The journal for the international Ada community

Ada User Journal

Volume 44
Number 4
December 2023

Editorial 251
Quarterly News Digest 252
Conference Calendar 261
Forthcoming Events 267

Proceedings of the Workshop on Challenges and New approaches for Dependable and Cyber-physical Systems Engineering of AEiC 2023

A. Moussaoui, A. Bagnato
The MORPHEMIC Project on the Data Intensive E-Brain Science Case Study 269

A. Pimentel et al.
The ADMORPH Approach for Adaptively Morphing Embedded Systems 274

M. García-Gordillo, J.J. Valls, J. Coronel, S. Sáez
Mode Change Management for Adaptive Cyber-physical Systems 280

A. Medaglini, S. Bartolini
Performance Study of Object Tracking with Multiple Kalman Filters in Autonomous Driving Systems 284

S. Royuela et al.
Multi-criteria Analysis and Optimisation in the AMPERE Ecosystem 288

A. Raynaud, T. Serru, N. Nguyen
Attack Scenarios Generation Algorithm Based on Discrete Event System Formalism 294

Produced by Ada-Europe

Information on Subscriptions and Advertisements

Ada User Journal (ISSN 1381-6551) is published in one volume of four issues. The Journal is provided free of charge to members of Ada-Europe. Library subscription details can be obtained direct from the Ada-Europe General Secretary (contact details above). Claims for missing issues will be honoured free of charge, if made within three months of the publication date for the issues. Mail order, subscription information and enquiries to the Ada-Europe General Secretary.

For details of advertisement rates please contact the Ada-Europe General Secretary (contact details above).

ADA USER JOURNAL

Volume 44
Number 4
December 2023

Contents

	Page
Editorial Policy for Ada User Journal	250
Editorial	251
Quarterly News Digest	252
Conference Calendar	261
Forthcoming Events	267
Proceedings of the Workshop on Challenges and New Approaches for Dependable and Cyber-physical Systems Engineering of AEiC 2023 (DeCPS 2023)	
A. Moussaoui, A. Bagnato *"The MORPHEMIC Project on the Data Intensive E-Brain Science Case Study"*	269
A. Pimentel et al. *"The ADMORPH Approach for Adaptively Morphing Embedded Systems"*	274
M. García-Gordillo, J. J. Valls, J. Coronel, S. Sáez *"Mode Change Management for Adaptive Cyber-physical Systems"*	280
A. Medaglini, S. Bartolini *"Performance Study of Object Tracking with Multiple Kalman Filters in Autonomous Driving Systems"*	284
S. Royuela, E. Quinones, A. Munera, T. Carvalho, L. M. Pinho, M. Samadi, T. Cucinotta, G. Ara, F. Paladino, S. Mazzola, T. Benz *"Multi-criteria Analysis and Optimisation in the AMPERE Ecosystem"*	288
A. Raynaud, T. Serru, N. Nguyen *"Attack Scenarios Generation Algorithm Based on Discrete Event System Formalism"*	294
Ada-Europe Associate Members (National Ada Organizations)	298
Ada-Europe Sponsors	Inside Back Cover

Editorial Policy for Ada User Journal

Publication

Ada User Journal — The Journal for the international Ada Community — is published by Ada-Europe. It appears four times a year, on the last days of March, June, September and December. Copy date is the last day of the month of publication.

Aims

Ada User Journal aims to inform readers of developments in the Ada programming language and its use, general Ada-related software engineering issues and Ada-related activities. The language of the journal is English.

Although the title of the Journal refers to the Ada language, related topics, such as reliable software technologies, are welcome. More information on the scope of the Journal is available on its website at *www.ada-europe.org/auj*.

The Journal publishes the following types of material:

- Refereed original articles on technical matters concerning Ada and related topics.
- Invited papers on Ada and the Ada standardization process.
- Proceedings of workshops and panels on topics relevant to the Journal.
- Reprints of articles published elsewhere that deserve a wider audience.
- News and miscellany of interest to the Ada community.
- Commentaries on matters relating to Ada and software engineering.
- Announcements and reports of conferences and workshops.
- Announcements regarding standards concerning Ada.
- Reviews of publications in the field of software engineering.

Further details on our approach to these are given below. More complete information is available in the website at *www.ada-europe.org/auj.*

Original Papers

Manuscripts should be submitted in accordance with the submission guidelines (below).

All original technical contributions are submitted to refereeing by at least two people. Names of referees will be kept confidential, but their comments will be relayed to the authors at the discretion of the Editor.

The first named author will receive a complimentary copy of the issue of the Journal in which their paper appears.

By submitting a manuscript, authors grant Ada-Europe an unlimited license to publish (and, if appropriate, republish) it, if and when the article is accepted for publication. We do not require that authors assign copyright to the Journal.

Unless the authors state explicitly otherwise, submission of an article is taken to imply that it represents original, unpublished work, not under consideration for publication elsewhere.

Proceedings and Special Issues

The *Ada User Journal* is open to consider the publication of proceedings of workshops or panels related to the Journal's aims and scope, as well as Special Issues on relevant topics.

Interested proponents are invited to contact the Editor-in-Chief.

News and Product Announcements

Ada User Journal is one of the ways in which people find out what is going on in the Ada community. Our readers need not surf the web or news groups to find out what is going on in the Ada world and in the neighbouring and/or competing communities. We will reprint or report on items that may be of interest to them.

Reprinted Articles

While original material is our first priority, we are willing to reprint (with the permission of the copyright holder) material previously submitted elsewhere if it is appropriate to give it a wider audience. This includes papers published in North America that are not easily available in Europe.

We have a reciprocal approach in granting permission for other publications to reprint papers originally published in *Ada User Journal.*

Commentaries

We publish commentaries on Ada and software engineering topics. These may represent the views either of individuals or of organisations. Such articles can be of any length – inclusion is at the discretion of the Editor.

Opinions expressed within the *Ada User Journal* do not necessarily represent the views of the Editor, Ada-Europe or its directors.

Announcements and Reports

We are happy to publicise and report on events that may be of interest to our readers.

Reviews

Inclusion of any review in the Journal is at the discretion of the Editor. A reviewer will be selected by the Editor to review any book or other publication sent to us. We are also prepared to print reviews submitted from elsewhere at the discretion of the Editor.

Submission Guidelines

All material for publication should be sent electronically. Authors are invited to contact the Editor-in-Chief by electronic mail to determine the best format for submission. The language of the journal is English.

Our refereeing process aims to be rapid. Currently, accepted papers submitted electronically are typically published 3-6 months after submission. Items of topical interest will normally appear in the next edition. There is no limitation on the length of papers, though a paper longer than 10,000 words would be regarded as exceptional.

Editorial

I would like to start this last issue of 2023 by wishing that the new year, 2024, will be a great year for all Ada-Europe members and AUJ readers, with health and success. Speaking of good news, we continue working towards the merge between the AUJ and the Ada Letters. I must say, however, that the process is not straightforward. In fact, it will be more complicated than initially expected and hence there is not much I can say for the moment. I will get back to this topic in a future issue. An immediate consequence of this situation is that the production of the AUJ will be kept as usual, at least concerning the first issue in 2024.

As for the technical contents, this issue provides the proceedings of the "Workshop on Challenges and New Approaches for Dependable and Cyber-physical Systems Engineering (DeCPS-2023)", which took place with AEiC 2023, in Lisbon, Portugal, last June. Out of the 6 papers included in these proceedings, 4 papers describe work that was done in the scope of the following H2020 European projects: MORPHEMIC, ADMORPH, AMPERE and TRANSACT. The first 3 ended in June 2023, hence the papers provide a global overview of the main results that were achieved. The goal of MORPHEMIC was to solve challenges in the deployment of cloud applications on multi-cloud platforms. ADMORPH tackled safety and security requirements of CPS, by developing solutions ranging from design to run-time monitoring and adaptation. AMPERE focused on the development of a model-driven engineering (MDE) framework to address cost and complexity challenges in the development of multi-component computing platforms. The 4th of these projects, TRANSACT, will end in May 2024 and aims at achieving safer and more secure CPS, much like ADMORPH. In this case, the project looks at safety-critical CPS that rely on edge and cloud computing and, besides providing an overview of the project objectives, the paper presents an operational mode change management solution for multimode applications. The proceedings also include a paper authored by A. Medaglini and S Bartolini, which addresses the problem of object tracking and provides a solution based on a Kalman filter that exploits parallelization to improve performance. The experimental evaluation is done using data from a real tramway use-case. The last paper, by A. Raynaud, T. Serru and N. Nguyen, proposes a tool to model the architecture and the behaviour of CPS in the presence of cyberattacks. It also presents an algorithm to generate all the attack scenarios, which is evaluated by considering the navigation system of an autonomous vessel as a target system.

Last but not the least, the issue includes the News Digest section and the Calendar and Events section, respectively prepared by Alejandro R. Mosteo and Dirk Craeynest, their editors.

Antonio Casimiro
Lisboa
December 2023
Email: AUJ_Editor@Ada-Europe.org

Quarterly News Digest

Alejandro R. Mosteo

Centro Universitario de la Defensa de Zaragoza, 50090, Zaragoza, Spain; Instituto de Investigación en Ingeniería de Aragón, Mariano Esquillor s/n, 50018, Zaragoza, Spain; email: amosteo@unizar.es

Contents

Preface by the News Editor	252
Ada-related Events	252
Ada-related Resources	253
Ada-related Tools	253
Ada and Operating Systems	253
Ada Inside	255
Ada and Other Languages	255
Ada Practice	257

[Messages without subject/newsgroups are replies from the same thread. Messages may have been edited for minor proofreading fixes. Quotations are trimmed where deemed too broad. Sender's signatures are omitted as a general rule. —arm]

Preface by the News Editor

Dear Reader,

One of the most significant features of Ada 2022 is the new light-weight parallelism. Although no Ada compiler implements it as of this writing, equivalent features are now available in library form thanks to Tucker Taft [1]. Let us hope this brings us near an actual Ada 2022 implementation.

While preparing this issue, I learned about the "Beaujolais Effect" [2], a challenge issued by Ada's original designer, Jean Ichbiah. He offered a bottle of Beaujolais wine to the person that found an example of changing the behavior of an Ada 83 program by adding/removing a "use" clause. Was there ever a winner? Find out at the referenced thread!

Sincerely,

Alejandro R. Mosteo.

[1] "Light-weight Parallelism Threading Library Based on Ada 2022 Features", in Ada-related Tools.

[2] "Beaujolais Challenge", in Ada Practice.

Ada-related Events

Ada Monthly Meetup 2023

From: Fernando Oleo / Irvise <irvise_ml@irvise.xyz>
Subject: Re: Ada Monthly Meetup 2023
Date: Wed, 11 Oct 2023 19:16:21 +0200
Newsgroups: comp.lang.ada

[Past events for the record. —arm]

I would like to announce the November Ada Monthly Meetup which will be taking place on the 4th of November at **14:00 UTC time (15:00 CET)**. As always the meetup will take place over at Jitsi. Hopefully this time I will not have the same amount of technical issues…

If someone would like to propose a talk or a topic, feel free to do so! Streaksu, the creator of the [Ironclad] (https://ironclad.cx/) kernel, has volunteered to give an introductory talk and demonstration of the OS :)

Here are the connection details from previous posts:

The meetup will take place over at Jitsi, a conferencing software that runs on any modern browser. The link is [Jitsi Meet] (https://meet.jit.si/AdaMonthlyMeetup) The room name is "AdaMonthlyMeetup" and in case it asks for a password, it will be set to "AdaRules".

I do not want to set up a password, but in case it is needed, it will be the one above without the quotes. The room name is generally not needed as the link should take you directly there, but I want to write it down just in case someone needs it.

Best regards and see you soon!
Fer

P.S: careful with the time! In the EU we will enter daylight savings time at the end of October. For that reason, I have decided to change the meeting to UTC 14:00h! Please, check your local timezones!

P.P.S: the October meeting went pretty well! We had quite a few people and two presentations, one from Francesc, who introduced [Alice] (https://github.com/alice-adventures/Alice) and Rod Kay, who showed his work on [SWIG4Ada] (https://github.com/charlie5/swig4ada).

The meeting was livestreamed to Youtube and can be watched [here] (https://www.youtube.com/watch?v=0Pnuy663gZM) (the video resolution is only 360p for the time being).

From: Fernando Oleo / Irvise <irvise_ml@irvise.xyz>
Date: Mon, 13 Nov 2023 22:19:27 +0100

I would like to announce the December Ada Monthly Meetup which will be taking place on the 2nd of December at **14:00 UTC time (15:00 CET)**. As always the meetup will take place over at Jitsi. The Meetup will also be livestreamed to Youtube :)

If someone would like to propose a talk or a topic, feel free to do so! We currently have no topics ;)

Here are the connection details from previous posts: [...]

2nd Call for Contributions - AEiC 2024

From: Dirk Craeynest <dirk@orka.cs.kuleuven.be>
Subject: Ada-Europe Conference - 2nd Call for Contributions - AEiC 2024
Date: Wed, 22 Nov 2023 16:54:55 -0000
Newsgroups: comp.lang.ada, fr.comp.lang.ada, comp.lang.misc

[CfC is included in the Forthcoming Events Section —arm]

Grants for Open Access - AEiC 2024

From: Dirk Craeynest <dirk@orka.cs.kuleuven.be>
Subject: AEiC 2024 - Ada-Europe conference - grants for Open Access
Date: Thu, 21 Dec 2023 17:14:52 -0000
Newsgroups: comp.lang.ada, fr.comp.lang.ada

Season's greetings from the organizers of the 28th Ada-Europe International Conference on Reliable Software Technologies (AEiC 2024), to be held 11-14 June 2024, in Barcelona, Spain!

Accepted Journal Track papers will be published in the conference's Special Issue of the Journal of Systems Architecture (JSA). Note that the Ada-Europe organization will waive the Open Access fees for the first four accepted papers, which do not already enjoy OA from other agreements with the Publisher.

www.ada-europe.org/conference2024/cfp.html#cfpjournal
(V3.1)

Ada-related Resources

[Delta counts are from October 10th to February 19th. —arm]

Ada on Social Media

From: Alejandro R. Mosteo <amosteo@unizar.es>
Subject: Ada on Social Media
Date: 19 Feb 2024 16:03 CET
To: Ada User Journal readership

Ada groups on various social media:

- Reddit: 8_561 (+139) members [1]
- LinkedIn: 3_479 (+25) members [2]
- Stack Overflow: 2_393 (+28) questions [3]
- Gitter: 243 (+14) people [4]
- Telegram: 173 (+15) users [5]
- Ada-lang.io: 182 (+36) users [6]
- Libera.Chat: 76 (+4) concurrent users [7]

[1] http://old.reddit.com/r/ada/

[2] https://www.linkedin.com/groups/114211/

[3] http://stackoverflow.com/questions/tagged/ada

[4] https://app.gitter.im/#/room/#ada-lang_Lobby:gitter.im

[5] https://t.me/ada_lang

[6] https://forum.ada-lang.io/u

[7] https://netsplit.de/channels/details.php?room=%23ada&net=Libera.Chat

Repositories of Open Source Software

From: Alejandro R. Mosteo <amosteo@unizar.es>
Subject: Repositories of Open Source software
Date: 19 Feb 2024 16:10 CET
To: Ada User Journal readership

GitHub: 1000* (=) developers [1]
Rosetta Code: 940 (=) examples [2]
38 (=) developers [3]
Alire: 393 (+23) crates [4]
Sourceforge: 248 (+1) projects [5]
Open Hub: 214 (=) projects [6]
Codelabs: 57 (=) repositories [7]
Bitbucket: 37 (=) repositories [8]

* This number is an unreliable lower bound due to GitHub search limitations.

[1] https://github.com/search?q=language%3AAda&type=Users

[2] https://rosettacode.org/wiki/Category:Ada

[3] https://rosettacode.org/wiki/Category:Ada_User

[4] https://alire.ada.dev/crates.html

[5] https://sourceforge.net/directory/language:ada/

[6] https://www.openhub.net/tags?names=ada

[7] https://git.codelabs.ch/?a=project_index

[8] https://bitbucket.org/repo/all?name=ada&language=ada

Language Popularity Rankings

From: Alejandro R. Mosteo <amosteo@unizar.es>
Subject: Ada in language popularity rankings
Date: 19 Feb 2024 16:13 CET
To: Ada User Journal readership

[Positive ranking changes mean to go up in the ranking. —arm]

- TIOBE Index: 25 (-2) 0.77% (=) [1]
- PYPL Index: 15 (+1) 1.08% (+0.04%) [2]
- Stack Overflow Survey: 42 (=) 0.77% (=) [3]
- IEEE Spectrum (general): 36 (=) Score: 0.0107 (=) [4]
- IEEE Spectrum (jobs): 29 (=) Score: 0.0173 (=) [4]
- IEEE Spectrum (trending): 30 (=) Score: 0.0122 (=) [4]

[1] https://www.tiobe.com/tiobe-index/

[2] http://pypl.github.io/PYPL.html

[3] https://survey.stackoverflow.co/2023/

[4] https://spectrum.ieee.org/top-programming-languages/

Ada-related Tools

UXStrings 0.6.0

From: Blady <p.p11@orange.fr>
Subject: [ANN] Release of UXStrings 0.6.0
Date: Sat, 14 Oct 2023 18:33:19 +0200
Newsgroups: comp.lang.ada

This Ada library provides Unicode character strings of dynamic length. It is now available on Alire [1] in version 0.6.0.

Changes:

- Add string convenient subprograms [2]: Contains, Ends_With, Starts_With, Is_Lower, Is_Upper, Is_Basic, Is_Empty, Remove, Replace.
- Add list of strings with convenient subprograms [3]: Append_Unique, Filter, Join, Remove_Duplicates, Replace, Slice, Sort, Is_Sorted, Merge and Split on strings.

So far in UXStrings, its API are similar to those of the strings Ada standard libraries. If you find some missing, make your proposals on Github.

NB: UXStrings3 is now the default implementation.

[1] https://alire.ada.dev/crates/uxstrings.html

[2] https://github.com/Blady-Com/UXStrings/blob/master/src/uxstrings3.ads#L346

[3] https://github.com/Blady-Com/UXStrings/blob/master/src/uxstrings-lists.ads

[4] https://github.com/Blady-Com/UXStrings/issues

Source Code for the ARM Formatting Tool

From: Vincent D. <vincent.diemunsch@gmail.com>
Subject: Source code for the ARM Formatting Tool
Date: Wed, 25 Oct 2023 14:15:00 -0700
Newsgroups: comp.lang.ada

I have tried to download the source code of the formatting tool from the site http://ada-auth.org/arm.html, but it seems that the package ARM_Paragraph is missing.

Does anyone know how to get this file?

From: Maxim Reznik <reznikmm@gmail.com>
Date: Thu, 26 Oct 2023 01:15:19 -0700

I have a github repository synced with ada-auth Web CVS. I was able to build the formatting tool from the source.

https://github.com/reznikmm/ada-auth/

From: Vincent D. <vincent.diemunsch@gmail.com>
Date: Thu, 26 Oct 2023 07:58:13 -0700

Thank you for the link on GitHub, but the build didn't work for me: I get the same error regarding "ARM_Paragraph" package missing.

```
$ git clone https://github.com/reznikmm/ada-auth.git
$ cd ada-auth
$ gprbuild -p -P ada_form.gpr
Setup
[mkdir] object directory for project Ada_Form
Compile
[Ada] arm_form.ada
arm_form.ada:6:06: error: file "arm_paragraph.ads" not found
arm_form.ada:6:06: error: "Arm_Formatter
```

```
(body)" depends on "Arm_Master (spec)"
arm_form.ada:6:06: error: "Arm_Master
(spec)" depends on "Arm_Format (spec)"
arm_form.ada:6:06: error: "Arm_Format
(spec)" depends on "Arm_Paragraph (spec)"
gprbuild: *** compilation phase failed
```

From: Vincent D.
<vincent.diemunsch@gmail.com>
Date: Thu, 26 Oct 2023 08:46:04 -0700

Versions that compile:

- 4500f560 Corrected note format for ISO version
- 29db0326 Split out the normative references clause.
- ff3db3ca Various updates for FDIS work and draft 34.

Versions that do not compile:

- 0e95e912 Various updates for FDIS 2.0.
- 4d93b18c A number of small formatting changes, mostly only for the FDIS.
- 260566bd Various updates for FDIS/Draft 35.

The problem appears in version 0e95e912 "Various updates for FDIS 2.0." from the 23/09/2022 where with ARM_Paragraph is added to arm_frm.adb but the package was not added to the sources.

From: Simon Wright
<simon@pushface.org>
Date: Thu, 26 Oct 2023 18:06:19 +0100

> the package was not added to the sources.

Because it's not in CVS.

From: Randy Brukardt
<randy@rrsoftware.com>
Date: Tue, 31 Oct 2023 20:56:54 -0500

> Because it's not in CVS.

It is now. And it always was in the ZIP file of the source. (That is made from the files that I use to compile the tool, so it should always be compilable.)

One of the big downsides of working at home is that some support functions get delayed until one gets into the office -- and that means that they're easily forgotten. As in this case, checking the new files into the CVS (the existing files were updated, of course, leaving a mess for anyone trying to build from the CVS).

Sorry about that.

From: Vincent D.
<vincent.diemunsch@gmail.com>
Date: Fri, 3 Nov 2023 15:02:02 -0700

> It is now. And it always was in the ZIP file of the source. (That is made from the files that I use to compile the tool, so it should always be compilable.)

I am sorry, but even if I put the package "arm_paragraph" in the source code, I get compile errors. For instance in version 0e95e9125e066ce564fe369221821452535b6260:

```
gprbuild -p -P ada_form.gpr
Compile
[Ada] arm_form.ada
[Ada] arm_cont.adb
[Ada] arm_frm.adb
arm_frm.adb:1163:13: error: missing case
value: "Usage"
arm_frm.adb:9357:53: error:
"Numbered_T_and_D_List" not declared in
"ARM_Database"
arm_frm.adb:9362:53: error: "T_and_D_List"
not declared in "ARM_Database"
arm_frms.adb:1415:33: error: unmatched
actual "Note1_Text" in call
gprbuild: *** compilation phase failed
```

And with the latest: gprbuild -p -P ada_form.gpr

```
Compile
[Ada] arm_form.ada
arm_form.ada:263:41: error: "Rest" not
declared in "ARM_Master"
gprbuild: *** compilation phase failed
```

What is the ZIP file of the source that you mentioned? Where can I find it?

From: Randy Brukardt
<randy@rrsoftware.com>
Date: Thu, 16 Nov 2023 19:17:08 -0600

On each of the individual Reference Manual pages (that is, Ada 2012, Ada 2022, etc.) on Ada-Auth.org, at the bottom, you will find links for the formatting tool, one for a Windows executable, one for the CVS, and one for a ZIP file containing the source.

I'm not sure why you are seeing compilation problems with the CVS; it appears complete and correct on my end. Did you make sure that you have the latest versions of all of the files (they were updated on October 3rd)?

For example, the ARM_Frm.Adb file should have a change entry of: - 9/11/23 - RLB - Added Usage category and commands. and of course have code for the Usage category and commands. I've started some work for the post-Ada 2022 RM (currently known as Ada 202y) - the tool is constantly evolving.

One of the advantages of using the ZIP files is that they reflect the tool as it was used to generate a specific version of the RM; the "current" version of the tool probably only has been tested on the "current" version of the RM source and thus it is not certain to work perfectly.

Light-weight Parallelism Threading Library Based on Ada 2022 Features

From: Tucker Taft
<tucker.taft@gmail.com>
Subject: Light-weight parallelism threading library based on Ada 2022 features
Date: Tue, 31 Oct 2023 15:43:44 -0700
Newsgroups: comp.lang.ada

A full implementation of the parallel features of Ada 2022 is yet to be released. In the meantime, here is a light-weight-threading library that provides essentially all of the parallel features of Ada 2022, using various generics, etc. Scheduling is provided using a plug-in architecture. If no scheduler is plugged in, the light-weight threads are simply executed sequentially. If a light-weight-thread scheduler is plugged in, then the light-weight threads spawned by instances of the various generic packages are managed by that scheduler.

There are currently two LWT scheduler plug-ins:

* a wrapper for the GNU implementation of OpenMP (lwt-openmp.ads)
* a work-stealing based plug-in, written entirely in Ada

Below is a link to the "readme.md" documentation for the GitHub lwt library. It is currently part of the ParaSail GitHub repository, but the files in "lwt" are actually independent of ParaSail. ParaSail has its own work-stealing-based scheduler built-in, but at some point we plan to shift over to using the "lwt" library. But at the moment, there is no dependence either way between the ParaSail interpreter/compiler and the lwt library.

https://github.com/parasail-lang/parasail/tree/main/lwt#light-weight-threading-library-for-ada-2022

Feel free to open GitHub Issues if you find problems with the implementation, or have suggestions for improvements.

Enjoy!

-Tucker Taft

The ParaSail GitHub repository was created by my colleague Olivier Henley, and he has also helped to improve the documentation and testing scripts. Much appreciated!

GtkAda Contributions 3.32

From: Dmitry A. Kazakov
<mailbox@dmitry-kazakov.de>
Subject: ANN: GtkAda contributions v3.32
Date: Mon, 18 Dec 2023 12:34:59 +0100
Newsgroups: comp.lang.ada

The library is an extension of GtkAda dealing with the following issues:

- Tasking support;
- Custom models for tree view widget;
- Custom cell renderers for tree view widget;
- Multi-columned derived model;
- Extension derived model (to add columns to an existing model);
- Abstract caching model for directory-like data;

- Tree view and list view widgets for navigational browsing of abstract caching models;
- File system navigation widgets with wildcard filtering;
- Resource styles;
- Capturing resources of a widget;
- Embeddable images;
- Some missing subprograms and bug fixes;
- Measurement unit selection widget and dialogs;
- Improved hue-luminance-saturation color model;
- Simplified image buttons and buttons customizable by style properties;
- Controlled Ada types for GTK+ strong and weak references;
- Simplified means to create lists of strings;
- Spawning processes synchronously and asynchronously with pipes;
- Capturing asynchronous process standard I/O by Ada tasks and by text buffers;
- Source view widget support;
- SVG images support.

http://www.dmitry-kazakov.de/ada/gtkada_contributions.htm

Changes (18 December 2023) to the version 3.31:

- Get_CSS_Name and Set_CSS_Name were added to Gtk.Missed;
- Gtk.Widget.Styles.CSS_Store changed to enumerate labels of Gtk_Notebook or its descendants;
- Since Gtk broke its CSS rules and the widget class cannot be used in the CSS style anymore and because unset widget names are defaulted to the class name but have no effect on the style name, Gtk.Widget.Styles.CSS_Store was changed to report only the widget name if different from the class names.

Ada and Operating Systems

Spurious Error with GNAT 13.2.0 on Intel MacOS 17.2

From: Moi <findlaybill@blueyonder.co.uk>
Subject: spurious error with GNAT 13.2.0 on Intel macOS 17.2
Date: Thu, 14 Dec 2023 23:49:15 +0000
Newsgroups: comp.lang.ada

The 17.2 update is accompanied by updated Command Line Tools, so, having made a copy of the current CLTs, I let it update.

With GNAT 13.2.0 on Intel macOS 17.2 this happens:

/Users/wf/KDF9/emulation/Testing: chmod 444 ST0

On compiling and running the minimum test case:

```
with Ada.Direct_IO;
with Ada.IO_Exceptions;
procedure failure is
 package my_IO is new
    Ada.Direct_IO(Integer);
  use my_IO;
     my_file : File_Type;
begin
   Open(my_file, Inout_File, "ST0");
exception
   when Ada.IO_Exceptions.Use_Error =>
     raise program_error with "Open
     failed to get RW access";
end failure;
```

I get:

A. with ”-largs -Wl,-ld_classic” in the linker parameters:

/Users/wf/KDF9/emulation/Testing: failure raised PROGRAM_ERROR: Open failed to get RW access

This is what should happen.

B. recompiled and relinked without ”-largs -Wl,-ld_classic”:

/Users/wf/KDF9/emulation/Testing: failure raised CONSTRAINT_ERROR: erroneous memory access

OOPS!

Strangely, this is now the ONLY one out of dozens of regression tests that fails in this software configuration. Previously, they all failed at the link stage.

From: Simon Wright <simon@pushface.org>
Date: Fri, 15 Dec 2023 15:23:31 +0000

> The 17.2 update is accompanied by updated Command Line Tools, so, having made a copy of the current CLTs, I let it update.

I think you mean 14.2!

GCC 13/CLT 15.* still has issues with exceptions, eg this unresolved issue: https://github.com/iains/gcc-13-branch/issues/10, and the ld-classic dance fixes them as far as I can tell.

I'm working with this fix: https://github.com/simonjwright/xcode_15_fix

which is included in my GCC 13.2.0 build: https://github.com/simonjwright/distributing-gcc/releases/tag/gcc-13.2.0-aarch64

> Strangely, this is now the ONLY one out of dozens of regression tests that fails in this software configuration. Previously, they all failed at the link stage.

Yes, the 15.1 betas fixed the linking problem. I've no idea why some exceptions don't get caught/raise memory access errors. I do think that the aarch64 GCC 14 has fixed the problem.

Ada Inside

First Ada DO-178 Certification

From: Jeffrey R.Carter <spam.jrcarter.not@spam.acm.org.not>
Subject: First Ada DO-178 Certification
Date: Wed, 4 Oct 2023 13:39:39 +0200
Newsgroups: comp.lang.ada

Does anyone remember when Ada avionics S/W was first certified to DO-178? My memory is 1980s, but I haven’t been able to find any information about it.

Ada and Other Languages

Upcasting Interfaces with CPP Convention in GNAT

From: Kura <kuraitou@gmail.com>
Subject: Upcasting interfaces with CPP convention in GNAT
Date: Thu, 2 Nov 2023 05:20:08 -0700
Newsgroups: comp.lang.ada

I'm trying to figure out how to correctly perform an upcast in GNAT using interface types that have the CPP convention. I have two types corresponding to C++ classes: IBase and IDerived, each with corresponding access types suffixed with _Ptr. The library that I'm wrapping returns a pointer to some concrete implementation of IDerived, but the moment I convert it to an IBase_Ptr to pass it to functions within the library, the program segfaults while performing some kind of tag check. I've found information on this topic to be very sparse, and the GNAT manual has no discussion about allowed conversions or examples showing how to do this. This seems like it should be an allowed conversion and the program works as expected if I either replace the usage of Base_Ptr with Derived_Ptr within the function wrapper specification (not feasible because there are many types deriving from Base_Ptr and adding overloads would affect its vtable) or use Unchecked_Conversion between the two pointer types (unsure if this is safe). I'm also aware I could lay out the vtables manually and do away with tagged types, but I would like to avoid that if possible.

Here is a minimal example that segfaults, all compiled with the same toolchain on windows/mingw64:

==== repro.adb

```
with Wrap; use Wrap;
procedure Repro is
   P : IDerived_Ptr := GetDerivedInstance;
   Q : IBase_Ptr := IBase_Ptr (P);
begin
  null;
end Repro;
```

==== wrap.ads

```
package Wrap is
   type IBase is interface;
   pragma Convention (C_Plus_Plus, IBase);
   type IBase_Ptr is access all IBase'Class;

   type IDerived is interface and IBase;
   pragma Convention (C_Plus_Plus,
   IDerived);
type IDerived_Ptr is access all
IDerived'Class;

   function GetDerivedInstance return
   IDerived_Ptr with Import => True,
   Convention => C_Plus_Plus,
   External_Name => "GetDerivedInstance";
end Wrap;
```

==== lib.cpp

```
class IBase {};
class IDerived : public IBase {};
class Impl : public IDerived {
public:
        Impl() = default;
};

extern "C"

IDerived* GetDerivedInstance() {

        return new Impl();
}
```

==== end example

Error and stack trace:

```
Thread 1 received signal SIGSEGV,
Segmentation fault.
ada.tags.offset_to_top
(this=(system.address) 0x6591710) at a-
tags.adb:806
806 a-tags.adb: No such file or directory.
(gdb) bt
#0 ada.tags.offset_to_top
(this=(system.address) 0x6591710) at
a-tags.adb:806
#1 ada.tags.base_address
(this=(system.address) 0x6591710) at
a-tags.adb:286
#2 ada.tags.displace (this=(system.address)
0x6591710, t=0x7ff7fe8301c8
<wrap__ibaseT+8> (wrap.ibase)) at
a-tags.adb:354
#3 0x00007ff7fe82649c in repro () at
C:\repro\src\repro.adb:5
```

Any ideas?

From: J-P. Rosen <rosen@adalog.fr>
Date: Thu, 2 Nov 2023 15:11:54 +0100

I don't know if this is the cause of your problem, but you should give convention C_Plus_Plus to the pointer types too (IBase_Ptr and IDerived_Ptr).

From: Kura <kuraitou@gmail.com>
Date: Thu, 2 Nov 2023 08:08:24 -0700

That did not solve my problem, but thank you for the tip.

From: Kura <kuraitou@gmail.com>
Date: Fri, 3 Nov 2023 14:13:05 -0700

I was able to work around the issue by using abstract tagged null records instead of interfaces - no other changes necessary. It seems that interfaces can only be at the very top of a hierarchy even if you're only extending another interface.

Ada vs. Rust for Low-level System Software

From: Nasser M. Abbasi <nma@12000.org>
Subject: Ada vs. Rust for low level system software
Date: Tue, 12 Dec 2023 22:28:40 -0600
Newsgroups: comp.lang.ada

Has anyone made a study of differences between Rust and Ada for low level hardware system software?

Since Ada is mainly used in this area, why has Rust, which is a much younger language, and target this same area has gained so much popularity but not Ada?

https://dl.acm.org/doi/fullHtml/10.1145/3551349.3559494

"Rust is a rising programming language designed to build system software [4, 10, 20]. On the one hand, Rust offers access to and control of the low-level system resources. On the other hand, unlike conventional systems programming languages, Rust ensures memory and concurrency safety"

"Rust often inserts bound checks at the execution time to rule out out-of-bound accesses"

Well, does not Ada also "ensures memory and concurrency safety" and checks for out-of-bound accesses?

I am just wondering what Rust brings to the table that Ada does not have and why is Rust becoming so popular when Ada is not.

I never used Rust myself, but used Ada.

Has anyone done a study comparing the two languages or knows both that can give some comments on this?

From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>
Date: Wed, 13 Dec 2023 09:27:34 +0100

> Has anyone made study [...]

What for? Any language comparisons lost their meaning long ago as the whole language business degraded into hobbyist/corporate bullshit.

> why has Rust [...] gained so much popularity but not Ada?

Because it is always someone's arbitrary decision.

In my view Rust brings nothing and moreover is a huge step back compared to Ada. Its main and only idea is to force the programmer to explicitly manage memory through references where Ada simply uses object notation regardless of the mechanism doing the same under the hood.

Safety comes not through references but by limiting the number of cases you must resort to using dynamic allocation for statically scoped objects. E.g. Unbounded_String in Ada.

As for my major concern - the type system and the abstraction mechanisms in general, there is nothing in Rust at all.

Then of course Rust continues the worst practices tried by Ada and C++: templates/generics, macros.

From: Jeffrey R.Carter <spam.jrcarter.not@spam.acm.org.not>
Date: Wed, 13 Dec 2023 09:44:34 +0100

> Has anyone made study of difference between Rust and Ada for low level hardware system software?

You might be interested in this discussion: https://www.reddit.com/r/ada/comments/18c2nr4/where_is_ada_safer_than_rust/

From: Luke A. Guest <laguest@archeia.com>
Date: Wed, 13 Dec 2023 09:10:20 +0000

>> why has Rust gained so much popularity but not Ada?

I would say because they aimed to be a C++ replacement.

[...]

> templates/generics, macros.

What's the alternative to generics?

From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>
Date: Wed, 13 Dec 2023 10:53:28 +0100

> What's the alternative to generics?

The question is what is the alternative to static/parametric polymorphism. The answer is dynamic polymorphism.

1. Dynamic polymorphism in Ada is as static as generics are. No run-time penalty unlikely to C++.
2. It covers cases generics do not, e.g. you can have class-wide run-time objects and proper class-wide subprograms.
3. It supports modular programming en large. E.g. you can put a class member in a dynamically linked library.
4. It is fully testable. Generics are fundamentally non-testable, only concrete instances are.

5. It does not create a meta language layer with complexities for the compiler and programmer. Advanced generic code is close to unmaintainable.

From: Kevin Chadwick <kc-usenet@chadwicks.me.uk>
Date: Mon, 18 Dec 2023 12:52:08 -0000

Ada's ranged type system coupled with its excellent record overlays make Ada a much better choice than Rust for safe and easy hardware register and network protocols.

I can't imagine that ownership for data structures on a single runtime is the reason (before SPARK got support).

A lot of Ada code will not run on any runtime but Rust also has nostd for embedded use.

They often say Rust has excellent C interfacing support but Ada's appears to be even better.

Perhaps it is ease of use and guide availability which has improved or simply perception and a lack of knowledge about Ada.

Rather than language merits, a lot of people only care about job availability, money and library availability today but might consider a risk if they perceive a future demand increase.

As to why the likes of Google and Microsoft are putting money behind it when Ada would have been a better investment. You would have to ask them. Tell me why Google continues to write security sensitive code like matter in C++? When it could be written in Ada or Rust with a C binding.

Even Javascript engines like Mozilla's spidermonkey still has so little Rust code. Though Mozilla does have more financial concerns and its competitors are already trying to say their browsers are faster. Yet Google's websites use umpteen domains slowing browsing down anyway.

Ada Practice

Get_immediate Echo Character

From: Richardthiebaud <thiebauddick2@aol.com>
Subject: get_immediate echoe character--compiled error?
Date: Sun, 1 Oct 2023 22:42:39 -0400
Newsgroups: comp.lang.ada

When I build and run the following program using GNAT 11 in Linux Mint 21.2, the keys I press are echoed on the console. According to the Ada Reference Manual, they should not be echoed. Is this a compiler error?

```
with ada.text_io; use ada.text_io;
procedure test3 is
   c: character;
   avail: boolean;
begin
   loop
      loop
         Get_Immediate(c, Avail);
         if Avail then
           exit;
         end if;
         delay 0.01;
      end loop;
   end loop;
end test3;
```

This does not happen if I call get_immediate without the avail parameter, i.e. get_immediate(c);

From: Keith Thompson <keith.s.thompson+u@gmail.com>
Date: Sun, 01 Oct 2023 22:48:36 -0700

> According to the Ada Reference Manual, they should not be echoed. Is this a compiler error?

Where does the ARM say that?

From: Richardthiebaud <thiebauddick2@aol.com>
Date: Mon, 2 Oct 2023 16:07:49 -0400

> Where does the ARM say that?

https://www.adaic.org/resources/add_content/standards/05rm/html/RM-A-10-7.html

From: Keith Thompson <keith.s.thompson+u@gmail.com>
Date: Mon, 02 Oct 2023 15:27:33 -0700

> https://www.adaic.org/resources/add_content/standards/05rm/html/RM-A-10-7.html

I don't see anything there about the character being echoed, or not.

> If a character, either control or graphic, is available from the specified File or the default input file, then the character is read; Available is True and Item contains the value of this character. If a character is not available, then Available is False and the value of Item is not specified. Mode_Error is propagated if the mode of the file is not In_File. End_Error is propagated if at the end of the file. The current column, line and page numbers for the file are not affected.

Are you assuming that not updating the current column, line, and page numbers for the file implies that the character is not echoed?

[...]

From: Richardthiebaud <thiebauddick2@aol.com>
Date: Mon, 2 Oct 2023 18:41:46 -0400

> Are you assuming that not updating the current column, line, and page numbers for the file implies that the character is not echoed?

Yes.

In any case, when it echoes the character, it increases the current column by 1, and that does contradict the Ada Reference Manual.

From: Niklas Holsti <niklas.holsti@tidorum.invalid>
Date: Tue, 3 Oct 2023 11:41:05 +0300

>> I don't see anything there about the character being echoed, or not.

Nor do I. But perhaps there should be something, since "not echoing" is useful behavior and the program can itself echo characters if that is desired.

Possibly this is why AdaCore have given different echoing behaviors to the two forms of Get_Immediate, with and without the "Available" parameter. If so, this echo difference is unfortunately coupled with the wait/no-wait behavior difference, and that coupling may be unwanted.

There are (or have been) computer terminals with local echo, where the program cannot prevent the display of each keystroke. So the "no echo" behavior cannot be an absolute requirement in the Ada manual, but it could be Implementation Advice.

> In any case, when it echos the character. it increases the current column by 1, and that does contradict the Ada Reference Manual.

You are assuming that "current column" in the Ada Reference Manual means the same as the "current column position of the terminal/screen cursor", which is not the case, so there is no formal contradiction. The Ada "current column" refers to an internal state of the file.

For an unknown type of terminal/screen, deducing the current cursor column from the stream of input characters and output characters is not feasible, because it depends on the device's interpretation of formatting control characters such as TABs and on the width of the screen or terminal window.

From: Simon Wright <simon@pushface.org>
Date: Tue, 03 Oct 2023 11:20:40 +0100

> Possibly this is why AdaCore have given different echoing behaviors [...]

The low-level Get_Immediate implementation is in sysdep.c (probably not in the adainclude/ directory in an installed compiler), in getc_immediate() and getc_immediate_nowait(), both of which call getc_immediate_common(), and I can't see any difference! ECHO gets turned off in getc_immediate_common(), regardless of caller - see link. https://github.com/gcc-mirror/gcc/blob/3ca09d684e496240a87c0327687e2898060c2363/gcc/ada/sysdep.c#L387

From: G.B.
<bauhaus@notmyhomepage.invalid>
Date: Tue, 3 Oct 2023 23:00:40 +0200

> When I build and run the following program using GNAT 11 in Linux Mint 21.2, the keys I press are echoed on the console.

Which console?

Can you try to run a C program in the same console that tests for it to be a TTY? See Simon Wright's link to GNAT's implementation. The C program would be calling isatty(0) or isatty(fileno(your_stream));

Some IDEs have a console window that is not a TTY in the sense of termios(4)/tcsetattr(3). Echoing is different, then.

From: Keith Thompson
<keith.s.thompson+u@gmail.com>
Date: Tue, 03 Oct 2023 17:13:17 -0700

> https://github.com/gcc-mirror/gcc/blob/3ca09d684e496240a87c0327687e2898060c2363/gcc/ada/sysdep.c#L387

I haven't really looked into this, but I *think* what's happening is that for the versions with the Available parameter, ECHO hasn't yet been turned off when the user types the character. If you type 'x', it echoes immediately, because the program has no way of knowing that the character will later be consumed by a call to Get_Immediate. Presumably if the user hasn't typed anything, causing Available to be set to false, Get_Immediate will turn echoing off and back on again very quickly. Echoing is disabled only for small fraction of a second it takes for Get_Immediate to be executed.

The Get_Immediate functions without the Available parameter block until a character is entered. They can disable echoing before the character is entered. Echoing will typically be disabled for minutes or seconds, from the time Get_Immediate is called and the time the user types something.

The only solution I can think of would be to disable echoing (in some non-portable manner; I don't think the standard library provides this) before the user starts typing. (Perhaps you want to run the Get_Immediate without the Available parameter in a separate task?)

From: Simon Wright
<simon@pushface.org>
Date: Wed, 04 Oct 2023 09:22:05 +0100

> I *think* what's happening is [...]

Great analysis! Is this worth raising a PR on GCC Bugzilla? (maybe only on the documentation?)

Or, alternatively, don't turn echoing off at all - what's the use case for turning it off? After all, the ARM says nothing about it.

From: Jeffrey R.Carter
<spam.jrcarter.not@spam.acm.org.not>
Date: Wed, 4 Oct 2023 12:48:41 +0200

The use case is inputting passwords and the like. See Password_Line (https://github.com/jrcarter/Encryption-utilities/blob/master/password_line.ads) for an example. Note that this has identical behavior with GNAT/Linux and ObjectAda/Windows.

From: Simon Wright
<simon@pushface.org>
Date: Wed, 04 Oct 2023 12:38:51 +0100

Obviously, you need to turn echoing off for password input. But neither the ARM nor the GNAT RM says anything about Get_Immediate's echoing behaviour, so it's hard to explain why OA does the same thing. Does its manual specify this behaviour?

From: Jeffrey R.Carter
<spam.jrcarter.not@spam.acm.org.not>
Date: Wed, 4 Oct 2023 15:05:03 +0200

Unfortunately, Ada does not provide a standard way to turn off echo.

I agree that the ARM says nothing about echo for any of its operations on Standard_Input, but clearly there is a broad consensus of Ada.Text_IO writers and users who think this is desirable behavior.

From: Niklas Holsti
<niklas.holsti@tidorum.invalid>
Date: Wed, 4 Oct 2023 19:55:51 +0300

> Great analysis!

Yes indeed.

A possible solution in Text_IO would be for Get_Immediate with Available not to enable echo when it exits. Get_Immediate with Available is typically called repeatedly, with no other input from the terminal in between these calls, so it should be ok to keep echo disabled from one such call to another. Any non-immediate input operation on the terminal (that is, on this Text_IO file) should start by re-enabling echo if it was disabled. Possibly the same should apply also to Get_Immediate without Available, that is, it should leave echo disabled, until some non-immediate input operation re-enables echo.

From: Keith Thompson
<keith.s.thompson+u@gmail.com>
Date: Wed, 04 Oct 2023 12:39:27 -0700

> A possible solution in Text_IO would be for Get_Immediate with Available not to enable echo when it exits

The *first* character typed would still echo.

I suggest that what's needed is a way to turn echoing on and off.

Meanwhile, would calling Get_Immediate *without* the Available parameter (which blocks and turns echoing off until after a character is typed) in a separate task work I haven't tried it. Of course you'd need to be careful not to have I/O calls from separate tasks interfere with each other.

From: Niklas Holsti
<niklas.holsti@tidorum.invalid>
Date: Thu, 5 Oct 2023 00:20:05 +0300

> The *first* character typed would still echo.

Only if the user is quick enough to type it before the first call of Get_Immediate.

If Get_Immediate is called for example to enter a password, usually the program will first prompt the user to "Enter password:" and then at once call Get_Immediate. Only a user who starts typing before the prompt is visible would have time to type something before the (first) call of Get_Immediate.

> I suggest that what's needed is a way to turn echoing on and off.

The user could still be quick enough to type characters before the echo is turned off, so they would echo.

> Meanwhile, would calling Get_Immediate *without* the Available parameter (which blocks and turns echoing off until after a character is typed) in a separate task work? I haven't tried it.

That should work, provided that the Ada run-time system does not block the whole program when one task blocks on an I/O request. There have been, and perhaps still are, Ada programming systems where the whole Ada program appears to the OS as a single OS thread so that one Ada task waiting on a blocking OS call blocks all other tasks in the program.

> Of course you'd need to be careful not to have I/O calls from separate tasks interfere with each other.

Yes, but other tasks should be able to output text through Standard_Output even while one task is reading Standard_Input using a blocking I/O call. Except under a one-thread run-time system.

From: Randy Brukardt
<randy@rrsoftware.com>
Date: Wed, 4 Oct 2023 19:43:55 -0500

> I agree that the ARM says nothing about echo for any of its operations on Standard_Input, but clearly there is a broad consensus of Ada.Text_IO writers and users who think this is desirable behavior.

For what it's worth, Janus/Ada turns off echoing, and that was decided without referring to any other implementation's choice in the matter. Rather, it was done to provide a way using standard calls to provide functionality that had always been available in Janus/Ada in a non-standard way.

Specifically, Janus/Ada has always had a predefined file name "KBD:" (or "/dev/kbd" on Unix), which provides raw access to the keyboard device (or standard input on more modern systems). This did not echo (or do any line editing) on CP/M and MS-DOS, and we carried that same behavior over into more modern systems.

For instance, the "Continue or Abort?" question in the compiler uses KBD: to take and discard input immediately without any waiting (usual standard input is line buffered and usually input is not processed until "enter" or similar is pressed). It seemed to us that the Get_Immediate function was intending the same sorts of uses. Note that implementing it this way makes it hard to get meaningful results if Get_Immediate is mixed with other input on the same file. (That's why we treated it as a special file in the beginning, but even that gets confused if someone else reads from Standard_Input.)

From: Keith Thompson <keith.s.thompson+u@gmail.com>
Date: Wed, 04 Oct 2023 15:12:39 -0700

>> [Initial example removed. —arm]

> I should have checked this earlier, but this does not echo with ObjectAda.

On what target system?

From: Jeffrey R.Carter <spam.jrcarter.not@spam.acm.org.not>
Date: Thu, 5 Oct 2023 11:51:15 +0200

> On what target system?

Windows.

Using Log_Float in Inline Assembler for ARM

From: Ahlan Marriott <ahlan@marriott.org>
Subject: Using Log_Float in inline assembler for ARM
Date: Sun, 19 Nov 2023 04:22:20 -0800
Newsgroups: comp.lang.ada

The following procedure Unbiased_Rounding for Float works as expected.

```
function Unbiased_Rounding (X : Float)
return Float is
  Y : Float;
begin
   Asm ("vrintn.f32 %0,%1",
   Outputs  => Float'asm_output ("=t", Y),
   Inputs   => Float'asm_input ("t", X));
   return Y;
end Unbiased_Rounding;
```

According to https://gcc.gnu.org/onlinedocs/gcc/Machine-Constraints.html the constraint t means "VFP floating-point registers s0-s31. Used for 32 bit values" and the constraint w means "VFP floating-point registers d0-d31 and the appropriate subset d0-d15 based on command line options. Used for 64 bit values only"

Therefore, we wrote our long_float version as

```
function Unbiased_Rounding
(X : Long_Float) return Long_Float is
   Y : Long_Float;
begin
    Asm ("vrintn.f64 %0,%1",
    Outputs  => Long_Float'asm_output
     ("=w", Y), Inputs   =>
     Long_Float'asm_input ("w", X));
     return Y;
 end Unbiased_Rounding;
```

However, this fails to compile. GNAT 11.2/0-4 (Alire) complains

```
Error: invalid instruction shape
-- `vrintn.f64 s14,s14'
```

presumably because the operands are S registers rather than double precisions D registers. Is this a bug or have we misunderstood something?

From: Ahlan Marriott <ahlan@marriott.org>
Date: Fri, 24 Nov 2023 01:09:38 -0800

The solution is to use %P to access the parameters constrained using “w”. Try as I might I can’t find this wonderful secret documented anywhere.

I stumbled on the solution in the NXP forum where jingpan replied to a question on how to use the ARM VSQRT instruction for double. When using the inline assembler from C and using named parameters you need to access parameters constrained by “w”, i.e. D registers using %P[name] rather than %[name] as everywhere else.

Using positional parameters one needs to use %Pn rather than %n

And yes it must be a capital P.

I fail to understand why one needs to do this because surely the assembler already knows that the parameter has been constrained to a D register - but I guess this is just an additional quirk to an already very quirky assembler.

My GNAT Ada code to implement the Unbiased_Rounding attribute efficiently using the VFLOATN instruction is therefore

```
subtype T is Long_Float;
function Unbiased_Rounding (X : T) return
T is
   Y : T;
begin
   Asm (“vrintn.f64 %P0,%P1”,
   Outputs => T’asm_output (“=w”, Y),
   Inputs => T’asm_input (“w”, X));
return Y;
end Unbiased_Rounding;
```

Of course we wouldn’t have to resort to assembler at all had there been a built-in intrinsic for VFLOATN as there is for all the other VFLOAT instructions. But I guess that is hoping for too much.

GNAT.Source_Info Volatile and SPARK

From: Kevin Chadwick <kc-usenet@chadwicks.me.uk>
Subject: GNAT.Source_Info Volatile and SPARK
Date: Fri, 8 Dec 2023 18:28:24 -0000
Newsgroups: comp.lang.ada

I guess the SPARK annotations in GNAT.Source_Info mark them as Volatile_Functions for good reason.

I'm not sure how to handle using them in SPARK. They produce compile time known constants but I guess SPARK does not know e.g. the String length.

I use them in a logging function which is everywhere. So I get error “Volatile function call as actual is not allowed in SPARK” when calling GNAT.Source_Info.Source_Location as a logger’s parameter. Perhaps I should just avoid using them for SPARK compatibility? I can get by with GNAT.Source_Info.Line which only produces warnings and not the above error but it is not ideal.

I can use the function File as a package global constant. Any other ideas?

From: Kevin Chadwick <kc-usenet@chadwicks.me.uk>
Date: Sat, 9 Dec 2023 14:16:17 -0000

> I can use the function File as a package global constant. Any other ideas?

I shall go with doing the above per package for Gnat.Source_Info.File and wrapping the Gnat.Source_Info.Line procedure with one marked with Global => null.

Where would I suggest that Global => null be added to Line?

From: Kevin Chadwick <kc-usenet@chadwicks.me.uk>
Date: Sat, 9 Dec 2023 14:33:22 -0000

Doh… Of course I can't wrap Line, ha ha. If I want the right line.

From: Jeffrey R.Carter <spam.jrcarter.not@spam.acm.org.not>
Date: Sat, 9 Dec 2023 15:57:41 +0100

> Doh… Of course I can't wrap Line

Perhaps

```
private with Gnat.Source_Info;
package Source_Line_Info with
SPARK_Mode is
   function Line ... with Global => null;
private -- Source_Line_Info
   pragma SPARK_Mode (Off);
   function Line ... renames
     Gnat.Source_Info.line;
end Source_Line_Info;
```

(Untested)

From: Kevin Chadwick <kc-usenet@chadwicks.me.uk>
Date: Sat, 9 Dec 2023 15:13:28 -0000

> (Untested)

Interesting, Thanks. I might just use random identifiers. With the added benefit of knowing it will work with any runtime, any platform and any compilation options.

Beaujolais Challenge

From: jklsemicolon@f172.n1.z21.fsxnet (Jklsemicolon)
Subject: Beaujolais Challenge
Date: Sun, 10 Dec 2023 12:34:03 +1300
Newsgroups: comp.lang.ada

More than twenty years ago as a high schooler digging into the stacks at a community college library, I came across a book on Ada where a chapter epigraph referenced a bug bounty where the finder of some variety of bug in the Ada language specification would receive a case of wine. Does this ring any bells? I realize that this is quite vague, but I didn't have the CS background then to appreciate what I was reading, and events have taken me quite far from that shelf on that day.

... We all live in a yellow subroutine...

From: J-P. Rosen <rosen@adalog.fr>
Date: Sun, 10 Dec 2023 19:39:54 +0100

Sure. Ichbiah bet that the addition or removal of a use clause could cause compilation errors, but could not give a working program with a different meaning (a different resolution).

John Goodenough came up with such a case (a very contrived case, involving several levels of generics). I'm not sure that Ichbiah offered the bottle... The so-called beaujolais effect was fixed in Ada95.

From: Dirk Craeynest <dirk@orka.cs.kuleuven.be>
Date: Sun, 10 Dec 2023 19:27:26 -0000

>The so-called beaujolais effect was fixed in Ada95.

See also: https://en.wikipedia.org/wiki/Beaujolais_effect.

And the reference given on that wiki-page: https://web.archive.org/web/20060823054957/http://www.adaic.com/learn/oldfaqs.html#beaujolais

From: Randy Brukardt <randy@rrsoftware.com>
Date: Tue, 12 Dec 2023 03:23:41 -0600

>...The so-called beaujolais effect was fixed in Ada95.

It's still something that is talked about today when new Ada features are proposed; we don't want to reintroduce it, or the related "Ripple effect" (which is associated with "with" clauses, and is named for a cheap American wine brand circa 1980).

Map Iteration and Modification

From: Drpi <314@drpi.fr>
Subject: Map iteration and modification
Date: Thu, 28 Dec 2023 14:53:16 +0100
Newsgroups: comp.lang.ada

I need to delete nodes from a Hashed_Map. I don't know which nodes to delete in advance. I have to iterate on the Map keys and delete the nodes which fulfill a condition. From the LRM I understand I can't delete nodes within a loop iterating the Map nodes. That makes sense. What's the recommended way of doing this? Iterate the Map and temporarily store the key nodes to be deleted then delete the nodes from the key list?

From: Drpi <314@drpi.fr>
Date: Thu, 28 Dec 2023 14:59:07 +0100

> Iterate the Map and temporarily store the key nodes to be deleted then delete the nodes from the key list?

Not clear. Rephrasing it.

Using 2 steps by iterating the Map and temporarily store the keys of nodes to be deleted then delete the Map nodes using the key list?

From: Randy Brukardt <randy@rrsoftware.com>
Date: Thu, 28 Dec 2023 21:08:47 -0600

If the keys are messy to save (as say with type String), it might be easier to save the cursor(s) of the nodes to delete. You would probably want to use a cursor iterator (that is, "in") to get the cursors. Code would be something like (declarations of the Map and List not shown, nor is the function Need_to_Delete which is obviously application specific, Save_List is a list of cursors for My_Map, everything else is standard, not checked for syntax errors):

```
Save_List.Empty; -- Clear list of saved
                 -- cursors.
-- Find the nodes of My_Map
-- that we don't need.
for C in My_Map.Iterate loop
   if Need_to_Delete (My_Map.Element(C))
   then
      Save_List.Append (C);
      -- else no need to do anything.
   end if;
end loop;
-- Delete the cursors of the nodes we don't
-- want anymore.
for C of Save_List loop
   My_Map.Delete(C);
end loop;
```

From: Drpi <314@drpi.fr>
Date: Fri, 29 Dec 2023 14:53:53 +0100

That's what I did but I saved the keys (String) instead of the cursors.

Does it make a difference? Performance maybe?

From: Randy Brukardt <randy@rrsoftware.com>
Date: Sat, 30 Dec 2023 00:29:07 -0600

> That's what I did but I saved the keys (String) instead of the cursors. Does it make a difference? Performance maybe?

It certainly will make a performance difference; whether that difference is significant of course depends on the implementation. There's two parts to it (one of which I thought of yesterday and the other which I forgot):

(1) The cost of storing keys vs. storing cursors. Cursors are going to be implemented as small record types (canonically, they are two pointers, one to the enclosing container and one to the specific node/element). A key can be most anything, and storing that can be more costly.

(2) The cost of looking up a key. A map is a set of nodes, and there needs to be some operation to associate a key with the correct node. Those operations take some time, of course: for a hashed map, the key has to be hashed and then some sort of lookup performed. Whereas a cursor generally contains an indication of the node, so the access is more direct.

For a lot of applications, this difference won't matter enough to be significant. But I'd probably lean toward using cursors for this sort of job as that would minimize performance problems down the line. (Of course, if the container gets modified after you save the cursors, then they could become dangling, which is a problem of it's own. If that's a possibility, saving the keys is better.).

Conference Calendar

Dirk Craeynest
KU Leuven, Belgium. Email: Dirk.Craeynest@cs.kuleuven.be

This is a list of European and large, worldwide events that may be of interest to the Ada community. Further information on items marked ♦ is available in the Forthcoming Events section of the Journal. Items in larger font denote events with specific Ada focus. Items marked with ☺ denote events with close relation to Ada.

The information in this section is extracted from the on-line *Conferences and events for the international Ada community* at http://www.cs.kuleuven.be/~dirk/ada-belgium/events/list.html on the Ada-Belgium Web site. These pages contain full announcements, calls for papers, calls for participation, programs, URLs, etc. and are updated regularly.

2024

January 15-16 — 25th **International Conference on Verification, Model Checking, and Abstract Interpretation** (VMCAI'2024), London, UK. Co-located with POPL'2024. Topics include: program verification, model checking, abstract interpretation, static analysis, type systems, program certification, detection of bugs and security vulnerabilities, hybrid and cyber-physical systems, concurrent and distributed systems, analysis of numerical properties, analysis of smart contracts, etc., case studies on all of the above topics.

☺ January 17-19 — 51st ACM SIGPLAN **Symposium on Principles of Programming Languages** (POPL'2024), London, UK. Topics include: fundamental principles and important innovations in the design, definition, analysis, transformation, implementation, and verification of programming languages, programming systems, and programming abstractions.

- Jan 15-16 — **International Conference on Certified Programs and Proofs** (CPP'2024). Topics include: new languages and tools for certified programming; program analysis, program verification, and program synthesis; program logics, type systems, and semantics for certified code; verification of correctness and security properties; etc.

January 17-19 — 19th **International Conference on High Performance and Embedded Architecture and Compilation** (HiPEAC'2024), Munich, Germany. Topics include: computer architecture, programming models, compilers and operating systems for general-purpose, embedded and cyber-physical systems. Areas include safety-critical dependencies, cybersecurity, energy efficiency and machine learning.

- ☺ Jan 17-19 — 5th **Workshop on Next Generation Real-Time Embedded Systems** (NG-RES'2024). Topics include: application of formal methods to distributed and/or parallel real-time systems; programming models, paradigms and frameworks for real-time computation on parallel and heterogeneous architectures; compiler-assisted solutions for distributed and/or parallel real-time systems; scheduling and schedulability analysis for distributed and/or parallel real-time systems; etc.

February 07-09 — 18th **International Working Conference on Variability Modelling of Software-Intensive Systems** (VaMoS'2019), Bern, Switzerland. Topics include: variability across the software lifecycle, test and verification of variable systems, evolution of variability-intensive systems, runtime variability, variability mining, reverse-engineering of variability, economic aspects of variability, variability and quality requirements, industrial development of variable systems, experience reports from managing variability in practice, etc.

March 02-06 — IEEE/ACM **International Symposium on Code Generation and Optimization** (CGO'2024), Edinburgh, UK.

☺ March 11-14 — **International Conference on the Art, Science, and Engineering of Programming** (Programming'2024), Lund, Sweden. Deadline for submissions: January 25, 2024 (student research competition abstracts).

March 12-15 — 31st IEEE **International Conference on Software Analysis, Evolution, and Reengineering** (SANER'2024), Rovaniemi, Finland. Topics include: software tools for software evolution and maintenance; software analysis, parsing, and fact extraction; software reverse engineering and reengineering; program comprehension; software evolution analysis; software architecture recovery and

reverse architecting; program transformation and refactoring; mining software repositories and software analytics; software reconstruction and migration; software maintenance and evolution; program repair; software release engineering, continuous integration and delivery; education related to all of the above topics; etc. Deadline for submissions: January 14, 2024 (Journal First papers).

March 20-22 32nd **Euromicro International Conference on Parallel, Distributed and Network-Based Processing** (PDP'2024), Dublin, Ireland. Topics include: embedded parallel systems, dependability, survivability, programming languages, compilers, middleware, runtime, and systems software, performance prediction and analysis, simulation and modelling of parallel/distributed systems, etc.

April 06-11 27th **European Joint Conferences on Theory and Practice of Software** (ETAPS'2024), Luxembourg City, Luxembourg. Events include: ESOP (European Symposium on Programming), FASE (Fundamental Approaches to Software Engineering), FoSSaCS (Foundations of Software Science and Computation Structures), TACAS (Tools and Algorithms for the Construction and Analysis of Systems). Deadline for submissions: January 4, 2024 (ESOP, FASE, FoSSaCS artifact submissions), February 5, 2024 (Test of Time Award nominations).

April 10-11 30th **International Symposium on Model Checking of Software** (SPIN'2024 Topics include: automated tool-based techniques for the analysis of software as well as models of software, for the purpose of verification and validation. Deadline for submissions: January 22, 2024.

April 08-11 30th **International Working Conference on Requirements Engineering: Foundation for Software Quality** (REFSQ'2024), Winterthur, Switzerland. Theme: "Out of the Lab, into the Wild!" Deadline for submissions: February 9, 2024 (workshop submissions, education and training track, posters, tools, doctoral symposium).

April 08-12 36th ACM **Symposium on Applied Computing** (SAC'2024), Avila, Spain.

☺ Apr 08-12 **Track on Programming Languages** (PL'2024). Topics include: technical ideas and experiences relating to implementation and application of programming languages, such as compiling techniques, domain-specific languages, garbage collection, language design and implementation, languages for modeling, model-driven development, new programming language ideas and concepts, practical experiences with programming languages, program analysis and verification, etc. Deadline for submissions: October 13, 2019 (regular papers, SRC research abstracts).

Apr 08-12 **Software Verification and Testing Track** (SVT'2024). Topics include: new results in formal verification and testing, technologies to improve the usability of formal methods in software engineering, applications of mechanical verification to large scale software, model checking, correct by construction development, model-based testing, software testing, static and dynamic analysis, abstract interpretation, analysis methods for dependable systems, software certification and proof carrying code, fault diagnosis and debugging, verification and validation of large scale software systems, real world applications and case studies applying software testing and verification, etc.

Apr 08-12 19th **Track on Dependable, Adaptive, and Secure Distributed Systems** (DADS'2024). Topics include: Dependable, Adaptive, and secure Distributed Systems (DADS); modeling, design, and engineering of DADS; foundations and formal methods for DADS; applications of DADS; etc.

April 14-20 46th **International Conference on Software Engineering** (ICSE'2024), Lisbon, Portugal.

April 14-15 12th **International Conference on Formal Methods in Software Engineering** (FormaliSE'2024). Topics include: approaches, methods, and tools for verification and validation; formal approaches to safety and security related issues; scalability of formal method applications; integration of formal methods within the software development lifecycle; model-based engineering approaches; correctness-by-construction approaches for software and systems engineering; application of formal methods to specific domains (such as, autonomous, cyber-physical, intelligent, and IoT systems); formal methods for certification; guidelines to use formal methods in practice; usability of formal methods; etc.

April 24-25 16th **Software Quality Days** (SWQD'2024), Vienna, Austria. Theme: "Software Quality as a Foundation for Security". Topics include: all topics related to software and systems quality, such as methods and tools for constructive and analytical quality assurance; testing of software and software-intensive systems;

process improvement for development and testing; automation in quality assurance and testing; domain specific quality issues such as embedded, medical, automotive systems; continuous integration, deployment, and delivery; project and risk management; secure coding, software engineering and system design; detection and prevention of vulnerabilities and security threats; etc.

May 06-10 27th **Ibero-American Conference on Software Engineering** (CIbSE'2024), Curitiba, Paraná, Brazil. Topics include: software architecture and variability; software quality, quality models and technical debt management; software reliability; software ecosystems and systems of systems; software Engineering (SE) education and training; software evolution and modernisation; SE for emerging application domains (cyber-physical systems, Internet of Things, ...); industrial experience reports in SE; software product lines and processes; software repository mining and software analytics; software processes; software reuse; software testing; etc. Deadline for submissions: January 16, 2024 (abstracts), January 30, 2024 (papers, doctoral symposium, journal first).

May 07-11 15th ACM/SPEC **International Conference on Performance Engineering** (ICPE'2024), London, UK. Deadline for submissions: January 22 - February 2, 2024 (workshop papers), January 26, 2024 (emerging research track, posters, demos, tutorials), February 2, 2024 (artifacts), February 9, 2024 (data challenge).

May 13-16 17th **Cyber-Physical Systems and Internet of Things Week** (CPS-IoT Week'2024), Hong Kong. Event includes: 5 top conferences, HSCC, ICCPS, IoTDI, IPSN, and RTAS, as well as poster and demo sessions, workshops, tutorials, competitions, industrial exhibitions, and PhD forums.

☺ May 13-16 29th IEEE **Real-Time and Embedded Technology and Applications Symposium** (RTAS'2024). Topics include: time-sensitive applications; real-time and embedded operating systems; application profiling, WCET analysis, compilers, tools, benchmarks and case studies; modelling languages, modelling methods, model learning, model validation and calibration; scheduling and resource allocation; verification and validation methodologies; etc. Deadline for submissions: January 25, 2024 (brief presentations).

May 13-16 15th ACM/IEEE **International Conference on Cyber-Physical Systems** (ICCPS'2024). Topics include: safety and resilience for CPS; software platforms and systems for CPS; specification languages and requirements; design, optimization, and synthesis; testing, verification, certification; security, trust, and privacy in CPS; tools, testbeds, demonstrations and deployments; etc. Deadline for submissions: February 8, 2024 (posters, demos).

May 27-31 38th IEEE **International Parallel and Distributed Processing Symposium** (IPDPS'2024), San Francisco, California, USA. Topics include: applications to solve problems using parallel and distributed computing concepts; programming models, compilers, and runtime systems (ranging from the design of parallel programming models and paradigms to languages and compilers supporting these models and paradigms, to runtime and middleware solutions); system software; existing and emerging architectures; experiments and performance-oriented studies in the practice of parallel and distributed computing; etc. Deadline for submissions: January 15 - February 16, 2024 (workshop papers).

May 27-31 17th IEEE **International Conference on Software Testing, Verification and Validation** (ICST'2024), Toronto, Canada. Topics include: manual testing practices and techniques, security testing, model-based testing, test automation, static analysis and symbolic execution, formal verification and model checking, software reliability, testability and design, testing and development processes, testing in specific domains (such as embedded/cyber-physical systems, concurrent, distributed, ..., and real-time systems), testing/debugging tools, empirical studies, experience reports, etc. Deadline for submissions: January 29, 2024 (workshop papers).

June 04-06 16th **NASA Formal Methods Symposium** (NFM'2024), Moffett Field, California, USA. Topics include: identifying challenges and providing solutions towards achieving assurance for critical systems; formal techniques for software and system assurance for applications in space, aviation, robotics, and other NASA-relevant safety-critical systems.

June 10-12 21st **International Conference on Software and Systems Reuse** (ICSR'2024), Limassol, Cyprus. Theme: "Sustainable Software Reuse". Topics include: new and innovative research results and industrial experience reports dealing with all aspects of software reuse within the context of the modern software development landscape, such as technical aspects of reuse (model-driven development, variability management and software product lines, domain-specific languages, new language abstractions for software reuse, software composition and modularization, technical debt and software reuse, ...), software

reuse in industry and in emerging domains (reuse success stories, reuse failures and lessons learned, reuse obstacles and success factors, return on investment studies, ...). Deadline for submissions: February 12, 2024 (abstracts), February 19, 2024 (full papers).

☺ June 11-12 12th **European Congress on Embedded Real Time Systems** (ERTS'2024), Toulouse, France. Topics include: all aspects of critical embedded real-time systems, such as model-based system and safety engineering, product line engineering, programming languages, verification methods, software development frameworks, dependability, safety, cyber security, quality of service, fault tolerance, maintainability, certification, etc. Deadline for submissions: April 3, 2024 (regular papers), May 5, 2024 (final short and regular papers).

♦ June 11-14 28th **Ada-Europe International Conference on Reliable Software Technologies** (AEiC'2024), Barcelona, Spain. Organized by Ada-Europe and Barcelona Supercomputing Center (BSC), in cooperation with ACM SIGAda, ACM SIGBED, ACM SIGPLAN, and Ada Resource Association (ARA). Deadline for submissions: January 31, 2024 (journal track papers), February 26, 2024 (industrial track and work-in-progress papers, tutorial and workshop proposals). #AEiC2024 #AdaEurope #AdaProgramming

June 19-21 28th **International Conference on Engineering of Complex Computer Systems** (ICECCS'2024), Limassol, Cyprus. Topics include: all areas related to complex computer-based systems, including the causes of complexity and means of avoiding, controlling, or coping with complexity, such as model-driven development, security, reliability and dependability, safety-critical and fault-tolerant architectures, formal methods, verification and validation, reverse engineering and refactoring, software architecture, agile methods, cyber-physical systems and Internet of Things (IoT), industrial case studies, etc.

July 09-12 6th **Euromicro Conference on Real-Time Systems** (ECRTS'2024), Lille, France. Topics include: all aspects of timing requirements in computer systems; elements of time-sensitive software systems, such as operating systems, hypervisors, middlewares and frameworks, programming languages and compilers, runtime environments, ...; real-time applications topics, such as modeling, design, simulation, testing, debugging, and evaluation in domains such as automotive, avionics, control systems, industrial automation, robotics, space, railways telecommunications, multimedia, ...; foundational scheduling and predictability questions, such as schedulability analysis, synchronization protocols, ...; static and dynamic techniques for resource demand estimation, such as classic worst-case execution time (WCET) analysis, ...; formal methods for the verification and validation of real-time systems; the interplay of timing predictability and other non-functional qualities, such as reliability, security, quality of control, testability, scalability, ...; etc. Deadline for submissions: February 29, 2024.

July 15-19 32nd ACM **International Conference on the Foundations of Software Engineering** (FSE'2024), Porto de Galinhas, Brazil. Topics include: debugging and fault localization; dependability, safety, and reliability; embedded software, safety-critical systems, and cyber-physical systems; model checking; model-driven engineering; parallel, distributed, and concurrent systems; program analysis; programming languages; software architectures; software engineering education; software evolution; software security; software testing; software traceability; symbolic execution; tools and environments; etc.

☺ August 26-30 30th **International European Conference on Parallel and Distributed Computing** (Euro-Par'2024), Madrid, Spain. Topics include: all aspects of parallel and distributed processing, ranging from theory to practice, from small to the largest parallel and distributed systems and infrastructures, from fundamental computational problems to applications, from architecture, compiler, language and interface design and implementation, to tools, support infrastructures, and application performance aspects. Deadline for submissions: February 5, 2024 (workshops, minisymposia), March 5, 2024 (abstracts), March 15, 2024 (papers), May 17, 2024 (posters, demos, PhD symposium).

September 09-13 26th **International Symposium on Formal Methods** (FM'2024), Milan, Italy.

Sep 09-10 **18th International Conference on Tests And Proofs (TAP'2024).** Topics include: many aspects of verification technology, including foundational work, tool development, and empirical research; the combination of static techniques such as proving and dynamic techniques such as testing; verification and analysis techniques combining proofs and tests; static analysis of programs with the aid of dynamic techniques; deductive techniques supporting the automated generation of test vectors and oracles, and supporting (novel) definitions of coverage criteria; specification inference by deductive or dynamic methods; testing and runtime analysis of formal specifications; verification of verification tools and

environments; applications of test and proof techniques in new domains; combined approaches of test and proof in the context of formal certifications; case studies, tool and framework descriptions, and experience reports; etc. Deadline for submissions: May 8, 2024 (abstracts), May 15, 2024 (papers), July 3, 2024 (artifacts).

☺ September 16-20 38th **European Conference on Object-Oriented Programming** (ECOOP'2024), Vienna, Austria. Co-located with ISSTA'2024. Topics include: programming languages, software development, systems and applications. Deadline for submissions: January 17, 2024 (submissions round 1), January 23, 2024 (artifacts round 1), April 17, 2024 (submissions round 2), April 23, 2024 (artifacts round 2).

September 17-20 43rd **International Conference on Computer Safety, Reliability and Security** (SafeComp'2024), Florence, Italy. Topics include: all aspects related to the development, assessment, operation, and maintenance of safety-related and safety-critical computer systems; safety guidelines and standards; safety/security co-engineering and tradeoffs; safety and security qualification, quantification, assurance and certification; model-based analysis, design, and assessment; formal methods for verification, validation, and fault tolerance; testing, verification, and validation methodologies and tools; etc. Domains of application include: railways, automotive, space, avionics & process industries; highly automated and autonomous systems; telecommunication and networks; critical infrastructures; medical devices and healthcare; surveillance, defense, emergency & rescue; logistics, industrial automation, off-shore technology; education & training; etc. Deadline for submissions: February 4, 2024 (workshops, abstracts), February 11, 2024 (full papers).

☺ October 20-25 ACM **Conference on Systems, Programming, Languages, and Applications: Software for Humanity** (SPLASH'2024), Pasadena, California, USA.

☺ Oct 20-25 **Conference on Object-Oriented Programming, Systems, Languages, and Applications** (OOPSLA'2024). Topics include: all practical and theoretical investigations of programming languages, systems and environments, targeting any stage of software development, including requirements, modelling, prototyping, design, implementation, generation, analysis, verification, testing, evaluation, maintenance, and reuse of software systems; development of new tools, techniques, principles, and evaluations. Deadline for submissions: April 5, 2024 (round 2).

December 10 Birthday of Lady Ada Lovelace, born in 1815. Happy Programmers' Day!

Join Ada-Europe!

Become a member of Ada-Europe and **support Ada-related activities** and the future **development of the Ada programming language**.

Membership benefits include **receiving the quarterly Ada User Journal** and a substantial **discount when registering for the annual Ada-Europe conference**.

To apply for membership, visit our web page at

http://www.ada-europe.org/**join**

28th Ada-Europe International Conference on Reliable Software Technologies

11-14 June 2024, Barcelona, Spain

BARCELONA
AEiC 2024

Conference Chair
Sara Royuela
sara.royuela@bsc.es
Barcelona Supercomputing Center, Spain

Journal track Chairs
Bjorn Andersson
baandersson@sei.cmu.edu
Carnegie Mellon University, USA

Luis Miguel Pinho
lmp@isep.ipp.pt
ISEP & INESC TEC, Portugal

Industrial track Chairs
Luciana Provenzano
luciana.provenzano@mdu.se
Mälardalen University, Sweden

Michael Pressler
Michael.Pressler@de.bosch.com
Robert Bosch GmbH, Germany

Work-In-Progress track Chairs
Alejandro R. Mosteo
amosteo@unizar.es
CUD Zaragoza, Spain

Ruben Martins
rubenm@andrew.cmu.edu
Carnegie Mellon University, USA

Tutorial Chair
Maria A. Serrano
maria.serrano@nearbycomputing.com
NearbyComputing, Spain

Workshop Chair
Sergio Saez
ssaez@disca.upv.es
Universitat Politècnica de València, Spain

Exhibition & Sponsorship Chair
Ahlan Marriott
ahlan@Ada-Switzerland.ch
White Elephant GmbH, Switzerland

Publicity Chair
Dirk Craeynest
Dirk.Craeynest@cs.kuleuven.be
Ada-Belgium & KU Leuven, Belgium

Webmaster
Hai Nam Tran
hai-nam.tran@univ-brest.fr
University of Brest, France

Local Chair
Nuria Sirvent
nuria.sirvent@bsc.es
Barcelona Supercomputing Center, Spain

The 28th Ada-Europe International Conference on Reliable Software Technologies (AEiC 2024) will take place in Barcelona, Spain.

AEiC is a leading international forum for providers, practitioners, and researchers in reliable software technologies. The conference presentations will illustrate current work in the theory and practice of the design, development, and maintenance of long-lived, high-quality software systems for a challenging variety of application domains. The program will also include keynotes, Q&A and discussion sessions, and social events. Participants include practitioners and researchers from industry, academia, and government organizations active in the development of reliable software technologies.

The topics of interest for the conference include but are not limited to (more specific topics are described on the conference web page):

- *Formal and model-based engineering of critical systems*
- *High-Integrity Systems and Reliability*
- *AI for High-Integrity Systems Engineering*
- *Real-Time Systems*
- *Ada Language*
- *Applications in relevant domains*

The conference comprises different tracks and co-located events:

- *Journal track* papers present research advances supported by solid theoretical foundation and thorough evaluation.
- *Industrial track* contributions highlight industrial open challenges and/or the practitioners' side of a relevant case study or industrial project.
- *Work-in-progress track* papers illustrate novel research ideas that are still at an initial stage, between conception and first prototype.
- *Tutorials* guide attenders through a hands-on familiarization with innovative developments or with useful features related to reliable software.
- *Workshops* provide discussion forums on themes related to the conference topics.
- *Vendor presentations and exhibitions* allow for companies to showcase their latest products and services.

Important Dates

31 January 2024	**Deadline for submission of journal track papers (extended)**
26 February 2024	Deadline for submission of industrial track papers, work-in-progress papers, tutorial and workshop proposals
22 March 2024	First round notification for journal track papers, and notification of acceptance for all other types of submissions
11-14 June 2024	Conference

Call for journal track submissions

Following a journal-first model, this edition of the conference includes a journal track, which seeks original and high-quality papers that describe mature research work on the conference topics. Accepted journal track papers will be published in a Special Issue of Elsevier JSA – the *Journal of Systems Architecture* (Q1 ranked, CiteScore 8.5, impact factor 4.5). Accordingly, the conference is listed as "Journal Published" in the latest update of the CORE Conference Ranking released in August 2023. Contributions must be submitted by **31 January 2024**. Submissions should be made online at https://www.editorialmanager.com/jsa/, selecting the "Ada-Europe AEiC 2024" option (submission page open from 15 November 2023) as article type of the paper. General information for submitting to the JSA can be found at the Journal of Systems Architecture website.

JSA has adopted the Virtual Special Issue model to speed up the publication process, where Special Issue papers are published in regular issues, but marked as SI papers. Acceptance decisions are made on a rolling basis. Therefore, authors are encouraged to submit papers early, and need not wait until the submission deadline. Authors who have successfully passed the first round of review will be invited to present their work at the conference. The abstract of the accepted contributions will be included in the conference booklet.

The Ada-Europe organization will waive the Open Access fees for the first four accepted papers (whose authors do not already enjoy Open Access agreements). Subsequent papers will follow JSA regular publishing track. Prospective authors may direct all enquiries regarding this track to the corresponding chairs, Bjorn Andersson (baandersson@sei.cmu.edu) and Luis Miguel Pinho (lmp@isep.ipp.pt).

Call for industrial track submissions

The conference seeks industrial practitioner presentations that deliver insight on the challenges of developing reliable software. Especially welcome kinds of submissions are listed on the conference web site. Given their applied nature, such contributions will be subject to a dedicated practitioner-peer review process. Interested authors shall submit a 1-to-2 pages abstract, by **26 February 2024**, via EasyChair at https://easychair.org/my/conference?conf=aeic2024, selecting the "Industrial Track". The format for submission is strictly in PDF, following the Ada User Journal style. Templates are available at http://www.ada-europe.org/auj/guide.

The abstract of the accepted contributions will be included in the conference booklet. The corresponding authors will get a presentation slot in the prime-time technical program of the conference and will also be invited to expand their contributions into full-fledged articles for publication in the *Ada User Journal*, which will form the proceedings of the industrial track of the Conference. Prospective authors may direct all enquiries regarding this track to its chairs Luciana Provenzano (luciana.provenzano@mdu.se) and Michael Pressler (Michael.Pressler@de.bosch.com).

Call for work-in-progress track submissions

The work-in-progress track seeks two kinds of submissions: (a) ongoing research and (b) early-stage ideas. Ongoing research submissions are 4-page papers describing research results that are not mature enough to be submitted to the journal track. Early-stage ideas are 1-page papers that pitch new research directions that fall within the scope of the conference. Both kinds of submissions must be original and shall undergo anonymous peer review. Submissions by recent MSc graduates and PhD students are especially sought. Authors shall submit their work by **26 February 2024**, via EasyChair at https://easychair.org/my/conference?conf=aeic2024, selecting the "Work-in-Progress Track". The format for submission is strictly in PDF, following the Ada User Journal style. Templates are available at http://www.ada-europe.org/auj/guide.

The abstract of the accepted contributions will be included in the conference booklet. The corresponding authors will get a presentation slot in the prime-time technical program of the conference and will also be offered the opportunity to expand their contributions into 4-page articles for publication in the *Ada User Journal*, which will form the proceedings of the WiP track of the Conference. Prospective authors may direct all enquiries regarding this track to the corresponding chairs Alejandro R. Mosteo (amosteo@unizar.es) and Ruben Martins (rubenm@andrew.cmu.edu).

Call for tutorials

The conference seeks tutorials in the form of educational seminars on themes falling within the conference scope, with an academic for practitioner slant, including hands-on or practical elements. Tutorial proposals shall include a title, an abstract, a description of the topic, an outline of the presentation, the proposed duration (half-day or full-day), the intended level of the contents (introductory, intermediate, or advanced), and a statement motivating attendance. Tutorial proposals shall be submitted at any time but no later than the **26 February 2024** to the respective chair Maria A. Serrano (maria.serrano@nearbycomputing.com), with subject line: "[AEiC 2024: tutorial proposal]". Once submitted, each tutorial proposal will be evaluated by the conference organizers as soon as possible, with decisions from January 1st. The authors of accepted full-day tutorials will receive a complimentary conference registration, halved for half-day tutorials. The *Ada User Journal* will offer space for the publication of summaries of the accepted tutorials.

Call for workshops

The conference welcomes satellite workshops centred on themes that fall within the conference scope. Proposals may be submitted for half- or full-day events, to be scheduled at either end of the AEiC conference. Workshop organizers shall also commit to producing the proceedings of the event, for publication in the *Ada User Journal*. Workshop proposals shall be submitted at any time but no later than the **26 February 2024** to the respective chair Sergio Saez (ssaez@disca.upv.es), with subject line: "[AEiC 2024: workshop proposal]". Once submit ted, each workshop proposal will be evaluated by the conference organizers as soon as possible, with decisions from January 1st.

Call for exhibitors and sponsors

The conference will include a vendor and technology exhibition with the option of a 20 minutes presentation as part of the conference program. Interested providers should direct inquiries to the Exhibition & Sponsorship Chair Ahlan Marriot (ahlan@ada-switzerland.ch).

Venue

The conference will take place in Barcelona, Spain. Barcelona is a major cultural, economic, and financial centre, known for its architecture, culture, and Mediterranean atmosphere, a hub for technology and innovation. There's plenty to see and visit in Barcelona, so plan in advance!

Organized by:

In cooperation with:

The MORPHEMIC Project on the Data Intensive E-Brain Science Case Study

Amina Moussaoui, Alessandra Bagnato

Softeam, Softeam Software, Docaposte Groupe, 3 avenue du Centre 78280 Guyancourt.;Tel: +33 1 30 12 18 58; e-mail: amina.moussaoui@softeam.fr, alessandra.bagnato@softeam.fr

Abstract

The MORPHEMIC project covers several features from modelling cross-cloud applications, continuous and autonomous optimization and deployment and providing access to several cloud capabilities for data intensive application needing on one hand more and more resources, and, on the other hand, to support different models, such as Edge and Fog, besides the Cloud.

In this paper we present the MORPHEMIC H2020 project and its way of adapting and optimising Cloud computing applications through one of the project case studies, the E-Brain Science case study, proposed by Centre Hospitalier Universitaire Vaudois (CHUV) giving details on the application and its modelling, deployment, and optimization through MORPHEMIC.

CHUV is one of the five Swiss University hospitals. Specifically, the use case can be applied to the work of the Laboratoire de recherche en neuroimagerie (LREN), which consists of a cross-disciplinary team of basic and clinical neuroscientists with an interest in the role of human brain structure and function in neurological disorders and healthy ageing. The aim of the E-Brain Science application within MORPHEMIC is to bring the pipeline application to the multi-cloud environment and to deploy it in the most optimal way, considering deployment time and cost.

Keywords: Cloud services, Cloud computing, model-driven engineering, multi-cloud platform.

1 Introduction

Funded by the European Union Horizon 2020 research and innovation programme, the project MORPHEMIC [1] began in January 2020 for a period of 36 months then extended to 42 months. The project involves 12 partners from 7 countries belonging to both academia and industry.

On the academic side, it includes FORTH[1], UiO[2], UPRC[3], ICCS[4].

From the industrial side, it includes Softeam[5], ICON[6], IS-Wireless[7], CHUV[8], InAccel[9], Activeon[10], 7bulls[11], Engineering[12]. The project is an extension of MELODIC multi-cloud platform developed in the H2020 project[2] and it introduces two novel concepts to it: "polymorphing architecture" that will allow for dynamic adaptation of the architecture of application to the current workload and "proactive adaptation «that will allow the reconfiguring of the application based forecasted metrics about usage and workload level[11] .

The MORPHEMIC open-source platform [8] provides to its users the optimization of the deployment and life-cycle management of data-intensive applications. Within this paper we will provide a step-by-step guide throughout the Centre Hospitalier Universitaire Vaudois (CHUV) use-case illustrating how to provide the necessary settings, how to deploy the application and more about monitoring and proactive adaptation.

The paper is structured as follows, we firstly describe the CHUV use-case application: 'E-Brain Science'. Then we describe the requirements to model an application: by explaining CAMEL Model Structure and the CAMEL 3.0 novelties, describing CAMEL Designer environment features throughout the use-case application. The main models of the application are described: Deployment model, Requirement model, Metric model. After that, we give an idea about how to make the configuration, deployment and monitoring of the application throughout MORPHEMIC GUI [5].

2 The CHUV Use case

The Centre Hospitalier Universitaire Vaudois (CHUV): Lausanne university hospital is one of the five Swiss university hospitals. Through its collaboration with the Faculty of Biology and Medicine of the University of Lausanne, CHUV plays a key role in the areas of medical care, bio-Prototype medical research and education [10].

The aim of the E-Brain Science application within MORPHEMIC is to bring the pipeline application to the

[1] https://www.ics.forth.gr/
[2] https:///www.uio.no/
[3] https://www.unipi.gr/unipi/en/
[4] https://www.iccs.gr/
[5] https://www.softeamgroup.fr
[6] https://www.iconcfd.com
[7] https://www.is-wireless.com/
[8] https://www.chuv.ch/fr
[9] htttps://www.inaccel.com
[10] https://www.activeeon.com/fr
[11] https://www.7bulls.com/en
[12] https://www.eng.it/en/

multi-cloud environment and to deploy it in the most optimal way, considering deployment time and cost.

The following elements in Figure 1 are describing the architecture and hardware requirements of the application, that consist of two main components:

- The scheduler where the workflow will besubmitted and orchestrated.
- The worker where the tasks will be executed.

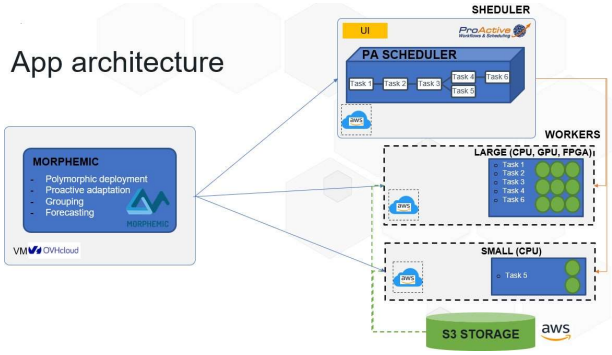


Figure 1: CHUV application architecture

The following elements are appearing in Figure 2 and are describing image pre-processing pipeline of the application:

- image format conversion,
- quantitative maps calculation,
- artefact removal,
- feature extraction, such as segmentations, brain parcellation.

3 CAMEL 3.0 Modelling language

The deployment and polymorphic adaptation of cross-cloud applications in the MORPHEMIC project is based on Cloud Application Modelling and Execution Language (CAMEL). CAMEL is a multi-domain-specific language (DSL) [9] allowing users to specify multiple aspects/domains related to multi-/cross-cloud applications, such as the domains of deployment, requirement, metric, scalability, security, organisation, and execution.

Furthermore, CAMEL has been extended within the MORPHEMIC project to cover the polymorphic modelling concepts that allow applications to have several possible deployment configurations [6].

The following elements (highlighted in green.) in Figure 3 are part of the new enhancements in the deployment model:

- Control flow relations: Precedes, Parallel, Sequence, Switch and Conditional.
- Communication Requirement.
- Configurations: Image Configuration, Container Configuration.

In order to cover the modelling of polymorphic applications, CAMEL was extended to draft version 3.0 (Deliverable D1.1 [7]). This change was driven by various requirements, drawn from the MORPHEMIC project.

An important domain in CAMEL is the Requirement domain as it enables to specify all application and component requirements that need to be met for a specific application by its management platform. Requirements can be hard or soft. Hard requirements need to be met at any cost by the application management platform. On the other hand, the platform will attempt to satisfy soft requirements on a best-effort basis. The following Figure 4 shows the requirement meta-model with the new enhancement of CAMEL 3.0 which is the LinkRequirement.

Similarly, the meta-data schema (MDS), a conceptual model for the Cloud and big data domains, was also similarly extended with the capability mainly to cover concepts and relations for various kinds of resources, including hardware-accelerated ones like field-programmable gate arrays (FPGAs) as well as network elements. Persistent applications will in fact experience variable demands and workloads, and Cloud resources may mitigate performance problems by allowing the necessary resources to be temporarily rented as needed. Cloud deployment decisions must consider not only the best-suited providers but also the possibility of beneficially using hardware accelerators like GPUs and FPGAs should these be useful for the application

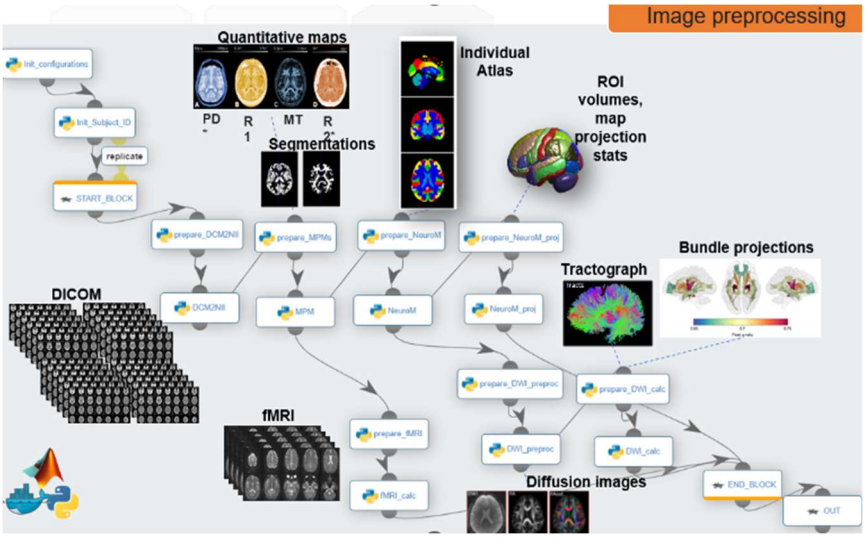


Figure 2: CHUV image pre-processing pipeline

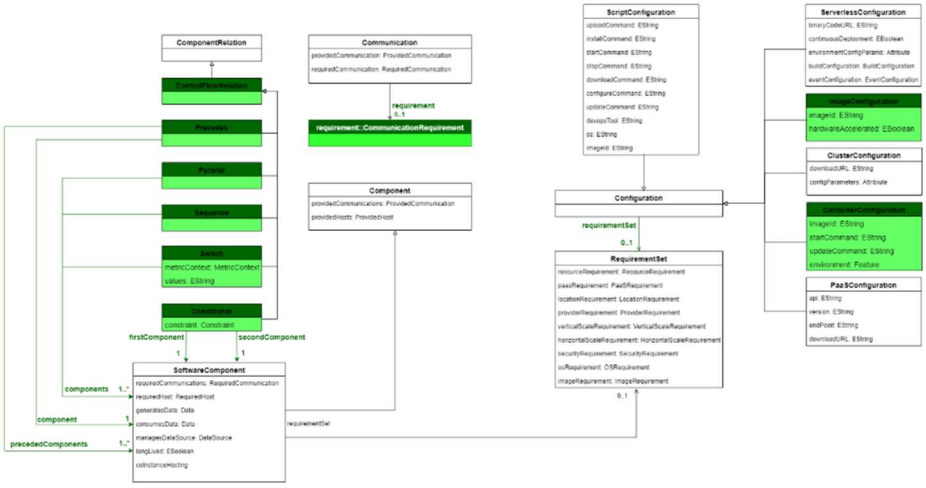


Figure 3: CAMEL 3.0 Deployment meta-model enhancements

and improve its performance. MDS as a formal vocabulary complements CAMEL in order to enable the enhancement of CAMEL models via feature hierarchies decorated via MDS elements. In this respect, as it now includes the modelling of hardware-accelerated resources like field-programmable gate arrays (FPGAs), this enables to re-use such elements in the modelling of multi-cloud, polymorphic applications, e.g., to specify requirements over their characteristics. The modelling-related work done is both provider and platform agnostic, making it suitable for use by any kind of (multi-)cloud (application) management platform.

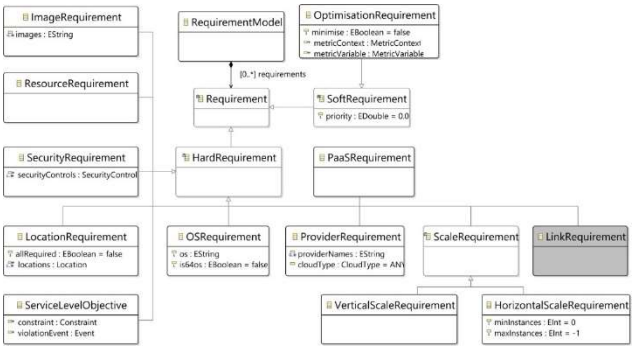


Figure 4: CAMEL 3.0 Requirement meta-model enhancements

4 Camel Designer: A Modelio environment for modelling The CHUV Case Study

The CAMEL Modelling task is performed by CAMEL Designer [3] tool which is a module for Modelio Open-Source modelling tool [4]. Modelio is an extensible tool and modules are the means of defining, implementing, and deploying extensions for Modelio.

The CAMEL model allows describing the requirement model of the CHUV use case. The following elements are appearing in Figure 5 and are describing the specific requirement model:

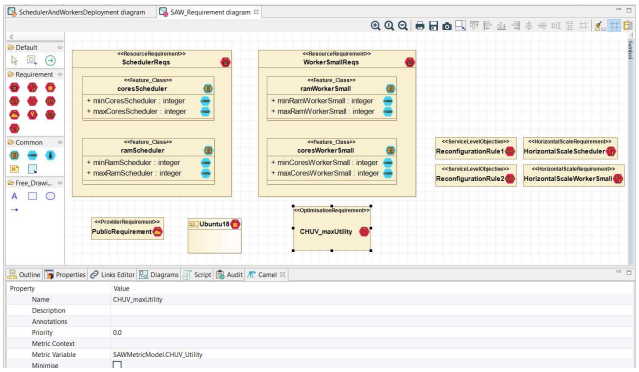


Figure 5: CHUV Requirement model

- Resource requirement: The scheduler requirements specify that the CPU should have at least 2 cores and at most 4 cores.
- Provider requirement: The Public requirement specifies that the cloud type should be public.
- Optimisation requirement: The CHUV_maxUtility requirement defines the CHUV-Utility variable of the Metric Model being the one that should be optimised in the deployment process.

The case study focuses on the MORPHEMIC Seamless Cloud Acceleration feature, which allows an easy deployment and utilisation of hardware accelerators (i.e., FPGAs) in the Cloud. The following FPGA resource requirements described in CHUV use case are appearing in Figure 6.

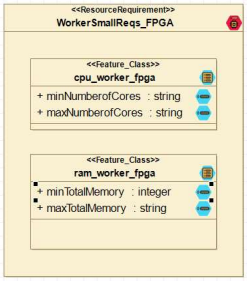


Figure 6: FPGA resource requirement

The following elements representing the metric domain of the application are modelled in the Figure 7:

- Measurable attributes,
- Metric templates,
- Sensors,
- Raw metric context,
- Object context,
- Composite metric context,
- Metric variables.

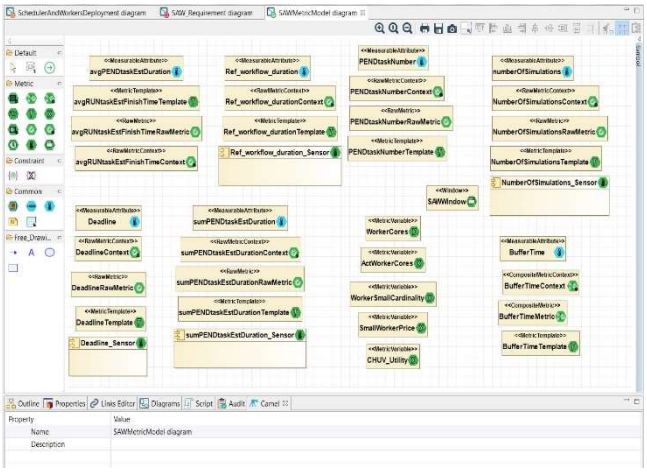


Figure 7: CHUV Metric model

5 Deployment on the platform

The MORPHEMIC Platform allows the user to choose the technical form of deployment during the optimization process to fulfil the user's requirements and needs. The quality of the deployment is measured by a user-defined and application-specific utility. Depending on the application's requirements and its current workload, its components could be deployed in various forms in different environments to maximise the utility of the application deployment and the satisfaction of the user.

Once the E-Brain Science application 's modelled,the model is imported on the platform. The configuration of the provided settings as the cloud providers (public /private) and their respective credentials is then mandatory for the deployment process. After that, the reasoning process takes place and the deployment view appears as shown in the following Figure 8.

In the process visualisation view, the user can go through all the reasoning steps after deploying the application.

The first step is Fetching Offers, the UI displays the current total number of offers previously selected providers. From these offers, MORPHEMIC will choose the best one for the application components.

The second step is Generating Constraint Problem. This generation is based on the requirements defined in the CAMEL model. The UI displays all variables from the constraint problem with their respective domain values. The utility formula is also displayed, it is used to measure the utility of each possible solution and use the best one.

MORPHEMIC can then make a decision about triggering the reconfiguration process which means creating new additional instances or deleting not fully used ones.

The platform also allows the user to find virtual machines and functions that are created by MORPHEMIC for each component. The polymorphic adaptation feature enables proposing the most suitable variant, hardware and resource boundaries and component grouping of running components, MORPHEMIC provides a visualisation tool showing for each application version a different configuration selected.

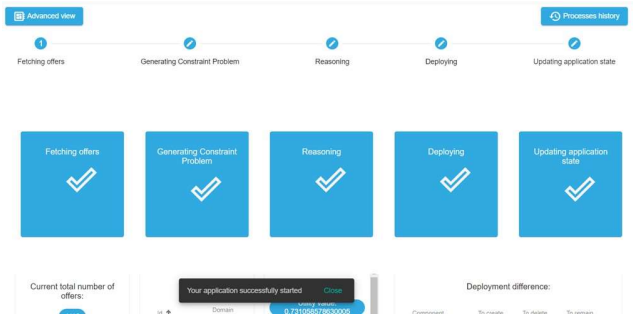


Figure 8: Application starting

Conclusions

Nowadays, Cloud application modelling languages do not supply the polymorphic application components and do not provide polymorphic infrastructure models. Therefore, MORPHEMIC goes beyond the state-of-the-art to introduce automatic DevOps capabilities for the efficient application life-cycle management in the dynamic Cloud computing continuum. Such capabilities are demonstrated in this paper through the CHUV use case.

References

[1] MORPHEMIC website: Available at: www.morphemic.cloud (Accessed: 22-04-2023)

[2] G. Horn and P. Skrzypek, "MELODIC: Utility Based Cross Cloud Deployment Optimisation", *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA),* Doi: 10.1109/WAINA.2018.00112, 2018.

[3] Modelio Camel Designer, Available at: github.com/Modelio-R-D/CamelDesigner (Accessed: 22-04-2023)

[4] Modelio website: Available at: www.modelio.org (Accessed: 22-04-2023)

[5] A.Moussaoui, A.Bagnato, E.Brosse J. Krasnodębska and P.Skrzypek, "The MORPHEMIC Project and its Unified User Interface", *Proc. of RCIS-WS&RP 2022 RCIS 2022 Workshops and Research Projects Track*, Barcelona, Spain, May 19, 2022, CEUR-WS.org, online:ceur-ws.org/Vol-3144/RP-paper13.pdf

[6] K. Kritikos et al., “D1.3 Final Data, Cloud Application & Resource Modelling,” MORPHEMIC Project Deliverable, Nov. 2022.

[7] K. Chaabouni et al., “D1.1 Data, Cloud Application & Resource Modelling,” MORPHEMIC Project Deliverable, Dec. 2020.

[8] CAMEL DSL Documentation, Available at: http://camel-dsl.org/documentation/ (Accessed: 22-04-2023)

[9] e-Brain Science Background and challenges, Available at: https://www.morphemic.cloud/e-brainscience/ (Accessed: 22-04-2023)

[10] A. Bagnato., J. Krasnodębska, “MORPHEMIC - Optimization of the deployment and life-cycle management of data-intensive applications in the Cloud computing continuum”, *Ada User Journal, vol. 43*, n. 4, pages 235-239, December 2022.

The ADMORPH Approach for Adaptively Morphing Embedded Systems

A. Pimentel, C. Grelck, L. Miedema, D. Sapra
Informatics Institute, University of Amsterdam; email: {a.d.pimentel, c.u.grelck, p.l.miedema, d.sapra}@uva.nl

M. Völp, F. Lucchetti, A. Matovic
SnT - Université du Luxembourg; email: {marcus.voelp, federico.lucchetti, aleksandar.matovic}@uni.lu

M. Maggio, N. Vreman
Department of Automatic Control, Lund University; email: {martina.maggio, nils.vreman}@control.lth.se

S. Altmeyer, F. Haas
Augsburg University; email: {altmeyer, haas}@es-augsburg.de

A. Casimiro, J. Cecílio, G. Jäger, A. Espindola
LASIGE, University of Lisbon; email: {casim, jmcecilio, gjager, adespindola}@ciencias.ulisboa.pt

S. Skalistis
Collins Aerospace; email: stefanos.skalistis@collins.com

J. Kouwer, G. de Lange
Thales Nederland B.V.; email: {Jeroen.Kouwer, Guus.delange}@nl.thalesgroup.com

J. Almeida, H. Blasum, M. Brotz, S. Wagner
SYSGO SAS and SYSGO GmbH; email: {jose.almeida, holger.blasum, mario.brotz, stephan.wagner}@sysgo.com

P. Novobílský
Q-media; email: pno@qma.cz

Abstract

Due to the increasing performance demands of mission- and safety-critical Cyber-Physical Systems (of Systems), these systems exhibit a rapidly growing complexity, manifested by an increasing number of (distributed) computational cores and application components connected via complex networks. However, with these systems' growing complexity and interconnectivity, the chances of hardware failures and disruptions due to cyber-attacks will also quickly increase.

In the ADMORPH project we explore system adaptivity, in terms of dynamically remapping application components to processing cores, to fuse fault and intrusion tolerance with the increasing performance requirements of mission- and safety-critical CPS(oS). This paper describes the overall ADMORPH architecture and provides an overview of the developed methodologies, methods and tools for the specification, design, analysis and runtime deployment of adaptive mission- and safety-critical CPS(oS) that are robust against both component failures and cyber-attacks.

Keywords: Cyber-Physical Systems, Adaptation, Resilient control, Design Space Exploration

1 Introduction

Cyber-Physical Systems (CPS) form a crucial information-technology domain worldwide, that covers many industrial sectors, including: health industries, industrial automation, avionics, and space. CPS often consists of heterogeneous, multi- or many-core systems that are distributed and connected via complex networks, creating what is known as Cyber-Physical Systems of Systems (CPSoS).

Designing CPS(oS) systems is challenging due to the stringent and often conflicting extra-functional design requirements they must meet. A single task in the system that misses its computational deadlines can have severe – sometimes even life-threatening – consequences for mission- or safety-critical CPS(oS). Safety-critical CPS(oS) demand ultra-high levels of dependability, which is becoming even more important as the levels of system autonomy rise.

To ensure reliability, availability, and safety, mission- and safety-critical CPS(oS) must be able to handle various disruptive events caused by hardware component failures or cyber-attacks like Denial-of-Service (DoS) attacks aimed at disrupting the system or compromising critical system functionality.

System adaptation offers a promising technique to maintain

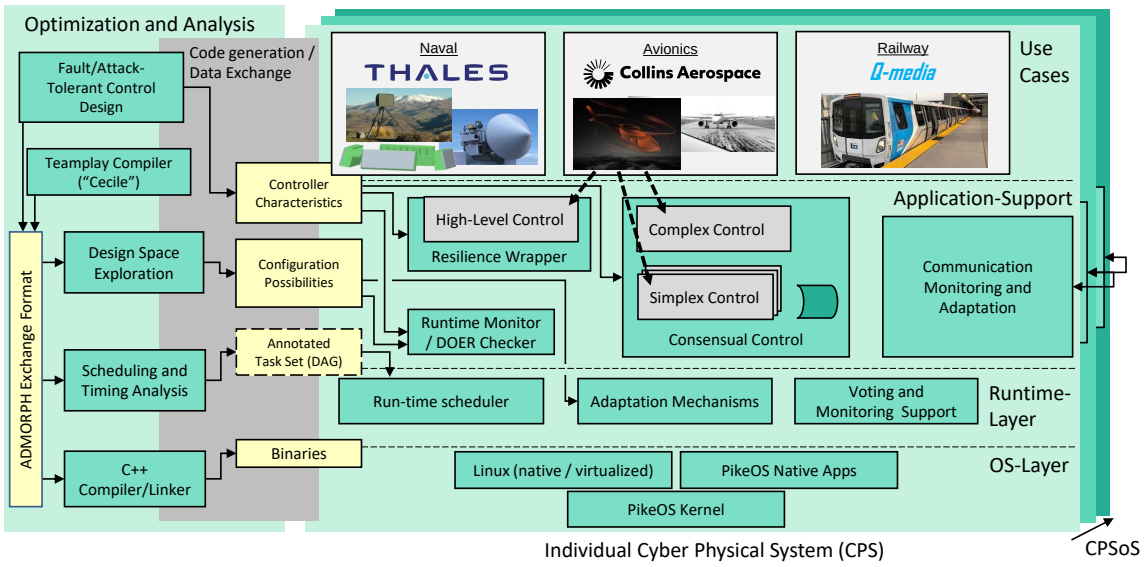


Figure 1: Global perspective of the ADMORPH architecture.

the system's operation at the required level of Quality of Service (QoS) or achieve graceful degradation when disruptive events hit the system. Allowing dynamic task relocation or replication between different processors, processor types, or even across hardware/software boundaries yields morphing systems. These enable applications to execute using a variety of different and dynamically interchangeable system configurations over heterogeneous system resources. Morphing systems are the key to providing a real breakthrough in establishing robustness against unpredictable, disruptive events that affect the dependability of systems, prolonging their life and maximizing their efficiency during this lifetime.

The absence of systematic methodologies to design and runtime-manage complex, adaptive embedded systems is holding back their progress. To address this issue, the ADMORPH project[1] is investigating holistic approaches to systematically designing, analyzing, and managing embedded computer systems in mission- or safety-critical CPS(oS). The approach uses the concept of system adaptation in the form of dynamic task-to-resource allocation to achieve fault and intrusion tolerance.

Adaptation is crucial for coping with faults over the long run, provided the system can tolerate faults long enough for adaptation to become effective. Moreover, it allows us to optimize both functional and non-functional properties of these systems. In ADMORPH, we design and develop adaptation strategies (1) to evade faults by relocating services to a different set of resources, (2) to improve resilience by including more resources in the tolerance of faults, and (3) to match the systems' resilience to the perceived threat by allocating more or less resources to the defense.

This paper presents an overview of the ADMORPH architecture and its key components (Figure 1). The architecture encompasses Optimization and Analysis components (including design space exploration, a coordination language, and scheduling and time analysis), Application Support components (providing various adaptation strategies, with the support of monitors), and considers Runtime and Operating System layers that are designed to provide real-time support for applications, for different adaptability requirements. All these can be used in multiple application scenarios, and in ADMORPH we consider three specific Use Cases to validate the several components of the architecture.

2 Optimization and Analysis

The embedded computer systems in CPS(oS) are often composed of multicore systems built for a specific application, operating a combination of homogeneous and heterogeneous cores on a single chip. Modelling and exploring embedded multicore systems is a time-consuming and a complex task. Moreover, these devices can be deployed for a long term and therefore system lifetime reliability is an important consideration while designing them. However, it is imperative that over long periods of deployment time, some of the cores in such systems will start to deteriorate owing to the ageing process and will eventually fail. In principle, placing extra cores on the chip can increase the lifetime reliability albeit at the cost of increased power consumption and chip area.

2.1 Design Space Exploration

In ADMORPH, we presented a framework [1] to explore the design space of platform architectures and their floorplans, with the contradictory objectives of increasing lifetime reliability and lowering the average power consumption. The exploration algorithm in this framework, based on a Genetic Algorithm, returns the *Pareto Set* from the population upon convergence. The design points in the *Pareto Set* exhibit the trade-off between two objectives and one cannot be considered better over the other w.r.t both the objectives [2]. The framework employs a high level simulator to calculate the Mean Time to Failure (MTTF) of the chip. A higher MTTF of a chip represents the fact that the chip will fail after longer duration and hence is predicted to operate for longer duration

[1] https://www.admorph.eu

after initial deployment. In this context, MTTF can be interpreted as an estimate of active lifetime of the chip and thus its reliability.

The simulator runs multiple times, also called Monte Carlo simulation, to take the averages of both failure times and power usage. The high number of simulations required makes the framework compute intensive. We therefore also proposed variations of the exploration methodology to reduce the total number of simulations needed for a faster convergence of the main algorithm.

2.2 TeamPlay Coordination Language

The TeamPlay Coordination Language [3] supports the specification of CPS(oS) at a very high level of abstraction. TeamPlay adopts the principle of *exogenous coordination* as a key design choice and achieves a complete separation of concerns between the specification of stateless components (aka tasks) and their orderly interaction in a streaming network. Components are characterized by a set of functional as well as non-functional contracts. The functional contracts consist of a set of typed input and output ports as well as a set of state ports, i.e. output ports that are short circuited to corresponding input ports, and hence allow us to mimic state in an otherwise stateless world. The non-functional contracts span from average and worst-case execution time (on a given hardware unit) to average and worst-case energy consumption (on a given hardware unit) to fault-tolerance execution regimes, such as dual or triple modular redundancy.

Components may have multiple versions that behave identically with respect to the functional contracts, but typically expose different behaviour with respect to the non-functional contracts, even when run on the same hardware. As a coordination language TeamPlay focuses on the specification and interaction of components and leaves the implementation of components to lower-level languages, usually C or C++ in the domain of cyber-physical systems.

This design permits TeamPlay to adapt CPS(oS) applications to the available hardware, usually commodity-off-the-shelf high-performance embedded systems, under varying objectives such as meeting deadlines, energy budgets or robustness requirements [4]. To this effect we have proposed various scheduling algorithms [5, 6], as well as the adaptive runtime environment YASMIN [7] and investigated the best use of constrained resources for fault-tolerance under a weakly-hard real-time regime [8], among others.

2.3 Scheduling and Time Analysis

For the scheduling, we assume an input AFC-file that models an application as a directed acyclic graph, divided into section with varying redundancy levels. This means that redundancy levels can change dynamically in between section boundaries. The level of redundancy at runtime depends on the number of available processing elements and the current fault rate. To enable the dynamic adaptation depending on the number of cores and fault-rate, we use a set of pre-computed schedules for the different sections that can be loaded and executed at runtime. For the computation of the various schedules, we use a modified version of the Heterogeneous Earliest Finish Time (HEFT) scheduler that ensures a suitable mapping of redundant tasks on the processing elements. Based on this scheduler, we have developed *faktum*, a scheduling and analysis tool that serves two purposes: Firstly, it computes offline a set of fault-tolerant schedules, using the HEFT scheduler as described above, that can be fed to the design-space exploration. Secondly, it serves as a scheduling verification and analysis tool, which estimates the feasibility of design candidates during the design-space exploration. After the design-space exploration, it derives the final verdict on the suitability of the chosen candidate architecture.

2.4 Analysis of the Impact of Deadline Misses on Control Systems

Feedback control is a central enabling technology in a wide range of applications. Control systems are at the core of energy distribution infrastructures, regulate the behaviour of engines in vehicles, and are embedded in household appliances like washing machines. Control is centred around the feedback mechanism. Sensors provide information about the current state of the physical environment. This is used to compute suitable control actions to fulfil performance requirements, that are then implemented by actuators. For example, adaptive cruise control systems use measurements from a range of sensors to determine how to adjust the throttle to automatically regulate the vehicle's speed, while maintaining a safe distance from vehicles ahead.

Control actions are often calculated using hardware and software. Hence, the computation of the new control signals is subject to accidental faults, systematic issues, and software bugs. In practice, these computational problems are often ignored. But when can this be done safely? In ADMORPH we introduce a framework for analyzing the behaviour of control software subject to computational problems. We started the analysis with the evaluation of the stability [9] and performance [10] of control systems subject to consecutive deadline misses. We then worked on generalising the analysis to all the other weakly-hard [11] task models [12, 13] and creating an experimental toolchain [14]. We also analysed the latency of complex pipelines in which tasks in a chain of dependent computations experience deadline misses [15] and discussed recovery strategies for control systems [16].

3 Adaptation Methods

Applications quite naturally adapt their functionality or performance to changing demands and environmental situations. For example, planes transition through modes for taking-off, flying, landing and taxiing from the runway to the parking position at the terminal. While the triggers for these changes are expected or at least well predictable, one cannot equally well predict when the system has to adapt to faults. In ADMORPH, we focus on exactly that prediction and on the adaptation of the fault and intrusion tolerance mechanisms that support such adaptation. Figure 2 shows the control architecture of ADMORPH.

In various scenarios, high-level controllers play a crucial role in guiding the behavior of lower-level control loops, which operate at a much higher frequency and prioritize the system's stability. Functional adaptation, involving transitions

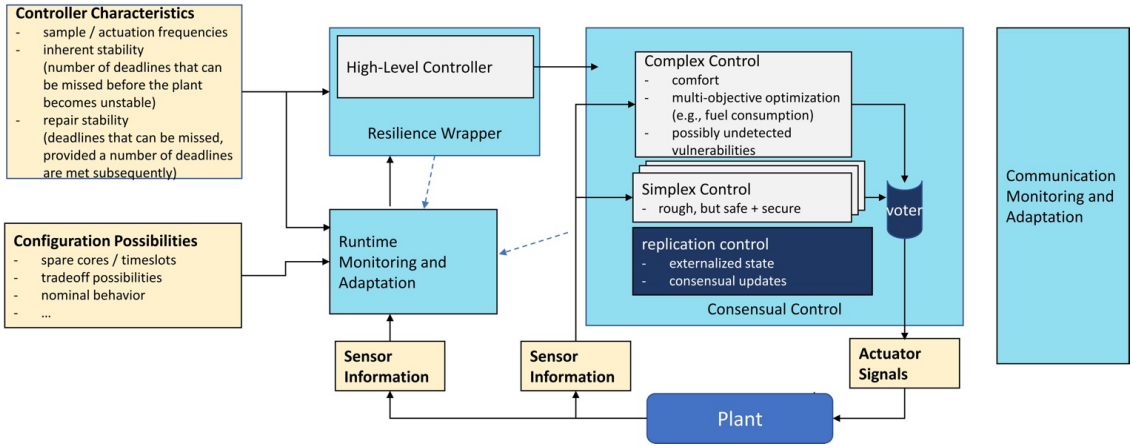


Figure 2: ADMORPH's control architecture.

between multiple high-level controllers, typically occurs during pre-defined configuration changes coordinated by the coordination language, its compiler, and toolchain. Threat-related adaptations, such as adjusting the internal resilience of components in response to perceived threat levels, require a combination of proactive resource planning for severe threats and swift runtime adjustments to ensure the system maintains the desired state when threats emerge. Both strategies require runtime adaptations within the control architecture to respond to unforeseen situations, utilizing available resources swiftly and planned configurations from design-space exploration. Such adaptations may include (a) activating the simplex subsystem if the complex fails to provide timely and accurate information, (b) modifying the simplex replication policy, changing the detection quorum that enhances resilience over subsequent control epochs, (c) relocating consistently failing controllers to spare resources and (d) adapting the frequency of rejuvenation. These runtime adaptations ensure efficient system response, leveraging redundancy and planned measures for effective performance in unforeseen circumstances.

However, control tasks are susceptible to failure or compromise over time as adversaries seek to gain control of the system and cause harm to the operating environment. Ensuring the recovery of these control tasks and the resources they utilize becomes crucial to maintain healthy majorities and withstanding failures. In ADMORPH, we have enhanced both the state-capturing capability of replicated controllers and their ability to restart replicas in a stateless manner, while also incorporating the capability to introduce additional replicas. These new replicas are initialized using pre-compiled and pre-analyzed binary images sourced from a diverse pool. This diversity thwarts adversarial knowledge regarding replica attacks. By starting the replicas without their previous states, the replication controller injects the captured state and maintains synchronization with the control tasks, ultimately transitioning the responsibility to control the device or cyber-physical system. This approach allows for the creation of additional replicas, which can later join the active consensus group once they are fully operational.

Another essential aspect that ADMORPH addresses is related to the time of adaptation and reconfiguration. In ADMORPH, we are considering finer-grained reconfiguration, and we do so by integrating fault models into the scheduling analysis of redundant dataflow tasks. This way, we determine the Worst-Case Execution Time (WCET) of such tasks in the presence of errors down to a specific probability.

Nevertheless, certain factors can still result in unbounded reconfiguration times, such as repeated resource reboots following a crash or the reactivation of the same software vulnerability. These situations often indicate a persistent fault within the hardware resource or the re-instantiated software component. In the ADMORPH approach, we are considering software diversification and relocation to address this issue. We have considered the implications of these actions on the WCET of the software component.

To address these concerns, the ADMORPH control architecture employs a three-step reconfiguration process: (1) the new configuration is initiated by creating new replicas, starting components that implement the new functionality, or utilizing spare resources, (2) the new configuration establishes connections with the existing setup, ensuring it receives state updates, sensor inputs, and is monitored effectively and (3) once the preparation phase is completed and all components confirm their readiness, control is transitioned atomically by updating the control system and associated components to consider the new subsystems instead of the previous configuration.

During this transition, the control system exclusively considers and applies proposals from the old configuration, leveraging its inherent fault tolerance for a specific duration. Once the new configuration is fully established, regardless of the time taken, the control system solely considers proposals from the new configuration at the beginning of the next control epoch. This decoupling of configurations allows timely control to be maintained by either the old configuration (until the transition point) or the new configuration (from the transition point onwards).

4 Runtime and OS support for resilient control

In ADMORPH, to ensure security and safety of the application workloads, we have used a separation kernel to provide

strict separation. Historically, a governing principle of a separation kernel is to enable static configurations for relatively simple embedded systems and applications [17]. With safety-critical embedded systems and CPS(oS) growing to more powerful hardware, more complex embedded systems become feasible, and, in part the ADMORPH use cases were helpful to gather demands for runtime and OS support.

We have shown that it is possible to extend the separation kernel on the sensing side by host intrusion detection by control flow integrity [18, 19], network intrusion detection with Suricata, and safety monitoring infrastructure for heterogeneous systems. For the acting side, we have extended the separation kernel experimental by more flexible scheduling (run-time adaptation of time windows and cross-CPU thread/task migration). We have also implemented mechanisms for FPGA reconfiguration. For secure update, we have demonstrated the use of Mender.

5 Use Cases

ADMORPH's adaptivity technology is being evaluated through three use cases. These use cases have been selected to cover a significant safety- and mission-critical CPS(oS) spectrum. They span different domains with varying system requirements and needs regarding robustness and quality of service.

5.1 Autonomous Aerospace Systems

Flight delays due to airport congestion create a costly ripple effect for airlines and airports. Enhancing aircraft autonomy in specific flight phases can mitigate these issues while ensuring safety remains crucial in unforeseen circumstances like system faults and security attacks. In a highly regulated environment, it is not yet clear who the decision maker for specific actions will be. In some cases, it will be Air Traffic Control (ATC), whereas in other cases, it will be the aircraft. This provides a perfect environment for System of Systems (SoS) demonstration where ADMORPH solution can be applied to provide adaptivity and safety.

Nowadays, in commercial aircrafts, the level of autonomy during the cruise is very high. However, takeoff, landing, and taxiing are still pending subjects and very critical. In this use case, we are implementing a hybrid simulation environment that combines model-in-the-loop and hardware-in-the-loop approaches. The co-simulation involves integrating Simulink and FlightGear, while real-time execution takes place on the Ultrascale+ platform. This configuration can be equipped with a safety-critical operating system or hypervisor, such as PikeOS, to ensure real-time characteristics. This setup mimics the functionality of a single computer unit within an Integrated Modular Avionics architecture, allowing for fault injection testing. These faults will simulate operational failures and attacks. Leveraging the expressive coordination language utilized in the project, detecting these faults will be used to trigger adaptivity measures, thereby ensuring the overall system's safety.

5.2 Radar Surveillance Systems

Radar surveillance systems are essential for ships as they provide crucial situational awareness, enabling vessels to detect other ships, obstacles, and navigational hazards, enhancing maritime safety and preventing collisions. This used case involves evaluating specific methods within the context of an industrial embedded software system (or subsystem) used for radar surveillance processing in a laboratory setup using a realistic processing platform. As command and control decisions require reliable and robust real-time data processing, the ability of the ADMORPH approach to achieve fault tolerance is substantially assessed and validated.

In this particular use case within the ADMORPH framework, the coordination language is crucial in specifying adaptive systems alongside adaptive runtimes. This approach ensures formal guarantees and facilitates comprehensive testing of the runtime system itself.

5.3 Transport system

This use case focuses on evaluating ADMORPH approaches within the context of a Train Supervision Surveillance System, which falls under the category of a System of Systems (SoS). The railway system operates multiple information flows between ground systems and moving trains, each serving specific purposes with varying levels of criticality. In the event of incidents, it becomes crucial to maintain communication channels associated with the most critical information flows. Hence, a system capable of adapting to diverse operating conditions such as signal level, channel interference, hardware faults, and line overload is essential. In this use case, the TeamPlay coordination language describes and generates adaptation targets. The PikeOS hypervisor is employed to partition the system into isolated and independent partitions that communicate through proprietary tools. It facilitates resource allocation and sharing while enabling the execution of multiple Linux instances within isolated partitions. This configuration allows the creation of replicas and seamless switching between them, facilitating necessary adaptations in the presence of faults or attacks.

6 Conclusion

This paper comprehensively overviews the ADMORPH architecture and its key components. We delve into the mechanisms employed by ADMORPH to facilitate real-time adaptation while upholding safety considerations. While adaptation strategies are generated offline, potentially through design-space exploration, the actual runtime adaptation necessitates careful attention to ensure swift response times within individual CPSoS building blocks. Tasks with strict timing requirements may require internal resilience mechanisms to effectively handle accidental and malicious faults. We identified and described various strategies for adapting to different scenarios, bounding reconfiguration times, and particularly decoupling reconfiguration from the operational behavior of components. These strategies enable efficient and reliable runtime adaptation within the ADMORPH solution. Lastly, we referred to the Use Cases that support the project by briefly describing the challenges and how the ADMORPH solutions can address them.

Acknowledgments

This work was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 871259 (ADMORPH project).

References

[1] D. Sapra and A. D. Pimentel, "Exploring multi-core systems with lifetime reliability and power consumption trade-offs," in *Embedded Computer Systems: Architectures, Modeling, and Simulation: 23rd International Conference, SAMOS 2023*, Springer, 2023.

[2] A. D. Pimentel, "Exploring exploration: A tutorial introduction to embedded systems design space exploration," *IEEE Design & Test*, vol. 34, no. 1, 2017.

[3] J. Roeder, B. Rouxel, S. Altmeyer, and C. Grelck, "Towards energy-, time- and security-aware multi-core coordination," in *22nd International Conference on Coordination Models and Languages (COORDINATION 2020), Malta* (S. Bliudze and L. Bocchi, eds.), vol. 12134 of *Lecture Notes in Computer Science*, pp. 57–74, Springer, 2020.

[4] B. Rouxel, U. Pagh Schultz, B. Akesson, J. Holst, O. Jorgensen, and C. Grelck, "PReGO: a generative methodology for satisfying real-time requirements on cots-based systems: Definition and experience report," in *19th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences (GPCE 2020), Chicago, USA*, pp. 70–83, ACM, 2020.

[5] J. Roeder, B. Rouxel, S. Altmeyer, and C. Grelck, "Energy-aware scheduling of multi-version tasks on heterogeneous real-time systems," in *36th ACM/SIGAPP Symposium on Applied Computing (SAC 2021)*, pp. 500–510, ACM, 2020.

[6] J. Roeder, B. Rouxel, and C. Grelck, "Scheduling dags of multi-version multi-phase tasks on heterogeneous real-time systems," in *14th IEEE International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC 2021), Singapore*, IEEE, 2021.

[7] B. Rouxel, S. Altmeyer, and C. Grelck, "YASMIN: a Real-time Middleware for COTS Heterogeneous Platforms," in *22nd ACM/IFIP International Middleware Conference (MIDDLEWARE 2021)*, ACM, 2021.

[8] L. Miedema and C. Grelck, "Strategy Switching: Smart Fault-tolerance for Weakly-hard Resource-constrained Real-time Applications," in *Software Engineering and Formal Methods, 20th International Conference, SEFM 2022*, pp. 129–145, Springer, LNCS 13550, 2022.

[9] M. Maggio, A. Hamann, E. Mayer-John, and D. Ziegenbein, "Control system stability under consecutive deadline misses constraints," in *ECRTS, Euromicro Conference on Real-Time Systems*, 2020.

[10] N. Vreman, A. Cervin, and M. Maggio, "Stability and performance analysis of control systems subject to bursts of deadline misses," in *ECRTS, Euromicro Conference on Real-Time Systems*, 2021.

[11] G. Bernat, A. Burns, and A. Liamosi, "Weakly hard real-time systems," *IEEE Transactions on Computers*, vol. 50, no. 4, pp. 308–321, 2001.

[12] N. Vreman, R. Pates, and M. Maggio, "WeaklyHard.jl: Scalable analysis of weakly-hard constraints," in *RTAS, IEEE Real-Time and Embedded Technology and Applications Symposium*, 2022.

[13] N. Vreman, P. Pazzaglia, V. Magron, J. Wang, and M. Maggio, "Stability of linear systems under extended weakly-hard constraints," *IEEE Control Systems Letters*, vol. 6, pp. 2900–2905, 2022.

[14] B. J. Josephrexon and M. Maggio, "Experimenting with networked control software subject to faults," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 1547–1552, 2022.

[15] P. Pazzaglia and M. Maggio, "Characterizing the effect of deadline misses on time-triggered task chains," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 3957–3968, 2022.

[16] P. Pazzaglia, A. Hamann, D. Ziegenbein, and M. Maggio, "Adaptive design of real-time control systems subject to sporadic overruns," in *DATE, Design, Automation and Test in Europe Conference*, 2021.

[17] S. Tverdyshev, H. Blasum, B. Langenstein, J. Maebe, B. De Sutter, B. Leconte, B. Triquet, K. Müller, M. Paulitsch, A. Söding-Freiherr von Blomberg, and A. Tillequin, "MILS Architecture," Sept. 2013.

[18] M. Kadar, *Integration Methods for Host Intrusion Detection into Embedded Mixed-Criticality Systems*. PhD thesis, TU Kaiserslautern, 2022.

[19] D. Kuzhiyelil, P. Zieris, M. Kadar, S. Tverdyshev, and G. Fohler, "Towards Transparent Control-Flow Integrity in Safety-Critical Systems," in *Information Security*, Lecture Notes in Computer Science, (Cham), pp. 290–311, Springer International Publishing, 2020.

Mode Change Management for Adaptive Cyber-physical Systems

Miguel García-Gordillo, Joan J. Valls, Javier Coronel

Instituto Tecnológico de Informática, Valencia, Spain; email: {miguelgarcia, jvalls, jcoronel}@iti.es

Sergio Sáez

Instituto Universitario Mixto Tecnológico de Informática, Universitat Politècnica de València, Valencia, Spain; email: ssaez@iti.es

Abstract

Cyber-physical systems are usually integrated with a highly changing environment. Because of the variability of their environment, these systems need to adapt their behaviour at runtime to achieve the expected results. In the case of distributed solutions, this behaviour must be consistent across the continuum, increasing the complexity of the design due to the need to synchronise the different components distributed in the multiple nodes.

A multimode application approach could be used to solve this need, assuming it is implemented as a distributed system. The approach involves applications with multiple behaviours (or operational modes) and a mechanism to change from one mode to another.

This article describes an operational mode change management solution based on that approach and the reference architecture presented in the TRANSACT project. This architecture integrates CPS end devices at the edge of the network with edge computing servers and cloud computing facilities. Static and dynamic views are included to describe the solution, and specific use case scenarios are added for better understanding.

Keywords: Cyber-physical systems, Distributed systems, Adaptive systems.

1 Introduction

Cyber-Physical Systems (CPS) are usually part of distributed systems that integrate physical and computational components to perform their operations. It is well known that this combination can lead to unpredictable failures due to the variability of the environmental conditions of the physical part. CPS usually must adapt their behaviour to these new conditions to achieve the expected results.

In recent years, several studies have focused on designing adaptive systems and describing their architecture patterns. Many of them have been developed from the framework of the MAPE-K loop [1] as a base architecture to enable the self-adaptation capability in these types of systems.

Regarding architectural patterns, Weyns et al. [2] have presented a selection of MAPE-K patterns to model different types of interactions between MAPE loops with different degrees of decentralisation. Focusing on decentralised systems, Weißbach et al. [3] have defined an approach to implement distributed adaptive systems without needing a central coordinator.

It is often necessary to define multiple behaviours to cover all functional needs when the system must adapt to possible variations. The design presented in this article proposes the use of multimode applications [4], for instance, applications with several operational modes, which are characterised by their own set of functionalities. During the execution, only one of the operational modes can be enabled, known as the active mode, which will define the current behaviour of the application.

This article proposes an approach to managing distributed multimode applications in the framework of the TRANSACT project. The project is introduced in section 2. The solution to adapt the behaviour of cyber-physical systems at runtime is presented in section 3. The static and dynamic views of the solution are described in sections 4 and 5, respectively. Finally, section 6 provides the conclusions and future works.

2 The TRANSACT project

The work described in this article is part of the solutions exposed in the TRANSACT project [5]. This project aims to develop a distributed architecture for transforming safety-critical CPS from localised standalone systems into safe and secure distributed solutions leveraging edge and cloud computing.

In order to manage the adaptability of the system from a continuum point of view, *the concepts and solutions for operational modes and change transitions* have been included as part of the safety and performance concepts to consider in the TRANSACT project when connecting CPS to the cloud [6]. This solution describes how to manage mode changes across the different tiers of a distributed system in a safe manner.

The TRANSACT tree-tier architecture (Figure 1) is based on a three-tier computing continuum that spans from the CPS device, through the edge, to the cloud. The TRANSACT architecture concept brings together CPS end devices at the edge of the network, with edge computing servers and cloud

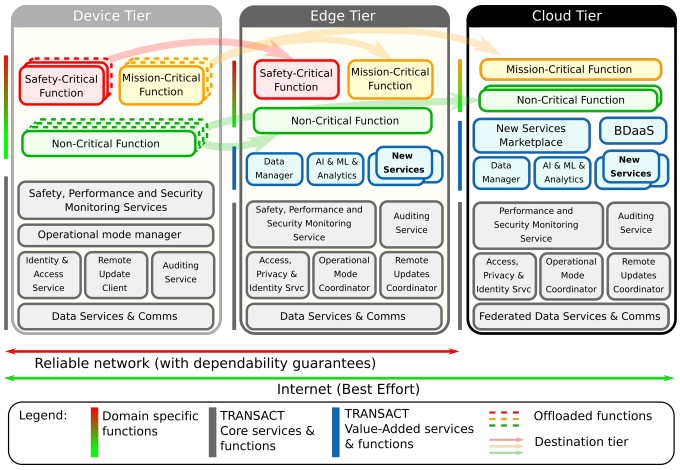


Figure 1: TRANSACT Reference architecture

computing facilities, hosting multiple mixed-criticality applications. This reference architecture defines the safety-critical and mission-critical functions, the core services, and further value-added services [6].

3 Proposed solution

This article presents a solution for adaptive cyber-physical systems, capable of continuously observing relevant information, to decide the actions to perform according to the expected results [7]. That is, the behaviour of the systems should be adapted to achieve those results. Their decisions depend on the system requirements and observing the systems themselves and their environment. A common approach from the architecture point of view is to design these adaptive systems based on the MAPE-K loop (Monitor-Analyse-Plan-Execute on a Knowledge base) [1].

In addition, multimode applications [4] comprise the different operational modes that configure the behaviour and functionality needed for the system to work as expected. Each operational mode defines a behaviour the system has to show in each phase of its execution, including both regular operation and failure mitigation mechanisms. A mode transition corresponds to a change in the system behaviour, responding to events or alarms provided by other services.

When a behavioural change is needed, a Mode Change Request (MCR) is generated to demand a transition between modes. Changing from the active mode to a new one requires that the functionalities of an outgoing mode be replaced by those of the incoming one. This process introduces a transient state where part of the old and new mode configurations can coexist.

Changing between modes is even more complicated when dealing with CPS solutions, where components are distributed across different nodes and must be synchronised while the transition is currently in progress. Designing a real-time multimode system is, therefore, a complex challenge since schedulability must be ensured, not only for each operational mode but also in the transitions between modes. To ensure the correctness during these transitions, Mode Change Protocols (MCP) define how a system must switch between modes using a set of predefined rules [8].

Selecting the best MCP for a given system is not easy. In SafeMC [9], a syntax is proposed to express MCPs. They also present a system that implements these language primitives and allows the evaluation of the protocols.

Regarding implementing multimode real-time systems based on those MCPs, Sáez et al. [10] propose a framework for multiprocessor, multimode real-time applications. The article describes how to model the different system modes and transitions using UML finite state machines (FSM). They present a code generation tool to implement the defined model in the Ada language. Additionally, Inam et al. [11] provide a generic skeleton for a two-level adaptive hierarchical scheduling, supporting multiple modes and multiple mode-change mechanisms over the FreeRTOS operating system.

This work presents a solution for adaptive cyber-physical systems whose design has been based on the MAPE-K loop and multimode applications. The presented solution covers the functionalities described in the following subsections, including mode change coordination and mode change management on the device.

3.1 Mode change coordination

As already mentioned, in the case of distributed systems, the difficulty of implementing multimode applications increases. Such difficulty is due to the necessary synchronisation between the different nodes during the transition from one operational mode to the next. It is at this point that a coordination mechanism becomes essential.

The proposed solution is in charge of synchronising the continuum between the nodes in a distributed system to guarantee the robustness of the system in terms of operational modes and mode change transitions.

The multimode distributed system comprises one or several coordinators distributed between the edge and the cloud, which define a set of global modes and a group of devices connected to it. Each device defines its set of local operational modes, which can be the same as the global modes of the coordinator or completely different.

From the point of view of the MAPE-K loop, the functionality of coordinating mode changes in a distributed system falls mainly in the planning (P) and execution (E) phases of the loop model. The planning phase is in charge of processing the incoming events provided by other services, such as analysis or monitoring services of the TRANSACT reference architecture, and generating, if necessary, the corresponding MCR. The execution phase propagates the MCR throughout the different nodes. It also adapts the requests to be compatible with the local operational modes when they differ from the global ones.

3.2 Mode change management on the device

The solution for managing mode changes in the device must allow receiving MCR, processing them based on a defined MCP and managing the activation of periodic tasks based on the MCP and the active mode configuration.

In the device, the planning phase (P) is in charge of defining when a mode change is necessary, which can be given by an internal event of the device or by an external MCR. In addition, it notifies the edge when a transition between modes is complete.

On the other hand, the execution phase (E) controls the activation of the tasks based on the current mode and is in charge of performing the action that allows the transition.

4 Static view of the reference architecture

The static view of the solution describes the components involved in managing the operational modes and deployed across the device-edge-cloud continuum. The proposed solution will be focused on the Operational Mode Manager and Coordinator services in the TRANSACT reference architecture (Figure 1).

The components involved in this solution are the following:

- **Operational Mode Coordinator in the cloud tier** aims to manage the transitions between modes at the system level. It also must be responsible for propagating requests to perform the mode change in the complete system, ensuring application consistency.
- **Operational Mode Coordinator at the edge tier** manages the transitions between modes at the edge level. It must follow the request from the cloud coordinator and adapt them to the devices. It is expected that devices could be legacy and not designed to fit into this architecture. For this reason, the coordinator in the edge must provide the adaptation layer to synchronise each connected device, ensuring application consistency.
- **Operational Mode Manager in the device tier** aims to control the transitions between operational modes at the device level. In addition, this component is responsible for both enabling tasks that correspond to the active mode and correctly executing the rules defined in the MCP to ensure a safe transition between modes.

5 Dynamic view of the reference architecture

The dynamic view of the reference architecture describes the relationship between the different components involved in the solution. It defines the workflow that the components must follow to establish and coordinate the operational modes of the system.

The mode change management should detect the need for a behavioural change from the events and alarms provided by other services on the system. It must also propagate the MCR and synchronise the different components involved, located in the different tiers of the architecture (Figure 1). These actions aim to maintain consistency across the device-edge-cloud continuum, ensuring that each part behaves as expected by the rest of the system.

The behaviour of the mode change management should not only consider the option of coordinating mode changes from higher tiers (edge or cloud), but also the capability to force them from a device, such as detecting a failure in the device tier and propagating it to the rest of the system.

Relevant scenarios have been described in the following subsections to clarify how the dynamic operation of the architecture behaves.

5.1 Scenario 1: Cloud coordination

In this scenario, *Cloud Operational Mode Coordinator* is configured to respond to events and alarms generated by the different services and detect the necessity of a behavioural change. *Edge Operational Mode Coordinator* will receive the MCR provided by the cloud and coordinate connected devices.

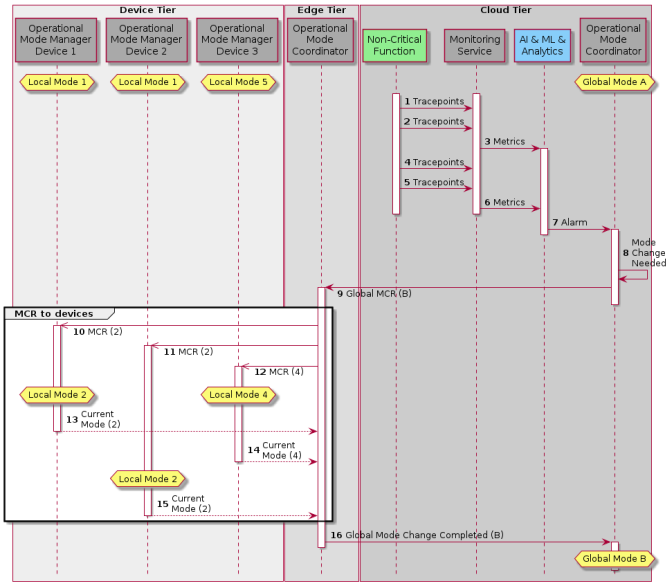


Figure 2: Sequence diagram - Cloud coordination

As depicted in Figure 2, *Monitoring Service* collects the tracepoints (1) from the *Non-Critical Function* and generates the metrics (3) with the parametrised behaviour. These metrics are analysed by the *Analytics Service*, which generates an alarm (7) to notify a change in behaviour. *Cloud Operational Mode Coordinator* receives that alarm indicating a necessary updating of the global operation (8). This coordinator initiates a process to request edge for the expected mode change (9) and waits for it to be completed (16) to ensure that the system is synchronised and consistent.

Because a device may implement different operational modes than those configured at the system level, the *Edge Coordinator* must define which *local mode* for each device complies with the requested *global mode*. In Figure 2, *Devices 1* and *Device 2* should be configured in *Local Mode 2* (10,11) and *Device 3* needs to change from *Local Mode 5* to *Local Mode 4* (12). Once all the devices are in the expected mode (13-15), *Edge Coordinator* notifies to *Cloud Coordinator* to complete the mode change operation (16).

5.2 Scenario 2: Loss of connection

In many cases, distributed applications on the device-edge-cloud continuum are designed to provide cloud services to enhance device-edge capabilities. These value-added services

in the cloud allow the execution of powerful algorithms that would not be possible otherwise. Critical functions in device and edge tiers usually require full availability, but cloud services do not ensure this capability. In case of a lack of connectivity between edge and cloud, the availability must be guaranteed in the device and edge tiers.

The proposed solution to this scenario involves modifying the behaviour of the system, allowing the functionality that cannot be executed in the cloud to be run on the device. This fallback mechanism will maintain the availability of the service, but with a reduced quality due to the lack of computing power in the device.

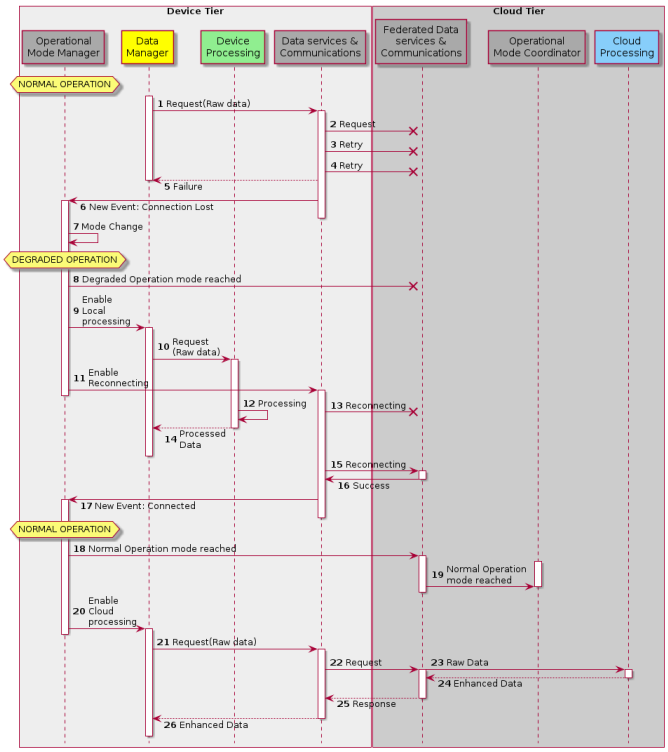


Figure 3: Sequence diagram - Loss of connection

In the example depicted in Figure 3, *Device Data Manager* aims to process the raw data in order to provided it to other services. In the *Normal Operation*, raw data is sent to the the *Cloud Processing* service. After processing, the enhanced data is returned to the device for further use. If connectivity with the cloud is interrupted (1-6), the availability of the processed data must be ensured. *Device Operational Mode Manager* must activate the *Degraded Operation* mode (7), updating the application behaviour (9,11) and enabling the low-quality data processing. Once the connection has been re-established (17), the *Operational Mode Coordinator* is notified (18,19), and the *Normal Operation* mode is enabled again (20-26).

6 Conclusions

Based on the TRANSACT reference architecture, the proposed solution is a framework designed to manage the adaptability of multimode CPS. It coordinates the system behaviour at the device-edge-cloud continuum, managing the mode change requests and the transitions between modes.

Two main advantages of this solution have been identified: i) it allows legacy device integration, in which operational modes can be predefined, thanks to the differentiation between *global modes* and *local modes*, and ii) mode change planning can be distributed, and the decisions about the necessity of a behavioural change can be taken throughout the continuum. The associated mode change requests can be generated from different tiers and propagated across them.

One of the planned next steps is to implement the proposed solution within a specific context, such as the TRANSACT industrial use cases. Analysing and implementing these new potential scenarios will help validate the solution.

References

[1] P. Arcaini, E. Riccobene, and P. Scandurra, "Modeling and analyzing MAPE-K feedback loops for self-adaptation," in *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, pp. 13–23, IEEE, 2015.

[2] D. Weyns, B. Schmerl, V. Grassi, S. Malek, R. Mirandola, C. Prehofer, J. Wuttke, J. Andersson, H. Giese, and K. M. Göschka, "On patterns for decentralized control in self-adaptive systems," in *Software Engineering for Self-Adaptive Systems II*, Springer, 2013.

[3] M. Weißbach and T. Springer, "Coordinated execution of adaptation operations in distributed role-based software systems," in *Proceedings of the Symposium on Applied Computing*, pp. 45–50, 2017.

[4] A. Burns, "System mode changes-general and criticality-based," in *2nd Workshop on Mixed Criticality Systems (WMC)*, pp. 3–8, 2014.

[5] "TRANSACT project. https://transact-ecsel.eu/," 2023.

[6] T. Hendriks, B. Akesson, J. Voeten, M. Hendriks, J. Coronel, M. García-Gordillo, S. Sáez, and J. J. Valls, "Thirteen concepts to play it safe with the cloud," in *The 17th IEEE International Systems Conference*, 2023.

[7] S. Kounev, P. Lewis, K. L. Bellman, N. Bencomo, J. Camara, A. Diaconescu, L. Esterle, K. Geihs, H. Giese, S. Götz, *et al.*, "The notion of self-aware computing," *Self-Aware Computing Systems*, pp. 3–16, 2017.

[8] J. Real and A. Crespo, "Mode change protocols for real-time systems: A survey and a new proposal," *Real-time systems*, vol. 26, pp. 161–197, 2004.

[9] T. Chen and L. T. X. Phan, "SafeMC: A system for the design and evaluation of mode-change protocols," in *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pp. 105–116, IEEE, 2018.

[10] S. Sáez, J. Real, and A. Crespo, "An integrated framework for multiprocessor, multimoded real-time applications," in *17th Ada-Europe International Conference on Reliable Software Technologies*, pp. 18–34, Springer, 2012.

[11] R. Inam, M. Sjödin, and R. J. Bril, "Mode-change mechanisms support for hierarchical freertos implementation," in *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation*, pp. 1–10, IEEE, 2013.

Performance Study of Object Tracking with Multiple Kalman Filters in Autonomous Driving Systems

Alessio Medaglini, Sandro Bartolini
Department of Information Engineering and Mathematics, University of Siena, Siena, Italy; email: {alessio.medaglini, sandro.bartolini}@unisi.it

Abstract

Object tracking is an important and central aspect of autonomous driving, as it underlies the obstacle detection and avoidance systems of any type of autonomous vehicles. A widely used method for tracking is based on Kalman filters, both for linear and non-linear cases, with different computational burden. Unfortunately, object tracking algorithms are computationally intensive, and they may not easily meet the efficiency and responsiveness requirements of real-time applications such as autonomous driving. This issue motivates ad-hoc investigations to speed up the computation and make Kalman filtering available even within limited computational power. This paper carry out a performance evaluation of a Kalman filter based object tracking system taken from a real tramway use-case, and aims at improving its performance efficiency by leveraging parallelization. In particular, this work analyzes the possibilities of execution parallelization on multi-core processors, proposing a target-specific optimization approach and comparing the obtained results, then summing them in general lessons learned. Our technique achieves up to 80% reduction of single frame processing time in the most crowded cases.

Keywords: Object Tracking, Kalman Filter, Parallel Processing, Autonomous Driving.

1 Introduction

Autonomous driving systems are revolutionizing the way we think about transportation. With the goal of making our roads safer and decreasing the amount of time we spend stuck in traffic, these systems use real-time information coming from a variety of sensors to guide vehicles without human input. One of the key challenges facing autonomous driving systems is the ability to accurately identify objects and obstacles in the scene. This requires sophisticated tracking algorithms that can process data in real-time, with an accuracy at the level required for safe and effective autonomous driving. Among the tools currently available for doing such kind of operation, many state-of-the-art methods for tracking are based on the use of Kalman filters (KF), which allow to precisely track the motion of objects even in non-linear cases. However, with recent developments in technology, the precision and accuracy of these object detection algorithms are no longer the only relevant factors. As we move towards embedding these systems into vehicles, the need for high-performance algorithms that can be executed on low-power embedded platforms still fitting the required performance targets becomes increasingly important. Accordingly, the main purpose of this work is to perform a performance analysis of a tracking system based on multiple KF instances, studying its behavior in different conditions, and the scalability of performance with respect to a variation of the parallelism level and in the number of target objects. Our aim is to increase the performance of tracking algorithms based on KF, to use them in a more effective way on low-power embedded platforms. The main contributions of this paper can be summarized as follows:

1. development of a use-case oriented optimization technique for low-dimensional Kalman filters
2. performance study on multi-core systems for Kalman filter based tracking algorithm
3. identification of general lessons to be exploited to increase performance and scalability in constrained architectures

2 Background and Application Use-Case

2.1 Kalman Filter Background

A Kalman filter [1] is an algorithm that uses a series of measurements observed over time to produce estimates of unknown variables that become more accurate over time. A KF works by means of two steps: *predict* and *update*. The *predict* step produces estimates of the current state variables by exploiting a system dynamic model and some uncertainties related to system evolution. Then, once the outcome of the next noisy measurement is received from sensors, these estimates are *updated* using a weighted average taking into account the reliability of the model and that of the sensors. To have a measurement to compare each prediction with, it is necessary to perform an *association* step between the two phases of the KF, to couple data coming from sensors with the different KFs in use. This is a critical aspect in many object tracking applications. The KF algorithm can be proven to be an optimal estimator under the assumption of Gaussian error distribution.

2.2 Tramway Use-Case

Our use-case is based on a real tramway Obstacle Detection and Avoidance System (ODAS), which collect data from multiple sensors, namely radar, lidar and camera. A tramway system is a challenging application for autonomous driving, since it moves inside urban traffic environment without being segregated from road traffic and pedestrians. For this reason, it is no so straightforward to discriminate whether an object on the track is to be considered a safety threat, making the object tracking module a fundamental one. For example a car could drive on the rail, in front of the tram, preceding it while driving the same direction. This tramway scenario is normal and should not cause any alert. In contrast, when the ODAS system detects a car or pedestrian whose future trajectory can be predicted to intersect the tram trajectory, it shall generate an alarm that allows taking appropriate corrective action to avoid a potential collision. Usually in this kind of context the KF is used to determine the future behavior of a *track*, associated with a real object. The KF's state is usually composed by six terms, which represent the position of the object and its velocity, along the three Cartesian directions. When allowed by sensors, it is possible to also measure the acceleration of objects to increase the object tracking precision at the cost of a heavier computational burden. In any case the state dimension of object tracking KFs is limited to no more than ten values, while the number of objects on the scene is typically from a few to several tens, and almost never beyond one hundred.

3 Related Works

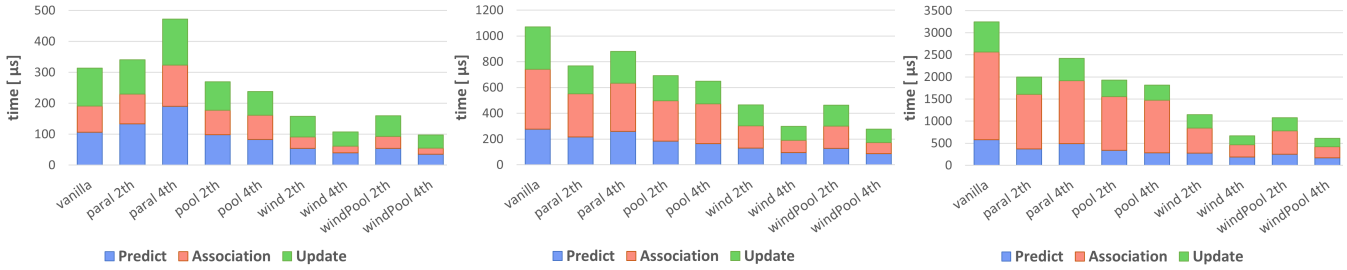
To increase the performance of object tracking by means of KFs, many techniques have been exploited. The most popular way of doing so is based on using the GPU to parallelize the operations between matrices, in particular inversions and multiplications. These techniques have been exploited in [2] for cases with large state dimensions, up to several thousand elements. Unfortunately, when the dimensions of the involved matrices are not large enough to fill all the GPU pipelines this kind of approach results in poor performance. In addition, data transfer between CPU and GPU can be a bottleneck in tracking application, since it requires data transfer at each step. To close this gap, there are some possible optimizations to reduce the memory data copying time [3] and improve performances for matrices of non-oversized dimensions [4]. Despite these improvements, the state dimension required to make these techniques effective remains significantly higher than that in our use-case reported in Sec. 2.2, making these solutions unsuitable for the application considered in this work. In contrast, on multi-core computational platforms, various attempts are made to parallelize the algorithm. Main efforts are aimed at optimizing the use of shared memory by reshaping the order of operations [5] or avoiding complex large-size matrix inversions through use of sparse matrices [6]. However, these attempts could not bring huge benefits in low state dimensional cases, since the size of matrices, being state-bound, is also small, and a few KB are therefore already enough to keep all the KF data in the cache. In [7] the particle tracking case is discussed, which involves KFs with low-dimensional states but applied on a huge scale. In their work parallelization is made through vectorization, a low-level technique based on SIMD (Single Instruction, Multiple Data) paradigm that uses a single instruction to perform the same operation on multiple data elements at the same time. Unfortunately, in the case of multi-sensor object tracking, irregular data patterns due to different processing paths required for each sensor's data and their different arrival times make it difficult to maintain the necessary dependencies between SIMD instructions. This paper explore several types of parallel implementations of KFs based on multi-core and multi-thread approaches to optimize performance in the object tracking use-case. These implementations have been developed from a real tramway use-case and are compared with each other and against it. Some of the difficulties encountered in KF algorithm parallelization are related to dealing with a small state size and to the presence of multiple decision and synchronization points in each iteration due to multi-sensor tracking. From the analysis performed, a target-specific parallelization proposal and some *lessons learned* emerge, that can be used to improve the performance of tracking via KF in generic applications.

4 Methodology

For each of the cases presented in the next section more tests were carried out, gradually increasing the level of parallelism and the number of target objects to track. In particular, we limit our test to an upper bound of 100 targets, i.e. 100 instances of KFs running simultaneously, since from real data of our use-case we discover that this can be considered as a stress-test of the system which is never reached. We start our analysis from a real implementation of tracking module taken from tramway ODAS use-case. We run our experiments on a NVIDIA Jetson TX2 board, the target platform of our real-world use-case, with 8 GB of LPDDR4 RAM memory, two 64-bit NVIDIA Denver CPU cores and four ARM Cortex-A57 CPU cores. These cores have L1 caches of 64Kb and 32Kb respectively, while L2 caches are 2Mb in both cases. The maximum frequency is 2 GHz for both, and there is only one thread available per core in each case. Although the fastest cores might be automatically used by the operating system, our experiments were conducted by setting affinity to pin threads to the highest performance cores. In each experiment, the timing results are obtained taking the median value of 100 execution. Regarding the results,we will evaluate them by considering as a metric the total processing time per frame divided among the various stages of tracking (prediction, association, update mainly). This metric will tell us which phase is the most critical during tracking, to understand the reasons for any performance bottlenecks. It will also allow us to compute the total processing time required to perform object tracking in the various configurations, highlighting which one maximizes the computational capacity of the system.

5 Experimental Evaluation

In this section, different strategies to improve tracking algorithm are tested with experiments carried out on some use cases collected on real trams. Three different portions of the dataset related to routes with different average number of objects on the scene, i.e., 20, 50 or 100 objects, were selected in order to evaluate the effectiveness and scalability of the



(a) 20 KFs running (b) 50 KFs running (c) 100 KFs running

Figure 1: Execution time measured per frame of different implementations with various number of KFs running simultaneously.

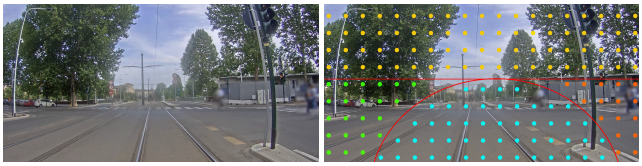
different techniques. The obtained results are shown in Fig. 1. We started from a straightforward *vanilla* implementation, which elaborates data from sensors sequentially, computing one after the other the three phases *predict*, *associate* and *update* that make up an iteration. While the *predict* phase is executed for all running KFs, *update* is required only for the associated ones. Therefore, each prediction is compared with data from sensors during the *associate* phase using a specific algorithm. This means that as the number of targets on the scene increases, the operations to be performed between successive frames grow proportionally. On the other hand, new and improved sensors with increasing frame rates are reducing the computation time available to the system. The strategies used to improve the KF performance within the ODAS tracking module are reported in the following sections.

5.1 Multi-threading

Multi-threading is a high-level technique that uses multiple threads of execution to perform different tasks simultaneously. Each thread runs on a separate CPU core and can access the same shared memory space. To realize it the KFs are implemented as an Array of Structures (AoS), in such a way it can be divided into chunks processed by different threads. Therefore, this technique divides the execution of N KFs among T threads so that each of them handles N/T threads. In particular, in our case, we parallelize the two phases *predict* and *update* by performing threads spawn at the beginning and threads join at the end of each of them. This introduces a large overhead, so that in cases with a limited amount of parallel work to be done no benefit is achieved, as can be seen in Fig. 1a. Moreover, in all other cases, increasing the number of threads from 2 to 4 degrades performance for the same reason. To overcome such limitation, we implemented multi-threading by using a *ThreadPool* class in charge of handling a pool of worker threads. When a thread completes its task, it returns to a queue of waiting threads and can be reused, avoiding the cost of creating a new thread for each task. Such kind of approach, especially for low-power platform, increase the performance when multiple threads are used, as can be seen from the 4-thread cases shown in Fig. 1. In any case, the speed-up achievable is still limited and does not scale proportionally to the number of threads employed. This is due to the operation of KFs, which performs two distinct phases interleaved by the association of new measures, and thus requires an amount of work that is too small to obtain an appreciable benefit from parallelization over multiple threads.

5.2 Thread-Window Approach

For this reason, starting from the multi-thread parallelism model, we explored a different way of parallelization based on some peculiarities of the use-case of interest. A key aspect of tracking is the associations between the active tracks, and thus the KFs, and the measurements coming from vehicle sensors. This phase is not easily parallelizable due to the need of performing a comparison between all objects in the scene and all possible KFs. This means that different threads managing different slices of the KFs vector need to communicate with each other, limiting performance scalability. The idea of dividing the various threads over disjointed, well-defined portions of the scene, stemmed from this observation. This way each thread would be almost completely independent of the others even during the *association* phase, and the necessity of reading predictions from each individual KF is removed. In fact, now this is only needed for objects within the same *window*. Furthermore, from observation of the data collected in the tramway use-case under analysis, it can be seen that displacements of objects over large distances between two successive frames are unlikely. This is due to the typical transmission rate of modern sensors, especially compared to tram speed in urban environments. This justifies some segregation between the various areas in front of the tram. Therefore, this strategy divide the scene into several sectors or *windows*, assigning all the KFs related to objects within each of them to a different thread. Likewise, the measurements arriving from the sensors are divided among the threads according to their location, making each window independent from the others.



(a) Tram FOV (b) Segmentation

Figure 2: Tram FOV segmentation for different threads

To partition the scene into windows, we relied on the specificity of autonomous driving in tramway use-case, resulting in the segmentation shown in Fig. 2. In this case, in fact, tracking is aimed at preventing obstacles from standing on the tracks in front of the tramway, putting passengers in hazard. For this reason, the scene area around the rail tracks is the most important, with the level of criticality gradually

decreases as we move away from them. We therefore chose to assign the upper band of the scene to a single thread, since it is likely to manage far less objects, while we divided the lower area into a main region around the rail tracks and two others at its edges.

5.3 Pros and Cons

As can be seen from charts in Fig. 1, in addition to improvements in the individual stages of the KF, this technique produce a remarkable improvement in the *association* phase, reducing timing up to seven times in the most crowded case. In fact, with a classical approach, the time related to this phase increases proportionally to the number of simultaneous KFs on the entire scene, whereas now it is limited to the size of the windows. This might suggest using smaller windows, especially in portion of the scene where the density of objects is greater. However, this would lead to an increased fragmentation, which would amplify a different drawback arising from object transitions between windows. In fact, when an object moves from one region to another, the KF associated with that object should be destroyed to be then re-created in the new window. This could produce some instants in which an object straddling two windows is not tracked. Although the number of windows is upper bounded by the maximum number of available threads/cores, which especially in embedded systems is usually not too high, this could significantly reduce the performance. To overcome this problem, our solution implements an overlap policy between adjacent windows. By setting an appropriate threshold we allow the thread that manages the KFs of one window to expand its bounds over a small portion of the neighboring window. Therefore, this technique allows threads to begin tracking objects just before they enter the assigned window, making transitions smooth and preventing object loosing. This may cause some overhead due to edge management policies, in relation to the number of windows used, and temporary duplication of some objects. We tested our technique by comparing the slice-parallel multithreaded case with the corresponding window version, using an overlap area between windows equal to 5% or 10% of the image size, which are far more than adequate values for our purpose. To stress the system, we always used the 4-thread parallel version with the maximum number of objects, i.e., 100 KFs, since this is the most critical case to handle. As can be seen from the graphs in Fig. 3, which show the results obtained with and without the use of *ThreadPool*, all versions with overlapping windows are significantly faster than the parallel sliced versions. In fact, there are performance improvements of at least 46% even in the maximum overlapping cases, which generate an overhead of 60% compared to the case without overlapping. We can conclude that the resulting overhead is negligible, since the portion of window area to be overlapped in order to find an object in advance is limited and the objects transitions between windows are sparse. Moreover, this technique reduces the number of objects handled by each thread, allowing full utilization of the cache locality of each core, and thus decreasing the computation time.

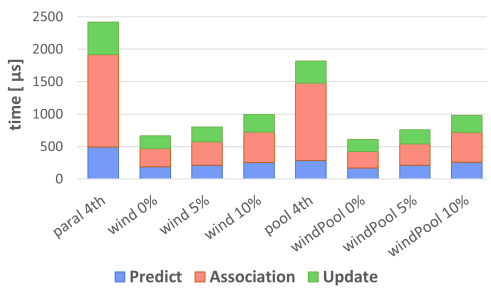


Figure 3: Performance of overlapping windows

6 Conclusion and Future Work

Object tracking is a central and underlying problem of autonomous vehicle guidance systems. This paper analyzed the possibilities of improving the performance efficiency of object tracking algorithms on embedded platforms by exploiting parallelization. We analyzed a realistic case from the tramway domain, testing various parallelization techniques and proposing one specifically intended for low-dimensional cases. The proposed solution is particularly optimized for the application considered in this work, but the presented idea can be applied to other situations where a limited number of low-size KFs are used. Below we list some heuristics and general rules learned:

- To take full advantage of embedded architectures, threads must be pinned to specific cores with higher computational power, thus maximizing performance.
- When the amount of parallel work to be done is limited, using a thread management class avoids the overhead of managing worker threads.
- Organize data structures and their size in memory according to the type of parallelization technique used, so as to facilitate access to data batches and fit the cache availability.
- Fully exploit the peculiarities of your use-case to divide the object tracking scenario into different sectors based on the purpose of the specific application task.

For the future, we plan to test our work on different use-cases and investigate how to develop parallelization techniques to take advantage of GPUs even in low-dimensional cases.

References

[1] R. E. Kálmán, “A new approach to linear filtering and prediction problems,” 1960.

[2] M.-Y. Huang, S.-C. Wei, B. Huang, and Y.-L. Chang, “Accelerating the kalman filter on a gpu,” 2011.

[3] A. Jarrah, “Optimized parallel architecture of kalman filter for radar tracking applications,” 2016.

[4] Z. Lin and D. Moore, “Gpu-based parallel kalman filter,” 2015.

[5] O. Rosén and A. Medvedev, “Efficient parallel implementation of a kalman filter for single output systems on multicore computational platforms,” 2011.

[6] O. Rosen and A. Medvedev, “Parallelization of the kalman filter for banded systems on multicore computational platforms,” 2012.

[7] D. H. C. Pérez and O. Awile, “An efficient low-rank kalman filter for modern SIMD architectures,” 2018.

Multi-criteria Analysis and Optimisation in the AMPERE Ecosystem

Sara Royuela, Adrian Munera, Eduardo Quinones
Barcelona Supercomputing Centre, Spain; email: {sara.royuela, adrian.munera, eduardo.quinones@bsc.es}@bsc.es
Tiago Carvalho, Luís Miguel Pinho, Mohammad Samadi
Instituto Superior de Engenharia do Porto, Portugal; email: {tdc, lmp, mmasa}@isep.ipp.pt
Tommaso Cucinotta, Gabriele Ara, Francesco Paladino
Scuola Superiore Sant'Anna, Italy; email: {tommaso.cucinotta, gabriele.ara, francesco.paladino}@santannapisa.it
Sergio Mazzola, Thomas Benz
ETH Zürich, Switzerland; email: {smazzola, tbenz}@iis.ee.ethz.ch

Abstract

The AMPERE project is developing the next generation of high-performance and energy efficient Cyber-Physical Systems supporting multi-criteria optimization. This paper provides the general overview of the AMPERE approach to analyse and optimise parallel real-time applications in heterogeneous platforms. More specifically, it details how parallel OpenMP programs are generated from AMALTHEA models, and the multi-criteria optimisation methodology, considering the fault-tolerance, time and energy properties of the targeted applications.

Keywords: Real-time Systems, Multi-criteria optimisation, AMPERE

1 Introduction

The growing computational demands of complex Cyber-Physical Systems (CPS) is hastening the introduction of parallel and heterogeneous architectures in domains with tight functional and non-functional requirements with respect to resilience, time and energy budgets, among other aspects. However, the parallel programming models, e.g. OpenMP, used to exploit parallelism multi-cores and accelerator devices are not compatible with the current Model-Driven Engineering (MDE) approaches used to develop CPS.

AMPERE strives to close the gap between the MDE techniques used in safety-critical automotive and railway systems and the parallel programming models used in high-performance systems by developing a complete software stack and development environment to help system developers leverage low-energy, highly-parallel, and heterogeneous systems in their development process, while fulfilling the non-functional constraints inherited from the cyber-physical interactions of safety-critical automotive and railway systems.

The paper is structured as follows. Section 2 presents a general overview of the AMPERE ecosystem and the flow used to analyse and optimise parallel applications considering the fault-tolerance, time and energy properties. The two main parts of the flow are provided next. Section 3 explains how OpenMP programs are generated from models, whilst Section 4 describes the multi-criteria optimisation approach.

2 The AMPERE project in a nutshell

AMPERE [1] is building a system design ecosystem and computing software to help system developers to leverage low-energy and highly-parallel and heterogeneous computation while fulfilling non-functional constraints.

2.1 The AMPERE ecosystem

One of the main challenges of the AMPERE project is to enable Model Driven Engineering (MDE) of CPS, accounting for parallelism and heterogeneity in high-performance embedded systems. As such, MDE tools provide the front-end to the entire AMPERE ecosystem (Figure 1). These tools include Domain Specific Modelling languages (DSML), e.g., AMALTHEA [2], which are used to describe the system in a modular and composable manner. Models are further annotated by system designers with the functional and non-functional requirements that determine how the system shall be generated. These annotations are key for the automatic optimization of the system of systems with respect to energy, timing guarantees, resilience, and heterogeneity [3].

Once the system has been modelled in AMALTHEA, a Synthetic Load Generator (SLG) [4] generates the corresponding source code, including OpenMP [5] annotations to exploit parallelism and heterogeneity. The source code is passed to an OpenMP compiler, for compilation. At this point, extensions provided in the OpenMP compiler allow for producing not only the binaries themselves, but also structured information of the system that is later used during the optimisation process [6] to ensure that the final system fulfils all requirements modelled in the DSML. The fundamental data structure generated as part of the structured information is the Task Dependency Graph (TDG) [7].

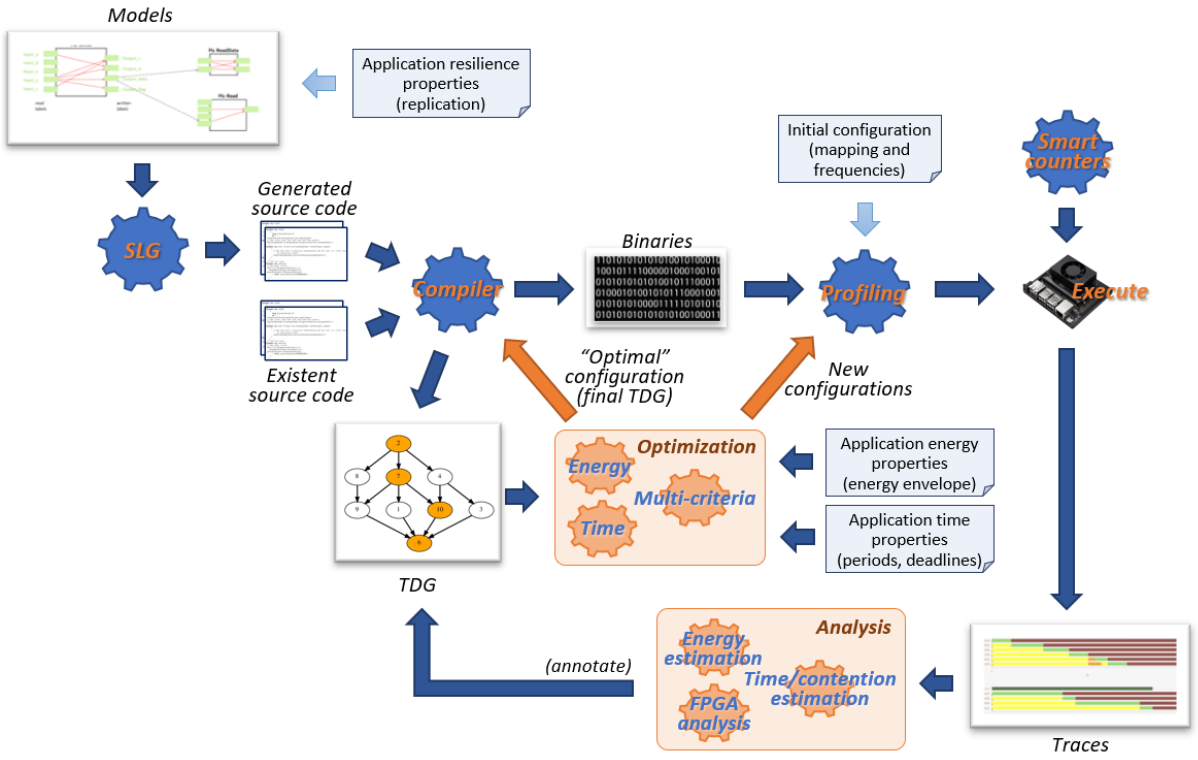


Figure 1: AMPERE software ecosystem flow.

The TDG provides the parallel structure of the different components of the system based on the dependencies described between the runnables of the AMALTHEA tasks, as outlined in the DSML. The TDG further contains meta-information that is used for the optimization (non-functional properties) annotated in the DSML towards which the system should be optimized. As part of the compilation process, the generated binary is profiled, and the information inserted in the TDG. As such, the TDG provides the necessary abstraction for determining the modelled requirements and dependencies of every task in the system, as well as wide information about the behaviour of each task as obtained through profiling.

2.2 The AMPERE analysis and optimisation flow

The analyses and optimization phases consist of multiple components operating in parallel: timing analysis and optimization, energy optimization, and scheduling. Heterogeneity and resilience techniques are implemented at the model and compiler levels, through the definition of function specializations (components which have multiple implementations, potentially for accelerators) and replication, respectively. This information is also included in the TDG and taken into consideration by the analyses.

Once the optimization phase has completed, it is either finished, i.e., all functional and non-functional requirements are guaranteed to be upheld, or another round of optimization is required. To that end, the AMPERE optimization relies on an optimization feedback loop that includes additional profiling information (red arrows in the figure).

The encoded information in the TDG allows for later use by the earlier components in the AMPERE pipeline, such that information in the model could either be updated, or warnings emitted to the MDE framework, and made available to the end user. At the end of the optimization pipeline, the TDG information can also be used to inject runtime hooks and configuration headers based on the optimization outcome into the generated source code. This enables actuation and monitoring of the non-functional requirements at runtime.

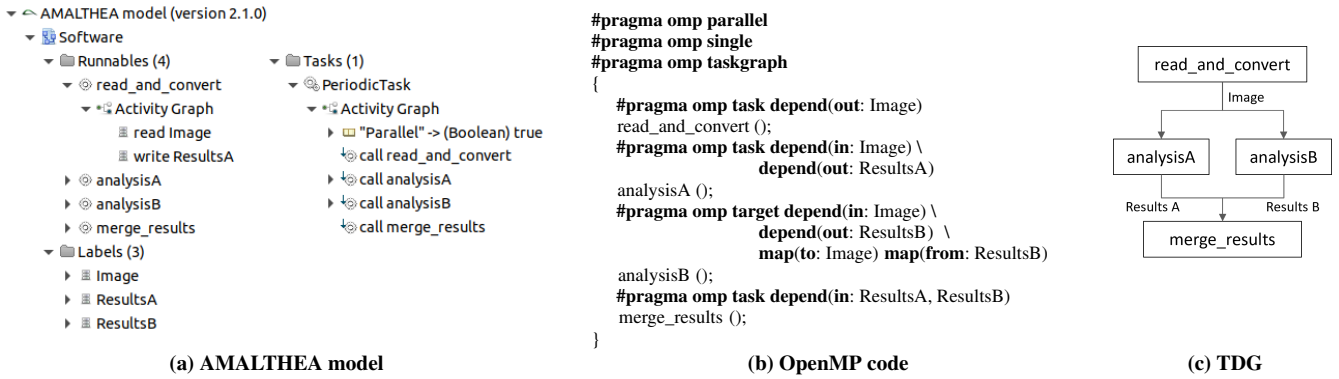
3 Model-to-code transformations

AMPERE defines model-to-code transformations targeting performance and fault tolerancein. These transformations are performed in two steps. First, AMALTHEA models are transformed into parallel code through the APP4MC SLG. Then, this code is analyzed and further transformed into a TDG to efficiently exploit the parallelism of the underlying processor architecture.

3.1 Performance

The synthesis tool included in the APP4MC framework processes AMALTHEA models by transforming runnables and tasks into C functions, and labels into global variables. In the frame of the AMPERE project, the SLG has been extended [8] to exploit concurrency among tasks no only with Pthreads for Linux systems, but also with ROS2 [9] primitives for ROS2 middleware communication. Moreover, extensions based on the OpenMP tasking model have been included to exploit parallelism within runnables from the same task.

Figure 2 illustrates the model-to-code transformation implemented in APP4MC in AMPERE, including a sample model in Figure 2a, the corresponding OpenMP code generated by the extended SLG in Figure 2b, and the TDG representing the OpenMP code in Figure 2c.



(a) AMALTHEA model **(b) OpenMP code** **(c) TDG**

Figure 2: Example of AMALTHEA to OpenMP transformation

A new custom property, named *Parallel*, triggers inter-runnable parallelism within AMALTHEA tasks (see igure 2a). For such tasks, the SLG wraps the associated code in a `parallel` directive followed by a `single` (see Figure 2b), to first spawn parallelism and second allow only one thread executing the inner region. Next, a taskgraph directive allows for optimizations towards predictability and performance [10]. Then, each runnable call is annotated with a tasking directive that depends on the processor defined for the runnable, i.e., a `task` directive for host runnables (AMPERE considers multi-core architectures) and a `target` directive for accelerator runnables (i.e., a GPU device). Finally, accesses to labels are used to define dependency clauses, transforming each read into an `in` dependency and each write into an `out` dependency. In the case of accelerated tasks, label accesses are also used to define the `map` clauses, which describe data movements between the host and the accelerator.

The code generated by the SLG is later compiled using an extended version of the LLVM compilation framework [11]. During this process, an extended version of the OpenMP tasking model [10] is used to replace the regions of OpenMP code that exploit the tasking model with a TDG (see Figure 2c). The TDG avoids the need for running the user code in order to instantiate and execute tasks, reducing time spent in context switching, and also enables optimizations at the runtime level that reduce contention due to accesses to shared resources (e.g., task queues) and overhead due to unnecessary computations (e.g., dependency ressolution). Besides enhancing the performance of the parallel orchestration, the TDG enables timing analysis techniques (see Section 4.1) for predictable execution. The TDG is described in a JSON format that is used as the interface between the different tools included in the multi-criteria optimization phase. Figure 3 illustrates a portion of the JSON corresponding to the example in Figure 2.

```
1  1: {
2      "ins": [ ],
3      "outs": [2,3]
4  },
5  2: {
6      "ins": [1],
7      "outs": [4]
8  },
```

```
1  3: {
2      "ins": [1] ,
3      "outs": [4]
4  },
5  4: {
6      "ins": [2,3],
7      "outs": [ ]
8  }
```

```
1: read_and_convert
2: analysisA
3: analysisB
4: merge_results
```

Figure 3: JSON format for describing TDG in Figure 2c.

3.2 Fault-tolerance

Fault-tolerance is addressed in AMPERE through software replication. The requirements for fault-tolerance are defined at the model level according to the Automotive Safety Integrity Level (ASIL), in the automotive use case, or the Safety Integrity Level (SIL), in the railway use case, of each component. Hence, runnables with an ASIL B or SIL 4 are defined with triple replication, while runnables with QM or SIL 0 are not replicated.

To express replication, the OpenMP `task` directive has been extended with the `replicated` clause [12]. This extension allows defining the number of replicas, the function used to check the results and the type of replication, with three different options: (a) *spatial*, which forces each replica and the original task to be executed in a different processor, allowing them to run in parallel, (b) *temporal*, which forces each replica and the original task to be executed in mutual exclusion among them, so they have to be sequentialized, and (c) *spatial_temporal*, which includes both cases.

The SLG has been extended to annotate tasks with a `replicated` clause when the ASIL B and SIL 4 levels are assigned to a runnable. The LLVM compiler has also been extended so it generates $n + 1$ tasks (where n is the replication level, i.e., 3 in AMPERE) when an annotated task is found. One of the tasks consumes the original data, while the rest consume copies of the data modified within the task to avoid race conditions. A synchronization task is inserted after the creation of the tasks, including as input dependencies all the tasks in the replication set, and inheriting as output dependencies those of the original task. Afterwards, a task performing the consolidation function is generated. This behavior is shown in Figure 4, where the application presented in Figure 2 defines $analysisB$ with ASIL B.

The replication mechanism has been optimized to support *MooN* optimizations, where M is the number of replicas that need to finish successfully out of a total of N replicas. A compilation flag is added to LLVM to enable this optimization and reduce the overhead of the replication.

4 Multi-criteria optimization

AMPERE defines a multi-criteria optimization phase, in which all components are co-operating to ensure that the

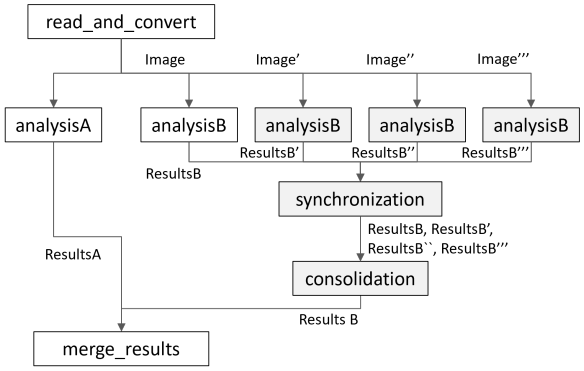


Figure 4: TDG when replicating $analysisB$.

system fulfils all requirements modelled in the DSML. During the analysis stage, two concurrent analysis are performed: a timing analysis (section 4.1) and a energy consumption analysis (section 4.2). During this stage, the analysis is performed for each TDG in the configuration file. The main purpose of the analysis stage is to annotate the TDGs with metrics that will be used in each target optimization phase. This stage does not change the configuration file, but it annotates the TDG files with new metrics. After the analysis, an optimization algorithm (section 4.3), is responsible to perform the multi-criteria optimization while each target optimization cannot find the most suitable configuration.

4.1 Timing analysis

The timing analysis phase of the optimization flow focuses on determining execution time metrics and the association of performance counters information, using measurement-based approaches, across all potential configurations and variants. The analysis is both for tasks as well as for the TDG as a whole.

For the tasks' execution time, the phase calculates the worst-case (WCET) and average execution times, which are then used in the optimization phase. It also determines fine grain information from performance counters (e.g. cache accesses, misses, etc.), which can be used for more detailed profiling of the applications.

Metrics related to a TDG use both existing information on the TDG and the new metrics calculated per task (the WCET of each task). The set of metrics outputted for the multi-criteria optimization flow includes volume, critical path length, potential maximum parallelism and average and worst-case makespan.

The volume corresponds to the aggregated WCET of all tasks within the TDG, whereas the critical path length signifies the overall cost of the path in the TDG that includes the longest route from the source task to the sink task.

The potential maximum parallelism serves as a metric that denotes the theoretical maximum parallelism achievable within a TDG, without considering any costs. It determines the highest number of tasks that could theoretically operate in parallel from all potential non-dependent siblings, even if they may not be executed simultaneously in practice.

The makespan represents the actual execution time of a TDG, spanning from the initiation of execution to its completion. This metric is pertinent when considering a specific task-to-thread mapping, as it provides the execution time of the TDG by taking into account the WCET of all tasks, given a certain number of available threads. The critical path length of a TDG can be regarded as the minimum duration the makespan of the TDG can take.

The timing analysis can be done also to reevaluate the TDGs, taking into account the additional information the optimization phase adds to the graph. It is also possible to use the analysis phase to optimize only for the time dimension, exploring different mapping algorithms, with an heuristic-based mapping approach [13].

4.2 Energy analysis

The aim of the energy analysis phase in the AMPERE multi-criteria optimization flow is to annotate the TDG with energy consumption information for each given task. Such energy consumption information relies on measurements from the profiling step embedded in the flow, as in figure 1. The energy annotations are then employed in the optimization step (section 4.3) to find the desired trade-off between energy consumption and execution time at the TDG level.

The AMPERE framework targets modern, heterogeneous, highly parallel systems. Measuring the energy consumption of a given workload running on these platforms is usually challenging for several reasons. Energy measurements require expensive external equipment, which impacts the scalability and flexibility of the system. Some platforms come equipped with built-in current sensors which can be used to estimate power consumption. However, analog sensors are slow with respect to the GHz regime of the digital hardware, and they can only provide coarse-grained measurements of the device sub-systems (e.g., entire CPU or GPU).

AMPERE's energy analysis is based on an approach to power modelling driven by the hardware performance monitoring counters (PMCs) of the target platform [14]. PMCs allow us to independently model the platform's power consumption at an arbitrary degree of granularity. Deriving directly from the digital domain, PMC-based power models expose high responsiveness, and match the workload execution in a reliable way, nevertheless impacting the optimization flow runtime with an extremely low overhead.

Thanks to a platform profile obtained in a one-time characterization step preceding the multi-criteria optimization, we calibrate our PMC-based power models to the desired target platform. Subsequently, we model each individual sub-system of the platform, at each one of its possible operating DVFS frequencies, with a set of representative PMCs coming from the platform characterization. The required PMCs are then sampled, in a low-overhead and non-invasive way, during the profiling in figure 1. Such measurements are employed in the energy analysis phase to estimate the average power and energy consumption of each task in the TDG.

Reliably supporting different devices and frequencies, the energy analysis step annotates each task's energy consumption

for each one of the task's functional specializations, and for each operating frequency of the platform.

AMPERE's energy analysis is fully automatized and flexible. Its data-driven nature requires minimal manual intervention: the best PMCs to model each sub-system of the target platform, at each frequency, are selected based on their correlation to their sub-system's power consumption. Additionally, requiring minimal architectural knowledge of the underlying hardware, the approach is easily extensible to additional devices.

4.3 Optimization approach

Starting from the annotated TDGs produced by the timing/energy analysis phases, the AMPERE workflow optimizer component [15] finds the system's optimal configuration. Considering the multi-criteria requirements of AMPERE, the developed tool supports two main optimization objectives.

The first objective is to find the system configuration that minimizes the target application's average energy consumption while preserving the system designer's timing constraints and, alternatively, maximizing the timing robustness of the application without exceeding its energy consumption budget.

The optimizer relies on a precise structural mathematical model of the target application, automatically derived from the TDG, that takes into account the resiliency requirements and the effects of selecting one or multiple heterogeneous hardware components of the target platform to perform (part of) the computations to find the optimal system configuration. The optimization is performed by applying mixed-integer quadratic constraint programming (MIQCP) and selecting one (or both) of the above objectives.

While the MIQCP formulation provides the requested optimality guarantees to generate the "optimal" configuration (final TDG, in fig. 1), it is also very complex, which may hinder its applicability to large-scale problems comprising more complex applications. For this reason, the optimizer also implements some simpler heuristic solvers that can be used to find sub-optimal configurations that may be fed back into the AMPERE loop or used as a starting point by the optimal MIQCP-based solver.

The final output of the optimization loop is the configuration of the target system in its entirety, including its multiple heterogeneous components (e.g., FPGA/GPU accelerators), and the placement of the individual runnables of each AMALTHEA task comprising the target application, including the choice between software and hardware implementation for tasks that can be hardware-accelerated, either for reaching the optimum (e.g., lowest average energy consumption) or because it is necessary to satisfy the application resiliency requirements.

5 Conclusions

This paper presented the approach used in the AMPERE project, to analyse and optimize the configuration of parallel real-time applications, in heterogeneous platforms, considering non-functional properties such as fault-tolerance, energy-efficiency and response time. The paper provides the project's multi-criteria optimization flow, which targets OpenMP parallel applications, presenting how each of the properties is considered in AMPERE ecosystem, and how the different dimensions are integrated in a single ecosystem.

6 Acknowledgments

This research has been co-funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 871669, in the context of the AMPERE project.

References

[1] E. Quiñones, S. Royuela, C. Scordino, P. Gai, L. M. Pinho, and L. Nogueira et al., "The ampere project: : A model-driven development framework for highly parallel and energy-efficient computation supporting multi-criteria optimization," in *2020 IEEE 23rd Intl. Symposium on Real-Time Distributed Computing (ISORC)*, pp. 201–206, 2020.

[2] Eclipse, "APP4MC." `https://www.eclipse.org/app4mc`, 2023.

[3] AMPERE, "D1.3. first release of the meta model-driven abstraction release," 2020.

[4] AMPERE, "D2.2. first release of the meta parallel programming abstraction and the single-criterion performance-aware." `https://ampere-euproject.eu/results/deliverables`, 2021. Accessed: 06-03-2023.

[5] OpenMP Arquitecture Review Board (ARB), "OpenMP Application Program Interface v5.2." `https://www.openmp.org/wp-content/uploads/OpenMP-API-Specification-5-2.pdf`, 2021.

[6] AMPERE, "D3.3. energy optimisation framework, predictable execution models and analysis, and software resilient techniques," 2022.

[7] AMPERE, "D2.3. programming model extensions and the multi-criteria performance-aware component." `https://ampere-euproject.eu/results/deliverables`, 2023.

[8] AMPERE, "Extended APP4MC SLG." `https://gitlab.bsc.es/ampere-sw/wp2/amalthea`, 2023.

[9] Open Source Robotics Foundation (OSRF), "ROS2." `https://github.com/ros2`, 2023.

[10] C. Yu, S. Royuela, and E. Quiñones, "Taskgraph: A low contention openmp tasking framework," *arXiv preprint arXiv:2212.04771*, 2022.

[11] AMPERE, "Extended LLVM 16.0." `https://gitlab.bsc.es/ampere-sw/wp2/llvm`, 2023.

[12] A. Munera, S. Royuela, and E. Quiñones, "Fault-tolerant applications through openmp," in *Proceedings of the 10th International BSC Severo Ochoa Doctoral Symposium*, 2023.

[13] M. S. Gharajeh, S. Royuela, L. M. Pinho, T. Carvalho, and E. Quiñones, "Heuristic-based task-to-thread mapping in multi-core processors," in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–4, IEEE, 2022.

[14] S. Mazzola, T. Benz, B. Forsberg, and L. Benini, "A data-driven approach to lightweight dvfs-aware counter-based power modeling for heterogeneous platforms," in *Embedded Computer Systems: Architectures, Modeling, and Simulation: 22nd International Conference, SAMOS 2022, Samos, Greece, July 3–7, 2022, Proceedings*, pp. 346–361, Springer, 2022.

[15] T. Cucinotta, A. Amory, G. Ara, F. Paladino, and M. D. Natale, "Multi-criteria optimization of real-time dags on heterogeneous platforms under p-edf," *ACM Trans. Embed. Comput. Syst.*, apr 2023. Just Accepted.

Attack Scenarios Generation Algorithm Based on Discrete Event System Formalism

Alexis Raynaud
Léonard de Vinci Pôle Universitaire, Research Center, France; email: alexis.raynaud@edu.devinci.fr
Théo Serru
ETIS laboratory - UMR8051, and Airbus Protect, France; email: theo.serru@ensea.fr
Nga Nguyen
Léonard de Vinci Pôle Universitaire, Research Center, France; email: nga.nguyen@devinci.fr

Abstract

To help automatize the security risk assessment process of Cyber-Physical Systems (CPS), we propose a tool based on Discrete Event Systems (DES) to model the architecture and the behavior of CPS in the presence of cyberattacks. Then, we present a lightweight algorithm to generate all the attack scenarios threatening a system, i.e. the sequences of attacks leading to a critical state (e.g. loss of control, collision, etc.). This kind of generation being prone to combinatorial explosion, our algorithm embeds state-space reduction capabilities focused on the specificities of cyber-physical attacks. Finally, we illustrate the performance of our algorithm on a case study: the navigation system of an autonomous vessel. This work can be seen as an alternative to heavy tools expressed in specific languages. It is open source and aims to give a good compromise between expressiveness, modeling time and computational power.

Keywords: Cyberattack, Attack Scenario, Cyber-Physical System, Discrete Event System, Model-Based Security Analysis

1 Introduction

Cybersecurity is a crucial issue in today's world, as the boundary between the physical and digital world is becoming increasingly tenuous. Cyberattacks can have direct effects on the safety of systems and even lead to the loss of human lives, mainly with Cyber-Physical Systems (CPS) mixing digital and mechanical components. With the expansion of connected devices, CPS offer an unprecedented growing attack surface. Engineers must therefore perform a security risk analysis to identify the impacts of cyber-threats and proactively protect systems. Formal models applied to cybersecurity, or model-based security, aim to protect dependability and security attributes such as safety, maintainability, availability, confidentiality, integrity, etc. in the event of attacks.

However, current approaches often lack of tools that are essential to support automated security verification and validation as well as to ensure that a certain level of security is maintained over time [1, 2]. Manual approaches are difficult, if not impossible, to analyze large systems, which emphasizes the need for tools and automation to enhance system security. Many formalisms have been developed in the literature to perform model-based assessment and overcome these limitations. Among them, attack trees [3,4], attack graphs [5,6,7], Markov processes [8, 9], Petri Nets [10, 11] are the most famous and used. In addition, several authors applied less common methods and tools to automatize the cybersecurity assessment, e.g. AADL in [12], Alloy in [13] and SysML in [14]. The popularity of this field led to different surveys in [1, 2, 15]. These approaches are not yet satisfying some industrial needs such as: providing an exhaustive generation of all the attack scenarios for large CPS models, reducing state-space explosion with heuristics based on security specificities and handling reconfiguration (all at once).

To satisfy these requirements, an experimental approach was proposed in [16, 17] to model the behavior of CPSs exposed to cyberattacks using Discrete Event System (DES) simulation. DES models can be used to evaluate the effectiveness of security measures and to design systems that are resilient to cyberattacks. The tool used for attack scenarios generation in these works is SimfiaNeo [18], an industrial software for safety analysis for which system engineers should have knowledge on the AltaRica formal language [19, 20]. AltaRica is a very expressive language and SimfiaNeo is consequently quite heavy to use.

The objective of this paper is thus to develop a lightweight and open-source algorithm for generating attack scenarios using the DES formalism. To face the state explosion problem, cutoff criteria such as the length of the attack path and the footprint that takes into account the dependency between attack steps have been implemented. To illustrate the applicability of this approach, a case study of an autonomous ship has been proposed. Using the algorithm, we are able to generate relevant sequences of attacks leading to critical states in a reasonable execution time.

The rest of the paper is organized as follows. Section 2 presents our attack scenario generation algorithm. Experimental results obtained with the autonomous navigation system in terms of generated sequences, minimal cutoffs, and

computing complexity are explained in Section 3. Section 4 discusses about the results and proposes some perspectives for the future works.

2 Attack Scenario Generation Algorithm

2.1 System Model

The objective of our algorithm is to generate attack scenarios for CPS. First, we model the system as a DES (see [21] for a definition), i.e. a five-tuple $\langle V, E, T, s_0, CS \rangle$. It is composed of components defined with variables $v \in V$ representing security attributes such as $integrity$, $availability$ or the presence of an attacker in the component ($attacker\_presence$). Variables take their values over a finite set of symbolic constants. The evolution of system's states depends on the events (set E) occurring in the system. Then, we describe the progress of the attacker within the system and the effects of the different attacks on the components by changing the value of these variables with transitions (set T). A transition from one state to another due to an attack is labeled by an event corresponding to an action carried out by the attacker or to an internal reconfiguration of the system (e.g. activate a fail-safe mode, isolate a component, etc.). s_0 represents the initial state and CS the set of critial states of the system.

In terms of modeling, the formalism is very permissive. One can add components, events into the model without the need to re-build it. However, the more refined is a model, the more combinatorial it will suffer. To this end, we choose a "system" level of abstraction, similar to the one used in model-based safety analyses. Components and subsystems are modeled with threats identified by the analyst during the threat modeling process. Security experts can also use publicly available databases that have a similar level of abstraction (e.g. the Common Vulnerabilities and Exposures [22], MITRE ATT&CK® [23], etc.). It is important to note that unknown attacks can be added to the model, as we can easily model the consequences and observe the behavior of the system.

Algorithm 1 is a breath-first search algorithm with state-space reduction capacities. It has five input parameters: the initial state of the system, the set of events, the set of critical states, and two cutoffs – the maximal length for a scenario to be generated and the maximum footprint of this scenario. Thus, the exploration of every scenario will dynamically stop depending on the length or the footprint. The output is the list of all scenarios leading to a critical state of the system.

2.2 State-space Reduction

Having an exponential complexity in this type of algorithm is expected, since we need to go through the set of all possible scenarios to find ones that end in a critical state. Let $|E|$ be the number of events, the number of scenarios whose length is equal to $|E|$ is $|E|!$. In addition, let $|V|$ be the number of variables of the system and supposing that each variable can only have a Boolean value, the number of possible states is $2^{|V|}$. Therefore, applying cutoffs is the only practical way of generating exploitable results in a given computation budget. In addition to max length, we have applied $footprint$, another cutoff introduced by Serru *et al.* [17] in order to reduce the state-space explosion. This cutoff is based on the heuristic that cyberattacks are intentional and directed towards a goal.

Algorithm 1 Algorithm to Generate Attack Scenarios

Input: Initial state of the system: $initialState$
Input: Set of events: $Events$
Input: Set of critical states: $CriticalStates$
Input: Maximum length: $MaxLength$
Input: Footprint threshold: $FootprintThreshold$
Output: List of critical sequences: Γ

```
workingList = [];
Γ = [];
firstEvents ← GetEvents(initialState, Events);
for event ∈ firstEvents do
    scenario = [event];
    workingList.append(scenario);
end for
while ∀scenario ∈ workingList, length(scenario) < MaxLength
  and footprint(scenario) < FootprintThrehold do
    newWorkingList = [];
    for scenario ∈ workingList do
        actualState ← GetState(scenario);
        if IsFinal(actualState, CriticalStates) then
            Γ.append(scenario);
            workingList.remove(scenario);
            break;
        end if
        possibleEvents ← GetEvents(actualState, Events);
        if possibleEvents == ∅ then
            workingList.remove(scenario);
        end if
        for event ∈ possibleEvents do
            newScenario = scenario;
            newScenario.append(event);
            newWorkingList.append(newScenario);
        end for
    end for
    workingList ← newWorkingList;
end while
return Γ
```

It introduces a weight to prioritize the more direct scenarios over the ones with attacks executed in a random order. By setting the right $footprint$ for the generation, the number of scenarios explored can be drastically reduced [21], as the exploration stops when the cutoff is reached. Moreover, the scenarios obtained tend to represent better the willfulness of the attacker. Proofs of the soundness of this criterion are given in [17].

Our algorithm has been implemented in Python and is available on GitHub [24]. The different parameters of the algorithm such as max length and footprint values are initialized in a configuration file. The description of the components (name, connected components and attributes) and events is specified in JavaScript Object Notation (JSON), a lightweight data-interchange format. We chose this type of file for its ease of use and clarity.

3 Case Study

We will now apply our algorithm to a concrete case study: the navigation system of an autonomous ship. See [16] for a deeper description of the case study.

3.1 Architecture

This system is composed of the following components: a satellite module (SAT), the Shore Control Center (SCC – on-ground station), the Autonomous Identification System (AIS – for sea monitoring), the Electronic Chart Display and Information System (ECDIS), a Global Positioning System (GPS), a RADAR and the Autonomous Navigation System (ANS).

A representation of the connections between the components is given in Figure 1. The components that can receive information from the outside are the SAT, SCC, and GPS. To model

the system, we have created seven components corresponding to the components mentioned above. We represent their security properties with variables, such as availability, integrity, role and the value of the component's output flows. There are a total of 51 variables embedded in these 7 components.

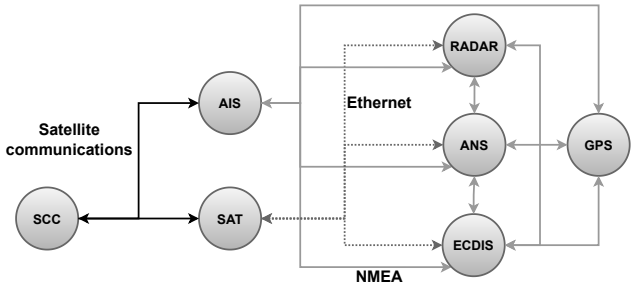


Figure 1: Architecture of the Vessel's Navigation System

3.2 Events

In the model, we consider 21 threats (from [16]), represented as events gathered in Table 1. Each attack is associated with a guard condition necessary to carry out the attack and an instruction that describes the variables modified by the attack and their associated new values. For example, the "privilege_escalation" attack on the AIS can be carried out if the role of the AIS is "user" and the event will result in "root" role. In the domain file, "user" is associated with the value 1 and "root" with the value 2. The condition of this attack is therefore: $AIS\_role = 1$ and the consequence of this event is: $AIS\_role \leftarrow 2$.

Table 1: Events Modeled in the Autonomous Vessel Case Study

Component	Events
ECDIS	Attacker access ECDIS, attacker exploits Apache vulnerability, attacker tempers with maps, attacker delete maps, download virus during chart update, lateral movement from ECDIS to radar via Ethernet
AIS	Malware deploys on AIS, privilege escalation, lateral movement to ECDIS, attacker send false information to ANS
ANS	Ship deviates from route, loss of incoming signal
SCC	Corrupted SCC sends malicious package to AIS
RADAR	Attacker accesses radar, attacker deletes radar targets, radar spoofing, attacker deactivates radar
GPS	GPS jamming, GPS spoofing
SAT	Receive email with infected payload, infected payload transferred to ecdis

3.3 Critical States

The first goal of the attacker is to deviate the ship from his original route. The second one is to maximize the damages, e.g. to sink the ship. In this case, the attacker will have to achieve two sub-objectives: to take control of the vessel and to disable the anti-collision system. The unwanted states are defined in the initialization file. Then, we can run the algorithm to explore, in a breath-first manner, all the sequences of events until the critical state is reached, or the cutoff is exceeded.

3.4 Results

The scenarios from this case study are generated as lists of events. For more readability, the algorithm renders two types of results: scenarios in text format and in graphs (each graph representing a single scenario). The graph format is illustrated in Figure 2. Here we see the attacker must follow two subsequences of events to achieve the goal. The footprint is here of great help to avoid generating the full shuffle between the sub-sequences.

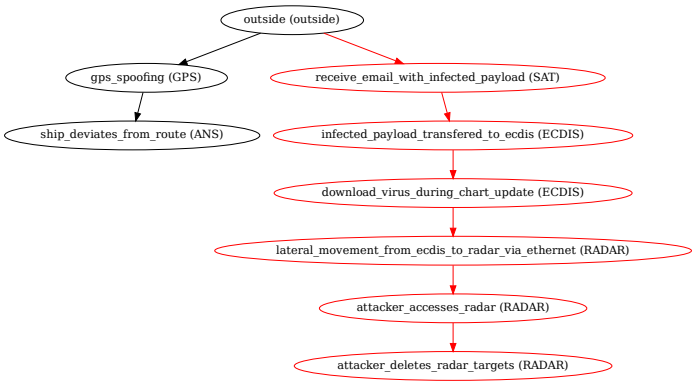


Figure 2: Attack Scenario with $footprint$ = 2

Table 2 represents the number of scenarios found as well as the execution time needed to explore the model according to the chosen maximum length of the scenarios. All experiments have been carried out using a CPU AMD Ryzen 5 3500U, 2.10 GHz, and 8.00Go of RAM. These results were generated without taking into account the $footprint$ cutoff. The longest scenarios have a length of 13 distinct events. In the case of an exhaustive generation (without length limitation), 293 attack scenarios are found.

Table 2: Results for the 2^{nd} Objective Without $footprint$

$MaxLength$	Number of scenarios	Execution time (sec)
8	9	0.85
9	21	1.38
10	31	2.19
12	173	7.71
13	293	12.69

It is important to note that we have an exhaustive generation of events. Many scenarios have parasitic events that do not help to achieve the critical state but have their guard activated at some point in the scenario.

3.5 Illustration of the Footprint

To test the effectiveness of the $footprint$, we performed several scenario generations without length limit, with only the maximal $footprint$ restriction. Table 3 shows that, by setting the right $footprint$, the execution time is drastically reduced. For the attacker's 2^{nd} objective, the scenarios are composed of at least two different sub-sequences. Setting the $footprint$ to 2 will give only the direct scenarios without unnecessary events. Thus, the usefulness of the $footprint$ is twofold. It allows to reduce the execution time and also to present clearer and more relevant results. With this prioritization, the analysts can: fix the identified sequences by applying security

measures, update the model and run simulations until the critical sequences are unlikely enough to be acceptable.

Table 3: Results for the 2^{nd} Objective with $footprint$ and without Length Limit

$Footprint$	Number of scenarios	Execution time (sec)
1	0	0.06
2	10	0.41
Without	293	11.35

We validated our results with the sequence generation of SimfiaNeo to ensure that they are the same. With this objective and a footprint $F = 2$ we obtained the same 10 sequences.

4 Conclusion

We proposed in this paper a model-based approach that describes CPS as an abstract model with discrete events representing attacks that can modify the state of the system. An algorithm that generates automatically the attack scenarios from an initial state to critical ones has been implemented, with max length and footprint as cutoff criteria to reduce the combinatorial number of critical sequences. This lightweight implementation takes as input the topology of the system, the list of all possible events with their pre- and post-conditions as well as the initial and critical states configuration, in JSON format. It computes different attack paths leading to these critical states, and displays them in either a text form or a graphical form.

The results obtained from a real case study on an autonomous vessel's navigation system are equivalent to those achieved with a much more complex industrial software who demands modeling skills with the AltaRica language. The Python code could be easily modified to add quantitative aspects on security metrics, such as the cost or the probability of an attack, in order to rank the generated sequences according to their relevance.

References

[1] P. Nguyen, S. Wang, and T. Yue, "Model-Based Security Engineering for Cyber-Physical Systems: A Systematic Mapping Study," *Information and Software Technology*, vol. 83, Nov. 2016.

[2] J. Geismann and E. Bodden, "A systematic literature review of model-driven security engineering for cyber–physical systems," *Journal of Systems and Software*, vol. 169, p. 110697, Nov. 2020.

[3] J. D. Weiss, "A System Security Engineering Process.," in *National Computer Security Conference*, vol. 249, pp. 572–581, 1991.

[4] A. Tantawy, S. Abdelwahed, A. Erradi, and K. Shaban, "Model-based risk assessment for cyber physical systems security," *Computers & Security*, vol. 96, p. 18, Sept. 2020.

[5] M. Dacier, *Vers une évaluation quantitative de la sécurité informatique*. phdthesis, Institut National Polytechnique de Toulouse - INPT, Jan. 1994.

[6] X. Ou, S. Govindavajhala, and A. W. Appel, "MulVAL: A logic-based network security analyzer," in *Conference on USENIX Security Symposium - Volume 14*, (USA), p. 8, July 2005.

[7] Z. Ye, Y. Guo, and A. Ju, "Zero-Day Vulnerability Risk Assessment and Attack Path Analysis Using Security Metric," in *Artificial Intelligence and Security*, vol. 11635, pp. 266–278, 2019.

[8] N. R. Pokhrel and C. P. Tsokos, "Cybersecurity: A Stochastic Predictive Model to Determine Overall Network Security Risk Using Markovian Process," *Journal of Information Security*, vol. 08, no. 02, pp. 91–105, 2017.

[9] A. B. Aissa, I. Abdalla, L. F. Hussein, and A. Elhadad, "A Novel Stochastic Model For Cybersecurity Metric Inspired By Markov Chain Model And Attack Graphs," *International Journal of Scientific & Technology Research*, vol. 9, no. 03, p. 7, 2020.

[10] T. M. Chen, J. C. Sanchez-Aarnoutse, and J. Buford, "Petri Net Modeling of Cyber-Physical Attacks on Smart Grid," *IEEE Transactions on Smart Grid*, vol. 2, pp. 741–749, Dec. 2011.

[11] R. Mitchell and I.-R. Chen, "Modeling and Analysis of Attacks and Counter Defense Mechanisms for Cyber Physical Systems," *IEEE Transactions on Reliability*, vol. 65, pp. 350–358, Mar. 2016.

[12] M. Ibrahim, Q. Al-Hindawi, R. Elhafiz, A. Alsheikh, and O. Alquq, "Attack Graph Implementation and Visualization for Cyber Physical Systems," *Processes*, vol. 8, p. 12, Jan. 2020.

[13] E. Kang, S. Adepu, D. Jackson, and A. P. Mathur, "Model-Based Security Analysis of a Water Treatment System," in *International Workshop on Software Engineering for Smart Cyber-Physical Systems*, (Austin, TX, USA), pp. 22–28, 2016.

[14] L. Li, *Safe and secure model-driven design for embedded systems*. Theses, Université Paris-Saclay, Sept. 2018.

[15] H. S. Lallie, K. Debattista, and J. Bal, "A review of attack graph and attack tree visual syntax in cyber security," *Computer Science Review*, vol. 35, p. 47, Feb. 2020.

[16] T. Serru, N. Nguyen, M. Batteux, and A. Rauzy, "Modeling cyberattack propagation and impacts on cyber-physical system safety: An experiment," *Electronics*, vol. 12, no. 1, 2023.

[17] T. Serru, N. Nguyen, M. Batteux, and A. Rauzy, "Minimal critical sequences in model-based safety and security analyses: Commonalities and differences," *ACM Trans. Cyber-Phys. Syst.*, May 2023.

[18] M. Machin, L. Sagaspe, and X. de Bossoreille, "SimfiaNeo, Complex Systems, yet Simple Safety," in *Embeded Real Time Software and System conference*, (Toulouse, France), p. 4, Feb. 2018.

[19] M. Boiteau, Y. Dutuit, and A. Rauzy, "The AltaRica Data-Flow Language in Use: Modeling of Production Availability of a MultiStates System," *Reliability Engineering & System Safety*, vol. 91, pp. 747–755, July 2006.

[20] T. Prosvirnova, *AltaRica 3.0: a Model-Based Approach for Safety Analyses*. PhD thesis, École Polytechnique, 2014.

[21] T. Serru, N. Nguyen, M. Batteux, A. Rauzy, R. Blaize, L. Sagaspe, and E. Arbaretier, "Generation of Cyberattacks Leading to Safety Top Event Using AltaRica: an Automotive Case Study," in *Congrès de Maîtrise des Risques et de Sûreté de Fonctionnement*, (Paris-Saclay, France), Oct. 2022.

[22] M. Corporation, "Common Vulnerability and Exposure, https://cve.org."

[23] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "MITRE ATT&CK™ : Design and Philosophy," tech. rep., MITRE, July 2018.

[24] A. Raynaud, "https://github.com/Alexis-Raynaud/attack_scenarios_generation."

National Ada Organizations

Ada-Belgium

attn. Dirk Craeynest
c/o KU Leuven
Dept. of Computer Science
Celestijnenlaan 200-A
B-3001 Leuven (Heverlee)
Belgium
Email: Dirk.Craeynest@cs.kuleuven.be
URL: www.cs.kuleuven.be/~dirk/ada-belgium

Ada in Denmark

attn. Jørgen Bundgaard

Ada-Deutschland

Dr. Hubert B. Keller CEO
ci-tec GmbH
Beuthener Str. 16
76139 Karlsruhe
Germany
+491712075269
Email: h.keller@ci-tec.de
URL: ada-deutschland.de

Ada-France

attn: J-P Rosen
115, avenue du Maine
75014 Paris
France
URL: www.ada-france.org

Ada-Spain

attn. Sergio Sáez
DISCA-ETSINF-Edificio 1G
Universitat Politècnica de València
Camino de Vera s/n
E46022 Valencia
Spain
Phone: +34-963-877-007, Ext. 75741
Email: ssaez@disca.upv.es
URL: www.adaspain.org

Ada-Switzerland

c/o Ahlan Marriott
Altweg 5
8450 Andelfingen
Switzerland
Phone: +41 52 624 2939
e-mail: president@ada-switzerland.ch
URL: www.ada-switzerland.ch

Ada-Europe Sponsors

Ada Edge

27 Rue Rasson
B-1030 Brussels
Belgium
Contact:Ludovic Brenta
ludovic@ludovic-brenta.org

AdaCore

46 Rue d'Amsterdam
F-75009 Paris
France
sales@adacore.com
www.adacore.com

AdaLabs
innovate.all

506 Royal Road
La Caverne, Vacoas 73310
Republic of Mauritius
Contact: David Sauvage
david.sauvage@adalabs.com

ADA LOG

2 Rue Docteur Lombard
92441 Issy-les-Moulineaux Cedex
France
Contact: Jean-Pierre Rosen
rosen@adalog.fr
www.adalog.fr/en/

Deep Blue Capital

Jacob Bontiusplaats 9
1018 LL Amsterdam
The Netherlands
Contact: Wido te Brake
wido.tebrake@deepbluecap.com
www.deepbluecap.com

Ellidiss Technologies

24 Quai de la Douane
29200 Brest, Brittany
France
Contact: Pierre Dissaux
pierre.dissaux@ellidiss.com
www.ellidiss.com

KONAD
Software for Control and Administration

In der Reiss 5
D-79232 March-Buchheim
Germany
Contact: Frank Piron
info@konad.de
www.konad.de

PTC®
Developer Tools

3271 Valley Centre Drive,Suite 300
San Diego, CA 92069
USA
Contact: Shawn Fanning
sfanning@ptc.com
www.ptc.com/developer-tools

SYSADA

Enterprise House
Baloo Avenue, Bangor
North Down BT19 7QT
Northern Ireland, UK
enquiries@sysada.co.uk
sysada.co.uk

systerel
Safe real-time solutions

1090 Rue René Descartes
13100 Aix en Provence
France
Contact: Patricia Langle
patricia.langle@systerel.fr
www.systerel.fr/en/

Tid rum

Tiirasaarentie 32
FI 00200 Helsinki
Finland
Contact: Niklas Holsti
niklas.holsti@tidorum.fi
www.tidorum.fi

WhiteElephant

Beckengässchen 1
8200 Schaffhausen
Switzerland
Contact: Ahlan Marriott
admin@white-elephant.ch
www.white-elephant.ch


Ada
europe