reproduce.sh` to reproduce everything in a single command;
- the expected result;
- a full log `reproduce.log` resulting from the execution of `reproduce.sh`.

All necessary information concerning the artifact and the way to reproduce the results can be found in the `README.md` file.

Solution using Fiacre/Tina. The artifact package contains:
- a `README` file, in text and markdown versions, with instructions on how to reproduce our results;
- a binary version of the latest version (v3.7) of sift, a state-space exploration tool for time Petri nets (TPN) and TTS, an extension of TPN with complex data types and priorities;
- Fiacre models, `maxlatency.fcr` and `between2lost.fcr`, used for challenges 1A and 1B respectively, as well as their compiled versions, `max.tts` and `bb.tts`;
- a basic shell script, `run_experiments.sh`, to reproduce our experiments in a single command.

Solution using MAST. The artifact package includes:
- a `README.txt` file that describe the contents plus a table describing the relation between its contents and the results included in the paper;
- a `README.pdf` file with similar but easier to read data;
- the original article where the full results were initially posted [11]: `FMTV15_Solution_MAST.pdf`;
- the binary version of the MAST tool used [10]: `mast-bin-win-1-5-0-1.zip`;
- a zip file with all the models used and the results obtained [9]: `mastmodels.zip`.

Finally, each solution is accompanied by its original paper from 2015 in PDF format. A sixth subdirectory corresponding to the CPAL solution (not part of this artifact) contains only their paper in PDF.

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 challenge description. The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the WATER 2015 forum website: `https://ecrts.org/forum/download/WATERS2015_Industrial_Challenge_Thales.zip`.

Solution using Uppaal. The artifact is (for the time being) not available anywhere else.

Solution using IMITATOR. The artifact is (for the time being) not available anywhere else.

Solution using Fiacre/Tina. All files are available on a dedicated GitLab repository at `https://gitlab.laas.fr/dalzilio/fmtvchallenge2015`. The artifact is distributed both as an archive, `https://gitlab.laas.fr/dalzilio/fmtvchallenge2015/-/releases/permalink/latest`, based on the v1.0 release; and as a lightweight virtual machine for VirtualBox users, in OVA format, `https://doi.org/10.5281/zenodo.7944508`.

Solution using MAST. The artifact can be downloaded from there:
`http://mast.unican.es/waters15challenge/ECRTS2023_MAST_solution_artifact.zip`

4   Tested platforms

The challenge description. Any Linux-based OS with 64 bits or Windows 64 bits. To the best of our knowledge, the UML MARTE profile is required to modify the UML model.

Solution using Uppaal. Any Linux-based OS with 64 bits.

Solution using IMITATOR. Any Linux-based OS with 64 bits. To the best of our knowledge, no dependencies are required to reproduce the results.

Solution using Fiacre/Tina. We assume that the code is executed on a computer running a 64 bit version of Linux using `glibc` 2.27 (the scripts have been tested on Ubuntu 20.04). For the virtual machine, based on Alpine Linux, a machine with 1 core and 4 GiB of RAM should be enough. You can connect to the machine use login: `root`, and password: `root`.

Solution using MAST. The artifact has been tested on Windows. The concrete version of the tool used and the analyses performed have been tested on Windows 7 but there is a very high probability that it runs in current versions of windows. A binary of the tool is also available for Linux versions (see the MAST Web page [8]) and shall also work fine, though it has not been fully tested in this case.

5   License

The challenge description. The artifact is available under the GNU General Public License.

Solution using Uppaal. The Uppaal models and scripts are available under the Creative Commons CC0 1.0 Universal license.

Solution using IMITATOR. The artifact is available under the GNU General Public License.

Solution using Fiacre/Tina. The models and scripts are available under the Creative Commons CC0 1.0 Universal license (see the accompanying `LICENSE` file). Tina is closed source software, free to use and distribute, with attribution, see `https://projects.laas.fr/tina/`.

Solution using MAST. The artifact is available under the same license that holds the MAST tool [8]. That is: This material is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

6 MD5 sum of the artifact

8a474ddcc9c80e4ffdbbd8f8b1742c1d

7 Size of the artifact

58.4 MiB

A Installing the Uppaal artifact

A.1 Requirements

This needs:
- Uppaal (tested with versions 4.1.19 (in interactive mode only) and 5.0.0 (both interactive and batch mode) on Linux (Ubuntu on a laptop)

A (free academic) license must be requested at `https://uppaal.org/downloads/`. A zip file with the version that we used can be downloaded from `https://download.uppaal.org/uppaal-5.0/uppaal-5.0.0-rc2/uppaal-5.0.0-rc2-linux64.zip`.

A.2 Make it run

Installing Uppaal only requires to unzip the downloaded zip-file. Running Uppaal in interactive mode requires Java Runtime Environment (JRE) version ≥ 11 (allows to visualize the model graphically).

For reproducing the results, the interactive mode is not needed. But it requires to replace the 3rd line of script file `run-all.sh`, that is,

`VERIFYTA=/home/graf/something/uppaal-5.0.0-rc2-linux64/bin/verifyta`

so that VERIFYTA points to `uppaal-5.0.0-rc2-linux64/bin/verifyta` in the unzipped Uppaal directory. Then follow the instructions in the `README` file to run the script.

──── **References** ────

1 Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. Parametric real-time reasoning. In S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal, editors, *STOC*, pages 592–601, New York, NY, USA, 1993. ACM. `doi:10.1145/167088.167242`.

2 Étienne André. IMITATOR 3: Synthesis of timing parameters beyond decidability. In Rustan Leino and Alexandra Silva, editors, *CAV*, volume 12759 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2021. `doi:10.1007/978-3-030-81685-8_26`.

3 Bernard Berthomieu, Jean-Paul Bodeveix, Patrick Farail, Mamoun Filali, Hubert Garavel, Pierre Gauffilet, Frédéric Lang, and François Vernadat. Fiacre: an intermediate language for model verifica-tion in the TOPCASED environment. In *Embedded Real Time Software*, 2008.

4 Bernard Berthomieu, Pierre-Olivier Ribet, and François Vernadat. The tool TINA – construction of abstract state spaces for Petri nets and time Petri nets. *International Journal of Production Research*, 42(14):2741–2756, 2004. `doi:10.1080/00207540412331312688`.

5 Silvano Dal Zilio, Bernard Berthomieu, and Didier Le Botlan. Latency Analysis of an Aerial Video Tracking System Using Fiacre and Tina. Technical Report 15355, LAAS-CNRS, 2015. This work was presented at the FMTV verification challenge of WATERS 2015, the 6th International Workshop on Analysis Tools and Methodologies for

Embedded and Real-time Systems. URL: `https://hal.science/hal-01202741`.

6 Michael González Harbour, J. Javier Gutiérrez García, José Carlos Palencia Gutiérrez, and J. M. Drake Moyano. MAST: Modeling and analysis suite for real time applications. In *ECRTS*, pages 125–134. IEEE Computer Society, 2001. `doi: 10.1109/EMRTS.2001.934015`.

7 Kim Guldstrand Larsen, Paul Pettersson, and Wang Yi. UPPAAL in a nutshell. *International Journal on Software Tools for Technology Transfer*, 1(1-2):134–152, 1997. `doi:10.1007/s100090050010`.

8 MAST. MAST home Page. URL: `https://mast.unican.es`.

9 MAST. MAST models used to get the associated results. URL: `http://mast.unican.es/waters15challenge/mastmodels.zip`.

10 MAST. Version of the MAST tool used to get the associated results. URL: `http://mast.unican.es/mast-bin-win-1-5-0-1.zip`.

11 Julio L. Medina, Juan M. Rivas, J. Javier Gutiérrez, and Michael González Harbour. Solving the 2015 FMTV challenge using response time analysis with MAST. Solutions to the FMTV Challenge, Informal proceedings of the 6th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS 2015), July 2015. URL: `https://www.ecrts.org/forum/download/FMTV15_Solution_MAST.pdf`.

12 Lijun Shan and Susanne Graf. Timing verification of an aerial video tracking system using UPPAAL. Solutions to the FMTV Challenge, Informal proceedings of the 6th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS 2015), July 2015. URL: `https://www.ecrts.org/forum/download/FMTV15_Solution_UPPAAL.pdf`.

13 Youcheng Sun, Étienne André, and Giuseppe Lipari. Verification of two real-time systems using parametric timed automata. Solutions to the FMTV Challenge, Informal proceedings of the 6th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS 2015), July 2015. URL: `https://www.ecrts.org/forum/download/FMTV15_Solution_Parametric_Timed_Automata.pdf`.