ADA USER JOURNAL

Volume 38

Number 4

December 2017

Contents

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Unless the authors state explicitly otherwise, submission of an article is taken to imply that it represents original, unpublished work, not under consideration for publication elsewhere.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

This issue of the Ada User Journal publishes the Proceedings of the "Workshop on Challenges and New Approaches for Dependable and Cyber-Physical System Engineering", co-located with Ada-Europe 2017, in Vienna, Austria. This workshop brought together industry and research participants, for a full-day discussion on dependability and critical issues of Cyber-Physical Systems (CPS), a good complement to the already rich program of the Ada-Europe conference.

The workshop program included presentations from academia and industry, on multiple interrelated aspects of dependability of CPS. The proceedings reflect part of this content, starting with a position paper on how an ecosystem is being created around two large European projects (Semi40 and Productive 4.0) to boost the industrial digitalization, by authors from Infineon Technologies, Austria. Then, the proceedings provide two papers on the implications of autonomous vehicles. In the first, a group of authors from KTH, Sweden, presents an analysis of existent safety-related challenges for fully autonomous vehicles, based on interviews with professionals. In the second paper, authors from ECE Paris and Université de Versailles Saint-Quentin-en-Yvelines, France, provide insights in the cognition of driving context, in semi-autonomous vehicles.

The workshop proceedings continue with a paper coming from Softeam, France, discussing the SysML-based profile being developed in the INTO-CPS project, for model-based design of Cyber-Physical Systems. Afterwards, a paper from the University of Petroleum and Energy Studies, India, proposes a new approach for information dissemination in Vehicular Ad-hoc Networks. The workshop proceedings conclude with a discussion paper, from authors coming from CEA LIST and Softeam, France, on the challenges faced by future transportation systems.

Adding to this strong set of contributions, the issue also continues the publication of contents of the industrial track of Ada-Europe 2017, with contributions from Stéphane Carrez, from France, describing Ada Embedded Network, a network stack for small embedded Ada applications on ARM platforms, and J. Germán Rivera, from the USA, presenting an approach for runtime data protection using a memory protection unit for Ada bare-metal software.

Apart from the technical contents, and as usual, the issue provides the News Digest, Calendar and Forthcoming Events sections, provided by the News and Events Editors, respectively Jacob Sparre Andersen and Dirk Craeynest. I would like to give a special mention to the Forthcoming Events section, providing information on the return of the Ada Developer Room at FOSDEM and the International Real-Time Ada Workshop (IRTAW), taking place in Benicàssim, Spain, 18-20 April 2018.

And obviously, a special note also to the forthcoming Ada-Europe 2018 conference, which returns to Portugal, taking place this time in Lisbon, 18-22 June 2018; we are looking forward to see the Ada community gathering there for the usual high-quality Ada-Europe week!

Luís Miguel Pinho
Porto
December 2017
Email: AUJ_Editor@Ada-Europe.org

Post scriptum: This issue is reaching the reader with some delay, due to difficulties in the final editing process. Please accept my apologies for this delay. And I take the opportunity to challenge the reader - to consider joining the team of volunteers that produces the Ada User Journal!

Ada User Journal

Call for Contributions

Topics: **Ada, Programming Languages**, **Software Engineering Issues** and **Reliable Software Technologies** in general.

Contributions: **Refereed Original Articles**, **Invited Papers**, **Proceedings** of workshops and panels and **News and Information** on Ada and reliable software technologies.

More information available on the
Journal web page at

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

Online archive of past issues at http://www.ada-europe.org/auj/archive/

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**

Quarterly News Digest

Jacob Sparre Andersen

Jacob Sparre Andersen Research & Innovation. Email: jacob@jacob-sparre.dk

Contents

Ada-related Organisations

2017 ACM SIGAda Awards - Call for Nominations

From: Ricky E. "Ranger" Sward
Date: Sat, 23 Sep 2017 11:18:37 -0000 (UTC)
Subject: 2017 SIGAda Awards - Call for Nominations

Dear Members of the Ada Community:

We welcome your nominations for the 2017 Robert Dewar Award for Outstanding Ada Community Contributions and the 2017 ACM SIGAda Distinguished Service Award.

This year's SIGAda Award winners will be announced via the SIGAda email lists and Linked-In page, since there is no workshop this year.

Award nominations are due on October 20th.

The ACM SIGAda Awards recognize individuals and organizations that have made outstanding contributions to the Ada community and to SIGAda. The two categories of awards are:

(1) Robert Dewar Award for Outstanding Ada Community Contributions

-- For broad, lasting contributions to Ada technology & usage.

(2) ACM SIGAda Distinguished Service Award

-- For exceptional contributions to SIGAda activities & products.

If there are individuals who you feel have made contributions that satisfy these criteria, please consider nominating them. You may nominate a person for either or both awards, and as many people as you think worthy.

Please visit the SIGAda Awards page <http://www.sigada.org/exec/awards/awards.html> and peruse the names of past winners. This may help you think about the measure of accomplishment that is appropriate. You may be aware of people who have made substantial contributions that have not yet been acknowledged. Nominate them. Consider what you believe to be the best developments in the Ada community or SIGAda in the last year; the last 5 years; since Ada's inception. Who was responsible? Nominate them.

Please note that anyone who has received either of the two awards remains eligible for the other. Perhaps there is an outstanding SIGAda volunteer who has won our Distinguished Service Award and who has also made important contributions to the advance of Ada technology, or vice versa. Nominate him or her!

The nomination form is available on the SIGAda Awards page: <http://www.sigada.org/exec/awards/awards.html>

Submit your nomination as an e-mail or e-mail attachment to <mailto:Ada-Award-Committee@acm.org>. From your nominations, the recipients of the awards are determined by a poll of previous award winners.

Call our attention to the people who are most deserving, by nominating them. And please nominate by October 20th!

Your participation in the nominations process will help maintain the prestige and honor of these awards.

Thank you,

Ricky E. "Ranger" Sward

Chair ACM SIGAda Awards Committee

ACM SIGAda Past Chair

[See also “ACM SIGAda Award”, AUJ 37-3, p. 124. —sparre]

Ada-related Events

[To give an idea about the many Ada-related events organised by local groups, some information is included here. If you are organising such an event feel free to inform us as soon as possible. If you attended one please consider writing a small report for the Ada User Journal. —sparre]

“Make with Ada” Winners

From: AdaCore Press Center
Date: Tue Oct 24 2017
Subject: AdaCore Announces Winners for Second Annual “Make with Ada” Programming Competition
URL: https://www.adacore.com/press/2nd-annual-mwac-winners

Entries demonstrate the ease of using Ada and SPARK languages for developing reliable, safe and secure software

NEW YORK, PARIS, SANTA CLARA, Calif., October 24, 2017 – ARM TechCon – AdaCore today announced the winners of its second annual Make with Ada programming competition for embedded projects. Make with Ada aims to show how the Ada and SPARK language technologies can significantly improve code quality for modern embedded systems without requiring a steep learning curve for developers unfamiliar with these languages. Prizes are awarded to the projects that best meet the overall criteria of software dependability, openness, collaborativeness and inventiveness.

This year the 1st place prize of €5000 was awarded to Jonas Attertun from Sweden for his Ada Motorcontrol. The project involved the design of a software platform for developing a brushless DC motor controller (BLDC/PMSM). He used a custom, open-source board with an STM32F446 microprocessor, a sensored field-oriented control algorithm, and a logging feature to simplify development and allow users to visualize what is happening.

The 2nd place prize of €2000 went to German Rivera from California, USA, who also won last year’s 2nd place prize. This year, his project was a Smartwatch. He developed the embedded software of a "Swiss Army Knife" watch in Ada 2012 using a Hexiwear IoT wearable development board with two NXP Kinetis microcontrollers.

The 3rd place prize of €1000 was awarded to Manuel Iglesias Abbatemarco from Ecuador for his Ada IoT Stack project. This project added several components to the Ada Drivers Library to support an IoT Framework based on an existing lwIP (lightweight IP) implementation ported to the embedded STM32 Ethernet family of devices.

“I thoroughly enjoyed the challenge of this competition, and was able to prove that Ada can be successfully used for bare-metal projects that require fast execution," said first place winner Jonas Attertun. "The design of my project was, thanks to Ada's many nice features, much easier to understand compared to a lot of the other C implementations out there. And the combination of increased design readability and the strictness of Ada made the resulting software safer and will simply further collaborative development and reuse."

"I really enjoyed being one of the judges for the Make with Ada competition," said Rich Nass, Executive VP, OpenSystems Media. "It was a very tough choice, as most of the entries were well thought out and each served a great purpose. I based my final selections on the entries that had the potential to turn into a real product down the road, or could serve as a stepping-stone to related projects/products. Congratulations to all the candidates on a job well done."

"I was really impressed by the quality of the entries this year," said Fabien Chouteau, AdaCore software engineer. "Despite the short time frame, the submissions included a number of innovative and well-engineered projects with documentation that was both comprehensive and comprehensible. And importantly, the competition attracted contestants with little or no previous experience with Ada or SPARK, showing that our efforts to bring these technologies to the broader community of embedded systems developers are paying off."

The Make with Ada competition ran from May 15, 2017, through September 15, 2017, and attracted 35 entries. Each entrant needed to design and implement an embedded software project, using Ada and/or SPARK as the principal language technologies. Entrants needed to demonstrate that their system met its requirements and was developed using sound software engineering practices.

Embedded systems experts Jack Ganssle (Principal Consultant at The Ganssle Group), Bill Wong (Technical Editor at Penton Media), Rich Nass (Executive Vice-President, Brand Director, Embedded and IoT Franchises, OpenSystems Media), Stephane Carrez (Software Engineer, Bouygues Telecom), and Cyrille Comar (AdaCore President) served as the competition judges.

One of AdaCore's goals in sponsoring the competition was to attract entries from embedded systems developers with no previous Ada experience, to show that programmers could come up to speed quickly and productively. As noted in the project log of 3rd-place winner Manuel Iglesias Abbatemarco, this goal was met: "This project was a very good challenge to me since I had to learn Ada as quickly as possible…. I enjoyed doing it."

The Make with Ada competition is part of an overall AdaCore initiative to foster the growth of Ada and SPARK for developing embedded systems and more generally for developing "software that matters". Other elements of this initiative include free on-line training available at AdaCore U (u.adacore.com), and various resources for free software developers and students/hobbyists at the GitHub repository (github.com/adacore) and the libre site (libre.adacore.com).

Information about next year's Make with Ada competition will be available during Q2 2018 at http://www.makewithada.org/.

[See also ""Make with Ada" Programming Competition", AUJ 38-2, p. 68. —sparre]

FOSDEM 2018

From: Dirk Craeynest
<dirk@cs.kuleuven.be>
Date: Thu, 26 Oct 2017 06:10:29 -0000 (UTC)
Subject: CfP - Ada Developer Room at FOSDEM 2018, Brussels, Belgium
Newsgroups: comp.lang.ada, fr.comp.lang.ada

---

Call for Presentations

8th Ada Developer Room at FOSDEM 2018

Saturday 3 February 2018, Brussels, Belgium

http://www.cs.kuleuven.be/~dirk/ada-belgium/events/18/180203-fosdem.html

Organized in cooperation with Ada-Europe

---

[See also "FOSDEM 2016", AUJ 36-4, p. 200. —sparre]

Ada-Europe 2018 in Lisbon

From: Dirk Craeynest
<dirk@cs.kuleuven.be>
Date: Wed, 8 Nov 2017 00:29:40 -0000 (UTC)
Subject: CfP 23rd Conf. Reliable Software Technologies, Ada-Europe 2018
Newsgroups: comp.lang.ada, fr.comp.lang.ada, comp.lang.misc

---

Call for Papers

23rd International Conference on

Reliable Software Technologies - Ada-Europe 2018

18-22 June 2018, Lisbon, Portugal

http://www.ada-europe.org/conference2018

Organized by U.Lisboa on behalf of Ada-Europe,

in cooperation with ACM SIGAda, SIGBED (pending), SIGPLAN (pending) and the Ada Resource Association (ARA)

---

Ada-related Resources

Ada on Social Media

From: Jacob Sparre Andersen
<jacob@jacob-sparre.dk>
Date: Sun Nov 26 2017
Subject: Ada on Social Media

Ada groups on various social media:

- LinkedIn:	2_677 members	[1]
- Reddit:	1_101 readers	[2]
- StackOverflow:	904 followers	[3]
- Google+:	739 members	[4]
- Freenode	75 participants	[5]
- Gitter:	54 people	[6]
- Twitter:	20 tweeters	[7]

[1] https://www.linkedin.com/groups?gid=114211

[2] http://www.reddit.com/r/ada/

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

[4] https://plus.google.com/communities/102688015980369378804

[5] #Ada on irc.freenode.net

[6] https://gitter.im/ada-lang

[7] https://twitter.com/search?f=realtime&q=%23AdaProgramming

[See also "Ada on Social Media", AUJ 38-3, p. 117. —sparre]

Repositories of Open Source Software

From: Jacob Sparre Andersen
<jacob@jacob-sparre.dk>
Date: Sun Nov 26 2017
Subject: Repositories of Open Source software

GitHub:	1_993 repositories	[1]
	462 developers	[2]
	2_113 issues	[3]
Rosetta Code:	644 examples	[4]
	33 developers	[5]
	0 issues	[6]
Sourceforge:	259 repositories	[7]
BlackDuck OpenHUB:	176 projects	[8]
Bitbucket:	92 repositories	[9]
Codelabs:	45 repositories	[10]
OpenDO Forge:	24 projects	[11]
	543 developers	[11]
AdaForge:	8 repositories	[12]

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

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

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

[4] http://rosettacode.org/wiki/Category:Ada

[5] http://rosettacode.org/wiki/Category:Ada_User

[6] http://rosettacode.org/wiki/Category:Ada_examples_needing_attention

[7] http://sourceforge.net/directory/language%3Aada/

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

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

[10] http://git.codelabs.ch/

[11] https://forge.open-do.org/

[12] http://forge.ada-ru.org/adaforge

[See also “Repositories of Open Source Software”, AUJ 38-3, p. 117. —sparre]

Ada-related Tools

Simple Components

From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>
Date: Mon, 4 Sep 2017 19:40:16 +0200
Subject: ANN: Simple Components for Ada v4.23
Newsgroups: comp.lang.ada

The current version provides implementations of smart pointers, directed graphs, sets, maps, B-trees, stacks, tables, string editing, unbounded arrays, expression analyzers, lock-free data structures, synchronization primitives (events, race condition free pulse events, arrays of events, reentrant mutexes, deadlock-free arrays of mutexes), pseudo-random non-repeating numbers, symmetric encoding and decoding, IEEE 754 representations support, multiple connections server/client designing tools. The library is kept conform to the Ada 95, Ada 2005, Ada 2012 language standards.

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

Changes to the previous version:

- Bug fix in the package Parsers.Generic_Operation.Generic_Stack. On an unexpected ligature Unexpected_Comma exception is propagated.

From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>
Date: Sun, 1 Oct 2017 15:44:11 +0200
Subject: ANN: Simple Components for Ada v4.24
Newsgroups: comp.lang.ada

[...]

Changes to the previous version:

- The package Object.Archived.Handle now accepts unconstrained object types;
- Procedure Received added to the package GNAT.Sockets.Server;
- All registered to 2017-09-13 URI schemes added to the HTTP server implementation;
- Upgrade_Insecure_Requests response header was added to the HTTP server implementation;
- Package GNAT.Sockets.NTP added to support simple NTP time queries.

From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>
Date: Sun, 26 Nov 2017 14:00:29 +0100
Subject: ANN: Simple Components for Ada v4.25
Newsgroups: comp.lang.ada

[...]

Changes to the previous version:

- The package GNAT.Sockets.Connection_State_Machine.ELV_MAX_Cube_Client.Stream_IO now supports reading and writing wall thermostat settings.

[See also “Simple Components”, AUJ 38-3, p. 118. —sparre]

AdaYaml

From: Felix Krause <contact@flyx.org>
Date: Thu, 7 Sep 2017 19:18:37 +0200
Subject: ANN: AdaYaml 0.2.0
Newsgroups: comp.lang.ada

I just have released AdaYaml 0.2.0. Documentation has been updated [1] and the code is available as tag of the GitHub repository [2].

This release reaches full compliance with the yaml-test-suite [3] and fixes some bugs. Thanks go to Jacob Sparre Andersen who provided feedback and bug reports. The API of the reference-counting smart pointers has changed to make them more usable, the Text subsystem has been extracted to an own project since I plan to use it for something else.

[1] https://ada.yaml.io

[2] https://github.com/yaml/AdaYaml/tags

[3] https://github.com/yaml/yaml-test-suite

[See also “AdaYaml”, AUJ 38-3, p. 119. —sparre]

RDF Processing

From: Victor Porton <porton@narod.ru>
Date: Fri, 22 Sep 2017 15:08:29 +0300
Subject: Ada2012 libRDF bindings created
Newsgroups: comp.lang.ada

I remind that RDF is a language for expressing semantic information on the web.

Earlier I implemented Ada2012 bindings for Raptor RDF parsing library. Today I've released bindings of its derived library Rasqal (which among other supports applying SPARQL queries to RDF datasets).

I did some unit testing, but test coverage isn't complete.

The release is at

https://github.com/vporton/redland-bindings/tree/ada2012

I am yet going to adjust the Ada API, it is not perfect now. So backward incompatible changes are possible.

[...]

[See also “RDF Processing”, AUJ 38-3, p. 118. —sparre]

OpenGLAda and FreeTypeAda

From: Felix Krause <contact@flyx.org>
Date: Thu, 28 Sep 2017 23:11:17 +0200
Subject: ANN: OpenGLAda 0.5 released
Newsgroups: comp.lang.ada

It has been a long time since I released an OpenGLAda version, and an even longer time since I announced it here. After laying around abandoned for quite some time, there was interest in OpenGLAda in the recent year, so I decided to put some work into it. The result is a new release that's available on GitHub as usual [1].

OpenGLAda is a wrapper for OpenGL, GLFW, FTGL and SOIL. It also includes functionality to load OpenGL function pointers at runtime similar to what the GLEW library does for C. As thick binding, it tries to make working with the OpenGL API as painless as possible.

In this release, the major change is that I rewrote the runtime-loading part to be more efficient. A welcome side effect is that I can now autogenerate a complete list of wrapped OpenGL functions along with code pointers to where in the OpenGLAda API the functionality can be found [2].

Since users struggled to get the demo programs to work, I included detailed instructions about how to setup dependencies in the Readme [3], especially for Windows (with GNAT GPL or TDM-GCC). I hope this makes the library more accessible for newcomers.

Thanks to Roger Mc Murtrie who contributed a lot to this release, there are now a lot of additional examples available besides the demo programs. These examples are taken from textbooks and translated into Ada. A good part of those examples have been taken from the OpenGL SuperBible [4] which is written for modern OpenGL, so they are most relevant for people who do want to use OpenGL 4.x functionality.

The examples have been tested and work on Windows 10 with both GNAT GPL

2017 and TDM-GCC-64, as well as on macOS High Sierra with GNAT GPL 2017. Sadly, I gave up on setting up a VM for testing on Linux (OpenGL support was too much of a pain to set up in the VM), so I could not test it on Linux, but I doubt that something has broken there since my last test. I would still be thankful for someone to test it.

Development is still ongoing and the next release will probably see the inclusion of a binding to parts of the FreeType library as FTGL is pretty dated and does not play well with modern OpenGL functionality. FreeType is necessary to render text using OTF/TTF fonts.

[1] https://github.com/flyx/OpenGLAda/tags

[2] http://flyx.github.io/OpenGLAda/mapping.html

[3] https://github.com/flyx/OpenGLAda#detailed-installation--compilation-instructions

[4] http://www.openglsuperbible.com

From: Felix Krause <contact@flyx.org>
Date: Mon, 30 Oct 2017 12:17:14 +0100
Subject: ANN: OpenGLAda 0.6 and FreeTypeAda released
Newsgroups: comp.lang.ada

In this version, the dated FTGL binding has been replaced by a binding to the FreeType library itself, which enables more flexibility for drawing text. Furthermore, a package GL.Text has been added which provides a more high-level API similar to FTGL. It depends on Dmitry A. Kazakov's excellent Strings_Edit [2] package for UTF-8 decoding so that it is easy to render any UTF-8 string.

Since the FreeType binding may be useful for purposes unrelated to OpenGL, it is also available as a separate project [3]; however, it currently has no own versioning scheme and no documentation and is simply synchronized with the OpenGLAda repository. The binding is not complete; it only wraps the parts that were necessary to use it with OpenGL. This may improve in the future.

The current OpenGLAda release is available as tag of its repository [1].

[1] https://github.com/flyx/OpenGLAda/releases

[2] http://www.dmitry-kazakov.de/ada/strings_edit.htm

[3] https://github.com/flyx/FreeTypeAda

[See also “OpenGLAda and OpenCLAda”, AUJ 35-1, p. 7. —sparre]

Industrial Control Widget Library

From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>
Date: Tue, 3 Oct 2017 10:22:15 +0200
Subject: ANN: Ada Industrial Control Widget Library v3.16
Newsgroups: comp.lang.ada

The library is provided for design high-quality industrial control widgets for Ada applications. The software is based on GtkAda, Ada bindings to GTK+ and cairo. The key features of the library:

- Widgets composed of transparent layers drawn by cairo;
- Fully scalable graphics;
- Support of time controlled refresh policy for real-time and heavy-duty applications;
- Caching graphical operations;
- Stream I/O support for serialization and deserialization;
- Ready-to-use gauge, meter, oscilloscope widgets;
- Editor widget for WYSIWYG design of complex dashboards.

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

Changes to the previous version:

- Bug fix in Gtk.Layered.Editor to work around GNAT compiler issues.

[See also “Industrial Control Widget Library”, AUJ 37-2, p. 71. —sparre]

Project Ideas

From: Gautier de Montmollin <gautier.de.montmollin@gmail.com>
Date: Tue, 3 Oct 2017 00:37:46 -0700
Subject: Re: Re : configure ssl in AWS on Windows?
Newsgroups: comp.lang.ada

> There was some exchange here in c.l.a about implementing SSL in Ada [+SPARK]. Is the idea dead?

In a similar vein a zlib-in-Ada (another AWS dependency) is a permanent low-hanging-fruit: the spec can (and must) be just taken from the zlib-Ada binding, and the compression/decompression part taken from Zip-Ada...

Emacs Ada Mode

From: Stephen Leake <stephen_leake@stephe-leake.org>
Date: Tue, 3 Oct 2017 20:06:08 -0700
Subject: Ada mode 5.3.1 released
Newsgroups: comp.lang.ada

Ada mode 5.3.1 is now available in GNU ELPA. See the homepage (http://www.nongnu.org/ada-mode/) for NEWS, or the project page (https://savannah.nongnu.org/projects/ada-mode) for tarball download.

This is a bug fix release; the version number should have incremented from 5.2 to 5.3 in the previous release, for the the GPS indentation engine.

[See also “Emacs Ada Mode”, AUJ 38-3, p. 118. —sparre]

Prove in the Cloud

From: Yannick Moy <moy@adacore.com>
Date: Wed Oct 18 2017
Subject: Prove in the Cloud
URL: http://www.spark-2014.org/entries/detail/prove-in-the-cloud

We have put together a byte (8 bits) of examples of SPARK code on a server in the cloud here[1]. (many thanks Nico Setton for that!)

Each example consists in very few lines, a few files at most, and demoes an interesting feature of SPARK. The initial version is incorrect, and hitting the "Prove" button will return with messages that point to the errors. By following the suggested fix in comments you should be able to get the code to prove automatically.

Of course, you could already do this by installing yourself SPARK GPL on a machine (download here and follow these instructions to install additional provers). The benefit with this webpage is that anyone can now experiment live with SPARK without installing first the toolset. This is very much inspired from what Microsoft Research has done with other verification tools as part of their rise4fun website.

Something particularly interesting for academics is that all the code for this widget is open source. So you can setup your own proof server for hands-on sessions, with your own exercises, in a matter of minutes! Just clone the code_examples_server project from GitHub, follow the instructions in the README, populate your server with your exercises and examples, and you're set! No need to ask for IT to setup all boxes for your students, they just need a browser to point to your server location. An exercise consists in a directory with:

- a file example.yaml with the name and description of the exercise (in YAML syntax)
- a GNAT project file main.gpr (could be almost empty, or force the use of SPARK_Mode so that all the code is analyzed)
- the source files for this exercise

For inspiration, see the examples from the Compile_And_Prove_Demo project from GitHub, inside directory 'examples', which were used to populate our little online proof webpage.

Feel free to report problems or suggest improvements to the widget or the examples on their respective GitHub project pages.

[1] https://cloudchecker.r53.adacore.com/

GNATColl.JSON Support Packages

From: Per Sandberg <per.s.sandberg@bahnhof.se>
Date: Fri, 27 Oct 2017 05:57:12 +0200
Subject: [ANN] gnatcoll-json 1.2.0
Newsgroups: comp.lang.ada

https://github.com/persan/gnatcoll-json/releases/tag/gnatcoll-JSON-v1.2.0

- More normalized JSON representation of items that could be serialized a a one-dimensional array.

- More packages added.

[See also “GNATColl.JSON Support Packages”, AUJ 38-2, p. 72. —sparre]

LEA

From: Gautier de Montmollin <gautier.de.montmollin@gmail.com>
Date: Sat, 4 Nov 2017 12:11:53 -0700
Subject: LEA - Lightweight Editor for Ada - First (early) binary release, v 0.5.
Newsgroups: comp.lang.ada

LEA is a Lightweight Editor for Ada.

(Currently for Windows, but is there is already everything on Linux and OS X, isn't there?)

Features:

- multi-document

- multiple undo's & redo's

- multi-line edit, rectangular selections

- color themes, easy to switch

- duplication of lines and selections

- syntax highlighting

- parenthesis matching

- bookmarks

URL: https://sourceforge.net/projects/l-e-a/

This is a very early release (project is 1 month old), but LEA can already used as a simple Ada editor.

Comments, remarks, ideas, ... are welcome.

From: Randy Brukardt <randy@rrsoftware.com>
Date: Wed, 15 Nov 2017 18:44:34 -0600
Subject: Re: LEA - Lightweight Editor for Ada - First (early) binary release, v 0.5.
Newsgroups: comp.lang.ada

> LEA is a Lightweight Editor for Ada.

This looks very promising. It might even allow me to retire the circa 1986 editor that I've been primarily using for programming since -- obviously -- 1986. (That's probably just a bit overdue. ;-)

One immediate concern: if LEA is given a file that contains tab characters, it ignores them completely, thus eliminating most or all of the indentation. I suppose I could open the file in my old editor first and manually eliminate the tabs, then use LEA after that, but that seems like a massive pain-in-the-a$$ (and it would mess up places where tabs are used in the middle of a construct, such as a table in a comment). It would be much better if I could specify somehow that any tabs in the source files used 8 character indentation and the editor could do whatever it wants to preserve the look of the original file.

I tried the "quick help" button to see what the editor is supposed to do (rather than trial-and-error), but nothing seemed to happen except a brief pause in my computer's responsiveness.

From: Gautier de Montmollin <gautier.de.montmollin@gmail.com>
Date: Fri, 17 Nov 2017 20:26:49 -0800
Subject: Re: LEA - Lightweight Editor for Ada - First (early) binary release, v 0.5.
Newsgroups: comp.lang.ada

Actually tabs are preserved. You can see them with the "Show special symbols" menu command (or the paragraph button). They are just displayed too short. If you choose 8 as indentation it will (hopefully) work exactly as the DOS tabs.

I see I should dissociate tab size setting and indentation setting...

I hope I can retrieve an Ada.Text_IO tool which got rid of all tabs - correctly...

Otherwise Notepad++ provides such conversions (I've not tested).

In-memory Streams

From: Victor Porton <porton@narod.ru>
Date: Mon, 13 Nov 2017 23:23:51 +0200
Subject: In-memory streams
Newsgroups: comp.lang.ada

Where can I find an implementation of in-memory streams of unbounded length?

From: Per Sandberg <per.s.sandberg@bahnhof.se>
Date: Tue, 14 Nov 2017 06:51:29 +0100
Subject: Re: In-memory streams
Newsgroups: comp.lang.ada

I got one implementation at:

https://github.com/persan/a-stream-tools

From: Brad Moore <bmoore.ada@gmail.com>
Date: Mon, 13 Nov 2017 22:22:02 -0800
Subject: Re: In-memory streams
Newsgroups: comp.lang.ada

Another implementation is the stream_buffers-simple_unbounded.ads package at dequesterity.sourceforge.net

From: Gautier de Montmollin <gautier.de.montmollin@gmail.com>
Date: Mon, 13 Nov 2017 23:17:13 -0800
Subject: Re: In-memory streams
Newsgroups: comp.lang.ada

Yet another one :-) :

http://unzip-ada.sf.net/za_html/zip_streams__ads.htm#121_9

From: Jacob Sparre Andersen <jacob@jacob-sparre.dk>
Date: Tue, 14 Nov 2017 10:08:20 +0100
Subject: Re: In-memory streams
Newsgroups: comp.lang.ada

https://github.com/sparre/black

From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>
Date: Tue, 14 Nov 2017 10:31:27 +0100
Subject: Re: In-memory streams
Newsgroups: comp.lang.ada

http://www.dmitry-kazakov.de/ada/components.htm#Storage_Streams

Zip-Ada

From: Gautier de Montmollin <gautier.de.montmollin@gmail.com>
Date: Sat, 18 Nov 2017 12:33:20 -0800
Subject: Ann: Zip-Ada v.53
Newsgroups: comp.lang.ada

New in v.53:

- Decompression and loading of an archive's directory are more resistant to fuzzing attacks (details & credits in zipada.txt)

- Zip.Compress & ReZip: fix in local header generation for LZMA format.

- (Tools) ZipAda tool has a more useful recursive directory search (-r2 option).

- (Tests) Added Fuzzip, a fuzzing tool for the compression side.

Main site:

http://unzip-ada.sf.net

Project site:

https://sf.net/projects/unzip-ada/

GitHub clone:

https://github.com/svn2github/unzip-ada

[See also “Zip-Ada”, AUJ 37-4, p. 186. —sparre]

Ada-related Products

GNAT Pro Product Lines

From: AdaCore Press Center
Date: Wed Nov 15 2017
Subject: AdaCore Launches New GNAT Pro Product Lines
URL: https://www.adacore.com/press/new-gnat-pro-product-lines

Flagship Ada Development Environment Addresses New Needs and Domains

NEW YORK and PARIS and BURLINGTON, Mass., November 15, 2017 – AdaCore Tech Days – In response to evolving requirements from existing customers and increasing interest in Ada from traditionally C-based application domains, AdaCore today announced the launch of three product lines for its GNAT Pro technology:

- GNAT Pro Enterprise, a full-featured environment supporting industrial-grade development of mission-critical software;

- GNAT Pro Assurance, an extension of GNAT Pro Enterprise oriented towards users with software certification requirements or the need for a stable development platform that is maintained (with repairs to critical issues) over the entire duration of a long-lived project; and
- GNAT Pro Developer, geared towards new users of Ada who want to take advantage of the language's software engineering support and early error detection.

GNAT Pro Enterprise combines and replaces the existing GNAT Pro native/cross and GNAT Pro Ada Safety-Critical products. It supports all versions of the Ada language standard and includes a comprehensive toolset (visual debugger, GNAT Programming Studio and GNATbench IDEs, a variety of static analysis tools, a multi-language build facility, and much more), with premium on-line support supplied by the product developers themselves.

GNAT Pro Assurance extends GNAT Pro Enterprise with a special service known as "sustained branches", which allows customers to continue to use a specific version of the technology over the lifetime of their subscription while receiving repairs to critical code generation issues. GNAT Pro Assurance also provides, as an option, a variety of software certification services such as certification material for high-integrity run-time libraries.

GNAT Pro Developer is a new product line, offering an entry-level solution for programmers interested in benefiting from the many advantages of Ada 2012, including contract-based programming, strong typing, support for low-level programming, type-safe generic templates, and structured concurrency features. The product is especially suited to C and C++ programmers implementing small-footprint embedded systems and looking for a simple transition path to a more reliable language.

All three GNAT Pro products include SPARK Discovery, a formal methods toolsuite that allows developers to demonstrate, with mathematical rigor, program properties ranging from data flow security to absence of run-time errors.

"The GNAT Pro product line realignment is a natural step in the growth and evolution of our Ada technology, simplifying things for our customers while bringing new benefits," said Jamie Ayre, Commercial Team Lead at AdaCore. "Anyone who is currently using GNAT Pro can continue with GNAT Pro Enterprise, or else move to GNAT Pro Assurance if they have certification requirements or need long-term support on a specific release of the technology. And for new Ada users, GNAT Pro Developer will be a cost-effective way to get started and become productive with Ada."

"AdaCore has always seen an interest in Ada as an alternative language for embedded systems programming, from customers in diverse application domains ranging from medical devices to industrial process control," said Quentin Ochem, Lead of Business Development at AdaCore. "What's new is the recent significant uptick in the number of requests from these and other areas including drones and autonomous vehicles / assisted driving. GNAT Pro Developer is our initial step in welcoming this wave of newcomers to the Ada technology."

GNAT Pro Assurance, GNAT Pro Enterprise, and GNAT Pro Developer are available now; please contact info@adacore.com for pricing, platform coverage or other information about these products.

Ada and Operating Systems

Debian: Ada Packages

From: Nicolas Boulenguez <nicolas.boulenguez@free.fr>
Date: Sat, 23 Sep 2017 20:31:23 +0200
Subject: transition to gcc-7, ALI version updates
Newsgroups: gmane.linux.debian.packages.ada

The migration of most [1] Ada packages to gcc-7 to the testing distribution should be completed in a few hours.

Thanks to all maintainers implied in this update.

You may be interested in what follows if you maintain a library.

The policy requires a change of the ALI version in the name of a -dev binary package when the Ada source change [2].

The attached python script may be run after a successful build of your source package. It attempts to automatically warn about such issues by downloading a previous version of each -dev package, and comparing the ALI checksums. If a package with the same ALI version is available with different ALI checksums, it exits with a non-zero exit status.

If a -dev package requires a renaming, the attached map should help finding its reverse dependencies. Each node represents a source package, and each arrow from A to B means that B mentions the ALI version of A in its build, run-time or test dependencies.

Feel free to test and report any problem or suggestion.

[1] The only exception for now is blocked by a trivial, unrelated and patched issue that I must admit I am fully responsible for :-).

[2] GNAT is granted an exception to spare a passage through the NEW queue for all GCC packages. In other words, libgnat7-dev will *not* change its name when it breaks all other Ada -dev packages.

[See also "Debian: GNAT", AUJ 38-3, p. 120. —sparre]

Ubuntu: Simple Components etc.

From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>
Date: Sun, 1 Oct 2017 15:39:43 +0200
Subject: ANN: Ubuntu ARM v7 support
Newsgroups: comp.lang.ada

Ubuntu 16.04 Xenian packages for ARM v7 of

- Ada industrial control widget library
- Fuzzy machine learning framework
- Fuzzy sets for Ada
- GtkAda 3.14.2
- GtkAda contributions
- MAX! home automation
- Interval arithmetic for Ada
- Units of measurement for Ada
- Simple components for Ada
- String edit
- Tables

are available at www.dmitry-kazakov.de. The packages can be downloaded individually or via repository. The repository can be added by placing the following line into /etc/apt/sources.list:

deb [trusted=yes] http://www.dmitry-kazakov.de/distributions/ubuntu xenian main

[See also "Ubuntu: Simple Components etc.", AUJ 38-3, p. 120. —sparre]

Mac OS X: XNAdaLib

From: Pascal Pignard <p.p14@orange.fr>
Date: Mon, 9 Oct 2017 09:24:20 -0700 (PDT)
Subject: [ANN] XNAdaLib 2017 binaries.
Newsgroups: comp.lang.ada

This is XNAdaLib 2017 built on MacOS X 10.11 El Capitan for Native Quartz including:

- GTKAda GPL 2017 with GTK+ 3.22.20 complete,
- Glade 3.20.0,
- GnatColl GPL 2017,
- Florist GPL 2017,

- AdaCurses 20110404,
- Gate 3-05-b,
- Components 4.23,
- AICWL 3.15,
- Zanyblue 1.3.0b,
- PragmARC 06-2017-10,
- GNOGA 1.3-beta,
- AdaControl 1.18r9,
- AdaDep 1.4r1
- AdaSubst 1.5r1
- SparForte 2.1 - NEW

and as side libraries:

- Template Parser GPL 2017,
- gtksourceview 3.24.4,
- GNUTLS 3.5.9,
- ASIS GPL 2017
- SDL 1.2.15 and SDL Image 1.2.12.

XNAdaLib binaries have been posted on Source Forge: https://sourceforge.net/projects/gnuada/files/GNAT_GPL%20Mac%20OS%20X/2017-el-capitan/

Feel free to send comments.

[See also "Mac OS X: XNAdaLib", AUJ 37-4, p. 190. —sparre]

NixOS: GNAT

From: Felix Krause <contact@flyx.org>
Date: Tue, 10 Oct 2017 12:13:45 +0200
Subject: Getting GNAT GPL 2017 to run on NixOS
Newsgroups: comp.lang.ada

NixOS provides a pretty old GNAT version, so I decided to try and install AdaCore's GNAT GPL 2017 release.

I came up with the following nix expression which fetches the 64bit release from AdaCore, patches the ELF binaries to use the correct interpreter for NixOS, and sets some environment variables so that the tools find resources:

[…]

I move all binaries into $out/share/gnat/bin because everything in $out/bin would be added to my environment which is undesirable (some tools collide with existing ones). Then I create wrappers for all gnat* and gpr* tools in $out/bin. The wrapper prepends $out/share/gnat/bin to PATH so that the tools can find the accompanying binaries.

So far, so good; this got me a working gnatmake. However, the binary gnatmake creates references /lib64/ld-linux-x86-64.so.2 as interpreter instead of the correct NixOS interpreter. Is there a way to teach gnatmake which interpreter to write into the resulting binary?

I also tried to use gprbuild, which fails with these lines when called on an existing project:

```
No valid configuration found
Generation of configuration files failed
GNAT-TEMP-000001.TMP:1:01:
 "project" expected
gprbuild: processing of configuration
project "/tmp/
GNAT-TEMP-000001.TMP" failed
```

The file in /tmp does not exist (and is not the project I called gprbuild with). I am unsure what goes wrong here, any suggestions?

I also did not get GPS to start yet due to a missing libncurses.so.5 which does not seem to be part of the distribution – strange as everything else is bundled. I will try to load NixOS' native ncurses.

iOS: Poll

From: Luke A. Guest <laguest@archeia.com>
Date: Mon, 13 Nov 2017 13:16:32 -0800
Subject: Poll: Would you use iOS GNAT?
Newsgroups: comp.lang.ada

https://plus.google.com/115665919057085310215/posts/TA461XCNhFy

Windows: GNAVI: GNU Ada Visual Interface

From: Gautier de Montmollin <gautier.de.montmollin@gmail.com>
Date: Sat, 18 Nov 2017 12:38:03 -0800
Subject: GWindows release, 11-Nov-2017
Newsgroups: comp.lang.ada

- Changes to the framework are detailed in changes.txt or in the News forum on the project site.
- The installer doesn't require administrator login anymore, and doesn't duplicate the framework anymore for special purposes.

Project site:

https://sf.net/projects/gnavi/

GitHub clone:

https://github.com/svn2github/gnavi

[See also "Windows: GNAVI: GNU Ada Visual Interface", AUJ 37-3, p. 128. —sparre]

References to Publications

Getting Started with AVR-Ada

From: Rolf Ebert <rolf.ebert.gcc@gmx.de>
Date: Sun, 10 Sep 2017 19:54:12 +0200
Subject: Docker Image for AVR-Ada
Newsgroups: gmane.comp.hardware.avr.ada

only now I discovered the following blog entry:

http://finisterra.motd.org/?p=272

This guy provides a docker image with AVR-Ada. this is probably the most simple way to get started with AVR-Ada. Thank you.

From: Tero Koskinen <tero.koskinen@iki.fi>
Date: Sun, 10 Sep 2017 22:06:08 +0300
Subject: Re: Docker Image for AVR-Ada
Newsgroups: gmane.comp.hardware.avr.ada

[...]

And just a reminder that I have AVR-Ada packages for Fedora 25, Ubuntu 12/14, and RPi1+Debian Wheezy at:

- http://rpi1.ada-language.com/
- http://ubuntu.ada-language.com/
- http://fedora.ada-language.com/

[See also "AVR-Ada", AUJ 36-1, p. 14. —sparre]

Assessing Ada for Audio Applications

From: Gustavo Hoffmann <gusthoff.ada@gmail.com>
Date: Fri, 10 Nov 2017 14:12:20 -0800
Subject: ANN: article on Ada language for audio applications
Newsgroups: comp.lang.ada

I'm just sharing an article on Ada language for audio applications that I've co-authored:

http://www.electronicdesign.com/embedded-revolution/assessing-ada-language-audio-applications

There's a Mini-RTOS in My Language

From: Fabien Chouteau <fabien.chouteau@gmail.com>
Date: Thu Nov 23 2017
Subject: There's a mini-RTOS in my language
URL: http://blog.adacore.com/theres-a-mini-rtos-in-my-language

The first thing that struck me when I started to learn about the Ada programing language was the tasking support. In Ada, creating tasks, synchronizing them, sharing access to resources, are part of the language

In this blog post I will focus on the embedded side of things. First because it's what I like, and also because it's much more simple :)

For real-time and embedded applications, Ada defines a profile called `Ravenscar`. It's a subset of the language designed to help schedulability analysis, it is also more compatible with platforms such as micro-controllers that have limited resources.

So this will not be a complete lecture on Ada tasking. I might do a follow-up with some more tasking features, if you ask for it in the comments ;)

Ada Inside

Pasta!

From: Gautier de Montmollin <gautier.de.montmollin@gmail.com>
Date: Fri, 1 Sep 2017 01:46:39 -0700
Subject: Ann: Pasta!, an online game in Ada
Newsgroups: comp.lang.ada

[The subject has popped in this forum in the sidelines, but so far I did not announce it properly, so I'm doing it right now...]

"Pasta!" is a match-3 type puzzle game: you align three (or more) pasta to make them disappear. It's likely that you already came across such a game...

http://pasta.phyrama.com/game.html

The game is fully programmed in Ada, with the exception of a few SQL commands for player persistence or recording high scores.

The server side uses the Gnoga library.

The client side is any good Web browser supporting HTML5 + JS, on any device (desktop, laptop, tablet, smartphone, ...) on any OS (Windows, Mac, Linux, Android, iOS, ...).

For interaction you can use a mouse or a touchscreen.

For moving pasta, you can choose to do a pair of clicks / touchs, or do a drag & drop.

Pasta! is developed from a game demo in the Gnoga distribution (Leaves) so the key parts are open-source.

Pasta! is free of charge and contains NO in-app payment at any point.

[See also “Pasta!”, AUJ 38-3, p. 212. —sparre]

MAX! Home Automation

From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>
Date: Wed, 6 Sep 2017 18:10:44 +0200
Subject: ANN: MAX! home automation v1.11
Newsgroups: comp.lang.ada

MAX! home automation is a GTK+ application to manage ELV/eQ-3 MAX! cubes. A cube is a gateway to a network of radiator thermostats, shutter contacts etc.

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

Changes to the previous version:

- Preview window added to the configuration file chooser dialog;
- Restoring configuration from a file is made more lenient towards data format errors.

From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>
Date: Mon, 27 Nov 2017 18:24:37 +0100
Subject: ANN: MAX! home automation v1.12
Newsgroups: comp.lang.ada

[...]

Changes to the previous version:

- Saving and restoring wall thermostat settings added.

[See also “MAX! Home Automation”, AUJ 38-3, p. 121. —sparre]

SparForte

From: Ken O. Burtch <koburtch@gmail.com>
Date: Wed, 6 Sep 2017 16:05:19 -0700
Subject: ANN: SparForte 2.1
Newsgroups: comp.lang.ada

SparForte is my Ada-based shell, scripting language and web template engine.

Version 2.1 introduces 50 changes, including:

- Support for JUnit reporting
- Abstract and Limited qualified types
- Many stability improvements

Also:

- New SparForte mailing list has been created
- SparCanto web framework prototype is available on GitHub
- There is a "Try It" page on the SparForte website.

What's New in SparForte 2.1?

http://www.pegasoft.ca/coder/coder_september_2017.html

The Change Log is here

http://www.sparforte.com/news/2017/news_sep2017.html

[See also “SparForte”, AUJ 38-2, p. 77. —sparre]

Wasabee

From: Gautier de Montmollin <gautier.de.montmollin@gmail.com>
Date: Tue, 12 Sep 2017 06:40:57 -0700
Subject: Wasabee 0.0.2 (Ada Web browser) downloads
Newsgroups: comp.lang.ada

Just an update on this project [1] which is sleeping but not completely dead...

To facilitate things, I've uploaded a few resources

- zipped sources

- binary executable for Windows [2]

- a few screenshots

[1] https://sourceforge.net/projects/wasabee/

[2] Before anyone begins to panic: Wasabee is multi-platform; currently there are the following targets in various states:

- Text (console output, good for testing the HTML parser...)
- SDL
- Windows (already multi-tab, multi-window, supports hyperlinks, back & forward navigation).

[See also “Wasabee”, AUJ 34-3, p. 146. —sparre]

Jobs

From: Wiljan Derks <wiljan.derks@gmail.com>
Date: Sat, 21 Oct 2017 08:03:15 -0700
Subject: Re: Where to find Ada roles?
Newsgroups: comp.lang.ada

We are searching for Ada developers :)

Feel free to send me your CV.

Some background:

That is for Nexperia, location Netherlands, Nijmegen.

We develop real time control systems and also big data systems in Ada.

GNAT Pro for Critical Avionics Software

From: AdaCore Press Center
Date: Tue Oct 31 2017
Subject: Ada on Board: Thales Using AdaCore’s GNAT Pro for Critical Avionics Software
URL: https://www.adacore.com/press/ada-on-board-thales-using-adacores-gnat-pro-for-critical-avionics-software

Qualified autocode generator implemented in Ada

MERIGNAC, France & PARIS & NEW YORK, October 31, 2017 - AdaCore today announced that its GNAT Pro Ada environment has been successfully used by Thales to develop and verify a qualified autocode generator for critical airborne software. Thales implemented the autocode generator in Ada; the tool takes an XML file and produces source code for an embedded avionics system that will be assessed against the Level B objectives in DO-178C / ED-12C.

The development process for the autocode generator has been performed according to the European Aviation Safety Agency (EASA) DO-330 / ED-215 Tool Qualification Considerations standard, and the tool has been qualified for the avionics project at tool qualification level TQL-2.

In addition to the compilation environment used for the development and verification of the autocode generator, several AdaCore tools have also been used for the avionics software itself.

These include the GNAT Programming Studio (GPS) Integrated Development Environment (IDE), the GNATcheck coding standard verifier, and the GNATcoverage structural coverage analyzer. According to the Thales engineer in charge of the autocode generator, using an automated coding standard verifier and a qualified structural code coverage analyzer greatly helped the project complete the Tool Verification Process.

“AdaCore has a long and successful history in the avionics industry, with a growing product range that has enabled customers to develop and verify safety-critical software at the highest levels of DO-178B/C / ED-12B/C certification,” said Jamie Ayre, Commercial Team Lead at AdaCore. “We are pleased to see GNAT Pro Ada being used by Thales, both for their application itself and also the qualified autocode generator that is producing the code.”

[...]

GNAT Pro for International Space Station Software

From: AdaCore Press Center
Date: Wed Nov 15 2017
Subject: MDA Selects AdaCore’s GNAT Pro Assurance Development Platform for International Space Station Software
URL: https://www.adacore.com/press/mda-gnatpro-space-station

NEW YORK and PARIS and BURLINGTON, Mass., November 15, 2017 – AdaCore Tech Days – AdaCore today announced that MDA, a business unit of Maxar Technologies, has selected the GNAT Pro Assurance Ada development environment for the LEON3 target processor, to produce the software for a Ku-Band communication subsystem that will replace the current version. This critical International Space Station (ISS) subsystem has to work reliably over the long term, a requirement that led MDA to maintain Ada as the implementation language. With GNAT Pro Assurance, a service known as sustained branches allows MDA to continue developing and maintaining their software over the long term using a specific version of the GNAT Pro technology, with access to code generator updates to correct critical issues.

The replacement Ku-Band subsystem, known as the Space to Ground Transmitter Receiver Controller (SGTRC) will interface with the existing International Space Station (ISS) Space-to-Ground Antenna, previously provided by MDA. The project includes a prototype and test unit. The new SGTRC communication subsystem will support the long-term mission of the ISS and ensure the reliability and availability of high speed data connections between the ISS, Mission Control Centers and science laboratories on the ground.

A number of tools in the GNAT Pro Assurance product can help MDA meet their project’s goals. These include the GNAT Programming Studio (GPS) tailorable Integrated Development Environment (IDE), static analysis tools for stack usage computation (gnatstack) and code metrics calculation (gnatmetric), an emulator (gnatemulator) that in effect executes LEON3 target code on the host, a testing harness generator (gnattest), and many others, backed by expert support provided by the AdaCore product developers themselves.

“We have elected to reuse the majority of the original source code for the SGTRC replacement in its original language of Ada as a conservative design choice,” said Bryan Tracy, Software Lead for the SGTRC Replacement program. “Partnering with AdaCore enables us to do this with increased confidence and efficiency.”

“Ada and AdaCore have a long and successful history in space applications in general, and with MDA in particular,” said Jamie Ayre, Commercial Team Lead at AdaCore. “We're pleased to see that history continued with MDA's selection of GNAT Pro Assurance for their critical ISS communication subsystem.”

[...]

Ada in Context

Creating C Bindings

From: Per Sandberg <per.s.sandberg@bahnhof.se>
Date: Fri, 18 Aug 2017 07:18:47 +0200
Subject: Re: Please evaluate tiny binding that does not use Interfaces.C
Newsgroups: comp.lang.ada

The proper way to generate bindings to libraries with interfaces defined in C/C++ is to use the [GNAT] compiler-switch -fdump-ada-spec. To trust a human to get all the bits and pieces correct only works if the interface is trivial. The way I usually does it is:

- generate valid a C/C++ file including all required headers.
- compile the file with gcc -c -fdfump-ada-spec ${file}
- If required, edit in the file using some script tool such as sed.

By using the above method you could regenerate the bindings when the underlying library evolves and you will get a correct binding every time.

Of course if you want to use one and only one simple method from a foreign library its always possible to just do a simple "import" in the code.

Bidirectional UART Design for Ravenscar

From: Simon Wright <simon@pushface.org>
Date: Fri, 25 Aug 2017 23:24:38 +0100
Subject: Bidirectional UART design for Ravenscar
Newsgroups: comp.lang.ada

I'm trying to implement a two-way interrupt-driven interface via a UART with Ravenscar, and having trouble because there is one interrupt with multiple causes; a writer must wait until transmission is possible before putting the next character, while a reader must wait until there's a character to be read.

This makes it hard to write a PO with a single entry (a Ravenscar restriction) that serves both requirements.

It occurs to me that the interrupt handler could release one of two suspension objects; if it's Tx_Transfer_Complete, release the SO that the writer waits on, if it's Rx_Transfer_Complete release the reader's SO.

Does that seem a reasonable approach?

From: Niklas Holsti <niklas.holsti@tidorum.fi>
Date: Sat, 26 Aug 2017 01:40:01 +0300
Subject: Re: Bidirectional UART design for Ravenscar
Newsgroups: comp.lang.ada

> [...] Does that seem a reasonable approach?

IMO yes.

I suppose that Tx and Rx completion can happen at the same time (in the same execution of the interrupt handler), in which case both suspension objects should be released.

From: Jeffrey R. Carter <jrcarter@acm.org>
Date: Sat, 26 Aug 2017 13:04:48 +0200
Subject: Re: Bidirectional UART design for Ravenscar
Newsgroups: comp.lang.ada

> [...] Does that seem a reasonable approach?

It should work. It seems unnecessarily low level, with the additional opportunities for error that that brings.

Under Ravenscar, something that conceptually would be a PO with multiple entries has to become multiple POs with 1 entry each. When the entry is just a wait/release, it can be replaced by an SO.

I'd probably do an initial Ravenscar design like (really, I do the design graphically, but the graphics then translate mechanically into Ada specifications):

```
protected type Waiter is
   procedure Signal;
   entry Wait;
private -- Waiter
   Occurred : Boolean := False;
```

```
end Waiter;
Rx_Waiter : Waiter;
Tx_Waiter : Waiter;
protected UART_Interrupt is
   procedure Handle;
   pragma Attach_Handler (Handle, ...);
end UART_Interrupt;
```

Handle calls Rx_Waiter.Signal or Tx_Waiter.Signal. (It's been a while since I looked at Ravenscar, but I think such calls are OK. If not, then you'd have to add an entry to UART_Interrupt and have a task between it and the Waiter POs.)

Then your Tx task might be

```
loop
   UART.Put (Byte => Output.Next);
   Tx_Waiter.Wait;
end loop;
```

and your Rx task,

```
loop
   Rx_Waiter.Wait;
   Input.Put (Byte => UART.Received);
end loop;
```

Whether the overhead of the extra POs is too much for your application depends on the timing requirements and measurement.

I worked with Bo Sandén once on what would be involved in a Ravenscar solution of his Flexible-Manufacturing-System problem (I don't think it was ever published). The full-Ada solution was full of protected queues of IDs for matching available resources with tasks that needed them. For example, a Q to match available autonomous guided vehicles (AGVs) with job tasks that needed them. When an AGV was available, its ID was put on the Q. When a job needed an AGV, it blocked on the Q's Get entry until the Q was not empty, then got an AGV ID. Multiple job tasks could be blocked on the Q, which is not allowed in Ravenscar[*]. That simple protected Q turned into 2 protected Qs, a PO per workstation, and a helper task to connect them all.

Personally, I found the single Q easier to understand and analyze, but I guess that's not true of automated tools.

[*] Neither are the dynamically created job tasks, so the design had to be inverted from job tasks that obtain the resources needed to process them to workstation tasks that find jobs that need their services. At least 1 other Ravenscar-compliant design is possible.

Renaming of a Non-primitive Operation Creates a Primitive Operation

From: Jere <jhb.chat@gmail.com>
Date: Tue, 29 Aug 2017 08:39:39 -0700 (PDT)
Subject: procedure renames creates a primitive operation?
Newsgroups: comp.lang.ada

I was curious if this was a bug or intended.

Given the following code:

```
procedure Main is
   package Some_Package is
      type Main_Type is tagged null record;
      package Non_Primitive_Ops is
         procedure Some_Op(
                 Obj : Main_Type) is null;
      end Non_Primitive_Ops;
      -- without this line, the file
      -- correctly generates compiler errors
      procedure Some_Op(Obj : Main_Type)
         renames Non_Primitive_Ops.
                               Some_Op;
   end Some_Package;
   Var : Some_Package.Main_Type;

   package Other_Package is
      type Derived_Type
        is new Some_Package.Main_Type
           with null record;
      -- Should this create a compiler error?
      overriding
      procedure Some_Op(
                 Obj : Derived_Type) is null;
   end Other_Package;

begin
   Var.Some_Op; -- Should this create a
                -- compiler error?
end Main;
```

I used a renames clause to bring a non primitive operation into scope (for client readability when calling the procedure). However, when I did that, it created a primitive operation. Is this correct behavior or a bug?

It seems odd that a procedure created through a renames clause would be primitive, but perhaps it is intended?

From: Christoph Karl Walter Grein <christ-usch.grein@t-online.de>
Date: Tue, 29 Aug 2017 10:42:36 -0700
Subject: Re: procedure renames creates a primitive operation?
Newsgroups: comp.lang.ada

[...]

As a related example, the following renaming of a primitive op. creates a separate slot in the dispatching table, i.e. you can override both operations differently:

```
package Pack is
   type T is tagged private;
   procedure P (X: T);
   procedure Q (Y: T) renames P;
end Pack;
with Pack;
package Qack is
   type S is new Pack.T with private;
   overriding procedure P (X: S);
   overriding procedure Q (X: S);
   -- no renaming of P!
end Qack;
```

From: Randy Brukardt <randy@rrsoftware.com>
Date: Wed, 30 Aug 2017 20:24:53 -0500
Subject: Re: procedure renames creates a primitive operation?
Newsgroups: comp.lang.ada

> [...] Is this correct behavior or a bug?

Yes, a renames of a non-primitive can be primitive. Similarly, a renames of a primitive can be non-primitive.

Its an intended bug. That is, it is generally acknowledged that this semantics is the worst possible choice (you've stumbled on one reason why). There are some really horrid consequences for things like class-wide preconditions. But it wasn't obvious what choice made sense at the time it was made in Ada 95. It's way too late to change now, of course.

Misuse of Address Aspects

From: Randy Brukardt <randy@rrsoftware.com>
Date: Sat, 2 Sep 2017 20:30:21 -0500
Subject: Re: Community Input for the Maintenance and Revision of the Ada Programming Language
Newsgroups: comp.lang.ada

> procedure Algorithm (Item : Integer_Array) is

> use Ada.Assertions;

> X : Integer_Array (0 .. Item'Length - 1) with Address => Item'Address;

The use of address clauses other than for interfacing is evil (and not guaranteed to work in the case of overlaying Ada code). Don't do that.

You can change the bounds of an array with a type conversion to a constrained subtype, which is likely to be cheaper and not break if you change compilers.

Profiling with Gprof

From: Stephen Leake <stephen_leake@stephe-leake.org>
Date: Sat, 2 Sep 2017 15:02:46 -0700
Subject: link errors with GNAT GPL 2016 and gprof
Newsgroups: comp.lang.ada

I'm trying to Do The Right Thing and profile my code before I start messing with it to make it faster. But GNAT is _not_ cooperating!

According to the gnat user guide, to use gprof, we compile as:

```
gnatmake -f -pg -P <project.gpr>
```

That fails with GNAT GPL 2016:

```
gprbuild: illegal option "-pg" on the command line
```

So I added "-pg" in the Builder package in the project file:

```
package Builder is
   case Profile is
   when "Full" =>
```

```
    for Default_Switches ("Ada") use
                                  ("-f", "-pg");
  when "On" =>
    for Default_Switches ("Ada") use
                                  ("-pg");
  when "Off" =>
    null;
  end case;
end Builder;
```

That gives -pg options on each 'gcc' command, but fails at link time:

gcc run_ada_lite_parser.o -o run_ada_lite_parser.exe

run_ada_lite_parser.o: In function `ada_run_ada_lite_parser':

C:/Projects/org.wisitoken.stephe-1/wisi/test/run_ada_lite_parser.adb:12: undefined reference to `mcount'

I noticed there is no '-pg' on the 'gcc' link command line, so I added it to the Linker package:

```
package Linker is
  case Profile is
  when "Full" | "On" =>
    for Default_Switches ("Ada") use
                                    ("-pg");
  when "Off" =>
    null;
  end case;
end Linker;
```

That did not help. Also tried adding it to the Binder package; no help there either.

How do I tell gprbuild to specify "-pg" for linking?

The gnat user guide has always been unhelpful about mapping command line options to project file packages.

From: Jacob Sparre Andersen <jacob@jacob-sparre.dk>
Date: Mon, 04 Sep 2017 07:52:34 +0200
Subject: Re: link errors with GNAT GPL 2016 and gprof
Newsgroups: comp.lang.ada

We're using "gcov" regularly on a project I'm working on. (For some reason I can't get "gprof" to give me meaningful data, even though "gcov" works fine.)

The whole setup is for various reasons a bit complex, but in its essense, it reduces to:

```
project Build is
  [...]
  package Compiler is
    for Default_Switches ("Ada") use ([...],
                          "-fprofile-arcs",
                          "-ftest-coverage");
  end Compiler;
  package Linker is
    for Default_Switches ("Ada") use ([...],
                          "-pg",
                          "-fprofile-arcs");
  end Linker;
end Build;
```

Ada.Iterator_Interfaces?

From: Stephen Leake <stephen_leake@stephe-leake.org>
Date: Wed, 6 Sep 2017 00:49:49 -0700
Subject: Why does Ada.Iterator_Interfaces specify Next as a function rather than a procedure?
Newsgroups: comp.lang.ada

I have a fairly complex collection of stuff that I want to define an iterator for, so all clients access the stuff in the same order. The stuff is token definitions for a lexer; keywords, punctuation, whitespace, identifiers, numbers.

Initially, I defined my own Cursor type, that stores iterators to the various pieces, and manages the transitions between them.

Thus the Cursor type has state, and the "Next" subprogram is a procedure. It is _not_ just "return node.next", but more like:

if keywords not done, do Next (Keywords)

elsif punctuation not done, do Next (Punctuation)

with more logic that does First (Punctuation) at the right time. And so on for the rest of the stuff.

Then I learned about Ada 2012 Iterators (allowing things like "for Token_Name of Tokens"), so naturally I tried to define one for this collection.

It all goes well until I hit the definition of Ada.Iterator_Interfaces.Next; it must be a function, with an "in" parameter.

I can work around this with an access type, but that's just annoying. And since I would be lying to the compiler, I'm not clear what subtle bugs I might encounter. For example, the compiler might assume that it can do:

Next_Cursor := Next (Cursor);

and Next_Cursor and Cursor have different values (ie refer to different items). I can't think of why it would want to do that for a loop, but since the current definition of Next allows it, I worry about it.

If I don't use an access type, I'll have to copy the Cursor state in each call to Next, then modify it and return that. This is not in a time critical loop, so that's not really a problem, but it seems a silly waste. And my next use case could very well be in a time critical loop.

The Annotated Ada Reference Manual doesn't shed any light on this.

Would it be a problem to change Ada.Iterator_Interfaces.Next (and Previous) to take an "in out" parameter for the Cursor?

Or provide an alternate procedure Next? I guess that would have to be in a different package.

From: Randy Brukardt <randy@rrsoftware.com>
Date: Wed, 6 Sep 2017 16:50:47 -0500
Subject: Re: Why does Ada.Iterator_Interfaces specify Next as a function rather than a procedure?
Newsgroups: comp.lang.ada

(1) The ARG is looking at alternative kinds of iterators for Ada 2020. See AI12-0188-1 and AI12-0189-1 (probably only one of these will be used).

(2) Ada 2012 changed the rules to reflect that the Rosen trick and its brother, the squirreling controlled type (used a lot by Claw) are legitimate techniques, even if they cause a constant to be modified. (Essentially, true constants are impossible for some kinds of types.)

(3) So, making the iterator type use one of these techniques, or simply have it contain a pointer to a writable state object, should solve the problem. (But I agree it is not optimal, thus item (1)).

That is, For Ada 2012 I'd define this something like:

```
type State is record
  -- The components of the state.
end record;
type My_Iterator is new
   Ada.Finalization.Controlled and
   Reversible_Iterator with record
  My_State : access State; -- Writable
                               -- state.
  The_State : aliased State;
end record;
procedure Initialize (Obj :
               in out My_Iterator) is
begin
  Obj.My_State := The_State'Access; -- (*)
end Initialize;
```

(*) This might need to be Unchecked_Access. If so, it is safe, since the designated data will always live as long as the pointer -- they're both parts of the same object! Unless someone copies the pointer to a global variable -- but hopefully this is used in a private type so clients can't do that -- in that case, only you could do something stupid. ;-)

Initialize will be called whenever any My_Iterator object is created, and the object is always a variable at that time. Thus, you can save a writable pointer at the time, and use it later even if you only have a constant view of the object. This is a legitimate technique -- it is NOT erroneous in Ada 2012 -- so if your compiler has problems with it, file a bug report and if they don't believe that, send me a proposed ACATS test!

From: Stephen Leake <stephen_leake@stephe-leake.org>
Date: Wed, 6 Sep 2017 18:04:13 -0700
Subject: Re: Why does Ada.Iterator_Interfaces specify Next as a function rather than a procedure?
Newsgroups: comp.lang.ada

> (1) [...]

Interesting, but neither addresses my issue; the definition of Next as a function does not change.

189 mentions my problem ("often using the Rosen trick"), but seems to think the Rosen trick is an acceptable solution.

A unified cursor does make sense for my application.

> [...] This is a legitimate technique -- it is NOT erroneous in Ada 2012 -- so if your compiler has problems with it, file a bug report and if they don't believe that, send me a proposed ACATS test!

You are implying that my worry about the compiler assuming the input and output of Next must point to different items is not valid.

I'll give it a try.

From: Stephen Leake
<stephen_leake@stephe-leake.org>
Date: Wed, 6 Sep 2017 18:29:06 -0700
Subject: Re: Why does Ada.Iterator_Interfaces specify Next as a function rather than a procedure?
Newsgroups: comp.lang.ada

> type My_Iterator is new Ada.Finalization.Controlled and Reversible_Iterator with record

> My_State : access State; -- Writable state.

> The_State : aliased State;

> end record;

I don't see how to do this; 'Reversible_Iterator' is defined in an instantiation of Ada.Iterator_Interfaces, which takes the type My_Iterator as a generic parameter.

Just leaving that off still fails, if I try to make the contents of the cursor private while exposing the Iterate function.

Are you relying on some new Ada 202x syntax here?

From: Randy Brukardt
<randy@rrsoftware.com>
Date: Thu, 7 Sep 2017 18:14:20 -0500
Subject: Re: Why does Ada.Iterator_Interfaces specify Next as a function rather than a procedure?
Newsgroups: comp.lang.ada

I think you're confusing the cursor and the iterator object. (Next has two parameters, after all.) I'm talking about the definition of the iterator object; that's where you keep any state necessary. The cursor just provides a reference to the current item, whatever that might be.

The instantiation takes the cursor type; it then is used to define the actual iterator type.

In almost all cases, the iterator object has to be separate from the underlying data structure; generally you define a function to create the initial iterator object. (There are aspects to automate this process, look at how the containers are structured if you want to try to use them.) One reason for this is the possibility that multiple tasks might iterate over the same read-only object.

Network Time Protocol

From: Dmitry A. Kazakov
<mailbox@dmitry-kazakov.de>
Date: Sat, 16 Sep 2017 11:29:09 +0200
Subject: Re: NTP
Newsgroups: comp.lang.ada

> Is there a way to get time via the Network Time Protocol in Ada? [...]

Do you mean send a single UDP request to an NTP server and convert the response to Ada time?

You can try this:

```
with Ada.Text_IO; use Ada.Text_IO;
with Ada.Calendar; use Ada.Calendar;
with Ada.Calendar.Formatting;
use  Ada.Calendar.Formatting;
with Ada.Exceptions; use Ada.Exceptions;
with Ada.Streams; use Ada.Streams;
with GNAT.Sockets; use GNAT.Sockets;
with Interfaces; use Interfaces;

procedure Test is
    function Get_NTP_Time
       ( Server  : String;
         Timeout : Timeval_Duration := 10.0
       ) return Time is
      NTP_Packet_Size : constant := 48;
     -- RFC 5905: Official NTP era begins at
     -- 1 Jan 1900. We cannot
     -- have it in Ada.Calendar.Time,
     -- so taking a later time. Note
     -- Time_Zone = 0 in order to have it UTC
      Era : constant Time :=
          Time_Of (1999, 12, 31,
                   Time_Zone => 0);
     -- RFC 5905: seconds since
     -- 1 Jan 1900 to 31 Dec 1999
      Era_Offset : constant :=
                         3_155_587_200;
      Socket   : Socket_Type;
      Address  : Sock_Addr_Type;
      Seconds  : Unsigned_32;
      Fraction : Unsigned_32;
      Last     : Stream_Element_Offset;
      Data     : Stream_Element_Array
            (1..NTP_Packet_Size) :=
           ( 1  => 2#1110_0011#, -- LI,
                          -- Version, Mode
             2  => 0, -- Stratum,
                     -- or type of clock
             3  => 0,        -- Polling Interval
             4  => 16#EC#,   -- Peer Clock
                              --  Precision
             13 => 49,
             14 => 16#4E#,
             15 => 49,
             16 => 52,
             others => 0
            );
    begin
      Address.Addr := Addresses
           (Get_Host_By_Name (Server), 1);
      Address.Port := 123; -- NTP port
      Create_Socket (Socket, Family_Inet,
               Socket_Datagram);
      Set_Socket_Option
      (  Socket,
         Socket_Level,
         (Receive_Timeout, Timeout)
      );
      Send_Socket (Socket, Data,
                        Last, Address);
      Receive_Socket (Socket, Data, Last,
                          Address);
      if Last /= Data'Last then
        Raise_Exception (Data_Error'Identity,
                          "Mangled response");
      end if;
      Seconds :=
        (  Unsigned_32 (Data (41)) * 2**24
          +  Unsigned_32 (Data (42)) * 2**16
          +  Unsigned_32 (Data (43)) * 2**8
          +  Unsigned_32 (Data (44))
          -  Era_OFfset
        );
      Fraction :=
      (  Unsigned_32 (Data (45)) * 2**24
          +  Unsigned_32 (Data (46)) * 2**16
          +  Unsigned_32 (Data (47)) * 2**8
          +  Unsigned_32 (Data (48))
       );
      return (  Era
          +  Duration (Seconds)
     --     +  Duration (Long_Float
     --     (Fraction) / 2.0**32)
          );
    end Get_NTP_Time;

    Stamp : Time;
  begin
    Stamp := Get_NTP_Time
                            ("time.nist.gov");
    Put_Line ("NTP time " & Image (Stamp));
  exception
    when Error : others =>
      Put_Line ("Error: " &
            Exception_Information (Error));
  end Test;
```

Exclusive_Functions

From: Randy Brukardt
<randy@rrsoftware.com>
Date: Mon, 2 Oct 2017 19:08:34 -0500
Subject: Re: how to use "in out" for the "self" parameter in a function in a protected object?
Newsgroups: comp.lang.ada

> [...]

Note that the Ada 2012 Corrigendum added aspect "Exclusive_Functions", which changes the behavior of protected functions to support exclusive read-only access. (This allows a PO to be used as a concurrency wrapper around some non-concurrent data structure without having to change all of the functions of the wrapped data structure into procedures.)

Some of us complained about tying read-only access to the use of protected functions long ago, but we got nowhere then and probably wouldn't get anywhere now, either. 98% of the time you can use a protected procedure instead, but there are cases where a procedure isn't possible and you are just stuck.

Ada.Containers.Vectors. Reference and Constrained Objects?

From: Stephen Leake
<stephen_leake@stephe-leake.org>
Date: Sat, 28 Oct 2017 04:10:45 -0700
Subject: Is Ada.Containers.Vectors. Reference_Type a constrained view?
Newsgroups: comp.lang.ada

I have the following vector type:

```
  type Indent_Labels is (Not_Set, Int,
Anchor,
                      Anchored, Nested_Anchor);
  type Indent_Type (Label : Indent_Labels :=
                       Not_Set) is record
    case Label is
    when Not_Set =>
      null;
    when Int =>
      Int_Offset : Integer;
    ...
    end case;
  end record;
  package Indent_Vectors
     is new Ada.Containers.Vectors
        (Line_Number_Type, Indent_Type);
```

Then I have code like this:

```
  for Line in First_Line .. Last_Line loop
    declare
      Indent : Indent_Type :=
                Data.Indents (Line);
    begin
      case Delta_Indent.Label is
      when Int =>
        Indent_Apply_Int (Indent,
                       Delta_Indent.Offset);
      ...
      end case;
      Data.Indents.Replace_Element (
                            Line, Indent);
    end;
  end loop;
```

The body of Indent_Apply_Int may change the Label from Not_Set to Int; the other branches of the case statement change Not_Set to the other label values.

I'd like to avoid copying Indent (mostly on general principle; this is not a large object). However, if I use:

```
Indent : Indent_Type renames
  Data.Indents.Reference (Line).Element.all;
```

and drop the Replace_Element call, then Constraint_Error is raised at the point in Indent_Apply_Int that changes the Label (I'm using GNAT GPL 2017).

I'm not clear if that Constraint_Error is allowed by the ARM. Reference returns a Reference_Type object:

```
type Reference_Type (
          Element : not null
           access Element_Type) is private
   with Implicit_Dereference => Element;
```

Note that Element is of an access type. AARM 4.8 (6) says that allocated objects are constrained by their initial value. AARM 3.10 (26.d/2) says most non-allocated objects accessed via an access-to-object type are not constrained.

So is the type of discriminant Element an access to object type? It doesn't have 'all' in it, so I guess not. Note that the syntax of discriminants does not allow 'all'.

This seems to be a flaw; it would be nice to be able to use Reference in the code above, especially if the copy operation is slow. I guess I could use Update_Element in that case.

In Ada.Containers.Indefinite_Vectors, the Element parameter of Update_Element is allowed to be constrained (AARM A.18.11 (8/2)); that statement is not made in Ada.Containers.Vectors, so the actual Element in a Vector cannot be constrained.

From: Randy Brukardt
<randy@rrsoftware.com>
Date: Tue, 14 Nov 2017 18:38:28 -0600
Subject: Re: Is Ada.Containers.Vectors. Reference_Type a constrained view?
Newsgroups: comp.lang.ada

[...]

But your question really is, is the object designated by the reference object allowed to be constrained (that is, allowed to be an allocated object). This doesn't seem to be answered by the RM.

The question IS answered for Update_Element (see A.18.2(142/2)).

I'd argue that the same rule is intended for Reference_Type, but since there is no wording requiring it in the RM, it's hard to say that the GNAT implementation is wrong. You might try two things:

(1) Try using Update_Element rather than Reference. (A pain, I know.) If that does not work either, then GNAT is wrong, file a bug report.

(2) Send a question to Ada-Comment so that the ARG considers the question. I don't think we ever intended Update_Element and Reference to work differently, but it ought to be discussed.

Note that the answer would be different had you been using Indefinite_Vectors. (Those definitely allow the elements to be constrained, as those pretty much have to be allocated individually when they are created.)

Array Sliding

From: Jerry <list_email@icloud.com>
Date: Fri, 24 Nov 2017 03:42:06 -0800
Subject: How to access an array using two different indexing schemes
Newsgroups: comp.lang.ada

I want to access an array such as Real_Vector (built-in for Ada >= 2005) with two different indexing schemes. For example, I would access a Real_Vector indexed (0 .. 4) when thinking of it as a times series and the same data indexed (1 .. 5) when thinking of it as a vector. Another application would be a vector indexed (-128 .. 127) because it fits my problem domain, perhaps a spatial variable, but I need to index it as (0 .. 255) when thinking of doing a Fast Fourier Transform on it.

[...]

I'm currently looking at System.Address_To_Access_Conversions but I thought I would query the list in the meantime.

From: Robert A Duff
<bobduff@TheWorld.com>
Date: Fri, 24 Nov 2017 17:12:04 -0500
Subject: Re: How to access an array using two different indexing schemes
Newsgroups: comp.lang.ada

> [...]

This is called "sliding". So long as X'Length = Y'Length, the bounds can change in this way on assignment statements.

I don't think the RM defines the term "sliding", but you can find it in the AARM, as part of array subtype conversions.

> but this has two problems. First, I have wasted memory and cycles by declaring y and copying x into it. ...

Sliding is also allowed in various other contexts, including parameter passing, which is typically done by reference for the types you're interested in.

Here's an example. P expects a String starting at 5. Q is a wrapper, that takes a String with any lower bound, and slides it as appropriate, and then passes it to P. The program prints " 5 5 5 5".

Strings with various lower bounds are passed to Q, but they all have 'First = 5 inside P. And there's no copying of the strings (on any sensible compiler).

This is admittedly kind of convoluted.

```
 with Text_IO; use Text_IO;
 package Sliding is
   subtype String_1 is String with
      Predicate => String_1'First = 1;
   subtype String_5 is String with
      Predicate => String_5'First = 5;
   procedure P (X : String_5);
   procedure Q (X : String);
 end Sliding;
 package body Sliding is
   procedure P (X : String_5) is
   begin
     Put (X'First'Image);
   end P;
   procedure Q(X: String) is
     subtype Slide_To_5 is
             String_5 (5 .. 5 + X'Length - 1);
     procedure Call_P (X : Slide_To_5) is
```

```
    begin
      P (X); -- Predicate check
           -- (that X'First = 5) occurs here.
    end Call_P;
  begin
    Call_P (X); -- Sliding to 5..something
                 -- occurs here.
  end Q;
end Sliding;
procedure Sliding.Main is
  S1 : constant String := "";
  S2 : constant String
         (-100 .. -1_000_000) := "";
  S3 : constant String := "Hello";
  S4 : constant
        String (101 .. 105) := "Hello";
begin
  Q (S1);
  Q (S2);
  Q (S3);
  Q (S4);
end Sliding.Main;
```

From: Randy Brukardt
<randy@rrsoftware.com>
Date: Mon, 27 Nov 2017 19:31:09 -0600
Subject: Re: How to access an array using two different indexing schemes
Newsgroups: comp.lang.ada

[...]

If you used the Address clause hack here, you'd be subject to termination. GIGO.

Proven Test Solutions for Reliable Embedded Software

VECTOR software

vectorcast.com

"VectorCAST is unique in that it provides us with the ability to increase the reliability and quality of our flight software."

-- Honeywell

VectorCAST is a TÜV SÜD Certified Software Tool for Safety Related Development

Vector Software, Inc.
Golden Cross House | 8 Duncannon Street | London WC2N 4JF UK | +44 203 603 0120 | sales@vectorcast.com

Conference Calendar

Dirk Craeynest

KU Leuven. 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.

2018

☺ January 08-13 45th ACM SIGPLAN **Symposium on Principles of Programming Languages** (POPL'2018), Los Angeles, California, USA.

January 08-09 ACM SIGPLAN **Workshop on Partial Evaluation and Program Manipulation** (PEPM'2018). Topics include: semantics based program synthesis and program optimisation; program and model manipulation techniques (such as: partial evaluation, slicing, symbolic execution, refactoring, ...); techniques that treat programs/models as data objects (including: metaprogramming, generative programming, embedded domain-specific languages, model-driven program generation and transformation, ...); program analysis techniques that are used to drive program/model manipulation (such as: abstract interpretation, termination checking, type systems, test case generation, ...); application of the above techniques (including case studies of program manipulation in real-world (industrial, open-source) projects and software development processes, descriptions of robust tools capable of effectively handling realistic applications, benchmarking; etc.

January 16-19 10th **Software Quality Days Conference** (SWQD'2018), Vienna, Austria. Theme: "Software Quality 4.0: Advanced Methods and Tools for better Software and Systems". Topics include: improvement of software development methods and processes; testing and quality assurance of software and software-intensive systems; domain specific quality issues such as embedded, medical, automotive systems; novel trends in software quality; etc.

Jan 29 – Feb 02 44th **International Conference on Current Trends in Theory and Practice of Computer Science** (SOFSEM'2018), Krems an der Donau, Austria. Topics include: foundations of computer science, software engineering, and data and knowledge-based systems.

☺ Jan 31 – Feb 02 9th **Embedded Real Time Software and Systems** (ERTS2'2018), Toulouse, France. Topics include: embedded computing platforms and networked systems (multi-core / many-core platforms, middleware, ...); processes, methods and tools (agile techniques, model-based system engineering, formal methods, product line engineering, new programming and verification languages. ...); dependability (safety, security, quality of service, fault tolerance, maintainability, certification, ...); etc.

♦ February 03 **Ada Developer Room at FOSDEM 2018**. Brussels, Belgium. FOSDEM 2018 is a two-day event (Sat 3 - Sun 4 Feb). This years' edition includes once more a full-day Ada Developer Room, organized by Ada-Belgium in cooperation with Ada-Europe, which will be held on Saturday 3 February.

February 07-09 12th **International Workshop on Variability Modelling of Software-Intensive Systems** (VaMoS'2018), Madrid, Spain. Topics include: variability across the software life cycle; runtime variability approaches; variability in software architecture; managing variability at post-deployment time; formal verification, testing, and debugging of variable software systems; refactoring and evolution of variable software systems; Reverse engineering approaches; formal reasoning and automated analysis on variability; software economic aspects of variability; etc.

February 09-11 11th **India Software Engineering Conference** (ISEC'2018), Hyderabad, India. Topics include: safety-critical software, program specification and modeling languages, engineering for quality requirements, model driven development, software evolution and maintenance, system verification and validation, etc.

February 21-24 49th ACM **Technical Symposium on Computer Science Education** (SIGCSE'2018), Baltimore, Maryland, USA.

February 24-25 27th **International Conference on Compiler Construction** (CC'2018), Vienna, Austria. Topics include: work on processing programs in the most general sense; compilation and interpretation techniques; run-time techniques (memory management, virtual machines, ...); programming tools (refactoring editors, checkers, verifiers, compilers, debuggers, profilers, ...); techniques for specific domains (secure, parallel, distributed, embedded, ...); design and implementation of novel language constructs, programming models, and domain-specific languages; etc.

☺ February 24-28 23th ACM SIGPLAN **Annual Symposium on Principles and Practice of Parallel Programming** (PPoPP'2018), Vienna, Austria. Topics include: all aspects of parallel programming, including theoretical foundations, techniques, languages, compilers, runtime systems, tools, and practical experience; such as compilers and runtime systems, concurrent data structures, development, analysis, or management tools, formal analysis and verification, parallel programming languages, programming tools for parallel systems, software engineering for parallel programs, synchronization and concurrency control, etc.

March 19-22 24th **International Working Conference on Requirements Engineering - Foundation for Software Quality** (REFSQ'2018), Utrecht, the Netherlands. Deadline for submissions: January 15, 2018 (workshop papers).

March 20-23 25th IEEE **International Conference on Software Analysis, Evolution, and Reengineering** (SANER'2018), Campobasso, Italy. Topics include: 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 maintenance and evolution; experience reports; education; tools and methods; etc. Deadline for submissions: January 5, 2018 (early research achievements (ERA) abstracts, industrial abstracts, tool abstracts, REproducibility studies and NEgative results (RENE) abstracts), January 5&12, 2018 (workshop abstracts), January 12, 2018 (early research achievements (ERA) papers, industrial papers, tool papers, REproducibility studies and NEgative results (RENE) papers), January 12&19, 2018 (workshop papers), January 26, 2018 (Journal First (J1C2) submissions)

March 21-23 26th Euromicro **International Conference on Parallel, Distributed and Network-Based Processing** (PDP'2018), Cambridge, UK. Topics include: embedded parallel and distributed systems, multi- and many-core systems, programming languages and environments, runtime support systems, performance prediction and analysis, shared-memory and message-passing systems, dependability and survivability, real-time distributed applications, etc.

☺ April 09-12 **The Art, Science, and Engineering of Programming Conference** (Programming'2018), Nice, France. Topics include: everything to do with programming, including the experience of programming; general-purpose programming; distributed systems programming; parallel and multi-core programming; security programming; interpreters, virtual machines and compilers; modularity and separation of concerns; model-based development; testing and debugging; program verification; programming education; programming environments; etc.

April 09-13 33rd ACM **Symposium on Applied Computing** (SAC'2018), Pau, France.

- ☺ April 09-13 **Track on Object-Oriented Programming Languages and Systems** (OOPS'2018). Topics include: aspects and components; code generation, and optimization; distribution and concurrency; evaluation; formal verification; Internet of Things technology and programming; integration with other paradigms; interoperability, versioning and software evolution and adaptation; language design and implementation; modular and generic programming; runtime verification and monitoring; safe, secure and dependable software; static analysis; testing and debugging; type systems; etc.
- ☺ April 09-13 **Track on Programming Languages** (PL'2018). Topics include: 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, programming languages from all paradigms, etc. Deadline for submissions: September 25, 2016 (full papers).

April 09-13 **Track on Software Verification and Testing** (SVT'2018). 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, 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.

April 09-13 13th **Track on Dependable, Adaptive, and Trustworthy Distributed Systems** (DADS'2018). Topics include: Dependable, Adaptive, and trustworthy Distributed Systems (DADS); middleware for DADS; modeling, design, and engineering of DADS; foundations and formal methods for DADS; etc.

April 09-13 9th ACM/SPEC **International Conference on Performance Engineering** (ICPE'2018), Berlin, Germany. Theme: "Continuous Performance Assurance in Agile Delivery". Deadline for submissions: January 3, 2018 (posters/demos), January 10, 2018 (work-in-progress/vision papers).

April 09-13 11th IEEE **International Conference on Software Testing, Verification and Validation** (ICST'2018), Västerås, Sweden. Topics include: experience reports, formal verification, model checking, security testing, software reliability, testing in specific domains (such as embedded, concurrent, distributed, real-time, ..., systems), testing/debugging tools, etc. Deadline for submissions: January 12, 2018 (workshop papers), January 31, 2018 (PhD Symposium).

April 10-13 11th **Cyber-Physical Systems Week** (CPS Week'2018), Porto, Portugal.

☺ April 11-13 24th IEEE **Real-Time and Embedded Technology and Applications Symposium** (RTAS'2018). Topics include: timing issues ranging from traditional hard real-time systems to latency-sensitive systems with soft real-time requirements; original systems and applications, case studies, methodologies and applied algorithms that contribute to the state of practice in the design, implementation and verification of real-time systems; embedded, networked and cyber-physical systems that consider real-time aspects; etc.

April 11-13 9th ACM/IEEE **International Conference on Cyber-Physical Systems** (ICCPS'2018). Topics include: development of technologies, tools, and architectures for building CPS systems; design, implementation, and investigation of CPS applications; secure and resilient CPS infrastructure; etc.

April 14-21 21st European **Joint Conferences on Theory and Practice of Software** (ETAPS'2018), Thessaloniki, Greece. Events include: ESOP (European Symposium on Programming), FASE (Fundamental Approaches to Software Engineering), FoSSaCS (Foundations of Software Science and Computation Structures), POST (Principles of Security and Trust), TACAS (Tools and Algorithms for the Construction and Analysis of Systems), SV-COMP (7th Competition on Software Verification).

April 17-19 10th NASA **Formal Methods Symposium** (NFM'2018), Newport News, Virginia, USA. Topics include: identify challenges and provide solutions for achieving assurance for critical systems; model checking, static analysis, use of formal methods in software and system testing, compositional techniques, parallel and/or distributed techniques, safety cases and system safety, fault tolerance, model-based development, etc.

♦ April 18-20 19th **International Real-Time Ada Workshop** (IRTAW'2018), Benicàssim, Spain. Deadline for submissions: February 4, 2018 (position papers).

April 23-27 21st **Ibero-American Conference on Software Engineering** (CIbSE'2018), Bogotá, Colombia. Event includes Software Engineering Track (SET) and Experimental Software Engineering Track (ESELAW). Deadline for submissions: February 5, 2018 (doctoral symposium).

Apr 30 – May 04 2nd **International Conference on Software Architecture** (ICSA'2018), Seattle, USA. Topics include: model driven engineering for continuous architecting; component based software engineering and architecture design; re-factoring and evolving architecture design decisions and solutions; architecture frameworks and architecture description languages; preserving architecture quality throughout the system lifetime; software architecture for legacy systems and systems integration; architecting families of products; software architects roles and responsibilities; training, education, and certification of software architects; industrial experiments and case studies; bold arguments against current research directions and results; results that challenge established results or beliefs giving evidence that call for fundamentally new directions, open up new research avenues where software architecture research can contribute; etc. Deadline for submissions: January 18, 2018 (technical abstracts), January 25, 2018 (full technical papers), March 8, 2018 (New and Emerging Ideas, engineering track, Early Career Researchers Forum abstracts, workshop papers), March 9, 2018 (tutorials), March 15, 2018 (New and Emerging Ideas, engineering track, Early Career Researchers Forum papers).

May 21-23 17th **International Conference on Software Reuse** (ICSR'2018), Madrid, Spain. Theme: "New Opportunities for Software Reuse". Topics include: component-based reuse techniques, generative reuse, systematic reuse approaches helping industries transitioning from ad-hoc approaches, reverse engineering of potentially reusable components, evolution and maintenance of reusable assets, development of reusable components for Product Line Engineering, software variability approaches for configuring and deriving reusable assets, dynamic aspects of reuse (i.e post-deployment time), etc. Deadline for submissions: January 29, 2018 (workshops, tutorials, doctoral symposium papers, tool demos).

May 21-25 19th **International Conference on Agile Software Development** (XP'2018), Porto, Portugal. Deadline for submissions: January 6, 2018 (experience reports), January 20, 2018 (research papers, research posters, Doctoral Symposium plans, industry & practice sessions, tools & demos, tutorials & workshops), January 29, 2018 (Agile in Education & Training sessions).

May 21-25 32nd IEEE **International Parallel and Distributed Processing Symposium** (IPDPS'2018), Vancouver, Canada.

May 27 – Jun 03 40th **International Conference on Software Engineering** (ICSE'2018), Gothenburg, Sweden. Deadline for submissions: January 8, 2018 (ACM Student Research Competition), January 15, 2018 (student contest on Software Engineering), January 22, 2018 (student volunteers), February 5, 2018 (posters).

May 27-29 13th IEEE/ACM **International Conference on Global Software Engineering** (ICGSE'2018), Gothenburg, Sweden. Deadline for submissions: January 15, 2018 (experience report abstracts), January 29, 2018 (research papers), February 15, 2018 (Doctoral Symposium submissions, industry talk proposals).

☺ May 29-31 21st IEEE **International Symposium On Real-Time Computing** (ISORC'2018), Singapore. Topics include: object/component/service-oriented real-time distributed computing (ORC) technology, programming and system engineering (real-time programming challenges, ORC paradigms, languages, ...), trusted and dependable systems, system software (real-time kernels, middleware support for ORC, extensibility, synchronization, scheduling, fault tolerance, security, ...), applications (medical devices, intelligent transportation systems, industrial automation systems, Internet of Things and Smart Grids, embedded systems in automotive, avionics, consumer electronics, ...), system evaluation (performance analysis, monitoring & timing, dependability, fault detection and recovery time, ...), cyber-physical systems, etc. Deadline for submissions: February 2, 2018.

June 11-15 30th **International Conference on Advanced Information Systems Engineering** (CAiSE'2018), Tallin, Estonia. Theme: "Information Systems in the Big Data Era". Topics include: methods, models, techniques, architectures and platforms for supporting the engineering and evolution of information systems and organizations in the big data era. Deadline for submissions: March 4, 2018 (forum).

♦ June 18-22 23rd **International Conference on Reliable Software Technologies - Ada-Europe'2018**, Lisbon, Portugal. Sponsored by Ada-Europe, in cooperation (pending) with ACM SIGAda, SIGPLAN, and the Ada Resource Association (ARA). Deadline for submissions: January 22, 2018 (regular papers, industrial presentation outlines, tutorials, workshops).

June 25-29 **Software Technologies: Applications and Foundations** (STAF'2018), Toulouse, France. Successor of the TOOLS federated event. Topics include: practical and foundational advances in software technology, such as object-oriented design, testing, formal approaches to modelling and verification, transformation, model-driven engineering, aspect-oriented techniques, and tools. Deadline for submissions: January 19, 2018 (workshops), April 20, 2018 (workshop papers).

June 27-29 16th **International Conference on Software Engineering and Formal Methods** (SEFM'2018). Topics include: light-weight and scalable formal methods; software evolution, maintenance, re-engineering and reuse; programming languages; abstraction and refinement; correctness-by-construction; model checking; verification and validation; testing; safety-critical, fault-tolerant and secure systems; software certification; real-time and embedded systems; application and technology transfer; case studies, best practices and experience reports; tool integration; education; etc. Deadline for submissions: February 23, 2018 (abstracts), March 2, 2018 (full papers).

June 27-29 12th **International Conference on Tests And Proofs** (TAP'2018). Topics include: many aspects of verification technology, including foundational work, tool development, and empirical research; the connection between proofs (and other static techniques) and testing (and other dynamic techniques); verification and analysis techniques combining proofs and tests; program proving with the aid of testing techniques; deductive techniques supporting the automated generation of test vectors and oracles; deductive techniques supporting novel definitions of coverage criteria; program analysis techniques combining static and dynamic analysis; testing and runtime analysis of formal specifications; verification of verification tools and environments; applications of test and proof techniques in new domains, such as security, configuration management, learning; combined approaches of test and proof in the context of formal certifications (Common Criteria, CENELEC, ...); case studies, tool and framework descriptions, and experience reports about combining tests and proofs; etc. Deadline for submissions: February 23, 2018 (abstracts), March 2, 2018 (papers).

June 25-29 12th ACM **International Conference on Distributed Event-Based Systems** (DEBS'2018), Hamilton, New Zealand. Topics include: systems dealing with detecting, processing and responding to events and with massively distributed middleware and applications, real-time analytics, complex event processing, distributed programming, fault tolerance, reliability, availability, scalability, internet of things, cyber-physical systems, transportation, enterprise application integration, etc. Deadline for submissions: February 21, 2018 (research abstracts), February 26, 2018 (full research papers).

June 27-29 22nd **International Conference on Evaluation and Assessment in Software Engineering** (EASE'2018), Christchurch, New Zealand. Deadline for submissions: January 21, 2018 (full paper abstracts), January 28, 2018 (full papers), March 18, 2018 (impact-to-industry papers, short papers), March 25, 2018 (Emerging Researchers' Forum papers).

☺ July 03-06 30th **Euromicro Conference on Real-Time Systems** (ECRTS'2018), Barcelona, Spain. Topics include: all aspects of real-time systems, such as scheduling design and analysis, real-time operating systems, hypervisors and middlewares, virtualization and timing isolation, mixed-criticality design & assurance, worst-case execution time analysis, modelling and/or formal methods, industrial use-cases and real-time applications, tools, compilers and benchmarks for embedded systems, etc. Event includes: Worst-Case Execution Time analysis (WCET), Workshop on Analysis Tools and methodologies for Embedded and Real-time Systems (WATERS). Deadline for submissions: February 1, 2018 (papers).

July 14-17 30th **International Conference on Computer-Aided Verification** (CAV'2018), Oxford, UK. Topics include: theory and practice of computer-aided formal analysis methods for hardware and software systems, algorithms and tools for verifying models and implementations, specifications and correctness criteria for programs and systems, deductive verification using proof assistants, program analysis and software verification, formal methods for cyber-physical systems, verification methods for parallel and concurrent systems, testing and run-time analysis based on verification technology, applications and case studies in verification and synthesis, verification in industrial practice, formal models and methods for security, etc. Deadline for submissions: January 31, 2018 (papers).

July 15-16 22nd **International Symposium on Formal Methods** (FM'2018), Oxford, UK. Topics include: formal methods for the engineering of computer-based systems and software; such as industrial applications of formal methods; experience with formal methods in industry; tool usage reports; advances in automated verification, model-checking, and testing with formal methods; tools integration; environments for formal methods; development processes with formal methods; usage guidelines for formal methods; etc. Deadline for submissions: January 8, 2018 (abstracts), January 22, 2018 (papers).

July 16-20 18th IEEE **International Conference on Software Quality, Reliability and Security** (QRS'2018), Lisbon, Portugal. Topics include: reliability, security, availability, and safety of software systems; software testing, verification and validation; program debugging and comprehension; fault tolerance for software reliability improvement; modeling, prediction, simulation, and evaluation; metrics, measurements, and analysis; software vulnerabilities; formal methods; benchmark, tools, and empirical studies; etc. Deadline for submissions: January 15, 2018 (workshops), March 1, 2018 (abstracts), March 8, 2018 (regular and short papers), April 1, 2018 (workshop papers), May 1, 2018 (Student Doctoral Program, fast abstracts, industry track).

☺ July 16-22 32nd **European Conference on Object-Oriented Programming** (ECOOP'2018), Amsterdam, the Netherlands.

July 23-27 42nd Annual IEEE **Conference on Computer Software and Applications** (COMPSAC'2018), Tokyo, Japan. Deadline for submissions: January 15, 2018 (papers), April 10, 2018 (workshop papers).

August 29-30 44th Euromicro **Conference on Software Engineering and Advanced Applications** (SEAA'2018), Prague, Czech Republic. Topics include: information technology for software-intensive systems; tentative conference tracks on DSLs and Model-Based Development (DSLMBD), Software Process and Product Improvement (SPPI), etc.; tentative special sessions on Cyber-Physical Systems (CPS), SEaTeD: Software Engineering and Technical Debt, MoLS: Monitoring Large-Scale Software Systems, etc. Deadline for submissions: February 22, 2018 (abstracts), March 1, 2018 (papers).

August 29-31 12th **International Symposium on Theoretical Aspects of Software Engineering** (TASE'2018), Guangzhou, China. Topics include: theoretical aspects of software engineering, such as abstract interpretation, component-based software engineering, cyber-physical systems, distributed and concurrent systems, embedded and real-time systems, formal verification and program semantics, integration of formal methods, language design, model checking and theorem proving, model-driven engineering, object-oriented systems, program analysis, reverse engineering and software maintenance, run-time verification and monitoring, software architectures and design, software testing and quality assurance, software safety, security and reliability, specification and verification, type systems, tools exploiting theoretical results, etc. Deadline for submissions: February 23, 2018 (abstracts), March 2, 2018 (papers).

September 09-12 **Federated Conference on Computer Science and Information Systems** (FedCSIS'2018), Poznan, Poland.

☺ November ACM SIGAda's **High Integrity Language Technology** International Workshop on Cyber-Security Interaction with High Integrity (HILT'2018), Boston area, Massachusetts, USA.

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

Tools to get you there. Safely.
Ada and The GNAT Pro High-Integrity Family
www.adacore.com
AdaCore
The GNAT Pro Company

Call for Participation
8th Ada Developer Room at FOSDEM 2018
Saturday 3 February 2018, Brussels, Belgium

Organized by Ada-Belgium
in cooperation with Ada-Europe

FOSDEM[1], the Free and Open source Software Developers' European Meeting, is a non-commercial two-day weekend event organized early each year in Brussels, Belgium. It is highly developer-oriented and brings together 8000+ participants from all over the world. The 2018 edition takes place on Saturday 3 and Sunday 4 February. It is free to attend and no registration is necessary.

In this edition, Ada-Belgium[2] organizes once more a series of presentations related to Ada and Free or Open Software in a s.c. Developer Room. The "Ada DevRoom" at FOSDEM 2018 is held on the first day of the event. The program offers introductory presentations on the Ada programming language, as well as more specialised presentations on focused topics, tools and projects.

Program overview:

- *Welcome*, by Dirk Craeynest, Ada-Belgium
- *An Introduction to Ada for Beginning and Experienced Programmers*, by Jean-Pierre Rosen, Adalog
- *Making the Ada_Drivers_Library: Embedded Programming with Ada*, by Fabien Chouteau, AdaCore
- *Shared Memory Parallelism in Ada: Load Balancing by Work Stealing*, by Jan Verschelde, University of Illinois at Chicago
- *Ada, or How to Enforce Safety Rules at Compile Time*, by Jean-Pierre Rosen, Adalog
- *Contract-based Programming: a Route to Finding Bugs Earlier*, by Jacob Sparre Andersen, JSA Research & Innovation
- *SPARK Language: Historical Perspective & FOSS Development*, by Yannick Moy, AdaCore
- *Writing REST APIs with OpenAPI and Swagger Ada*, by Stephane Carrez, Bouygues Telecom
- *Browser-as-GUI and Web Applications with Gnoga*, by Jeffrey R. Carter, Atos Belgium
- *Easy Ada Tooling with Libadalang*, by Raphael Amiard & Pierre-Marie De Rodat, AdaCore

The Ada at FOSDEM 2018 web-page has all details, such as the full schedule, abstracts of presentations, biographies of speakers, and pointers to more info. For the latest information at any time, contact <Dirk.Craeynest@cs.kuleuven.be>, or see:

http://www.cs.kuleuven.be/~dirk/ada-belgium/events/18/180203-fosdem.html

[1] https://fosdem.org/2018
[2] http://www.cs.kuleuven.be/~dirk/ada-belgium

19th International Real-Time Ada Workshop – IRTAW 2018

Hotel Voramar, Benicàssim, Spain
18-20 April 2018
http://www.ada-europe.org/irtaw2018

Call for Papers

The International Real-Time Ada Workshop series has provided a forum for identifying issues with real-time system support in Ada and for exploring possible approaches and solutions, and has attracted participation from key members of the research, user, and implementer communities worldwide. Recent International Real-Time Ada Workshop meetings contributed to the Ada 2005/Ada 2012 standards, especially with respect to the tasking features, the real-time and high-integrity systems annexes, and the standardization of the Ravenscar Tasking Profile.

In keeping with this tradition, the goals of the 19th edition of IRTAW will be to:

- Review Ada 2012 Issues *vis-a-vis* real-time systems;
- Examine experiences in using Ada 2012 for real-time systems and applications;
- Implementation approaches for Ada 2012 real-time features;
- Consider developing other real-time Ada profiles in addition to the Ravenscar profile;
- Implications to Ada with multiprocessors in development of real-time systems;
- Paradigms for using Ada for real-time distributed systems, with special emphasis on robustness as well as hard, flexible and application-defined scheduling;
- Analysis of specific patterns and libraries for real-time systems development in Ada;
- Ada in context of the certification of safety-critical and/or security-critical real-time systems;
- Examine the Real-Time Specification for Java and other languages for real-time systems development, their current implementations and their interoperability with Ada in embedded real-time systems;
- Industrial experience with Ada and the Ravenscar Profile in real-time projects;
- Consider the language vulnerabilities of the Ravenscar and full language definitions;
- Consider testing for compliance with the Real-Time Annex.

Participation at the 19th IRTAW is by invitation following the submission of a position paper addressing one or more of the above topics or related real-time Ada issues. Alternatively, anyone wishing to receive an invitation, but for one reason or another is unable to produce a position paper, may send in a one-page position statement indicating their interests. Priority will be given to submitted papers.

Submission Requirements

Position papers should not exceed ten pages in typical IEEE conference layout, excluding code inserts. All accepted papers will appear, in their final form, in the Workshop Proceedings, which will be published as a special issue of Ada Letters (ACM Press). Selected papers will also appear in the Ada User Journal. Authors with a relevant paper submitted to the 23rd International Conference on Reliable Software Technologies – Ada-Europe 2018 (deadline 24 January, 2018) may offer an extended abstract of the same material to IRTAW. Please submit position papers, in PDF format, to the Program Chair by e-mail: *brad.moore@shaw.ca*

Important Dates

Paper Submission: **4 February, 2018**
Notification of Acceptance: 23 February, 2018
Confirmation of Attendance: 9 March, 2018
Final Paper Due: 30 March, 2018
Workshop: April 18-20, 2018

Program Chair

Brad Moore, *General Dynamics Mission Systems, Canada*

Workshop Chair

Jorge Real *Universitat Politècnica de València, Spain*

Ada-Europe 2018

23rd International Conference on Reliable Software Technologies
18-22 June 2018, Lisbon, Portugal

Conference Chair

Nuno Neves
nuno@di.fc.ul.pt
LASIGE/U. Lisboa, Portugal

Program Chair

António Casimiro
casim@ciencias.ulisboa.pt
LASIGE/U. Lisboa, Portugal

Special Session Chair

Marcus Völp
marcus.voelp@uni.lu
University of Luxembourg, Luxembourg

Tutorial and Workshop Chair

David Pereira
dmrpe@isep.ipp.pt
CISTER/ISEP, Portugal

Industrial Co-Chairs

Marco Panunzio
marco.panunzio@thalesaleniaspace.com
Thales Alenia Space, France

José Rufino
ruf@ciencias.ulisboa.pt
LASIGE/U. Lisboa, Portugal

Publication Chair

Pedro Ferreira
pmf@ciencias.ulisboa.pt
LASIGE/U. Lisboa, Portugal

Exhibition Co-Chairs

José Neves
jose.neves@gmv.com
GMV, Portugal

Ahlan Marriott
ahlan@Ada-Switzerland.ch
White Elephant GmbH, Switzerland

Publicity Chair

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

Local Secretariat

Madalena Almeida
madalena.almeida@abreu.pt
Viagens Abreu S.A., Portugal

General Information

The **23rd International Conference on Reliable Software Technologies – Ada-Europe 2018** will take place in Lisbon, Portugal. Following its traditional style, the conference will span a full week, including a three-day technical program and vendor exhibition from Tuesday to Thursday, along with parallel tutorials and workshops on Monday and Friday.

Schedule

22 January 2018	Submission of papers, industrial presentation outlines, and tutorial and workshop proposals.
9 March 2018	Notification of acceptance to all authors
24 March 2018	Camera-ready version of papers required
8 May 2018	Industrial presentations, tutorial and workshop material required

Topics

The conference 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 allow ample time for keynotes, Q&A sessions and discussions, and social events. Participants include practitioners and researchers representing industry, academia and government organizations active in the promotion and development of reliable software technologies.

This edition of Ada-Europe features a focused **Special Session on Security in Safety-Critical Systems**. Safety-critical systems, on which we daily bet our lives, have become increasingly more complex, networked and distributed. In combination with the growing professionalism of adversarial teams, this demands not only for safe systems but systems that remain safe while under attacks. This session seeks (but is not limited to) contributions aiming at bridging the safety and security gap in cyber-physical and other safety-critical systems. Topics include: **Software and System Aspects of Secure and Dependable CPS**, **Vulnerabilities and Protective Measures for Safety-Critical System Infrastructures**, and **Fault and Intrusion Tolerance and Long-Term Unattended Operation for Safety-Critical Systems**. For further information please contact the Special Session Chair directly.

For the **general track of the conference**, topics of interest include but are not limited to (full list on the website): Real-Time and Embedded Systems, Mixed-Criticality Systems, Theory and Practice of High-Integrity Systems, Software Architectures, Methods and Techniques for Software Development and Maintenance, Formal Methods, Ada Language and Technologies, Software Quality, Mainstream and Emerging Applications, Experience Reports in Reliable System Development, Experiences with Ada.

http://www.ada-europe.org/conference2018

Program Committee

Mario Aldea, *Univ. de Cantabria, SP*
Ezio Bartocci, *Vienna Univ. of Technology, AT*
Johann Blieberger, *Vienna Univ. of Tech., AT*
Rakesh Bobba, *Oregon State Univ., USA*
Bernd Burgstaller, *Yonsei Univ., KO*
António Casimiro, *LASIGE, Univ. Lisboa, PT*
Juan A. de la Puente, *Univ. Pol. de Madrid, SP*
Virgil Gligor, *Carnegie Mellon University, USA*
M. González Harbour, *Univ. de Cantabria, SP*
J. Javier Gutiérrez, *Univ. de Cantabria, SP*
Jérôme Hugues, *ISAE, FR*
Ruediger Kapitza, *Tech Univ. Braunschweig, DE*
Hubert Keller, *Karlsruhe Inst. of Technology, DE*
Raimund Kirner, *Univ. of Hertfordshire, UK*
Adam Lackorzynski, *TU Dresden & Kernkonzept GmbH, DE*
Kristina Lundkvist, *Mälardalen Univ., SE*
Franco Mazzanti, *ISTI-CNR, IT*
Laurent Pautet, *Telecom ParisTech, FR*
Luís Miguel Pinho, *CISTER, ISEP, PT*
Erhard Plödereder, *Univ. Stuttgart, DE*
Jorge Real, *Univ. Politècnica de València, SP*
José Ruiz, *AdaCore, FR*
Sergio Sáez, *Univ. Politècnica de València, SP*
Elad Schiller, *Chalmers Univ. of Technology, SE*
Frank Singhoff, *Univ. Bretagne Occidentale, FR*
Jorge Sousa Pinto, *Univ. of Minho, PT*
Tucker Taft, *AdaCore, USA*
Elena Troubitsyna, *Åbo Akademi Univ., FI*
Santiago Urueña, *GMV, SP*
Tullio Vardanega, *Univ. di Padova, IT*
Marcus Völp, *Univ. of Luxembourg, LU*

Industrial Committee

Ian Broster, *Rapita Systems, UK*
Luís Correia, *EMPORDEF-TI, PT*
Dirk Craeynest, *Ada-Belgium & KU Leuven, BE*
Thomas Gruber, *Austrian Inst. of Tech, AT*
Andreas Jung, *European Space Agency, NL*
Ismael Lafoz, *Airbus Defence and Space, SP*
Ahlan Marriott, *White Elephant, CH*
Maurizio Martignano, *Spazio IT, IT*
Marco Panunzio, *Thales Alenia Space, FR*
Paul Parkinson, *Wind River, UK*
Jean Pierre Rosen, *Adalog, FR*
José Rufino, *LASIGE, Univ. Lisboa, PT*
Emilio Salazar, *GMV, SP*
Helder Silva, *EDISOFT, PT*
Jacob Sparre Andersen, *JSA Consulting, DK*
Andreas Wortmann, *OHB System, DE*

In cooperation with
ACM SIGAda, SIGPLAN, SIGBED

and
Ada Resource Association (ARA)

Call for Regular and Special Session Papers

Authors of papers which are to undergo peer review for acceptance are invited to submit original contributions by 22 January 201. Paper submissions shall be 14 LNCS-style pages in length. Authors for both the general track and the special session shall submit their work via EasyChair at https://easychair.org/conferences/?conf=adaeurope2018. The format for submission is solely PDF.

The International Conference on Reliable Software Technologies is listed in DBLP, SCOPUS, Web of Science Conference Proceedings Citation index, Google Scholar and Microsoft Academic Search, among others.

Proceedings

The conference proceedings will be published in the Lecture Notes in Computer Science (LNCS) series by Springer, and will be available at the conference. Camera-ready accepted papers must be in conformance with the LNCS style, not exceeding 14 pages and are due strictly by 24 March 201. For format and style guidelines authors should refer to http://www.springer.de/comp/lncs/authors.html. Failure to comply and to register for the conference by that date will prevent the paper from appearing in the proceedings.

Call for Industrial Presentations

The conference seeks industrial presentations which deliver value and insight but may not fit the selection process for regular papers. Authors are invited to submit a presentation outline of at least one page in length by 22 January 201, at https://easychair.org/conferences/?conf=adaeurope2018. The format for submission is solely PDF.

The Industrial Committee will review the submissions and make the selection. The authors of selected presentations shall prepare a final short abstract and submit it by 8 May 201, aiming at a 20-minute talk. Authors will be also invited to submit corresponding articles for publication in the *Ada User Journal* (http://www.ada-europe.org/auj/), which will host the proceedings of the Industrial Program. For any further information please contact the Industrial Co-chairs directly.

Awards

Ada-Europe will offer honorary awards for the best regular paper and the best presentation.

Call for Tutorials

Tutorials should address subjects that fall within the scope of the conference and may be proposed as either half- or full-day. Proposals should include a title, an abstract, a description of the topic, a detailed outline of the presentation, a description of the presenter's lecturing expertise in general and with the proposed topic in particular, the proposed duration (half day or full day), the intended level of the tutorial (introductory, intermediate, or advanced), the recommended audience experience and background, and a statement of the reasons for attending. Proposals should be submitted by e-mail to the Tutorial Chair. The authors of accepted full-day tutorials will receive a complimentary conference registration as well as a fee for every paying participant in excess of 5; for half-day tutorials, these benefits will be accordingly halved. The *Ada User Journal* will offer space for the publication of summaries of the accepted tutorials.

Call for Workshops

Workshops on themes that fall within the conference scope may be proposed. Proposals may be submitted for half- or full-day events, to be scheduled at either end of the conference week. Workshop proposals should be submitted to the Tutorial and Workshop Chair. The workshop organizer shall also commit to preparing proceedings for timely publication in the *Ada User Journal*.

Call for Exhibitors

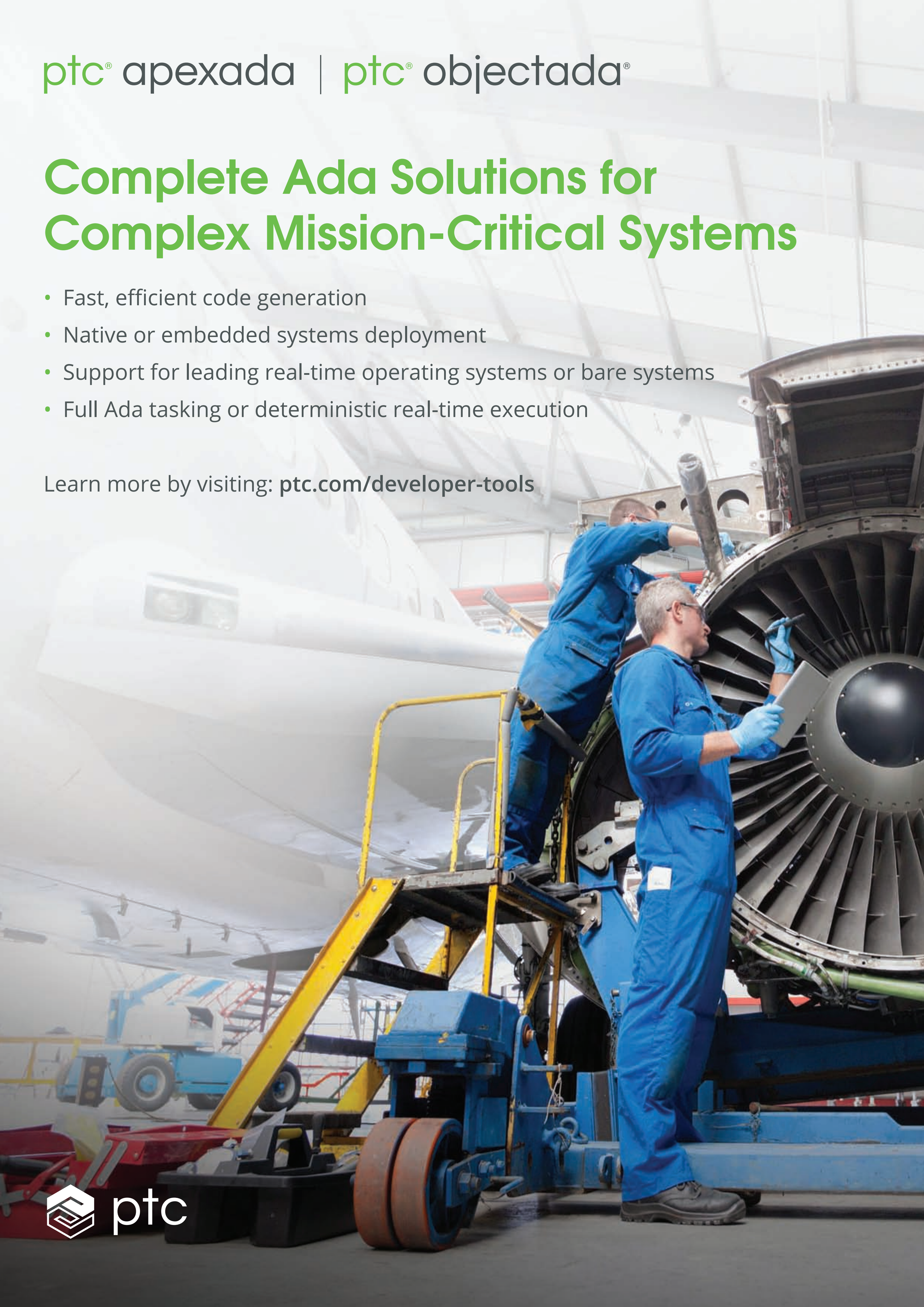
The commercial exhibition will span the three days of the main conference. Vendors and providers of software products and services should contact the Exhibition Chair for information and for allowing suitable planning of the exhibition space and time.

Grants for Reduced Student Fees

A limited number of sponsored grants for reduced fees is expected to be available for students who would like to attend the conference or tutorials. Contact the Conference Chair for details.

Venue

The conference will take place at the VIP Executive Art's Hotel (leftmost image), in the Parque das Nações area of Lisbon (central images), Portugal. June is full of events in Lisbon, including the festivities in honour of St. António, with music, grilled sardines and popular parties in Alfama and Bairro Alto old neighbourhoods, downtown (image on the right). Plan in advance! It is absolutely worth it!



ptc® apexada | ptc® objectada®

Complete Ada Solutions for Complex Mission-Critical Systems

- Fast, efficient code generation
- Native or embedded systems deployment
- Support for leading real-time operating systems or bare systems
- Full Ada tasking or deterministic real-time execution

Learn more by visiting: **ptc.com/developer-tools**

ptc

IP Network Stack in Ada 2012 and the Ravenscar Profile

Stéphane Carrez
Issy Les Moulineaux, France; email: Stephane.Carrez@gmail.com

Abstract

This article presents Ada Embedded Network, a small network stack intended to be used by small embedded Ada applications running on ARM. It implements the standard ARP, IPv4, UDP, DNS and DHCP protocols on top of an Ethernet driver. Its memory efficient design allows it to run on the STM32F746 board.

The article presents the components from the Ada implementation point of view. It highlights the Ada features that have been used and shows some benefits of the Ravenscar profile that have helped the project.

Keywords: Ethernet, IPv4, networking, protocols, Ravenscar.

1 Introduction

The Ada Embedded Network [1] project is an Open Source project that was created as part of the EtherScope MakeWithAda 2016 competition. The goal was to get a reliable, safe and secure network stack for embedded IoT applications.

The network stack has been designed with several goals in mind. First, the objective was to use the Ada 2012 language with a number of new features such as pre- and postconditions in order to make the network stack dependable. Another design goal was to define an architecture that avoids memory copies when sending and receiving packets. Whenever possible the choice was to promote asynchronous operations for sending and receiving network packets. To keep the architecture simple and re-usable, tasks are under the responsibility of the application layer.

The target board was a STM32F746 ARM board with the smallest Ada runtime available still with the ability to use tasks. To make sure the memory footprint is small, no exceptions are used so that we can use the Ravenscar sfp profile defined by Ada Drivers Library [2]. This allows the Ada Embedded Network to have a reasonable footprint as it does not exceed 50kb.

1.1 Outline

This article is structured as follow. Section 2 presents the architecture, which is then used in section 3 around a small echo server example. Section 4 presents the challenges encountered in the project, before concluding in section 5.

2 Architecture

A network stack is made of several layers that deal with specific protocols. The OSI Reference Model [3] depicts the different layers and their roles.

Ada package brings an important feature to set up and organize the architecture of the network stack. Ada package allows providing a representation and an organization that is close to the OSI model and to the available and supported network protocols.

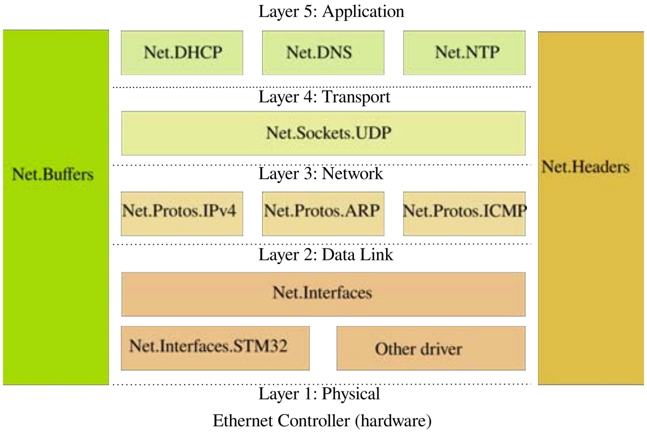


Figure 1: Architecture and Ada packages

Used by all the network protocol stacks the `Net.Buffers` and `Net.Headers` packages provide global data types and abstractions to represent a network packet or a message protocol header. At the bottom of the network stack, the `Net.Interfaces` and `Net.Interfaces.STM32` packages correspond to the data link layer of the OSI model responsible for sending and receiving raw packets. On top of it (transport layer in OSI model) sits the `Net.Protos.IPv4` package that deals with the IPv4 protocol [4], the `Net.Protos.ICMP` package that deals with the ICMP protocol [5] and the `Net.Protos.ARP` [6] package responsible for converting an IPv4 address into an Ethernet address with the ARP protocol.

At the top is the OSI model application layer which provides the `Net.DHCP`, `Net.DNS` and `Net.NTP` application protocols. These three protocols are implemented on top of the UDP protocol layer. The DHCP protocol [7] is used to retrieve the IPv4 network configuration including the IPv4 address, the default gateway and the DNS server address. The DNS protocol [8] is used to resolve a name into an IPv4 or

IPv6 address. Finaly the NTP protocol [9] is used to retrieve the GMT time and have the system clock synchronized with NTP servers.

Ada package dependency constraint was a challenge for the network stack implementation. Very soon dependencies arise between different protocols. The IPv4 package implements the operations to send and receive IP packets. In particular, it provides operations to fill and construct the IP header in packets. The IPv4 layer is also responsible for handling packet reception and dispatching to the upper layer such as UDP.

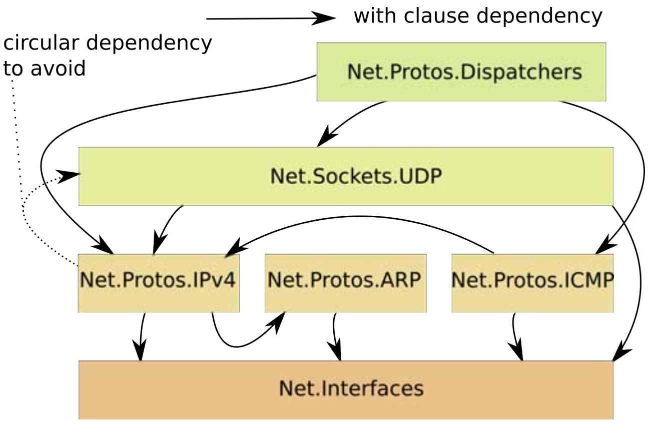


Figure 2: Ada package dependencies

The circular dependency was resolved by introducing the `Net.Protos.Dispatchers` package that provides the packet reception and which handles the dispatching to upper layer protocols.

2.1 Network buffers

Before looking at how the network stack is working and what happens when we send or receive a packet, it is important to understand how the network buffers are represented and controlled.

A network buffer is used by applications to prepare the data to be sent or to receive it. Then, the network stack has to prepend some protocol headers and the low level Ethernet driver has to put the packet in the hardware ring for transmission. One of our goal is to avoid memory copies when sending and receiving packets because copying data will slow down the network stack. Memory is also a scarce resource on embedded systems and the buffer management is key to obtain good performance.

The `Net.Buffers` package provides support for network buffer management. A network buffer can hold a single packet frame so that it is limited to 1500 bytes of payload with 14 or 18 bytes for the Ethernet header. As shown in figure 3, a packet has several parts that are controlled by the different layers of the OSI model. For a UDP packet, the application can fill up to 1458 bytes of data. Several checksums are defined by the Ethernet, IP and UDP protocols. All of them are under the control of the hardware by the STM32F746 chip.

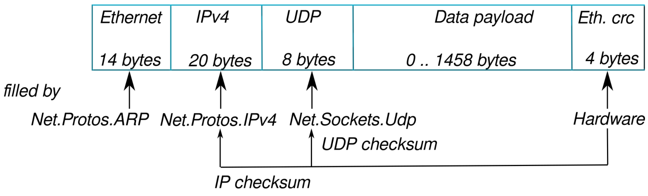


Figure 3: Ethernet Packet Layout

The package defines two important types: `Buffer_Type` and `Buffer_List`. These two types are limited types to forbid copies and force a strict design to applications. The `Buffer_Type` describes the packet frame and it provides various operations to access the buffer. The `Buffer_List` defines a list of buffers.

The network buffers are kept within a single linked list managed by a protected object. The protected operations to allocate and release buffers are in O(1) as the linked lists are used as queues.

An application will allocate a buffer by using the `Allocate` operation as follows:

```
Packet : Net.Buffers.Buffer_Type;
...
  Net.Buffers.Allocate (Packet);

  if Packet.Is_Null then
     null; -- Unsuccessful allocation
  end if;
```

Applications have to check that the buffer was successfully allocated because no exception is raised if there is no available buffer. You have to check if the allocation succeeded by using the `Is_Null` function.

Before receiving a packet, the application has to allocate a network buffer. Upon successful reception of a packet by the `Receive` procedure, the allocated network buffer will be given to the Ethernet receive queue and the application will get back the received buffer. There is no memory copy.

To send a packet, the application will first allocate a buffer, fill the data payload and pass the packet to the network stack. The network stack fills the different protocol specific headers and puts the packet in a transmit queue. When this happens, the application loses the ownership of the packet buffer. This ownership transfer is expressed by a postcondition on the interface driver operation which indicates that the buffer becomes empty.

```
procedure Send (Ifnet : in out Ifnet_Type;
                Buf   : in out Buffer_Type) is abstract
   with Pre'Class => not Buf.Is_Null,
        Post'Class => Buf.Is_Null;
```

2.2 Sending a packet

Let's consider what happens when we send a simple packet. Figure 4 shows the execution flow taken by the UDP packet through the different Ada packages.

The network packet buffer is first allocated by the application layer. The application fills the packet data with the information to send. The `Send` procedure is then invoked in cascade

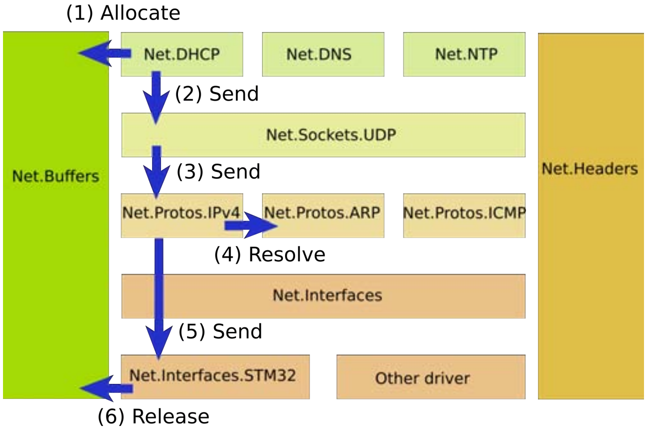


Figure 4: Sending a packet

through different layers to send the data. The UDP layer fills the UDP header of the packet and the IP layer takes care of the IP header. Before sending the packet to the interface layer it is necessary to resolve the IPv4 destination address into an Ethernet address. This is done by the ARP protocol [6]. The `Resolve` procedure takes care of this resolution by first looking at some ARP database table or by performing the ARP resolution when the IPv4 address is unkown. In that case, the packet is added into a queue until the ARP response is received or a timeout occurs.

As soon as the IPv4 address is resolved into an Ethernet address, the packet is sent to the Ethernet driver that puts the packet into the send queue. Once the packet is sent, the buffer is released and put back to the buffer management free queue.

2.3 Receiving a packet

Because the reception of a packet is a blocking operation, a task is necessary to handle the packet reception and this work must be done by the application. Indeed, the choice made for the design and implementation of the Ada Embedded Network was to avoid imposing a task model to handle the network packet reception and instead leave that to the application. This allows an application to choose different strategies to handle the network packet reception. For example, an application could decide to assign the packet to different tasks based on the packet type.

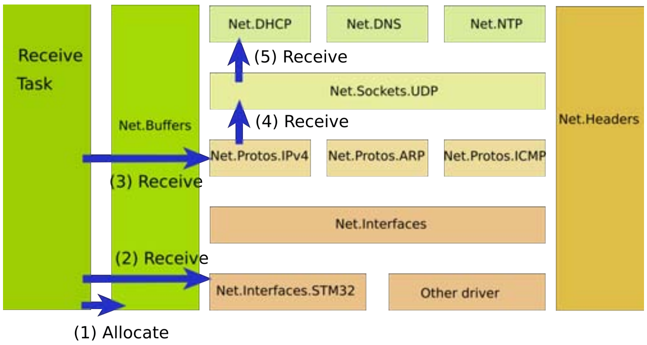


Figure 5: Receiving a packet

The reception task must allocate a network buffer and then calls the `Receive` procedure of the Ethernet interface driver to wait for a packet. That operation will return only after a packet is received. Once the packet is received, the reception task has to call the network stack with the packet so that the packet is processed by the appropriate upper layers.

The packet is first dispatched according to the Ethernet packet type and then eventually according to the IP protocol header. For example if the Ethernet packet type is `16#806#` it corresponds to the ARP protocol and the packet is dispatched to the `Net.Protos.ARP` layer. If the Ethernet packet type is `16#800#` it means the packet is for the IPv4 layer and it is handled by `Net.Protos.IPv4`.

The IPv4 layer dispatches to upper layers according to the IP protocol header. The UDP layer dispatches according to the destination UDP port up to the application.

2.4 Network stack housekeeping

Several network protocols need some housekeeping to manage timeouts or retransmissions. For the ARP resolution a retransmission is sometimes necessary and the ARP database must be periodically cleaned to drop old entries. On its side, the DHCP protocol needs to manage retransmission of queries but also take into account lease time renewal and expiration. The application has the choice to use a dedicated task to handle this, have a task for each protocol or integrate the network housekeeping in an existing loop. Most timers are in the order of seconds or minutes and they don't need any realtime constraint. Because the Ravenscar profile enforces the `No_Relative_Delay` pragma the protocol timeouts are all represented and managed by using deadlines. It is safe to call them before the deadline is reached. If they are called too late, they will handle the timeout anyway. A possible main loop could be the following:

```
with Ada.Synchronous_Task_Control;
use Ada.Synchronous_Task_Control;
 ...
Ready : Suspension_Object;
task body Housekeeping is
   Deadline : Ada.Real_Time.Time;
begin
   Suspend_Until_True (Ready);
   loop
      Net.Protos.Arp.Timeout (Ifnet);
      Dhcp.Process (Deadline);
      delay until Deadline;
   end loop;
end Houstkeeping;
```

With the Ravenscar profile, tasks are declared and created statically, they start immediately. This is why a suspension object is used to wait for the network stack to be fully initialized.

2.5 Ethernet driver design

The Ethernet driver is an interesting component to study from an Ada point of view. The Ethernet hardware has a receive and a transmit queue organized as a ring. The receive ring is initialized at the beginning and contains a number of network buffers that are ready to be used by the hardware. When a packet is received, the hardware uses a free ring entry buffer, fills the buffer with the packet data and marks the entry with flags indicating that some data is available. The transmit ring is also initialized at the beginning but it does not contain any buffer until there is a packet to send.

The Ravenscar profile allows at most one entry to be declared in a protected object or protected type. Because we need one entry to wait for a packet to be received and another entry to send a packet, we are going to declare and use two protected objects. The Ravenscar profile constraint is in fact good for us because it forces the design and implementation to manage separately the transmission and the reception. This is also a good design to maximize the concurrency between transmission and reception.

```
protected Transmit_Queue
   with Priority  => Net.Network_Priority is

   entry Send (Buf : in out Buffer_Type);
   procedure Transmit_Interrupt;
   procedure Initialize ;
   function Is_Ready return Boolean;
private
   ...
end Transmit_Queue;

protected Receive_Queue
   with Interrupt_Priority  => Net.Network_Priority is

   entry Wait_Packet (Buf : in out Buffer_Type);
   procedure Initialize  ( List  : in out Buffer_List );
   procedure Receive_Interrupt;
   procedure Interrupt
     with
       Attach_Handler => Ada.Interrupts.Names.ETH_Interrupt,
       Unreferenced;
   function Is_Ready return Boolean;
private
   ...
end Receive_Queue;
```

The receive and transmit rings are shared between the Ethernet driver (our Ada package) and the hardware. Furthermore, we are using interrupts to be notified when some hardware event occurs. Such interrupt occurs when a packet is received, a packet was sent or some other hardware event is present.

The Ethernet receive ring is represented by the Rx_Ring record and the transmit ring represented by the Tx_Ring record. Each of them contains a Buffer_Type object that represents the packet buffer associated with the ring descriptor. The hardware descriptor itself points to the beginning of the data buffer representing the Ethernet frame. When a packet is received, the Ethernet controller fills the Ethernet frame and sets in the hardware descriptor with information such as the received size and other validity flags.

```
type Tx_Ring is limited record
   Buffer  :  Net.Buffers.Buffer_Type;
   Desc   :  Eth.Tx_Desc_Type;
end record;
type Rx_Ring is limited record
   Buffer  :  Net.Buffers.Buffer_Type;
   Desc   :  Eth.Rx_Desc_Type;
end record;
```

In the Wait_Packet operation, we avoid the memory copy by switching the buffer between the application's buffer and the Ethernet receive ring buffer. This way the Ethernet receive ring always contains a valid buffer and we have avoided the copy.

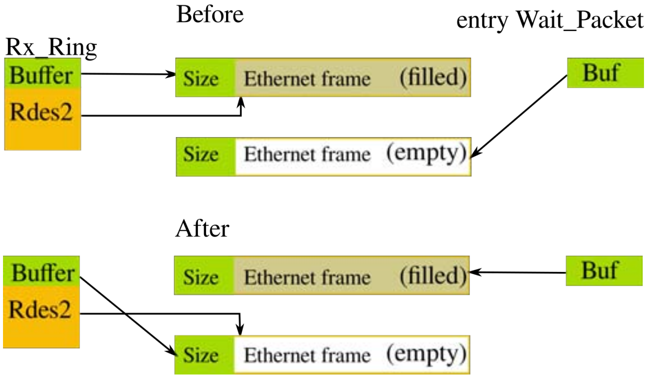


Figure 6: Buffer switch in receive operation

3 Writing an echo server

The simple echo server that is presented here corresponds to the Echo Protocol [10]. When a client sends a UDP message to the echo server, the server replies with the exact same UDP message. This is probably one of the simplest server that we can implement and it is also helpful to verify the network connectivity between the client and the server. In real life this protocol should not be activated on a server due to security issues.

To define our server we have to use the Net.Sockets.Udp.Socket tagged type as the extension of the echo server type and then override the Receive procedure. We have seen that the Receive procedure is called by the network stack when a packet is available for the UDP socket. We will get our echo message sent by the client and just have to send it back to him.

The full package specification of the echo server is the following:

```
with Net.Sockets.Udp;
with Net.Buffers;
package Echo_Server is

   type Echo_Server is
      new Net.Sockets.Udp.Socket with null record;

   overriding procedure Receive
     (Endpoint :  in out Echo_Server;
      From      :  in  Net.Sockets.Sockaddr_In;
      Packet    :  in out Net.Buffers.Buffer_Type);

   Server :  aliased Echo_Server;

end Echo_Server;
```

When the Receive procedure is called, it gets as parameter the packet that was received as well as the socket network address of the sender. It is possible to obtain the packet payload size by using the Get_Data_Size function. We can use the Get_String procedure to extract from the payload a string of the desired size.

An instance of our server is declared and must be registered to the network stack. This operation, known as binding a socket to a port in traditional Unix systems, is made by the Bind procedure provided by the Socket tagged type. The Bind procedure needs access to the network interface instance as well as to the UDP port and the optional IP address to bind. The network stack will keep an access type to the server instance and will use it when a UDP packet to the destination port 7 is received as specified by the echo protocol. Network stacks rely on integers to be represented in network byte order which is traditionally big endian. Hence, the port number value is converted by using the To_Network function.

```
Echo_Server.Server.Bind
   ( Ifnet  => Ifnet'Access,
    Addr  => (Port => Net.Headers.To_Network (7),
              Addr => (others => 0)));
```

When a UDP packet is received by the network stack, it looks at the socket list to find a matching UDP port, in our case port 7. When such instance is found, the associated Receive procedure is called.

```
package body Echo_Server is
   ...
   overriding procedure Receive
      (Endpoint : in out Echo_Server;
       From     : in  Net.Sockets.Sockaddr_In;
       Packet   : in out Net.Buffers.Buffer_Type) is

      Size    : constant Net.Uint16
        := Packet.Get_Data_Size (Net.Buffers.UDP_PACKET);
      Status  : Net.Error_Code;
   begin
      Packet.Set_Data_Size (Size);
      Endpoint.Send (To => From, Packet => Packet,
                     Status => Status);
   end Receive;

end Echo_Server;
```

The echo server has to send back the original packet. For this our packet payload part contains already the correct content and we only have to set up the packet size before sending the data. The packet is sent back to the originator so this is why we give to the Send procedure the originator's address in its To parameter.

After the Send procedure is finished, the packet data is transfered without copy to the Ethernet driver.

4 Difficulties found and solutions

This section presents several difficulties that were found during the project and it highlights the solutions that have been used to resolve them.

4.1 No Random Numbers

Random numbers are used by network protocols to generate cryptographic keys to encrypt data sent over networks. But they are also used by several non secure protocols to reduce the scope and make attacks harder. This is the case for DNS protocol which uses transaction IDs (TXID) for tracking queries and their responses. The DNS transaction ID is a 16-bit integer in the request packet and it is used together with the UDP source port to identify the DNS response. An attacker can forge a DNS response with the appropriate transaction ID and UDP source port to spoof the DNS server. By using a random number for the DNS transaction ID and the UDP source port, we are making the spoofing harder since a 32-bit value must be guessed by the attacker.

The Ravenscar sfp profile in the GNAT compiler does not provide the Ada.Numerics.Discrete_Random generic package but fortunately the STM32F746 board has a hardware random generator that can be used. Our random operation can be implemented as follows:

```
with STM32.RNG.Interrupts;
procedure Initialize is
begin
   STM32.RNG.Interrupts.Initialize_RNG;
end Initialize ;

function Random return Uint32 is
begin
   return STM32.RNG.Interrupts.Random;
end Random;
```

Unfortunately this simple operation had unexpected behavior: it raised the Program_Error exception in some situations. The reason and a solution is explained in the next section.

4.2 Restriction Detect_Blocking

The Ravenscar profile activates the pragma Detect_Blocking which forces the detection of potentially blocking operations within a protected operation. Such detection is made by the runtime which would raise the Program_Error exception when this happens. The blocking typically occurs if the protected operation calls an entry operation directly or indirectly. This situation occurred on the project when the Random function was called from a protected operation. Indeed, the implementation is using an internal protected object and uses an entry call to retrieve the random value. Nothing in the Ada 2012 declaration shows this implementation detail and this results in an exception being raised during the execution and not during the compilation.

We fixed the implementation so it calls Random before and add a parameter to the protected operation to get our random value.

The Ada Issue AI12-0064, if adopted, is intended to solve this problem by adding the Nonblocking aspect which indicates whether the operation is not blocking. Our Random function would be declared as follows:

```
function Random return Uint32
   with Nonblocking => False;
```

And we could expect the compiler to forbid us to use it in a protected operation.

4.3 Restriction Ceiling_Locking policy

The Ceiling_Locking policy is another pragma that is enforced by the Ravenscar profile. In short, it states that a protected object cannot call another protected object if the active priority of the calling object is less than the called object priority. When this contraint is not met, the Ada runtime raises the Program_Error exception. In fact, this constraint could be detected by a careful static analysis of the complete program. For the Ada Embedded Network, such analysis is rather simple because the number of protected objects are rather small and a simplified call graph can be drawn easily.

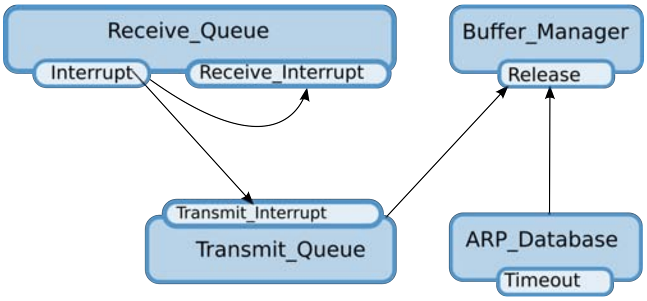


Figure 7: Static analysis of protected objects call graph

From the call graph and the ceiling locking policy rules, we can write the constraints that each protected object has to meet as summarized in table 1.

1	Receive_Queue'Interrupt_Priority	<=	Transmit_Queue'Priority
2	Transmit_Queue'Priority	<=	Buffer_Manager'Priority
3	ARP_Database'Priority	<=	Buffer_Manager'Priority

Table 1: Protected objects priority rules

When an Ethernet interrupt is raised, it is handled by a protected object that has interrupt level priority. The interrupt is then dispatched either to the transmit queue or the receive queue protected objects, this implies rules 1. The transmit queue protected object can call the buffer manager protected object to put back a buffer to the pool, this implies rules 2. Finally, the ARP database that maintains a table between the IPv4 address and the Ethernet address can also call the buffer manager protected object, hence the last rule 3. The last rule does not bring any new constraint because the ARP database protected object does need any high priority. With these rules, it is possible to assign different priorities to each protected object but the choice was to reduce this to a single priority defined as the network priority.

```
with System; use System;
package Net is
   Network_Priority : constant Interrupt_Priority
     := Interrupt_Priority ' First ;
end Net;
```

And then the protected objects are assigned the same priority:

```
protected Buffer_Manager
  with Priority => Net.Network_Priority is ...
```

4.4 Memory management and SDRAM

The memory management was an interesting issue for the project specially when we started to use the dynamic memory (SDRAM) to store the network buffers and Ethernet ring descriptors. The STM32F746 board has 8Mb of dynamic memory and 340Kb of static memory (SRAM). The SRAM is immediately available when the CPU boots but the SDRAM needs particular controller initialization to set up the memory timings (access times and refresh times). By default, the STM32F746 Ada runtime does not initialize the SDRAM and this initialization is left to the application. Furthermore, the Ada runtime that was used has the smallest runtime and it does not support Ada storage pools.

The allocation of simple types from SDRAM is not really an issue because it is possible to initialize all the fields in Ada after the allocation. The challenge comes when we start to use more complex types such as tagged records. In that case the initialization of the tag information in the object cannot be done.

Unlike C++ which provides a placement new operator, there is no way to tell the Ada compiler to build an object at a given address.

On the STM32F746 board, the application can allocate a dynamic memory region by using the `STM32.SDRAM.Reserve` function. Because buffers are represented using simple Ada types, we can use it to allocate them. The network buffers management provides a `Add_Region` procedure that splits the memory region in several network packets and adds them to the free list. The memory region must be a multiple of `NET_ALLOC_SIZE` constant which represents an Ethernet frame size. An allocation of 32 network buffers is done as follows:

```
NET_BUFFER_SIZE : constant Interfaces.Unsigned_32
   := Net.Buffers.NET_ALLOC_SIZE * 32;
  ...
  Net.Buffers.Add_Region
     (Addr => STM32.SDRAM.Reserve
                (Amount => NET_BUFFER_SIZE),
      Size => NET_BUFFER_SIZE);
```

The issue comes for the Ethernet transmit and receive ring descriptors because each descriptor holds a `Buffer_Type` object that points to the packet buffer. We cannot use the SDRAM to store them and we have to use the static ram.

4.5 CPU Cache

CPU cache introduces a complexity when we have to work with the hardware. This complexity is not due to Ada and is present in all languages. For the Ethernet driver, it is important to flush the CPU cache before giving the control to the hardware. If this is not done, the hardware could use data that is out-of-date and send an invalid data packet.

The `Send` operation transfers the buffer ownership from the application to the Ethernet transmit ring to avoid memory copy. After the `Transfer` operation we are sure that the application has a null buffer and it is not able to access the packet data. We can flush the data cache by using the `Clean_DCache` operation provided by the Ada Drivers Library.

```
with Cortex_M.Cache;

entry Send (Buf : in out Net.Buffers.Buffer_Type)
 when Tx_Ready is
   Addr : constant System.Address := Buf.Get_Data_Address;
   Size : constant UInt16 := Buf.Get_Length;
begin
   Tx.Buffer.Transfer (Buf);
```

```
      Cortex_M.Cache.Clean_DCache (Addr, Integer (Size));
      ...
   end Send;
```

There are other situations where the data cache has to be invalidated to make sure we are reading the physical memory. For our Ethernet driver this happens when we read the transmit or receive descriptor rings which are allocated in the SRAM. Indeed, when we look at a transmit or receive descriptor to check if it is available for us, the descriptor could have been changed by the hardware and we must ignore the data that could be available from the cache: we have to invalidate the cache by using the `Invalidate_Cache` procedure.

```
procedure Transmit_Interrupt is
   Tx : Tx_Ring_Access;
begin
   loop
      Tx := Tx_Ring (Dma_Tx)'Access;
      Cortex_M.Cache.Invalidate_DCache
         (Tx.Desc'Address, Tx.Desc'Size / 8);
      exit when Tx.Desc.Tdes0.Own = 1;
      ...
   end loop;
   ...
end Transmit_Interrupt;
```

5 Conclusion

Ada and the Ravenscar profile bring several contraints and challenges for a developper to implement a network stack. These contraints force the developer to have a clear and well-defined architecture and implementation. None of the challenges were impossible to solve and the final benefits on the design and implementation are clear. Pre- and postconditions are very helpful to express the behavior of some operations such as on the `Send` operation with the buffer ownership transfer.

Cooperation between hardware and software is still an interesting challenge with many traps. Ada and C are still at the same level when they have to control hardware. Forgetting to flush the CPU cache before giving the data to the hardware can result in random data being sent over the network.

The Constrained Application Protocol (CoAP) [11] is a specialized web transfer protocol for use with constrained devices. Based on UDP and Datagram TLS (DTLS) [12] it is probably the next network protocol that should be implemented to provide a complete, secure, and dependable network stack for IoT devices.

Acknowledgements

This work was possible with the availability of the GNAT Ada compiler for Open Source projects and the MakeWithAda competition organized by the AdaCore company. The Ada Drivers Library project [2] with its Ravenscar sfp profile and its numerous examples have been key to quickly set up the foundation of the Ada Embedded Network project, to understand and spot several issues described in this article.

References

[1] S. Carrez, “Ada embedded network,” GitHub https://github.com/stcarrez/ada-enet, 2017.

[2] F. Chouteau, P. Rogers, J. Lambourg, “Ada drivers library,” GitHub https://github.com/AdaCore/Ada_Drivers_Library, AdaCore, 2017.

[3] H. Zimmermann, “Osi reference model — the iso model of architecture for open systems interconnection,” *IEEE Transactions on Communications*, vol. 28, pp. 425–432, April 1980.

[4] J. Postel, “Internet protocol,” RFC 791, RFC Editor, September 1981.

[5] J. Postel, “Internet control message protocol,” RFC 792, RFC Editor, September 1981.

[6] D. C. Plummer, “Ethernet address resolution protocol: Or converting network protocol addresses to 48.bit ethernet address for transmission on ethernet hardware,” RFC 826, RFC Editor, November 1982.

[7] R. Droms, “Dynamic host configuration protocol,” RFC 2131, RFC Editor, March 1997.

[8] P. Mockapetris, “Domain names - implementation and specification,” RFC 1035, RFC Editor, November 1987.

[9] D. Mills, J. Martin, J. Burbank, and W. Kasch, “Network time protocol version 4: Protocol and algorithms specification,” RFC 5905, RFC Editor, June 2010.

[10] J. Postel, “Echo protocol,” RFC 862, RFC Editor, May 1983.

[11] Z. Shelby, K. Hartke, and C. Bormann, “The constrained application protocol (coap),” RFC 7252, RFC Editor, June 2014.

[12] E. Rescorla and N. Modadugu, “Datagram transport layer security version 1.2,” RFC 6347, RFC Editor, January 2012.

Hardware-Based Data Protection/Isolation at Runtime in Ada Code for Microcontrollers

J. Germán Rivera
Sunnyvale, CA, USA; email: jgrivera67@gmail.com

Abstract

This article describes an approach for using a memory protection unit (MPU) to enforce data protection/isolation at runtime, for individual data structures and memory-mapped peripherals, in Ada bare-metal embedded software for microcontrollers. First, an MPU-based data protection architecture for Ada programs is described. Then, the changes required for the GNAT small-foot-print Ravenscar Ada runtime library, to implement this data protection architecture, are described.

1 Introduction

Many modern low-end microcontrollers come with a memory protection unit (MPU) as an alternative to the memory management unit (MMU) from high-end microcontrollers. A memory protection unit (MPU) is a hardware module that enables software to control access to areas of the physical address space known as regions. The number of regions that can be defined in the MPU is limited by the maximum number of region descriptors supported by the MPU (typically 8-16). Each region descriptor defines a region as an address range along with read/write/execute permissions to access it. These regions can vary in size and typically can be as small as 32 bytes, in modern MPUs. With this level of granularity, it becomes possible to control access at the individual data object or individual data structure level. This means that, for code architected following the information hiding and encapsulation principles, the MPU can be used to enforce at runtime that a code module's internal data structures can only be modified by the module's code, and thus data of safety-critical modules cannot be corrupted from bugs in other code modules or from malicious attacks. Even further, for a device driver of a safety-critical device, the MPU can enforce at runtime that only the driver's code can modify the I/O registers of the device, thus ensuring that they cannot be accidentally or intentionally modified from other code. The MPU can also be used to restrict read accesses, to enforce that security-sensitive/privacy-sensitive data be accessed only by code it is supposed to.

This article describes an approach for using a microcontroller's MPU to provide runtime data protection/isolation within a single address space, for bare-metal embedded software, written in Ada. Although accidental data corruption is less likely in code written in Ada than in code written in C/C++, it is still possible. Besides, Ada programs that call libraries written in C/C++ need to protect themselves from buggy or malicious C/C++ code that can corrupt the Ada data structures. Also, Ada programs need to protect themselves from data corruption caused by buggy or malicious DMA transfers that may write to memory that they are not supposed to.

The proposed approach is based on the idea that all RAM data that is not a local variable should be read-only by default. So, an Ada package needs to ask permission to the MPU to be able to modify its own private global variables. This may sound a little inconvenient, but it is a small price to pay to ensure the package's data integrity. To support this approach, the MPU must be programmed at boot time. MPU region descriptors need to be allocated to specific regions. Some regions are fixed and defined at boot time, such as the *background data region*, the *text region*, and the *interrupt handler stack region*. The *background data region* covers the entire address space and is configured to be read-only by default. The *text region* covers the text segment of the program and it is the only region configured with execute permission. If this region is in RAM, it is also configured to be read-only, so that code cannot be accidentally or maliciously modified. Other regions are defined on a per Ada task basis, such as the *task-private stack region* and the *private data region*. The corresponding MPU region descriptors are saved/restored during task context switches. The *private data region* changes dynamically as the various code modules (Ada packages) in the program get invoked. The public subprograms of each Ada package call the `Set_Private_Data_Region` and `Restore_Private_Data_Region` primitives to temporarily acquire write permission to the package's private data, if necessary. The remaining MPU region descriptors are left to be statically allocated to DMA-capable devices. Some extensions to the basic approach will also be discussed to support hiding security-sensitive data and restricting executability of safety-critical code.

The rest of this article is organized as follows. Section 2 explains how an MPU works and presents examples of three actual MPUs. Section 3 describes an MPU-based data protection architecture for bare-metal Ada programs and the Ada code design implications to use this data protection architecture. Section 4 discusses some advanced variations of the basic data protection architecture. Section 5 describes the changes required for the GNAT small-foot-print Ravenscar Ada runtime system to support this data protection architecture.

2 Memory Protection with an MPU

Many modern small embedded processors (microcontrollers) come with a memory protection unit (MPU) as an alternative to the memory management unit (MMU) from larger processors. An MPU is a hardware block that sits between the microcontroller's CPU core and the hardware blocks addressable in the physical address space, as shown on figure 1.

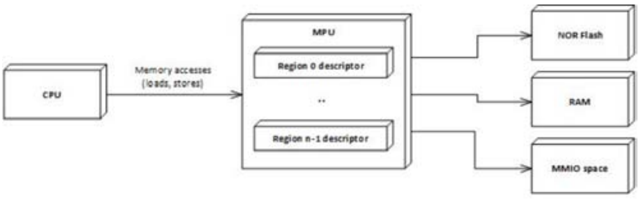


Figure 1: A Basic Memory Protection Unit

All load/store instructions executed by the CPU are intercepted by the MPU to grant or deny access to the target address, based on the current settings of the MPU. The MPU enables software to control access to areas of the physical memory address space, known as regions. The number of regions that can be defined in the MPU is limited by the number of region descriptors supported by the MPU (typically 8-16). Each region descriptor is a grouping of I/O registers that defines a region as an address range in the physical address space, along with read/write/execute permissions to access it. If the target address of the load/store instruction being executed is in any of the regions currently defined in the MPU and the type of access is allowed by the region's permissions, the access succeeds. Otherwise, the access fails and a hardware exception (e.g. Bus fault, MemManage fault) is triggered by the CPU.

There are MPUs that, in addition to controlling accesses from the CPU, also control accesses from DMA-capable peripherals, as show on figure 2.

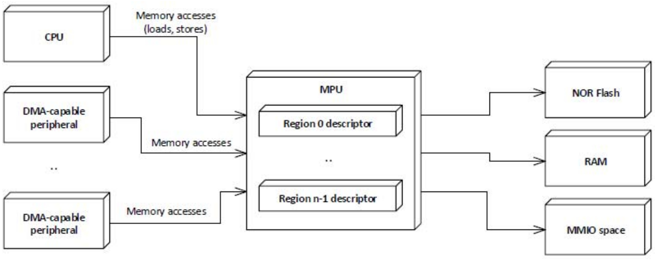


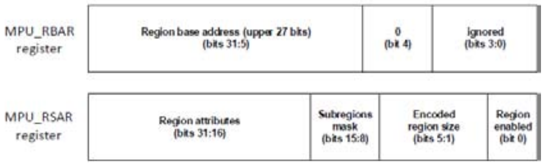
Figure 2: A Memory Protection Unit with support for both CPU and DMA access control

A key feature of modern MPUs is that they support regions as small as 32 bytes long. With this region size granularity, it becomes possible to control access at the individual data object or data structure level. This level of fine-grained access control is not possible with an MMU, which has page size granularity (typically 4KiB). By dynamically defining/undefining small regions in the MPU, access to private data structures and memory-mapped I/O registers can be restricted to only the code modules that are supposed to access them, to prevent accidental or malicious data corruption from other code modules, as well as to prevent information stealing from malicious code. In a similar way, invocation of safety-critical code can be restricted to only code modules that are supposed to invoke it, to prevent accidental calls from invalid function pointers or calls from malicious code.

Below, three examples of actual MPUs are described.

2.1 ARMv7-M Architecture MPU

The ARMv7-M architecture [1] includes an optional MPU that only controls memory accesses from the CPU core, not from DMA-capable peripherals. The region size can be as small as 32 bytes but needs to be a power of two. A region's starting address needs to be a multiple of its size. A region can have up to 8 sub-regions, which can be used to compensate for the alignment limitations. When overlapped, regions with higher region IDs have higher precedence over regions with lower region IDs. For this MPU, each region descriptor consists of the registers shown on figure 3.



MPU_RBAR register	Region base address (upper 27 bits) (bits 31:5)	0 (bit 4)	ignored (bits 3:0)

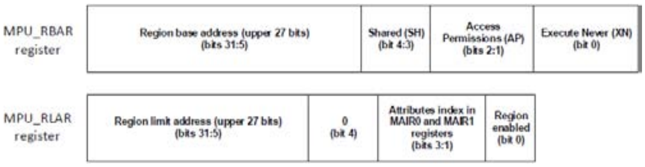
MPU_RSAR register	Region attributes (bits 31:16)	Subregions mask (bits 15:8)	Encoded region size (bits 5:1)	Region enabled (bit 0)

Figure 3: ARMv7-M MPU region descriptor

The MPU_RBAR register is used to specify the base address of the region. Only the highest 27 bits of the base address need to be specified, since the lowest 5 bits are always 0 (minimum 32-byte alignment). The MPU_RSAR register is used to specify the size of the region (encoded as a power of two) and its read/write/execute permissions. Only one region descriptor is addressable at a time through the MPU register interface. The MPU_RNR register is used to specify the index of the region descriptor to access. Bit 0 of register MPU_RSAR is used to enable/disable the region descriptor.

2.2 ARMv8-M Architecture MPU

The ARMv8-M architecture [2] also includes an optional MPU that only controls memory accesses from the CPU core, not from DMA-capable peripherals. An enhancement in the ARMv8-M MPU is that the region size does not need to be a power of two, but just a multiple of 32 bytes. Also, a region's starting address does not need to be a multiple of its size, but just 32-byte aligned. For this MPU, each region descriptor consists of the registers shown on figure 4.



MPU_RBAR register	Region base address (upper 27 bits) (bits 31:5)	Shared (SH) (bit 4:3)	Access Permissions (AP) (bits 2:1)	Execute Never (XN) (bit 0)

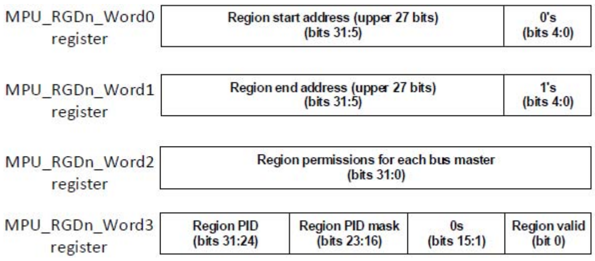
MPU_RLAR register	Region limit address (upper 27 bits) (bits 31:5)	0 (bit 4)	Attributes index in MAIR0 and MAIR1 registers (bits 3:1)	Region enabled (bit 0)

Figure 4: ARMv8-M MPU region descriptor

The MPU_RBAR register is used to specify the highest 27 bits of the base address of a region. The lowest 5 bits do not need to be explicitly specified as they are always 0, due to the 32-byte alignment. The read/write/execute permissions are specified in the lowest three bits of the MPU_RBAR register. The MPU_RLAR register is used to specfiy the highest 27 bits of the limit (last) address of the region. The lowest 5 bits do not need to be explicitly specified as they are always 1. Bit 0 of register MPU_RLAR is used to enable/disable the region descriptor.

2.3 NXP Kinetis MPU

Several NXP Kinetis microcontrollers such as the K64F [3] come with an MPU that supports access control both from the CPU core and from DMA-capable peripherals. For this MPU, the region size just needs to be a multiple of 32 bytes and the region starting address just needs to be a multiple of 32 (32-byte aligned). Also, this MPU supports other bus masters, besides one CPU core, including other CPUs and several DMA-capable peripherals. Each region descriptor consists of the registers shown on figure 5.



MPU_RGDn_Word0 register	Region start address (upper 27 bits) (bits 31:5)			0's (bits 4:0)
MPU_RGDn_Word1 register	Region end address (upper 27 bits) (bits 31:5)			1's (bits 4:0)
MPU_RGDn_Word2 register	Region permissions for each bus master (bits 31:0)			
MPU_RGDn_Word3 register	Region PID (bits 31:24)	Region PID mask (bits 23:16)	0s (bits 15:1)	Region valid (bit 0)

Figure 5: NXP Kinetis MPU region descriptor

Register MPU_RGDn_Word0 is used to specify the base address of a region and register MPU_RGDn_Word1 is used to specify the last address of the region. The lowest 5 bits of the region's base address are always forced to 0s and the lowest 5 bits of the region's last address are always forced to 1s. Register MPU_RGDn_Word2 is used to specify the read/write/execute permissions for bus masters 0-3 and read/write permissions for bus masters 4-7. Bit 0 of register MPU_RGDn_Word3 is used to enable/disable the region descriptor.

3 MPU-based Data Protection Architecture for Ada Programs

The MPU is programmed so that all RAM data that is not a local variable is read-only by default. By default, the only writable area for an Ada task is its own stack, nothing else. The same is true for an interrupt service routine. Non-local variables (statically and dynamically allocated globals) and MMIO registers are not writable by default. An Ada package needs to ask permission to the MPU to be able to modify its own private global variables. This may sound a little inconvenient, but it is a small price to pay to ensure the package's data integrity at runtime.

3.1 Allocation of MPU Region Descriptors

To support this data protection architecture, the following allocation of region descriptors in the MPU is used:

- *Current task's private data region descriptor:* defines the region containing private non-local variables or memory-mapped I/O registers accessible by the Ada package currently invoked by the current Ada task. Each package's public subprogram is responsible for calling the `Set_Private_Data_Region` and `Restore_Private_Data_Region` primitives to acquire and to give up access to the package's private data, respectively, by setting/restoring the private data region descriptor. This region can have read-write or read-only permissions, but not execution permission. This region descriptor is to be saved/restored upon task context switches. Interrupt handers (ISRs) can also use this region descriptor to gain access to private non-local variables or I/O registers that need to be modified as part of processing the interrupt.

- *Current task's stack region descriptor:* defines the region containing the stack for the current Ada task and it is set automatically, upon task creation. The stack region has read-write permissions, but not execute permissions. Thus, local variables can always be accessed and modified, but execution cannot branch to execute code form the stack. This region descriptor is to be saved/restored upon task context switches. This ensures that task has access only to its own stack and not the stacks of other tasks. Indeed, when a task is created, it is only granted access to its stack region and to the global regions.

- *Global MPU I/O registers region descriptor:* defines the region that contains the MPU's own memory-mapped I/O registers, with read/write permissions. For the ARMv7-M MPU and the ARMv8-M MPU, ths region descriptor is not needed, as, for these MPUs, their I/O registers are always accessible, even if no MPU region covers them.

- *Global interrupt stack region descriptor:* defines the region that contains the stacks shared by all interrupt handlers. This region has read-write permissions, but not execute permission.

- *Global background data region descriptor:* defines the region that contains the whole address space with just read-only permission by default. The Ada runtime library can temporarily change the permissions of this region to be read-write when needed, but no other code should change the permissions of this region. Having this read-only background region helps simplify the code, as read-only accesses to non-local variables and memory-mapped I/O registers are always allowed. As discussed in section 4, if default read-only access needs to be forbidden for a specific area (for security reasons), then this area needs to be excluded form the area covered by the global background data region.

- *Global text region descriptor:* defines the region that contains the executable code and constants of the program (text segment), in flash memory (if the program runs directly from NOR flash) or RAM. This region has read-only and execute permissions. It is the only region that has execute permission.

Note that the allocation of region descriptors proposed above scales well regardless of the number of Ada tasks in the application. Only a total of six region descriptors in the MPU are used, regardless of the number of tasks. Four region descriptors are global, meaning that they do not change upon task context switches. Only the two task-specific region descriptors need to be saved/restored on a task context switch. The global region descriptors need to be configured at boot time, as part of the Ada runtime initialization (see section 5).

3.2 An MPU-based Data Protection API for Ada Programs

3.2.1 Set_Private_Data_Region

```
procedure Set_Private_Data_Region (
   Start_Address : System.Address;
   Size_In_Bits : Integer_Address;
   Permissions : Data_Permissions_Type;
   Old_Region : out MPU_Region_Descriptor_Type);
```

This primitive sets the private data region descriptor in the MPU according to the address range specified by $(Start\_Address, Size\_In\_Bits)$ and with the access permissions specified by $Permissions$. It saves the previous value of this MPU region descriptor in Old_Region. It is to be invoked upon entry to every public subprogram that modifies private data. The address range specified by $(Start\_Address, Size\_In\_Bits)$ should correspond to a properly aligned MPU region, to ensure that access is granted to only the intended range. Otherwise, access will be granted to the smallest MPU region that encloses the given address range, which will be slightly bigger, opening the door for the calling code to modify data outside of the intended address range.

Although this primitive can be invoked with either read-write or read-only permissions, in general, it is expected to be called with read-write permissions, since read-only access is always allowed thanks to the global background data region. However, as discussed in section 4, a variation of the basic data protection architecture includes "secret" data areas that are not covered by the global background data region, and thus are not accessible by default. In that case, this primitive can be invoked for a "secret" data area with read-only permissions, if only read access is required.

3.2.2 Set_Private_Data_Region - without saving old region descriptor

```
procedure Set_Private_Data_Region (
   Start_Address : System.Address;
   Size_In_Bits : Integer_Address;
   Permissions : Data_Permissions_Type);
```

This is a variation of the primitive of the same name described above, in which the previous value of the MPU's private data region descriptor is not saved. It is intended to be used to switch to a different private data region in the middle of a public subprogram, when the subprogram has already set the private data region descriptor (by calling the primitive that saves the old descriptor).

3.2.3 Restore_Private_Data_Region

```
procedure Restore_Private_Data_Region (
   Saved_Region : MPU_Region_Descriptor_Type);
```

This primitive restores the private data region descriptor in the MPU, to a previously saved value. It must be invoked before returning from public subprograms that modified the private data region descriptor and saved the previous one.

3.3 Ada Code Design Implications

To make coding easier, the private global data of each Ada package must be grouped into a contiguous area of memory. A simple way to ensure this is to use a global record data type. This record should occupy a whole MPU region. That is, its start address and size should be aligned according to the specific MPU's region alignment and size constraints. So, for example, for the ARMv8-M MPU and the Kinetis MPU, the record's start address should be 32-byte aligned and the record's size should be a multiple of 32 bytes. For the ARMv7-M MPU, the record alignment requirements are more strict. The record size should be a power of two not smaller than 32 bytes and the record's start address mus be a multiple of the record's size. Examples:

- A record with size less than or equal to the smallest region size allowed

  ```
  type My_Global_Data_Type is record
     Field_1 : Type_1
     ..
     Field_N : Type_N;
  end record with
      Alignment => MPU_Region_Alignment,
      Size => MPU_Region_Alignment * Byte'Size;
  ```

  The record's size in bytes is smaller than or equal to $MPU\_Region\_Alignment$ (e.g., 32 bytes):

- A record with size greater than the smallest region size allowed, aligned for he ARMv8-M MPU or the Kinetis MPU:

  ```
  type My_Global_Data_Type is record
        ...
  end record with
      Alignment => MPU_Region_Alignment,
      Size => 3 * MPU_Region_Alignment * Byte'Size;
  ```

  The record's size in bytes is a multiple of $MPU\_Region\_Alignment$. The alignment aspect ensures that the record's starting address is $MPU\_Region\_Alignment$-aligned.

- A record with size greater than the smallest region size allowed, aligned for the ARMv7-M MPU:

```
type My_Global_Data_Type is record
    ...
  end record with
    Alignment => 4 * MPU_Region_Alignment,
    Size => 4 * MPU_Region_Alignment * Byte'Size;
```

The record's size in bytes must be a power of two greater than *MPU_Region_Alignment*. The alignment aspect ensures that the record's starting address is multiple of the record's size in bytes.

For each Ada package in the program, the package's public subprograms that modify non-local variables need to call the MPU primitives specified above, as follows:

- Upon entry, call `Set_Private_Data_Region`, to set the private data region descriptor in the MPU, to point to the package's private global variables or to point to output parameters, with read-write permissions. This call also saves the previous value of the MPU private data region descriptor, so that it can be restored before returning to the caller.
- If the subprogram needs to access more than one private data region, for the others, it can call the flavor of `Set_Private_Data_Region` that does not save the previous value of MPU private data region descriptor.
- Upon exit, call `Restore_Private_Data_Region`, to restore the caller's private data region descriptor in the MPU.

Below are some examples of skeletal public subprograms calling these primitives:

- Example 1:

```
procedure My_Public_Proc1(Output_Arg : out Arg_Type) is
   Old_Region : MPU_Region_descriptor_Type;
begin
   Set_Private_Data_Region (My_Globals'Address,
                            My_Globals'Size,
                            Read_Write, Old_Region);
   ...
   Set_Private_Data_Region (Output_Arg'Address,
                            Out_Arg'Size,
                            Read_Write);
   ...
   Restore_Private_Data_Region (Old_Region);
end My_Public_Proc1;
```

  In this example, note that the second invocation to `Set_Private_Data_Region` is necessary to gain read-write access to *Output_Arg*, but we do not need to save the previous value of the MPU's private data region descriptor.

- Example 2:

```
procedure My_Public_Proc2(Input_Arg : in Arg_Type) is
   Old_Region : MPU_Region_descriptor_Type;
begin
   Set_Private_Data_Region (My_Data'Address,
                            My_Data'Size,
                            Read_Write, Old_Region);
   ...
   Set_Private_Data_Region (MMIO_registers'Address,
                            MMIO_Registers'Size,
                            Read_Write);
   ...
   Restore_Private_Data_Region (Old_Region);
end My_Public_Proc2;
```

  In this example, note that input arguments do not need to use the private data region, since read-only access is always allowed. Also, note that for the purpose of gaining read-write access, memory-mapped I/O registers are treated the same as non-local variables.

4 Variations of the Basic Data Protection Architecture

4.1 Support for Execution from both Flash and RAM

If part of the code of the program runs from NOR flash and part from RAM, a separate global region descriptor needs to be dedicated for the each of the two executable regions. No changes to the API are necessary. However, some changes in the linker script and the code of the program are necessary, to specify which parts run from flash and which parts run from RAM. Below is an example in which most of the program runs from flash and only one subprogram runs from RAM.

```
procedure My_RAM_Proc
   with Linker_Section => ".ram_text";
```

Using the GNAT compiler, the *Linker_Section* aspect is used in the specification of the subprogram to indicate that code of the subprogram is to be placed in the ".ram_text" section by the linker, according to the following linker script fragment:

```
.data : AT (__rom_end) {
          ...
          __ram_text_start = .;
          *(.ram_text)
          . = ALIGN(MPU_REGION_ALIGNMENT);
          __ram_text_end = .;
          __background_data_region_start = .;
          ...
```

4.2 Support for "Hidden" Data Regions

No changes to the API are necessary. However, some changes in the linker script and the code of the program are necessary, to specify which global variables are to be non-accessible by default. Below is an example:

```
My_Secret_Data : My_Secret_Data_Type
   with Linker_Section => ".secret_data";
```

Using the GNAT compiler, the *Linker_Section* aspect is used in the specification of the subprogram to indicate that is to be placed in the ".secret_data" section by the linker, according to the following linker script fragment:

```
.data : AT (__rom_end) {
            ...
            __secret_data_area_start = .;
            *(.secret_data)
            . = ALIGN(MPU_REGION_ALIGNMENT);
            __secret_data_area_end = .;
            __background_data_region_start = .;
          ...
```

To make a "hidden" global variable accessible the `Set_Private_Data_Region` primitive needs to be invoked with either read-only or read-write permissions.

4.3 Support for "Hidden" Code Regions

To support "hidden" code regions, a new per-task region descriptor is needed: the private code region descriptor, which is similar to the private data region descriptor, but to control code execution, instead of data access. Also, two new primitives need to be added to the API: Set_Private_Code_Region and Restore_Private_Code_Region. When an Ada subprogram wants to call a "hidden" subprogram, it first needs to gain access to the subprogram's code by calling the Set_Private_Code_Region primitive, to set the private code region descriptor in the MPU. Then, it can call the "hidden" subprogram. Before returning, it needs to hide again the "hidden" subprogram, by calling the Restore_Private_Code_Region, which will restore the previous private code region descriptor into the MPU.

Finally, some changes in the linker script and the code of the program are necessary, to specify which subprograms are "hidden" by default. Below is an example:

```
procedure My_Secret_Flash_Code
   with Linker_Section => ".secret_flash_text";

procedure My_Secret_RAM_Code
   with Linker_Section => ".secret_ram_text";
```

Using the GNAT compiler, the $Linker\_Section$ aspect is used in the specification of the subprogram to indicate that is to be placed in either the ".secret_flash_text" section or the ".secret_ram_text" section, by the linker, according to the following linker script fragment:

```
.text : {
             . = ALIGN(MPU_REGION_ALIGNMENT);
             __secret_flash_text_start = .;
             *(.secret_flash_text)
             . = ALIGN(MPU_REGION_ALIGNMENT);
             __secret_flash_text_end = .;
             __flash_text_start = .;
             ...

data : AT (__rom_end) {
             . = ALIGN(MPU_REGION_ALIGNMENT);
             __data_start = .;
             __secret_ram_text_start = .;
             *(.secret_ram_text)
             . = ALIGN(MPU_REGION_ALIGNMENT);
             __secret_ram_text_end = .;
             ...
```

4.3.1 Set_Private_Code_Region

```
procedure Set_Private_Code_Region (
   First_Address : System.Address;
   Last_Address : System.Address;
   Old_Region : out MPU_Region_Descriptor_Type);
```

This primitive sets the private code region descriptor in the MPU according to the address range specified by $(First\_Address, Last\_Address)$, with read-only and execute permissions. It saves the previous value of this MPU region descriptor in Old_Region. It is to be invoked upon entry to a public subprogram that needs to invoke a"hidden" subprogram.

4.3.2 Set_Private_Code_Region - without saving old region descriptor

```
procedure Set_Private_Code_Region (
   First_Address : System.Address;
   Last_Address : System.Address);
```

This is a variation of the primitive of the same name described above, in which the previous value of the MPU's private code region descriptor is not saved. It is intended to be used to switch to a different private code region in the middle of a subprogram, when the subprogram has already set the private code region descriptor (by calling the primitive that saves the old descriptor).

4.3.3 Restore_Private_Code_Region

```
procedure Restore_Private_Code_Region (
   Saved_Region : MPU_Region_Descriptor_Type);
```

This primitive restores the private code region descriptor in the MPU, to a previously saved value. It must be invoked before returning from public subprograms that modified the private code region descriptor and saved the previous one.

4.4 Support for DMA access control

This requires that the MPU supports controlling access from DMA-capable peripherals. Of the MPUs mentioned in this article, only the Kinetis MPU provides this kind of support. The Kinetis MPU can control access to the memory space from multiple bus masters, which can be up to four CPU cores and up to four DMA-capable peripherals, or one CPU core and up to seven DMA-capable peripherals (see [3], section 19.3.5).

Since DMA transfers can be initiated synchronously or asynchronously with respect to instruction execution, at least one MPU region descriptor needs to be dedicated at boot time (or before DMA-capable peripherals are activated), to specify the source/destination address range for DMA transfers. If isolation between DMA-capable peripherals is wanted, then a separate MPU region descriptor for each peripheral would be required, assuming that there are enough free descriptors in the MPU. For example, in the implementation for the Kinetis MPU, the last three region descriptors are reserved for the DMA access control.

To support DMA access control, the following primitives are added to the API described in section 3.2:

4.4.1 Set_DMA_Region

```
procedure Set_DMA_Region (
   Region_Id : MPU_Region_Id_Type;
   DMA_Master : Bus_Master_Type;
   Start_Address : System.Address;
   Size_In_Bits : Integer_Address;
   Permissions : Data_Permissions_Type);
```

This primitive sets the $Region\_Id$ region descriptor in the MPU to control access to the address range specified by $(Start\_Address, Size\_In\_Bits)$, with the given read-write or read-only permissions, from the DMA-capable peripheral with the DMA_Master bus master Id. $Region\_Id$ must refer to one of the region descriptors reserved for DMA access control.

4.4.2 Unset_DMA_Region

```
procedure Unset_DMA_Region (
   Region_Id : MPU_Region_Id_Type);
```

This primitive disables the given region descriptor in the MPU. *Region_Id* must refer to one of the region descriptors reserved for DMA access control.

5 Changes Required for the GNAT Small-Foot-Print (SFP) Ravenscar Ada Runtime System

5.1 Ada Startup Code

The Ada startup code (i.e., the reset Exception Handler) needs to be extended to include the initialization of the MPU hardware and the configuration of the global MPU regions. To provide more control to the application for deciding when to enable the MPU, the MPU may initialized in a disabled state in the startup code. Then, the application's main subprogram need to explicityl enable the MPU.

5.2 Ada Task Control Block

The GNAT Ada runtime library keeps a control block for each Ada task in the program, as an instance of the *System.BB.Threads.Thread_Descriptor* record type. The following fields need to be added to this record type, so that the per-task MPU region descriptors can be saved/restored upon context switches:

```
Stack_Region : MPU_Region_Descriptor_Type;
Private_Data_Region : MPU_Region_Descriptor_Type;
Private_Code_Region : MPU_Region_Descriptor_Type;
Writable_Background_Region_Enabled : Boolean := False;
```

MPU_Region_Descriptor_Type is a record type that represents the state of the MPU I/O registers that form a region descriptor. So, the declaration of this record type is MPU specific. The *Private_Code_Region* only needs to be included if "hidden" code regions are supported. The *Writable_Background_Region_Enabled* is necessary only if Ada runtime library code invoked from tasks makes the global background data region writable, to access its internal data structures.

5.3 Task Creation

When an Ada task is created, its stack region descriptor needs to be initialized, so that when the task starts executing, it can access its own stack (for calling subprograms and accessing local variables). This can be done in the `System.BB.Threads.Initialize_Thread` subprogram.

5.4 Task Context Switch

The context switch logic for Ada tasks needs to be extended to save/restore the per-task MPU region descriptors and the "writable" on/off state of the global background data region descriptor. For the ARM Cortex-M ports of the GNAT Ravenscar SFP Runtime library, this change needs to be made in the `System.BB.CPU_Primitives.Pend_SV_Handler` subprogram.

5.5 Exception Handling of Memory Protection Faults

Logic to handle Memory protection faults needs to be added in package `Cpu_Exception_Handlers`. Some of the exception handling may be MPU specific, such as querying MPU error status registers to provide diagnostic information for the cause of the memory protection fault.

5.6 Other Ada Runtime Library Code Changes

All code of the Ada runtime library that modifies non-local variables needs to either temporarily make the global background data region writable or set the private data region accordingly, with read-write permissions. Ada Runtime library subprograms that need to do this include the following:

```
Ada.Synchronous_Task_Control.Set_False
Ada.Synchronous_Task_Control.Set_True
Ada.Synchronous_Task_Control.Suspend_Until_True
System.BB.Board_Support.Read_Clock
System.BB.CPU_Primitives.Sys_Tick_Handler
System.BB.Interrupts.Interrupt_Wrapper
System.BB.Threads.Set_Priority
System.BB.Threads.Sleep
System.BB.Threads.Wakeup
System.BB.Time.Alarm_Handler
System.BB.Time.Delay_Until
System.Tasking.Protected_Objects.Lock
System.Tasking.Protected_Objects.Unlock
System.Task_Primitives.Operations.Delay_Until
System.Task_Primitives.Operations.Enter_Task
System.Tasking.Restricted.Stages.Task_Wrapper
```

6 Sample Implementation

A fully-functional implementation of the MPU-based data protection architecture described here can be found in github at:

```
https :// github.com/jgrivera67/embedded−runtimes/tree/gpl2017
```

This implementation is for the Kinetis MPU. It is available as part of the Kinetis K64F ports of the GNAT Ravenscar SFP Runtime library for the Hexiwear and FRDM-K64F boards. The code can be found in the following source files:

```
bsps/kinetis_k64f_common/bsp/memory_protection.ads
bsps/kinetis_k64f_common/bsp/memory_protection.adb
```

7 Conclusions

Data protection at the individual data object level is a novel approach of using a memory protection unit (MPU) in bare-metal single-address-space microcontroller software. Code modules can be protected from corrupting each other's data structures, even in single-threaded programs. For object-oriented code, data protection can be done at the individual object instance.

Ideally, a safety-critical bare-metal program should be architected from the beginning to use MPU-based data protection, as opposed to adding it as an afterthought. However, MPU-based data protection does not have to be an all or nothing approach. For example, Ada tasks executing trusted or legacy code could set the global background data region as writable (as if the MPU was not being used), for the lifetime of the task. Then, only some untrusted components (e.g., third-party libraries or C/C++ code invoked from Ada code) would need to be wrapped in a data protection layer.

Although this article focused on Ada, the same ideas can be adapted to other programming languages commonly used for bare-metal embedded software, particularly less safe languages such as C and C++. If an RTOS is used with these languages, similar changes to those needed for the Ada runtime system would need to be done in the RTOS kernel.

The same approach described here for bare-metal Ada programs running on microcontrollers, could also be used to protect code modules inside of a process running in an MMU-based operating system (e.g., Linux), if fine-grained MPU functionality for virtual addresses were available as part of the MMU.

One weakness of the data protection architecture described here is that the MPU registers can potentially be modified directly, bypassing the proposed API, either accidentally (e.g., corrupting the MPU registers) or maliciously (e.g., to disable or cripple the MPU). One way to mitigate this risk is to have Ada tasks run in unprivileged CPU mode, since the MPU registers can only be modified when running in privileged mode. However, this introduces additional complexity, as now a system call mechanism is necessary to switch from unprivileged to privilege mode, to access the MPU registers. Also, still the MPU can be corrupted from interrupt handlers, as interrupt handlers always run in privileged mode. Ideally, a better way to solve this problem would be to have hardware support in the MPU to restrict access to the MPU registers to only the code that implements the data protection architecture API.

References

[1] *ARM v7-M Architecture Reference Manual.* ARM, 2010.

[2] *ARM v8-M Architecture Reference Manual.* ARM, 2016.

[3] *K64 Sub-Family Reference Manual.* NXP, 2017.

Proceedings

Workshop

Challenges and New Approaches for Dependable and Cyber-Physical System Engineering (DeCPS 2017)

Ada-Europe 2017

16 June 2017

Vienna, Austria

Organizers:

Daniela Cancila, CEA LIST, France (Chair)
Valeria Nuzzo, ECE Paris School of engineering, France
Assia Soukane, ECE Paris School of engineering, France
Martin Torngren, KTH Royal Institute of Technology in Stockholm, Sweden

Industrial Co-Chairs:

Alessandra Bagnato, SOFTEAM, France
Philippa Ryan, Adelard, UK
Cristina De Luca, Infineon Technologies Austria AG Austria
Silvia Mazzini, INTECS Italy
Laurent Rioux, Thales, France

Program Committee:

Katrina Attwood, University of York, UK; Alessandra Bagnato, SOFTEAM, France; Daniela Cancila, CEA LIST, France; Hakima Chouchi, Institut Mines Telecom, France; Vincent David, Krono-Safe, France; Huascar Espinoza, Tecnalia, Spain; Barbara Gallina, Malardalen University, Sweden; Silvia Mazzini, Intecs, Italy; Elisabeth Métais, Laboratoire Cédric CNAM, France; Luis Miguel Pinho, Polytechnic of Porto, Portugal; Valeria Nuzzo, ECE Paris School of Engineering, France; Amar Ramdane Cherif, University of Versailles/Paris-Saclay, France; Laurent Rioux, Thales, France; Assia Soukane, ECE Paris School of engineering, France; Martin Torngren, KTH, Sweden

Sponsors

ECE PARIS
ÉCOLE D'INGÉNIEURS

An European Ecosystem To Boost the Digitalization of EU Industry

Semi40 & Productive 4.0 ECSEL JU Projects (2016-2020)

Cristina De Luca, Knut Hufeld

Infineon Technologies Austria AG; email: Cristina.DeLuca@infineon.com, knut.hufeld@infineon.com

Adapting business, operational models to the incoming changes determined by Internet-of-Things (IoT) is becoming a business priority. But what does it mean for industry in term of business model, market approach and which is the impact on the society? All these questions are open. The technological challenges in order to link the digital with the real world, new technologies and solutions, new standards, are not fully understood and developed, for sure, it is not possible to overcome them as single enterprise. No one possesses the full range of skills, know-how and a so huge budget and human resources, necessary to manage the future IoT roadmap. It is an overall understanding that an Industry-Research-SMEs framework and ecosystem is an indispensable approach to meet the European Industry needs to fulfil the digital revolution challenges. “Smart Sustainable and Integrated Production” (SemI40) and “Electronics and ICT as enabler for industry and optimized supply chain management covering the entire product lifecycle” (Productive4.0), ECSEL JU Projects, will fully support and boost the transformation, preparing the European industry for the digital future.

The two projects are tightly joined, but on one side addressing “Smart Sustainable and Integrated Production” and “Semiconductor Manufacturing”, SemI40 will particularly concentrate on developing essential manufacturing capabilities. Competitive production in Europa will be leveraged by a well-focused approach of automation and smart production system integration in the domains of technologies, tools and methodologies which are complemented by innovations in the area of secure communication, knowledge management, automated decision-making, and smart (agile) production execution. Similarly, the aspects of security, comprehensive information management during the entire production lifecycle, management of critical knowledge for decision making and maintenance are key issues in Productive4.0. A particular attention on SemI40 is posed in understanding the technical/technological outcomes of the research from a different angle, i.e. that of the economic and the social significance and impact, taking into account that the above outcomes can have not only a financial or efficiency nature, but also – and in some cases especially – an intangible character. On the other side, in comparison to SemI40, Productive4.0 is targeting different applications domains, encompassing entire supply chains and considering the entire production lifecycle (development, production, maintenance).

A major strength is posed, not only in addressing single machines or production lines, but also in company networks, both in supply chain point of view and in costumers’ or industrial costumers’ point of view. To underline the technologies and framework of Productive4.0 that can be seen as a gateway to new kinds of ecosystems and new kinds of business models. The idea is to furnish the industry with a System-of-Systems (SoS) architecture supporting automation and digitalization for a sustainable production. This platform will boost the overall efficiency providing an independent domain enhancing application development, deployment, operation and maintenance, manufacturing or lifecycle management. Last but not least the two projects, among others, will impact improving operational efficiency, innovating in hardware and software platforms that overcome traditional industry boundaries, processes and markets.

Industrial Safety-Related Considerations to Introducing Full Autonomy in the Automotive Domain

Masoumeh Parseh, Fredrik Asplund, Martin Törngren

Department of Machine Design, Division of Mechatronics, KTH Royal Institute of Technology, Stockholm, Sweden; email: {mparseh, fasplund, martint}@kth.se

Abstract

Organizations in the automotive domain, which aim to transition into developing fully autonomous vehicles face many challenges. These range from organizational issues to engineering concerns. This paper builds on structured interviews with professionals from industry and academia to provide a deeper understanding of existing problems. Standards, safety analysis, legacy assumptions related to having a human driver, and increased complexity and complexity handling were raised as important concerns. The analysis of these concern leads us to consider the current relationship between academia and industry as too disconnected. There is a risk that new techniques developed by academia end up irrelevant to industry. This underlying problem, and others relevant to autonomy, might be solved by collaborative research between different automotive companies. However, there are experts that challenge the underlying need for such collaboration. Therefore, externally to automotive companies, new expert arenas might be required in order to facilitate an exchange of ideas that lead to new collaboration efforts. Internally to automotive companies, the changes brought on by autonomy will lead to organizational changes and the creation of new roles. These organizational changes will have to be managed, or otherwise unnecessary conflict might occur between new and old roles.

Keywords: safety; autonomy; standards; driver; complexity; methods; organization

1 Introduction

How to enable fully autonomous vehicles has become a hot topic for recent research in the automotive domain, driven by the significant industrial efforts to introduce such technology. The introduction of autonomous vehicles exposes industrial stakeholders to significant challenges. These challenges impact both engineering disciplines and business considerations. Systems engineering is an example of a much affected discipline, with new requirements for novel technologies, processes and organization. The multitude of vehicles and business cases are problematic commercial aspects. Furthermore, technical and business processes notwithstanding, the engineering output needs to meet user and legal requirements while minimizing cost.

In this paper, the focus is on considerations associated with introducing higher levels of automation, where the autonomous vehicle itself is responsible for driving. The challenges and associated solutions that are discussed have been elicited from industrial professionals. Special attention is on standards, safety analysis, legacy assumptions related to having a human driver, and increased complexity and complexity handling.

In the remainder of the paper, the study design is described in section 2. Findings from the study are presented in section 3, followed by a discussion of the corresponding considerations. Finally, conclusions are put forward.

2 Method

This paper is based on interviews, primarily with professionals from the industry. However, a few experts from academia are also included. Interviews were performed in two phases.

2.1 First Phase

The first round of interviews was performed with the purpose of identifying obstacles in implementing continuous integration for autonomous CPS, and the demands that autonomy put on the development process with a focus on verification. In total 13 interviews were performed as a part of a M.Sc. degree. For readers interested in the details of that part of the study, we refer to the associated M.Sc. thesis [1].

2.2 Second Phase

In the next phase, 3 interviews were performed by a PhD student to elaborate further on the findings. Interviews were designed based on Kvale and Brinkmann’s methodology for designing qualitative interviews [2]. During each interview, an observer with interviewing experience was present to control for interviewer bias. This at times provided a secondary perspective on replies, and ensured consistency in the interviews. Interviews were recorded, transcribed, coded and analyzed based on the conventional content analysis method described in [3].

3 Results

In this section we summarize some important considerations for industrial stakeholders in the automotive industry, intent on developing fully autonomous products. This summary is based on the perspectives of professional experts in the field, and structured around different themes identified through the interviews.

3.1 Standards

The increasing interest in highly automated systems in the automotive domain has triggered the introduction/revision of a number of standards/recommendations. Topics that have triggered recent revisions include security and autonomous driving. In particular, for autonomous driving, a working group is set up to address what is referred to as Safety of the Intended Function (SOTIF). Their work is addressing the lack of guidelines for dealing with the performance and failure modes of complex perception. However, the safety standard most widely referred to in the automotive industry today is the ISO 26262 standard, designed for the functional safety of the E/E systems for the road vehicles [4].

There exist different perspectives in the automotive industry regarding standards.

Some derive much confidence and expectations from following standards. They believe that aligning proprietary safety analysis processes with standards is beneficial for development processes. In the long run it will be a way of achieving reliable safe autonomous vehicles; it will lead to vehicles with a well-known behavior; a shared, up-to-date methodology for handling of autonomous vehicles; and an increased quality of safety analyses. This will in part rest on improvements to traceability of specifications, requirements and testing.

Others see standards as checklists. They make sure that the development process is structured and provide evidence needed due to legal obligations. They believe what is important is having a good safety culture.

However, professionals from both perspectives are worried that introducing new standards will put heavy requirements on e.g. documentation, which could be prohibitive for small to medium-sized companies.

3.2 Safety Analysis

Safety is one of the top priorities in the automotive industry. Recently, experts from academia have suggested that there is a need for new methods in hazard analysis and risk assessment to handle the increasing complexity of systems [5]. Autonomy most likely to increase the complexity of automotive systems even further.

However, there are problems that precede complexity by setting limits on safety even before an analysis is started. This includes the reuse of components, especially hardware; insufficient requirements engineering leading to the need to implement a system before any analysis can be performed; and a lack of formal documentation.

Regardless, some experts in automotive industry do not believe that there is a need for new hazard analysis methods. Others propose to look for the solution elsewhere, such as in the networking and collaboration in different research projects. This would e.g. give input on how to perform the required safety analyses as soon as new methods become available or are updated by experts in the field.

An interesting fact highlighted by our interviews is that safety experts often are mentors, moderators and reviewers of safety analyses, rather than implementers of them. Hazard analyses often strongly rely on domain experts, who can even be experts in and responsible for the safety of systems.

3.3 Legacy Assumptions Related to Having a Human Driver

One of the largest difficulties with introducing full autonomy is that the *assumption* that a driver exists becomes invalid. This assumption is often *implicit* in many engineering analyses, and the impact of it becoming invalid is therefore difficult to gauge.

Invalidating this assumption will at least influence hazard analysis/risk assessment and the design process.

For risk assessment, driver’s absence would imply a lack of controllability, i.e. the manner in which a hazardous event can be avoided by human action. One could assume no controllability. However, this would imply more stringent development is often required, which would lead to higher cost. According to the interviewees the driver is currently a large part of the hazard analysis and risk assessment processes. It is assumed that the driver is educated and experienced enough to handle the vehicle in a hazardous situation. Even if there is no explicit responsibility on the driver, he or she is implicitly found at the end of the degradation chain. Also for semi-autonomous functionality the driver must be aware of the environment at all times, even when e.g. redundancy is provided to deal with potentially dangerous situations.

Removing the driver also influences design choices. This is obvious for the design of functionality that is in direct contact with the driver, such as braking and steering [6]. However, for other functionality the influence, if any, often remains unknown.

3.4 Increased Complexity and Complexity Handling

One of the major challenges that autonomy brings, is an increased level of complexity. It is especially challenging to ensure that such complex systems can guarantee safety.

While the complexity itself and the associated challenges are to some extent known in the automotive industry, how to handle some aspects of complexity is unknown. For example, one way to handle complexity is to introduce monitoring systems [7], [8], [9]. The interviewees all knew about the basic idea of such systems, but when discussing examples, they rather referred to supporting systems with diagnostic capabilities. How to design, test or verify such systems was also unclear.

Furthermore, methods for handling complexity in place today might not work for fully autonomous, complex systems.

Interactions between components which can create dangerous situations is one example. Leveson points out that traditional hazard analysis methods do not provide engineers with the option to investigate the hazards that can arise from these interactions [5]. The interviewees agreed that interactions between functionalities are not always considered during hazard analysis. They believe that such interactions must be addressed at a high level of design to mitigate this problem. However, currently the automotive industry handles this problem through continuous collaboration between developers, whose functionality interacts within the vehicle. Identification of all relevant interactions is difficult and currently relies on the experience of the involved engineers and extensive testing.

However, also for the issue of increased complexity there exists those within the industry that believe that no large changes are required. In other words, increased complexity can be solved by already existing methods, as long as they are applied more stringently and frequently.

4 Discussion

Firstly, we uncovered industrial perspectives on standards to some extent were contradictory: do the standards prepare us for the difficulties of introducing full autonomy or are they just checklists? The different perspectives lead to different conclusions on how to introduce standards in an effective manner. Regardless of the perspective, standards may easily lead to large transitions in which companies have to adopt drastic measures, e.g. process automation, a large increase in documentation, a need for more and different expert knowledge, etc. These changes can be met with resistance, both from top management and individual engineers. However, while the former perspective points at a natural transition, where learning and sharing can be left to the organizations themselves, the latter hint at the need for legal and market demands to force firms to change.

Secondly, safety analysis processes and techniques can be poorly-informed and heavily impacted by requirements e.g. due to legacy. Safety is then to a high degree reliant on the experience of domain engineers, with safety engineers or experts acting as mentors. Does academia understand the industrial context, or do they simply assume that what is needed is a “silver bullet” in the form of a novel method for hazard analysis? How useful are actually these novel methods? As an example, STPA is a novel technique for safety analysis, which is claimed to be more suitable for complex systems [5]. It is very exhaustive in identifying hazards and risks. However, standards and industrial practice are trying to identify specific hazards and risks; If the consequences of the identified risks are below a certain threshold, following the standards and industrial practice mean that these risks will be ignored, regardless of the identification technique employed.

Comparing academia and industry, a gap seems to exist between the two. If industry is unaware of the finer points of what is achieved by academia, then the road towards full autonomy will be needlessly challenging. The opposite is also true; academic research which does not consider the industrial context risks leading to nothing. Even superior techniques might get rejected in this way; one only has to consider the willingness of engineers to learn new techniques. Throughout the interviews, it was discovered that engineers are result-oriented, indicating that new methods and techniques would be evaluated in regard to the support they would bring into producing better results, while reducing workload. A technique that is not obviously meeting these criteria might easily be met by resistance. This could be solved by management involving engineers in decision making, and introducing smaller changes gradually rather than drastically. However, it should not be required – applied research should not create two new problems for industry as it attempts to solve a third.

Thirdly, automation of safety techniques is one way to improve processes with regard to risk prioritization. If nothing else, then analyses can be quickly revisited in case new evidence turns up. However, what will then happen to the mentoring role of safety experts? On one hand, it would seem safety experts would benefit from studying and introducing automation. On the other hand, when building such automation one would benefit from considering the interaction between safety and domain experts. Roles might have to change: safety experts might become involved more in mentoring, guiding domain experts through the correct application of building models and analyzing them; or they could end up taking charge of the complete safety analysis, by relying on the knowledge of where and when models have to be populated by the right information, made available by the domain experts. It seems as safety roles could be affected, when an organization starts to build autonomous systems, even if techniques and technology does not change.

Especially removing the driver will affect the process of hazard analysis and design by affecting implicit assumptions. This is likely to bring changes to the body of the organization, e.g. by requiring different parts of the organization to hire experts with the specific knowledge in automation. Alternatively, collaborative research between different automotive companies could serve this purpose. According to the Federal Automated Vehicles Policy [10], taking advantage of the resources available by National Highway Traffic Safety Administration (NHTSA) is one way to handle the diversity in the automotive industry. NHTSA insists on an increased use and sharing of scientific and expert knowledge related to autonomy. Adding collaboration on subject areas such as safety-related methodology is then not a large step away.

However, when it comes to complexity, we have again identified two distinct views within the industry. One group sees the need for new techniques and methods, while the other group believes complexity will be solved by increasing the effort put into current practices. Controlled and official collaboration across companies are suggested by industrial professionals as a way of dealing with complexity. However, one might raise the question regarding how this collaborative relationship will overcome contradictory, fundamental beliefs among experts. If their experts do not see the need for new methods, why should companies invest more time, energy and money searching for novel methods? The automotive community might need *new arenas* where experts can come together with fresh and unbiased perspectives.

Finally, the introduction of new standards, methods or techniques; changes to design assumptions; and the increase of complexity to the current safety-critical systems might lead to organizational changes and the introduction of new roles. There will be a need to consolidate new and old methods, roles and processes. This points at a strong need for a systems engineering approach which go beyond a process-centric view. New methods will have to be compatible with existing ones, but people have to accept the changes too. Even there might be a need that the advantages and reasons for bringing new people to the organization be explained to senior experts. Otherwise a good working relationship between senior experts and new experts in automation will be difficult to achieve.

5 Conclusion

Large changes are coming to the automotive industry with the introduction of full autonomy. New standards seem likely to ensure this, whether through a natural transition or by force.

The relationship between academia and industry is then a cause for concern. New techniques developed by academia risk missing the mark and coming to nothing, or at least meet unnecessary resistance from engineers. If research is applied, then it should consider the applicability of its output in industrial context as of primary importance. At the same time, industry needs to find new ways to integrate and support research or new engineering techniques will not be seen for a long time.

Furthermore, changes brought on by full autonomy will not be limited to engineering methods – they will most likely influence organizational structures and roles. Collaborative research between different automotive companies has been highlighted as one way to avoid or mitigate this type of change. However, automotive experts challenge the fundamental need for this collaboration, as they do not agree on whether the underlying issues driving these changes are a concern. New expert arenas for the automotive community might be the only solution.

Finally, we conclude that the automotive industry needs to go beyond technology, and look into the challenges that introducing full autonomous vehicles bring to the core beliefs of their organizations. People can make the most brilliant ideas fail if they are not ready for them.

6 Limitation

The authors would like to mention that the results and discussions presented in this paper are based on interviews with professionals from the automotive industry and associated academia in Europe. Perspectives may differ outside of the interviewed population.

Acknowledgment

Support from Vinnova FFI, through ARCHER (proj. No. 2014-06260) is acknowledged. The authors also would like to thank Aina Johansson for her part in the interviewing process.

References

[1] A. Johansson, Obstacles to Continuous Integration for Autonomous Vehicles, KTH Royal Institute of Technology, 2016.

[2] S. Kvale and S. Brinkmann, Interviews: Learning the Craft of the Qualitative Research Interviewing, 2nd ed., SAGE Publication, 2009.

[3] L. Cohen, L. Manion, and K. Morrison, Research Method in Education, chapter 23rd, 6th ed., Routledge: Taylor & Francis Group, London and New York, 2009.

[4] ISO 26262: 2011, Road Vehicles-Functional Fafety, 2011.

[5] N.G. Leveson, Engineers to a Safer World: System Thinking Applied to Safety, MIT, 2011.

[6] R. Parasuraman, T.B. Sheridan, and C.D. Wickens, "A model for types and levels of human interaction with automation", IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, Vol 30, NO 3, May 2000, pp. 286-297.

[7] A.M. Mokhtar, J.P. Blanquart, J. Guiochet, D. Powell, and M. Roy, "Safety trigger conditions for critical autonomous systems", IEEE 18th Pacific Rim International Symposium on Dependable Computing, 2012.

[8] M. Machin, F. Dufosse, J.P. Blanquart, D. Powell, and H. Waeselynck, "Specifying safety monitors for autonomous systems using model-checking", International Conference on Computer Safety, Reliability, Security, SAFECOMP, 2014, pp. 262-277.

[9] IEC 61508: 2010, Functional Safety of Electrical/Electronic/Programmable Safety-Related Systems, 2010.

[10] Federal Automated Vehicle Policy, US Department of Transportation, National Highway Traffic Safety Administration (NHTSA), Sep 2016.

Cognition of Driving Context in a Connected and Semi-autonomous Vehicle: A Perspective

Manolo Dulva Hina, Assia Soukane

ECE Paris Ecole d’Ingénieurs, 37 quai de Grenelle, 75015 Paris, France email: manolo-dulva.hina@ece.fr | soukane@ece.fr

Amar Ramdane-Cherif

Université de Versailles Saint-Quentin-en-Yvelines, LISV Laboratory, Vélizy, France, mail : rca@lisv.uvsq.fr

Abstract

In this paper, we present our approach for the cognition of driving context in a connected and semi-autonomous vehicle. This cyber-physical vehicle has three main components: the embedded system, the networking and real-time system, and the intelligent system that communicate with each other via Cloud computing infrastructure. This paper is about the functionalities of the intelligent system in the cognition of driving context and the actions that it invokes for each of these driving situations. The driving context is determined as a result of the fusion of various parameters representing the context of the environment, the vehicle and the driver. The fission process yields the action that must be implemented with regards to the driving situation in consideration. Ontology is used as a tool for knowledge representation. This work is a contribution to safe driving in a connected and semi-autonomous vehicle.

Keywords: Cyber-physical vehicle; Intelligent transportation system; Ontology; Multimodal fusion and fission; Context cognition; System modelling

1 Introduction

Cyber-physical systems (CPS) are physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core. CPS is usually composed of interconnected clusters of processing elements and large-scale wired and wireless networks that connect a variety of smart sensors and actuators [1]. This work is about our cyber-physical vehicle [2] and its goal is to contribute to the reduction of road traffic accident by offering an intelligent, connected, semi-autonomous transportation, capable of detecting its driving context and offer safe driving, green driving and comfortable driving assistance [3], whenever possible. Our innovative vehicle is shown in Fig. 1. This vehicle has three main components, signifying the three main axes of research in ECE Paris, and they communicate with one another via Cloud computing infrastructure:

- *Embedded system*: it is responsible for capturing data and signals from the environment, from the driver and from the vehicle. Various sensors, gadgets and actuators of different modalities are associated with embedded system. Altogether, these components form connected objects and Internet of things (IoT) [4].
- *Intelligent System*: it is responsible for obtaining all input data from the embedded system, fusion them in order to deduce the driving situation and determine what action must be undertaken to the specified driving situation
- *Network and Real-time System*: it is concerned with all protocols of communication between all components of the system. The communication between embedded system and intelligent system is handled by this component.

In this paper we are concerned with the functionalities of the intelligent system component as the authors are part of the intelligent system team and work independently but in collaboration with other teams.

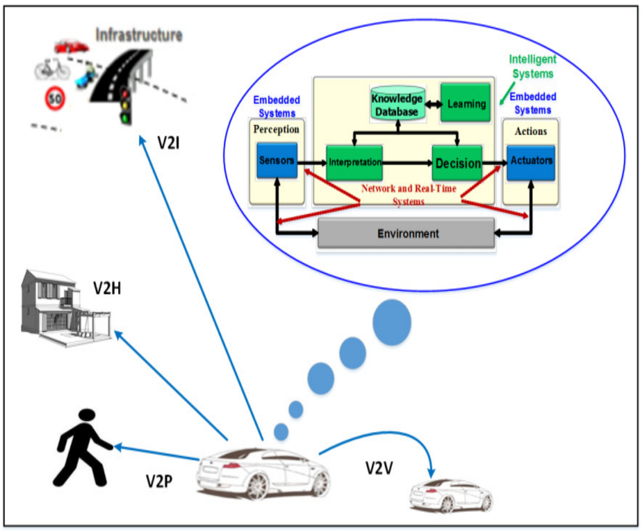


Fig. 1. Smart services for connected and semi-autonomous vehicle

2 Related Work

As early as 1920’s, various efforts have been made to automate vehicle [5]. Some promising trials happened in 1950’s and research on this domain has never ceased since then. The first autonomous vehicle was developed in 1980’s in Carnegie Mellon University Navlab [6] and AVL in 1984 [7]. It is followed by Mercedes-Benz and Bundeswehr University Munich's Eureka Prometheus

Project in 1987. Since then, various car manufacturers, such as Mercedes Benz, Toyota, Nissan, Audi, Volvo, etc. In July 2013, Vislab demonstrated BRAiVE, a vehicle that moved autonomously on a mixed traffic route open to public traffic [8]. Connected and autonomous vehicles (CAV) are a technological revolution, combining radical changes in the design of the road vehicles and in the understanding of their interactions with the networked infrastructure.

The core science and technology required to support CPS and cyber-physical vehicles are essential for future economic competitiveness. Creating the scientific and technological basis for CPS can pay dividends across a wide variety of domains. This is where this work of ours lies and we intend to contribute to its advancement.

3 Infrastructural Framework

Cloud computing is becoming the de-facto hosting platform for all types of applications and all social innovations. This is true for industry, government, organizations and society (especially social media) [9]. We use Cloud computing [10] as well for this project. The overall interactions of components of our connected and semi-autonomous vehicle are described below, in a step-by-step manner:

1. Our vehicle is equipped with various sensors, media, actuators and objects that capture parameters related to the context of the vehicle, the environment and the driver. These objects form the internet of things (IoT) for our vehicular application. In the IoT paradigm, many of the objects that surround us will be on the network in one form or another [4]. The embedded system in the Cloud receives relevant signals from IoT, infrastructure, embedded products and driver monitoring data. The communication protocol involved in this transaction is handled by the networking and real-time component of our system.

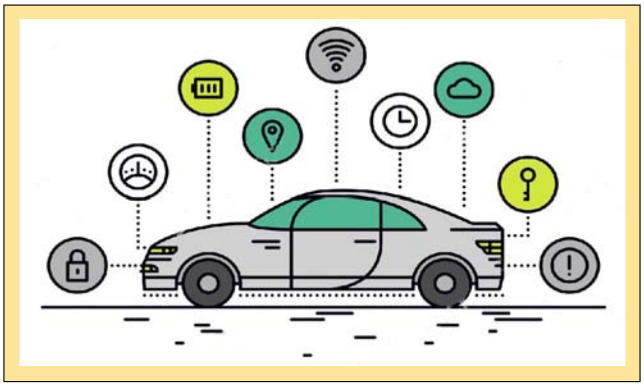


Fig. 2. Internet of things: anything, anytime, anywhere. The IoT and Cloud makes cognition of driving context of autonomous vehicle possible.

2. The values of the parameters are taken care of the embedded system component. These signals are processed such that its output is the interpretation of the signal in the higher level of abstraction. For example, a sensor may be used to detect visibility on the road. The values obtained from such sensor are processed and the result is the visibility value and its interpretation (e.g. visibility = 1000 meters → clear). This signal (pair of "name of signal" and "value of the signal") is sent (i.e. published) to the repository in the cloud
3. The intelligent system is subscribed to various signals (e.g. vehicle speed, speed limit, visibility, etc.) in the repository. Once there is a change in the value in any of these subscribed signals, the intelligent system is informed; kit takes the signal and its value and does its processing (i.e. fusion of parameters to detect the driving situation).
4. The fusion process [11-14] yields a driving situation that may need an action. If this is so, the intelligent system component performs the fission process [15-17] and decide what action to take. This is done by sending (i.e. publish) the appropriate signal and its value into the repository
5. The embedded system component is subscribed to some signals in the repository. If it finds one (e.g. activate fog light because it is detected that visibility on the road is poor) then it gets it.
6. The human-computer interface of our system is also subscribed to the signals in the repository. If it senses a pertinent data (i.e. alert or notification message intended for the driver) then it gets it and broadcasts the message via smartphone that is installed in the vehicle.
7. Likewise, the message intended for the vehicle (e.g. activate fog light) is retrieved and the corresponding action is implemented. This process is shown in Fig. 3

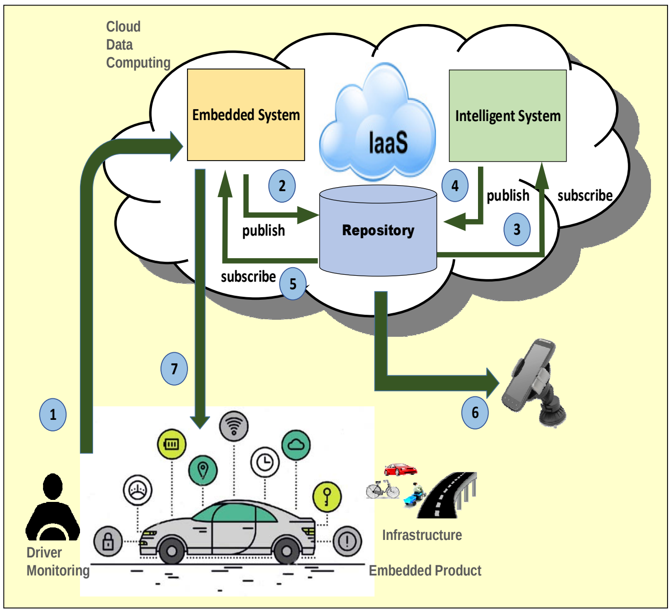


Fig. 3. Step-by-step process depicting transactions among our system components for the cognition of driving context of semi-autonomous vehicle.

4 Signal Processing for Cognition of Driving Context

Here, we focus on the details of the cognition of driving context for our connected and semi-autonomous vehicle.

4.1 Knowledge Representation using Ontology

Ontology [18-20] is the structure of concepts and the relations representing the meaning of a given domain. Ontologies are partial and formal specifications of a conceptualization. Ontologies are formal because they are expressed as formalism with formal semantics. They are partial because a conceptualization cannot always be fully formalized in such a framework, given the fact of ambiguities or of the fact that no representation of their semantics exist in the language of chosen representation. For visualization purposes, we use a Protégé plug-in, called VOWL (visual notation for OWL ontologies) in order to describe the ontology components.

4.2 The Driving Context

The driving context is the model of traffic/driving situation and rules of conduct for these situations. We use ontology for modelling to put in place a common conceptual language between the driver and the assistance system. Here, we are interested in the fusion of three main contexts: the vehicle context, the driver context and the environment context. The "Environment" is the ontology class that describes the external environment where the human vehicle interaction takes place. The "Vehicle" is the class that represents a vehicle that has to interact with its driver while the "Driver" is the class that describes the driver of a vehicle. "Environment" and "Vehicle" are related through *hasVehicle* object property while *hasDriver* is the object property that links with its driver; it is functional because only one driver can own a vehicle. See Fig. 4.

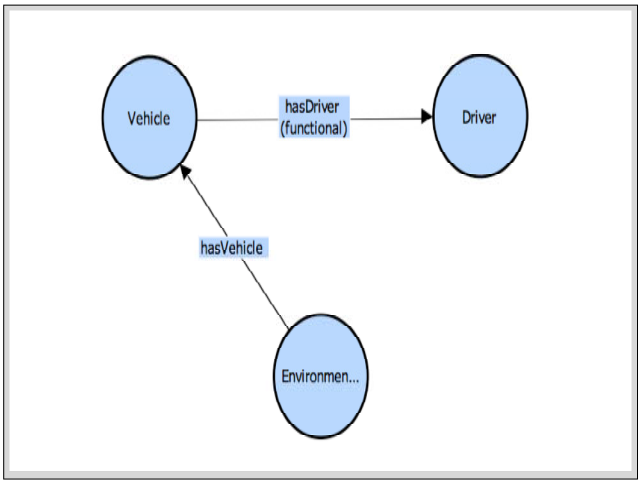


Fig. 4. Ontology for the driving context.

4.3 The Context of the Driver

The Driver class is related to many classes that contain necessary information to describe his context in the ontology, such as:

- *DriverProfile*: it is related to the "Driver" through "*hasDriverProfile*" functional property.
- *FocusOnDriving*:has data properties '*hasEyesOnTheRoad*', '*isLookingToTheRight*', and '*isLookingToTheLeft*'.
- *DriverViolation*: linked to the driver via '*hasViolation*' property. It is also linked to the 'Road' class via '*hasRoadViolation*' data property. The violations can be of type '*GiveRightToPass*', '*RedLightViolation*' and '*OverSpeedViolation*'
- *MentalState*: stores the mental state components that can negatively influence the behavior of the driver while driving a vehicle. It has three subclasses, namely '*Stress*', '*Fatigue*', and '*Faint*'. The value that it can take is one of the following: '*Low*', '*Average*', or '*High*'.

4.4 The Context of the Vehicle

As shown in Fig. 5, the Vehicle class has three subclasses that represent the three types of vehicles considered in the ontology: '*Car*', '*Truck/Bus*' or '*MotorBike*'. A vehicle has some data properties, such as '*hasPlateNumbe*r' and '*hasInsurance*'. The Vehicle class is linked to other classes, such as:

- *Cockpit*: a class that contains the status of all elements that are found in a vehicle's cockpit. For example, '*hasWindowsOpen*' is a data property that has a Boolean value.
- *ComponentStatus*: this contains as subclasses all components that we have to check to guarantee a good driving experience. Among these subclasses are "*DirectionIndicator*" (with values '*NoIndicator*' '*RightIndicator*', '*LeftIndicator*', and '*DoubleIndicators*'), "*TyresPression*", "*LubricantTemperature*", "*EngineLubricantLevel*" (with values '*LowLevel*', '*HalfLevel*' and '*FullLevel*'), and "*FuelQuantity*". The class has also some Boolean properties to check if some components are active or not. Example is '*hasFogLightsOn*'.
- *TechnicalData*: it is made up of three subclasses, namely "*FuelType*" (values are '*Petrol*', '*Diesel*', '*Electricity*' and '*GPL*'), "*EmissionClass*" (values are '*euro0*', '*euro1*', …, '*euro6*') and "*TractionType*" (values are '*Front-WheelDrive*', '*Rear-WheelDrive*' and '*All-WheelDrive*')

4.5 The Context of the Environment

The Environment is related to all the elements that belong to the scene where the human-vehicle interaction takes place. Here, the environment is an abstract class and general concept made up of cities where vehicles are present. The classes related to the Environment are given below:

- *City*: In this work, an Environment is an area or region where we can find many cities. A city has two data properties, namely '*hasCityName*' and '*hasLimitedTrafficZone*' which is a Boolean value indicating if the city can be accessed only during some intervals of the day.
- *DistrictArea*: it contains as objects the different districts of a city, linked through '*hasDistrictArea*' property. The position of the "*Driver*" is stored in the "*PositionArea*", a subclass of "*Physics*" and equivalent to "*DistrictArea*".
- *Road*: a road has many data properties, such as '*hasMinSpeedLimit*', '*hasMaxSpeedLimit*', '*hasNumberOfLanes*', '*hasContinuousLine*' and '*hasLength*'. A road is made up of three subclasses, as

follows: (1) ‘*Urban*’, (2) ‘*ExtraUrban*’, and (3) ‘*Highway*’. Every subclass of a road has its minimum and maximum speed limit.

- *RoadProperty*: it stores all the properties that belong to a particular road. This includes “*Visibility*” (values are ‘Low’, ‘Average’ or ‘High’), “*Weather*” (values are ‘*Fog*’, ‘*Sun*’, ‘*Rain*’ and ‘*Snow*’), “*AccidentHistory*” (values are ‘*Unusual*’ or ‘*Frequent*’), “*TrafficCongestionHistory*” (values are ‘*Low*’, ‘*Average*’ or ‘*Intense*’) and “*CurrentTrafficCongestion*” (values are ‘*Low*’, ‘*Average*’ or ‘*Intense*’).
- *Lane*: this represents the different lanes that a Road can have.
- *Position*: this class contains the exact position of the referenced object. It has two data value properties, namely ‘*hasLatitude*’ and ‘*hasLongitude*’.
- *Time*: it has data value properties, such as ‘*hasDate*’ and ‘*hasTime*’

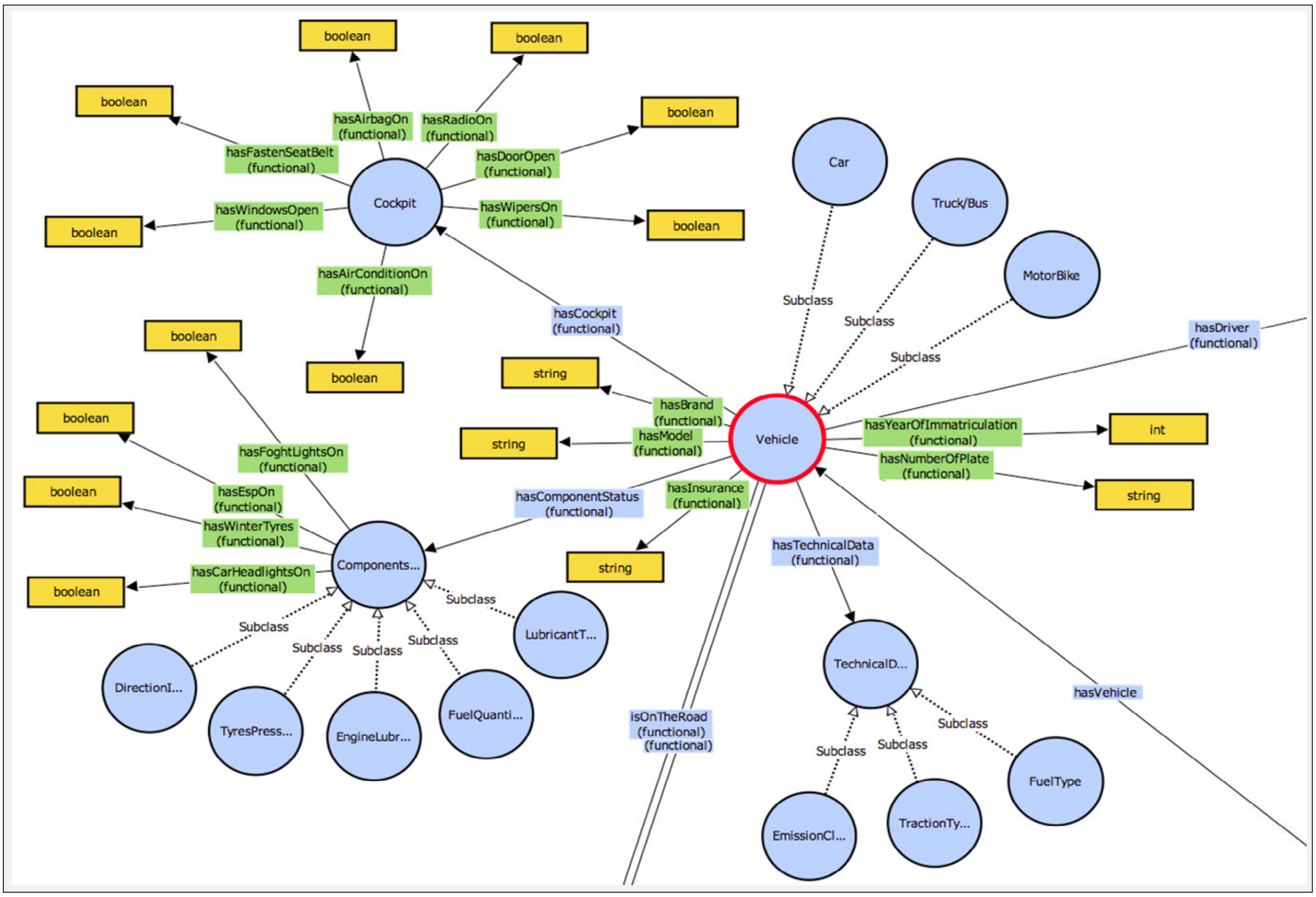


Fig. 5. The context of the vehicle

5 Conclusion

In this paper, we have demonstrated the framework of our connected and semi-autonomous vehicle, showing how embedded system and intelligent system interact with one another via the networking and real-time component. Internet of things and Cloud computing infrastructure are added components that make this intelligent vehicle possible. In the paper, we focus on the functionalities of the intelligent system component given that the authors are part of this team. We presented how the cloud infrastructure is mitigated to detect the driving situation of the semi-autonomous vehicle. We then presented the representation of context using ontology. This is an ongoing work and it will evolve in the days ahead. Future works include the machine learning component of our system, in particular, the cognitive user interface design [21] and the cognitive component [22] that learns new driving situation, reason with purpose and interact with humans naturally. This component will learn from its interaction with the system users and from its experiences with the environment

References

[1] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "*Cyber-Physical Systems: The Next Computing Revolution*," presented at the Design Automation Conference, Anaheim, CA, USA, 2010.

[2] S. Wang, "*Develop Vehicle Control Systems as CPS for Next- Generation Automobiles*," March 2015 2015.

[3] H. Estl, "*Paving the way to self-driving cars with advanced driver assistance systems*," August 2015 2015.

[4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "*Internet of Things (IoT): A vision, architectural elements and future directions*," Elsevier Future

Generation Computer Systems, vol. 29, pp. 1645-1660, 2013.

[5] T. M. Sentinel, "*Phantom Auto' will tour city,*" in Google News Archive, ed, 1926.

[6] C. M. University, "*Navlab: The Carnegie Mellon University Navigation Laboratory,*" in The Robotics Institute, ed.

[7] T. Kanade, "*Autonomous land vehicle project at CMU,*" presented at the 14th ACM Annual Conference on Computer Science 1986.

[8] University of Parma "*Vislab, Italy - Public Road Urban Driverless-Car Test 2013 - World premiere of BRAiVE*"

[9] M. Yousif, "*The State of the Cloud,*" IEEE Cloud Computing, vol. 4, pp. 4 - 5, January-February 2017 2017.

[10] Oracle, "*Architectural Strategies for Cloud Computing,*" August 2009

[11] A. Wehbi, M. D. Hina, A. Ramdane-Cherif, C. Tadj, and A. Zaguia, "*Patterns Architecture for Fusion Engines,*" presented at the ICOST 2011, 9th International Conference on Smart Homes and Health Telematics, Montreal, Canada.

[12] A. Zaguia, M. D. Hina, C. Tadj, and A. Ramdane-Cherif, "*Using Multimodal Fusion in Accessing Web Services,*" Journal of Emerging Trends in Computing and Information Sciences, vol. 1, 2010.

[13] L. Frédéric, "*Physical, semantic and pragmatics levels for multimodal fusion and fission,*" presented at the Seventh International Workshop on Computational Semantics, Tilburg, The Netherlands, 2007.

[14] A. S. Fulvio Mastrogiovanni, Renato Zaccaria, "*A Distributed Architecture for Symbolic Data Fusion,*" in 20th international joint conference on Artifical intelligence (IJCAI'07), San Francisco, CA, USA, 2007.

[15] O. Adjali, M. D. Hina, S. Dourlens, and A. Ramdane-Cherif, "*Multimodal Fusion, Fission and Virtual Reality Simulation for an Ambient Robotic Intelligence,*" presented at the The 6th International Conference on Ambient Systems, Networks and Technologies (ANT- 2015), London, UK, 2015.

[16] O. Adjali, M. D. Hina, S. Dourlens, and A. Ramdane-Cherif, "*Techniques for Multimodal Fusion and Fission for an Intelligent Robotic Application,*" Advances in Robotics and Automation, vol. S2, 2015.

[17] D. F. Costa and C. Duarte, "*Adapting Multimodal Fission to User's Abilities,*" presented at the UAHCI 2011 - 6th International Conference in Universal Access in Human-Computer Interaction, Orlando, FL, USA, 2011.

[18] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, et al. (1991) *Enabling Technology for Knowledge Sharing*. AI Magazine. 36-56.

[19] T. R. Gruber, "*A Translation Approach to Portable Ontology Specifications,*" Knowledge Acquisition, vol. 5, pp. 199 - 220, 1993.

[20] N. Guarino, "*Formal ontology, conceptual analysis and knowledge representation,*" Human-Computer Studies, vol. 43, pp. 625-640, 1995.

[21] M. F. Peschl and C. Stary, "*The Role of Cognitive Modeling for User Interface Design Representations,*" Mind and Machines, vol. 8, pp. 203- 236, 1998.

[22] John E. Kelley III, "*Computing, cognition and the future of knowing: How humans and machines are forging a new age of understanding,*" 2015.

The INTO-CPS Cyber-Physical System Profile

Alessandra Bagnato, Etienne Brosse, Imran Quadri, Andrey Sadovykh

Softeam – France; E-mail: alessandra.bagnato@softeam.fr

Abstract

This paper presents the INTO-CPS Cyber-Physical System (CPS) profile produced by the INTO-CPS H2020 EU project and readily available for public dissemination and analysis. The paper outlines how the EU Project is dealing with CPS modelling and analyses its open-source available CPS model examples

Keywords: Model-Based Design, Cyber-Physical Systems, Co-Simulation, FMU, DSE, Co-Simulation, Code generation, Test Automation.

1 Introduction

The paper looks at currently running Horizon2020 INTO-CPS project focusing on the “model-based design” of Cyber-Physical Systems (CPSs) and on the profile developed to model CPS and analyses some publicly available CPSs models drawn with the profile.

The paper first describes the INTO-CPS project. Then the paper investigates the foundations of SysML and proposes a SysML profile with a formal semantics for cyber-physical systems used in the context of the INTO-CPS project. The profile is based on a subset of SysML notations, namely, block definition and internal block diagrams, and is designed to embrace the project themes on heterogeneous modelling and co-simulation. The paper then illustrates visual modelling using the profile with several examples, and presents how the profile has been implemented in the Modelio tool [1] to enable the construction of SysML/INTO-CPS diagrams. Afterwards a discussion on the modelling choices in the various examples is given followed by a conclusion.

2 INTO-CPS

INTO-CPS aims to develop a model based tool chain for CPS development. The project integrates both Hardware in the Loop (HiL) and Software in the Loop (SiL).

The INTO-CPS project makes use of UML and SysML profile to develop an INTO-CPS profile that is used to map to FMU concepts. MBSE is used for test automation, high level simulation using the FMU model exchange mechanism with simulation engine and related tools; while code generation and test automation is also in the scope of the project via the usage of tools such as RT-Tester Model-Checker [2], Overture [3] resulting in platform-independent C source code.

The project aims to develop an integrated tool chain that focuses around a core Co-Simulation Orchestration Engine and involves tools such as Modelio [1], Overture [3], 20-Sim [4], OpenModelica [5] and RT-Tester [2]. Test Automation, Design Space Exploration and Code generation is also in the scope of the INTO-CPS project.

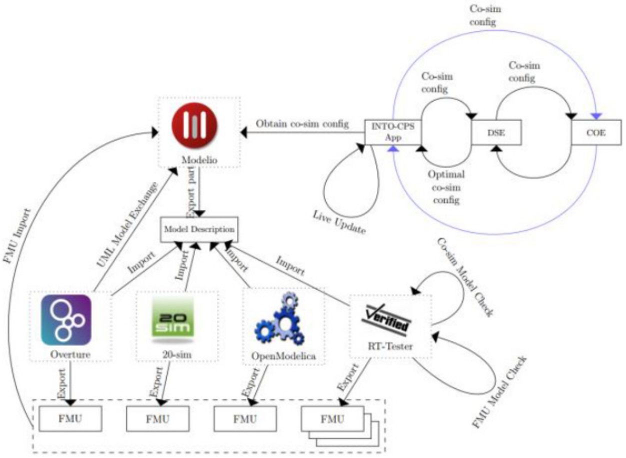


Fig.1. INTO-CPS Overview

INTO-CPS is built around the FMI (Functional Mockup Interface) [6] standard that allows FMI co-simulation and model exchange. Model-exchange takes place when the FMU is imported into a simulation tool and handled as a black box; whereas FMI’s co-simulation is based on a master-slave architecture.

The SysML profile is used in the project to develop Architecture Structure Diagrams as well as Connection diagrams, resulting in the INTO-CPS UML profile, which use a subset of SysML block and internal block diagrams respectively and define specific semantics for their usage.

The INTO-CPS profile is used to develop CPS models in Modelio UML environment. SysML is also mapped to FMI concepts via the INTO-CPS profile, while allows to export the high-level models in form of automatically generated FMI which are then taken as input by the Co-Simulation Orchestration Engine (COE) for FMI co-simulation purpose.

3 The INTO-CPS SysML profile

This section gives some details on Modelio's implementation of the profile. The SysML/INTO-CPS diagrams running examples drawn using the current version of Modelio's implementation are given in Section 4.

Within Modelio's extension mechanisms, a module has been developed to accommodate SysML/INTOCPS. The Modelio SysML/INTO-CPS module include an INTO-CPS/SysML profile which is organized around the following logical groups: block, type, instance, library and diagram. Only the block group is presented in this paper.

The next two diagrams depict the block group. INTO-CPS profile specializes SysML Block concept into one sub-concept named “Component”, with is also specializes into four sub-concepts respectively named “System”, “Subsystem”, “Cyber”, and “Physical”.

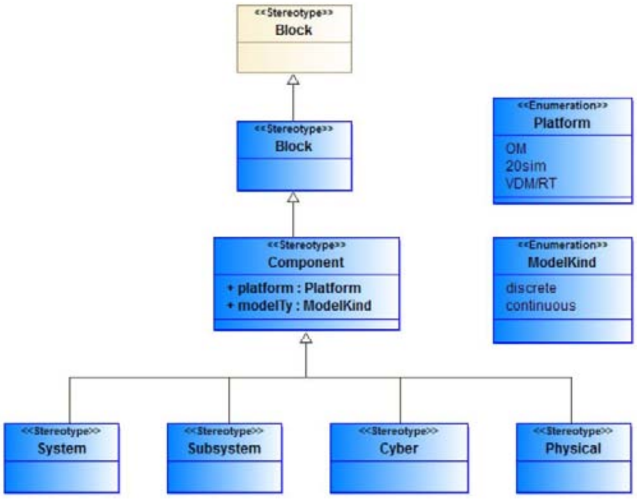


Fig. 2. The Block group

The next diagram presents the possible properties of INTOCPS/SysML Block. As depicted a Block can be composed, through the “Composition” Association, of another Block. A Block can be owned INTOCPS Flow Port. These later are a specialization of SysML Flow Port with the possibility to add dependency to other Flow Port. Variable, which are a specialization of UML Property, can be owned by Component.

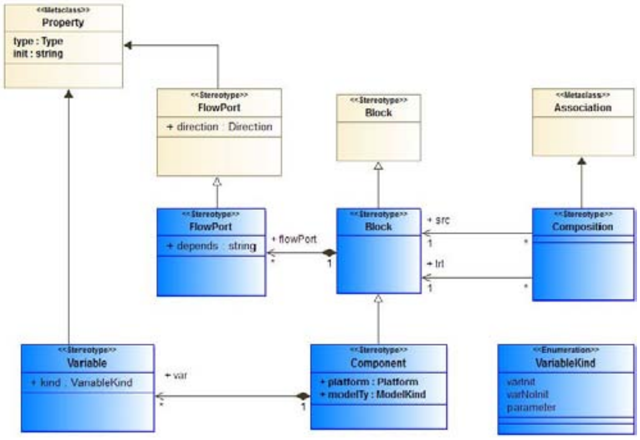


Fig 3. Block properties

The INTO-CPS profile identifies two kinds of diagrams Architectural Structure Diagrams (ASD) and Connection Diagrams.

As shows in the following figure, these diagram types respectively extend UML Class and Object diagrams.

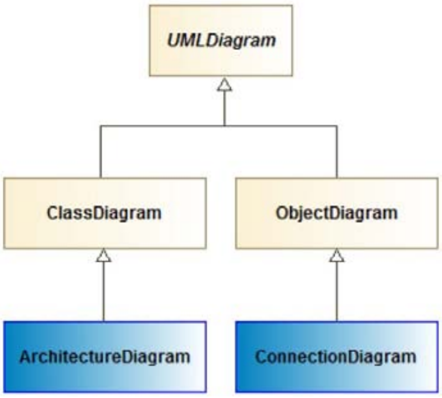


Fig. 4. INTO-CPS Diagrams

4 Analysis of available open-source models

The INTO-CPS project deliverables [7] and GitHub page [8] provide some useful modelling examples produced used the Modelio Profile, all models are designed with Modelio open source INTO-CPS module. For each example the Architecture and Connection diagrams available in the INTO-CPS profile are analysed.

4.1 LineFollowRobot Example

The application intent is to control the robot’s body by means of a controller and have sensors for the movement control (an optional 3D component is present for visualization during FMU co-simulation). The example models one CPS and uses an Architecture and two Connection diagrams available in the INTO-CPS profile. The example comes from Robotics domain. The ASD indicates a system consisting of a Robot that contains a controller, body and a sensor (all subsystems). There is also a 3D visualization cyber component that is present.

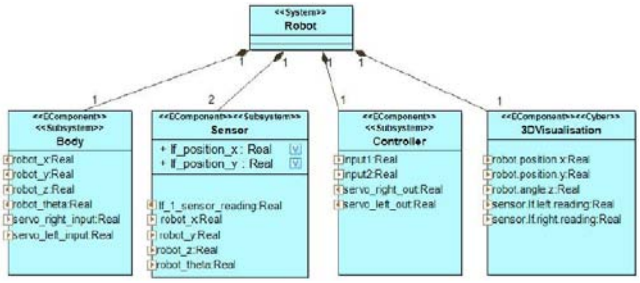


Fig. 5. Robot Example

The first connection diagram shows an instance of the robot, where the controller is connected to the body which in turn is connected to two sensor instances by connectors. The input from the sensor instances go to the controller.

The second connection diagram shows a second instance of the robot, where the controller is connected to the body which in turn is connected to the two instances of the sensor, and a 3D visualization component by connectors. The input from the sensors go to the controller and the 3D visualization component.

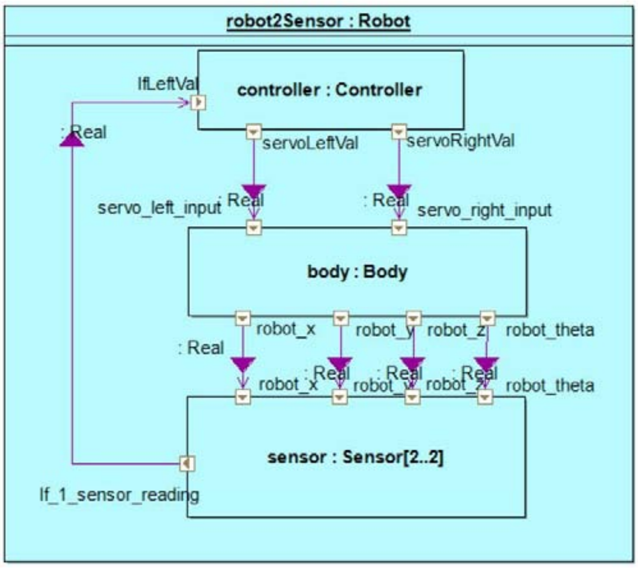


Fig. 6. Robot Example First Diagram

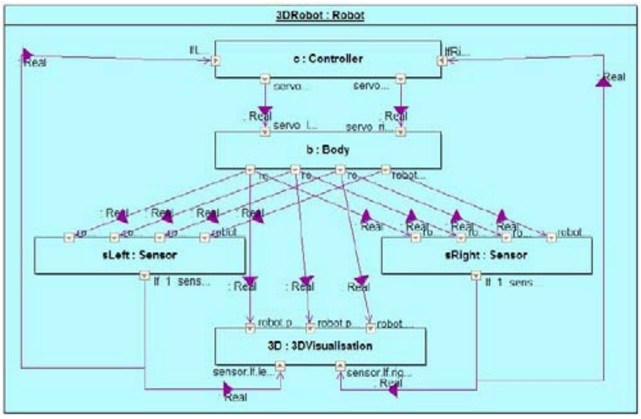


Fig. 7. Robot Example Second Diagram

4.2 Three Tank Example

The Three Tank example models one CPS and uses ASD and CD available in the INTO-CPS profile. The example comes from the water tank application to control the water level of the tanks by means of a controller.

As compared to the single tank example, this example uses three tanks. The ASD indicates a system consisting of two water tanks (subsystem components) and a Controller (cyber component) of the CPS.

Each water tank is then composed of Tanks (physical components) and other physical components such as Pipe, Drain etc. So, there is an additional hierarchical layer as compared to the single tank example. Additional data types are also specified such as Flowrate and water level along with a user defined enumeration.

The connection diagram indicates the flow between the water tank 1 to water tank 2 and then from water tank 2 to the controller component) of the CPS. The connection diagram indicates the flow between the water tank and the controller component.

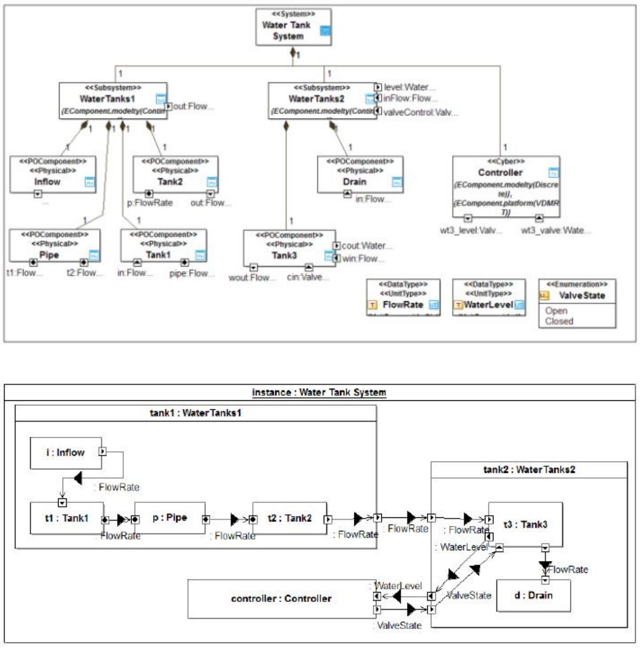


Fig. 8. Three Tank Example

4.3 Three Tank 3D Example

The Three Tank 3D example models one CPS and uses an ASD and two CDs available. The example comes from the water tank application to control the water level of the tanks by means of a controller and carry out 3D visualization of behavior simulation. As compared to the previous example, this example introduces an additional 3D animation visualization component in the global example.

The ASD indicates a system consisting of two water tanks (subsystem components) and a Controller (cyber component) of the CPS; along with a 3D animation component (physical component).

Each water tank is then composed of Tanks (physical components) and other physical components such as Pipe, Drain, etc. Data types are also specified such as FlowRate and water level along with a user defined enumeration.

The first connection diagram indicates the flow between the water tank 1, water tank 2, controller and the 3D animation component. While the second connection diagram indicates a flow where the 3D animation component is omitted, resulting in a flow from water tank 1 to water tank 2 and bi-directional flow from water tank 2 to the controller component.

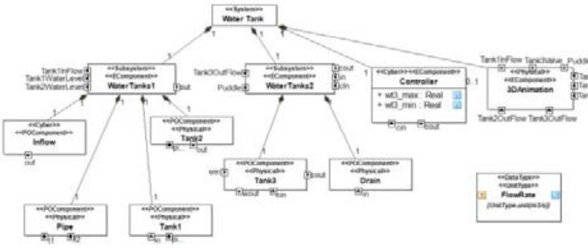


Fig. 9. Three Tank 3D First Diagram

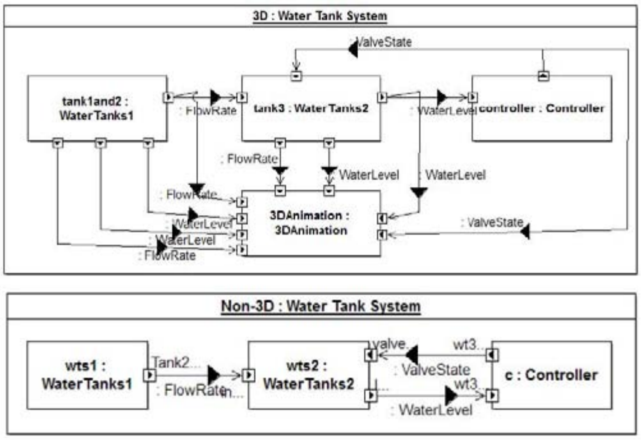


Fig. 10. Three Tank 3D Second Diagram

5 Discussion

By looking at the ASD made in these example, we can firstly remark that we always have a "System" Block on top of the model. This is imposed by an *Object Constraint Language* (*OCL*) defined in the profile itself. Then the block composition level which is at least 2 and at most 3. In INTO-CPS context, a CPS System is composed of a set of FMU, used for co-simulation purpose, this is shown in every example in this paper. Intermediate layer between the System and the set of FMU is never used. This could be because the available examples are too simple and do not need an intermediate layer between the System and the FMU. In the case of s System composed of hundreds of FMU, we expect to have some grouping (by domain or sub component for example) of the FMUs. The Three Tank examples shows a third layer of composition. The tank1 Sub Component (or FMU) owned several Physical Component (an Inflow, two Tanks and one Pipe). This level of description is no need for a FMU Co-simulation but help in the System comprehension and FMU implementation by listing the owned physical elements.

Available CDS always shows the existing connection for a future co-simulation between the FMU which is the aim of this kind of diagram. In the case of the Three Tank examples, where FMU expose internal Physical element the internal connections have also been specified. This should have been done in a documentation perspective, to exchange between the SysML designer in charge of the SysML design and the 20-sim or Modelica designer in charge of the FMU implementation. In some case a type (Real in most the case) is associated of the connection. This is done to force the type of element going through a connection between two ports having different types.

As notable difference, we can also highlight the choice made, for the sensor instances representation, in the two CDs of the Line Following Robot. The first CD shows one instance with 2 as multiplicity so no differences are made between the two sensor instances. At contrary in the second CD, the modeler creates two instances of the sensor named *leftSensor* and r*ightSensor* making a difference between them.

6 Conclusion

The paper analysis the Modelio profile produced within used applied into some available open-source models produced the INTO-CPS project applied to some of the publicly available CPS. For each model the Architecture and Connection diagrams available in the INTO-CPS profile and in the specific model are presented and analyzed. All the open source examples of the models and related information are available at INTO-CPS GitHub portal [8] as well as in related INTO-CPS deliverables [7].

Acknowledgment

This research presented in this chapter is funded by the European Community's H2020 framework under grant agreement no 644047 (INTO-CPS).

References

[1] https://www.modelio.org/

[2] https://www.verified.de/

[3] http://overturetool.org/

[4] http://www.20-sim.com/

[5] http://OpenModelica.org/

[6] http://fmi-standard.org/

[7] http://projects.au.dk/into-cps/

[8] https://github.com/into-cps

A Novel Approach to Multicast in VANET Using MQTT

Ravi Tomar, Manish Prateek, Hanumat G. Sastry

School of Computer Science and Engineering, University of Petroleum and Energy Studies, Dehradun, India;
E-mail: rtomar@ddn.upes.ac.in, mprateek@ddn.upes.ac.in, hsastry@ddn.upes.ac.in

Abstract

Intelligent Transportation Systems (ITS) are the future of vehicles, roads and society, and to make it possible we have Vehicular Ad-hoc Network or VANET as a key component in realizing the ITS. VANET comes out to be a promising field in Intelligent Transportation System. VANET is a special variant of Mobile Ad-hoc Network (MANET). VANET is in evaluation stage, among the various challenges in this field; Information dissémination is very prominent and critical. This paper proposes a novel framework for efficient information dissemination over VANET. This work emphasizes on Vehicle to Infrastructure (V2I) approach using existing Internet and cloud services available, where fixed roadside infrastructure units are not required. The proposed model has been designed by employing Message Queue Telemetry Transport (MQTT) protocol of IoT.

Keywords—information dissemination, MQTT in VANET, VANET, IoT.

1 Introduction

In today's era, we cannot imagine our life without transportation system, and this is the reason behind increase in number of vehicles on road. There has been a long history on evolution of vehicles, where we have witnessed the research in field of luxury, mechanics and speed of vehicles. As vehicles are growing in numbers on road. Traffic congestion, distraction of driver, mechanical failure, foggy weather etc. [1] are the several reasons responsible for mishaps. To minimize such incidents, we can promote safety transportation by utilizing the advances in Information Technology and here comes Vehicular Ad-hoc Network (VANET) in picture, VANET study focuses on vehicles to communicate with each other and thus enabling sharing of information. The information is used further in making real time decisions. Modern vehicles today come equipped with lots of sensors like GPS, Engine Monitoring Sensors, Fuel Consumption, Braking Assistance etc. the data collected from these sensors is processed by an in-vehicle installed micro-controller which helps in local monitoring and control of vehicle. The output is displayed in form of voice feedback, light indicators or LCD screen. VANET enables sharing of this local data with other vehicles in specific coverage area, so that all vehicles are aware of state, movement, speed, location of each other. [2] This type of communication can help in avoidance of various critical situations like accident due to malfunction of engine.

This paper is divided into V sections, Section I presents the introduction. Section II gives background information for proposed model, which explains about Cloud Computing, IoT, MQTT, and QoS. Section III depicts the scenario. Section IV presents our system model and followed by the concluding remarks with future scope of work.

2 Background

2.1 Cloud Computing

Cloud computing is the most evolving technology in the world of information technology. One of the key factors for which the Cloud is know is its accessibility to never ending resources. [3] Cloud computing refers to a model where resources are rendered on a utility basis just like electricity. Cloud computing provides virtualization of IT resources which can be subscribed by any user at any point of time and on pay as you use mode. The major advantage of cloud service is that we can scale our cloud as per our requirement, and this is the feature we are using it in our work. The cloud technology comes in various flavor but in abstract, Cloud is classified into three major categories i.e. Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). This work focusses on IaaS due to its nature of Vehicle to Infrastructure communication. We use cloud infrastructure to deploy our MQTT server, which will be used as a broker among all the vehicles. To maintain a low cost and highly efficient infrastructure cloud is the best choice today. According to recent Forbes studies [4] worldwide spending on public cloud services will grow at a 19.4% compound annual growth rate (CAGR) from nearly $70B in 2015 to more than $141B in 2019. This indicates that in coming years cloud services would be more stable and cheaper. It is foreseen that future of computing will be cloud computing and all the enterprises would move to cloud in coming years.

2.2 Internet of Things(IoT)

Kevin Ashton first used the term Internet of Things (IoT) in 2009 for interconnecting physical devices over the internet [5]. IoT is built upon a very simple idea of letting physical devices communicate with each other and thus controlling each other. Examples of such devices would be a refrigerator, a car, a building or any other electronic device. The scope of IoT is unlimited and can be implemented on almost every electronic equipment. One such example is home automation, where our home appliances can take decisions and start or stop as per their understanding of our behavior. In our proposed work, we take car as the things and propose a mechanism through which they can exchange their state information with others.

2.2.1 Messege Queue Telemtry Transport (MQTT)

Andy Stanford-Clark (IBM) and Arlen Nipper (Eurotech; now Cirrus Link) developed MQTT in 1999 for the monitoring of an oil pipeline through the desert. [6] The MQTT protocol uses pubsub model in contrast to request/response model used in HTTP. Publish/Subscribe model is event based model and a central broker is responsible for message delivery. MQTT is an event driven architecture, where clients pushes (publish) the message to broker and all subscribed clients receives the message. To make a link between the producer and consumer a topic is used, which acts like a channel for communication. Figure 1 explains the entire process, whenever a client publish some message to a certain topic all subscribers receives the message, therefore there is no need of a direct path between data producer and data consumer. MQTT architecture provides a very efficient and scalable solution for information exchange. Therefore, it suits the requirements of VANET being scalable and connected. This work uses MQTT protocol as the core component of information exchange between the vehicles.

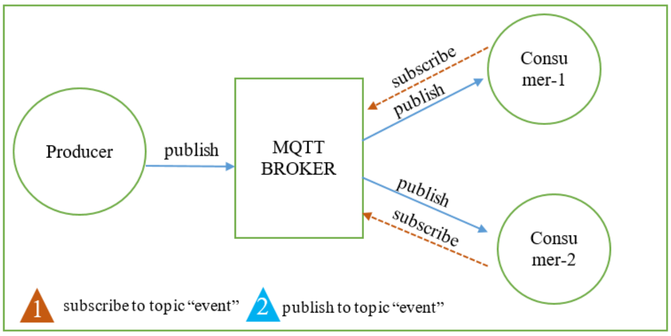


Fig. 1. MQTT Publish/Subscribe Architecture

2.2.2 Quality of Service (QoS) in MQTT

We are using MQTT in our work and are taking advantage of QoS mechanism of MQTT for prioritizing our messages; so, it is necessary to understand QoS in MQTT. QoS comes to be the major feature of MQTT Protocol, QoS enables a reliable communication in unreliable network a lot easier because the protocol has capability handle the retransmission and can guarantee the delivery of the message. [7] The delivery is not affected by the lower layer transport protocol. In addition, we can prioritize the QoS based on our data.

MQTT 3.1.1 provides 3 types of QoS:

QoS (0): In this mode, the message is delivered at most once, the best effort to deliver the messages are applied and no rebroadcast happens. It is like fire and forget. This mode gives similar QoS as the underlying layer. By default, MQTT works in QoS (0) only and this is most unreliable but fastest.

QoS (1): In this mode, the message will be delivered at least once, but can be delivered more than once. In this mode, the receiver sends a special control packet named PUBACK to sender, which ensures the delivery of packet. If PUBACK is not received in a defined time interval, then rebroadcasting is done. This mode ensures the delivery by adding a small overhead of sending PUBACK control packet.

QoS (2): In this mode, we receive highest QoS and slowest message delivery because it involves four-way message exchange. Here, receiver sends PUBREC to sender, which works as packet acknowledgement. Now the sender can drop the message as it already knows the delivery status of packet and then sends a PUBREL to let receiver know about receiving the PUBREC. Finally, the receiver sends PUBCOMP completing the entire process.

Downgrading of QoS: QoS in MQTT protocol depends on sender and receiver both, so if a sender sends a packet of QoS (2) then the receiver should also be subscribed to QoS (2) otherwise downgrading happens till receiver has subscribed. Figure 2 explains the downgrading process.

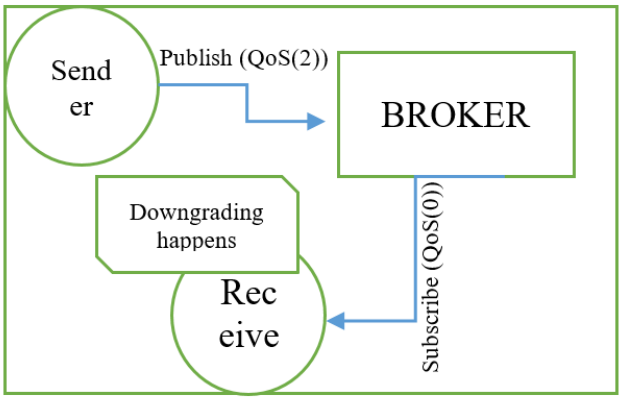


Fig. 2. Downgrading Process

In our proposed work we use QoS(0), QoS(1) or QoS(2) for publishing the message based on priority and QoS(2) for receiving so that downgrading do not happen in case of critical message.

3 Scenario

Let us consider an accident event happening on the highway depicted in Figure 3. We envision a system where vehicle is connected and the event information is sent to all the following or approaching vehicles. The timely received information can help other vehicles to decide the route and get warned about the hazardous situation on way further. When the Vehicle (red) meets an accident, a message will be published to the broker by accident vehicle or may be by any other vehicle who sense the accident. Once message is published now the Broker sends the message to nearest SOS service available and then all other vehicles who are subscribed on the same topic.

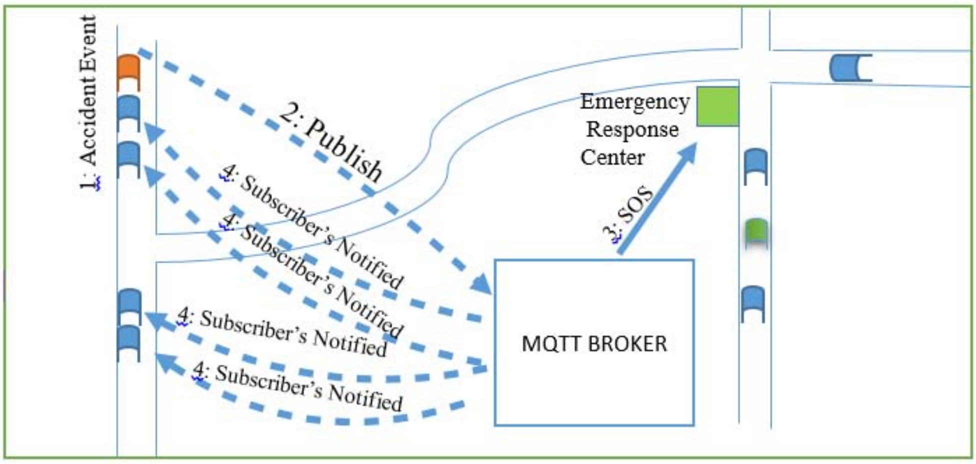


Fig. 3. Scenario depicting accident node and message delivery through MQTT.

4 System model

Our Model of Information Dissemination for VANET uses cloud enabled IoT approach, it uses MQTT protocol in application layer. We have implemented our model of information dissemination on Scenario explained in Section 2. We assume that each vehicle is equipped with an On-Board Unit (OBU), which is capable of gathering and processing the data from various sensors fitted in the car. OBU also possess location service like GPS [8] that can provide location (p) in the form of two co-ordinates such that:

$$p \in C \text{ with } C \in \mathbb{R}^2$$

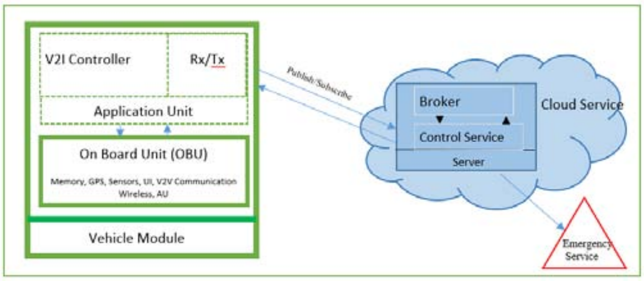


Fig. 4. Internal structure of each module.

We propose a separate layer on existing OBU, which is responsible for V2I communication. Figure 4 shows the flow chart of the entire process. This layer is responsible for Information transmission and reception from vehicle to Infrastructure i.e. a cloud-based infrastructure explained earlier where our MQTT Broker sits. Furthermore, MQTT broker server is responsible for disseminating the information to all the subscribed vehicles on the same topic. Our model works in three phases:

4.1 Subscription Phase

Subscription phase is essential phase and it starts by asking for co-ordinates set C from the OBU. These coordinates namely latitude and longitude are used to identify the Road_id. The Road_id is used to establish a common property among the vehicles moving on the same road. A google API [9] takes these latitude and longitude as input and provides a JSON[10] output with very detailed geographical data like route, locality, country, postal code etc. we use route name as our Road_Id, which is always unique and static for that area. This process is repeated based on time intervals, which keeps Road_Id field updated. This Road_Id is passed to the subscribe function of MQTT Server, which will now deliver all messages coming to the Road_Id topic. On every new subscription the previous subscription is unsubscribed which ensures that unnecessary messages are not received. Figure 5 shows the flow diagram of the entire process.

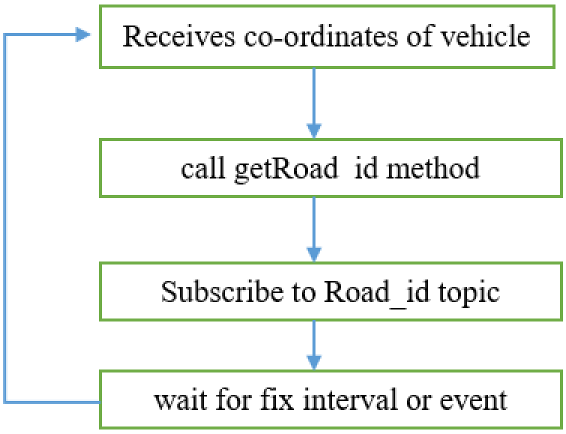

Receives co-ordinates of vehicle
call getRoad id method
Subscribe to Road_id topic
wait for fix interval or event

Fig. 5. Subscription Phase Flow

In other words, this phase is managing the mobility of vehicle and informing the infrastructure about its current presence

4.2 Information Dissemination Phase

Information Dissemination Phase works when there is an event or information that is required to be disseminated. The module will receive data from OBU along with priority of message, this data is published using the publish method on the topic Road_Id with priority as Quality of Service (QoS) [7] to the MQTT Server. Now, since server will receive the message will send the message to all subscribed clients, and the node itself receives the same message, which was sent. This ensures the delivery of the message to all nodes and hence apart from QoS provided by MQTT we can be double sure of message/information being disseminated to all the subscribed vehicle. Entire flow is depicted in Figure 6.

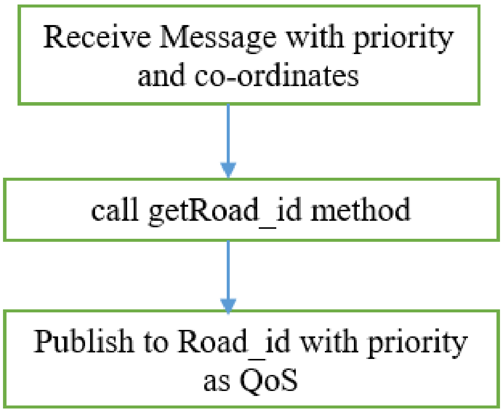

Receive Message with priority and co-ordinates
call getRoad_id method
Publish to Road_id with priority as QoS

Fig. 6. Information Dissemination Phase

4.2 Information Reception Phase

Information reception phase works when there is some event received on the subscribed topic, we ensure the subscription of topic having its QoS set to 2, this is done so that no downgrade of QoS happens when receiving the information through MQTT. Once the message is received, the data is parsed and sent forward to OBU for further processing and necessary actions. Figure 7 shows the flow of Information Reception.

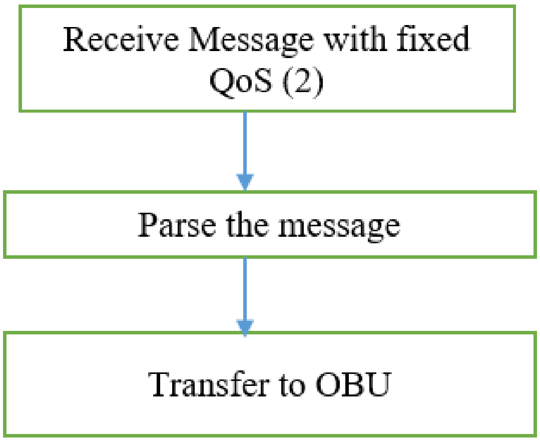


Fig. 7. Information Reception Phase

5 Conclusion and future work

In the proposed model, physical Infrastructure like Road side Units are not required; we implemented our model using IaaS from Amazon. Amazon Web Service (AWS)[11][12] are used to deploy Server for MQTT. Each vehicle is equipped with a ESP826612E microcontroller [14] and Ublox Neo6mv2 [14] for GPS co-ordinates. We built two prototypes for testing and got very accurate results. We considering that all vehicles are equipped with same device. Each Vehicle will subscribe to the topic of Road_Id they are travelling on, this we termed as Subscription Phase. Now, any vehicle publishes the message to the network and all subscribed vehicles receives the message, which was sent. This approach works on Pub/Sub Model so it is implicitly Multicast. Apart from this intra communication each vehicle can publish any emergency event to Emergency channel with QoS attained, which ensures quick and guaranteed delivery of message so that further action can be taken. Our work contributes to the information dissemination part of VANET [15]. As a future work, we can make vehicle subscribe the approaching road, so it can be informed about the events happening on approaching road in advance. We can also optimize the Road_id finding method by locally providing the data from specific city.

References

[1] Fitzgerald, Emma Mary. "Achieving dynamic road traffic management by distributed risk estimation in vehicular networks." (2013).

[2] Toor, Yasser, Paul Muhlethaler, and Anis Laouiti. "Vehicle ad hoc networks: Applications and related technical issues." *IEEE communications surveys & tutorials* 10.3 (2008).

[3] Khanna A, Tomar Ravi, An architectural view towards Autonomic Cloud Computing, In IC3T 2016 Third Springer International Conference on Computer & Communication. Technologies,Vijayawada, Andhra Pradesh, India, 5th-6th November, Springer Publication

[4] Louis Columbus , Roundup Of Cloud Computing Forecasts And Market Estimates, 2016 https://www.forbes.com/sites/louiscolumbus/2016/03/13/roundup-of-cloud-computing-forecasts-and-market-estimates-2016/#516f99252187. Retrieved 31 March 2017

[5] Gubbi, Jayavardhana, et al. "Internet of Things (IoT): A vision, architectural elements, and future directions." *Future generation computer systems* 29.7 (2013): 1645-1660.

[6] Roy, Sarbani, and Chandreyee Chowdhury. "Integration of Internet of Everything (IoE) with Cloud." *Beyond the Internet of Things*. Springer International Publishing, 2017. 199-222.

[7] Lee, Shinho, et al. "Correlation analysis of MQTT loss and delay according to QoS level." *Information Networking (ICOIN), 2013 International Conference on*. IEEE, 2013.

[8] Costa, Paolo, et al. "Towards lightweight information dissemination in inter-vehicular networks." *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*. ACM, 2006.

[9] Google API, Geocode, http://maps.googleapis.com/maps/api/geocode/json?latlng=40.714224,-73.961452 . Retrieved April 6 (2017):2017

[10] Crockford, Douglas. "The application/json media type for javascript object notation (json)." (2006).

[11] Cloud, Amazon Elastic Compute. "Amazon web services." *Retrieved April* 9 (2017): 2017.

[12] Luo, Jun-Zhou, et al. "Cloud computing: architecture and key technologies." *Journal of China Institute of Communications* 32.7 (2011): 3-21.

[13] Platform, Espressif Smart Connectivity. "ESP8266." *Espressif Systems* (2013).

[14] Pham, Hoang Dat, Micheal Drieberg, and Chi Cuong Nguyen. "Development of vehicle tracking system using GPS and GSM modem." *Open Systems (ICOS), 2013 IEEE Conference on*. IEEE, 2013.

[15] Ravi Tomar, Manish Prateek, G. H. Sastry. Vehicular Adhoc Network (VANET) - An Introduction. *International Journal of Control Theory and Applications*, International Science Press 2016, 9 (18), pp.8883-8888. < hal-01496806>

Dependability of the Transport of the Future

Daniela Cancila, Emine Laarouchi

CEA, LIST, CEA Saclay - F91191 Gif-sur-Yvette Cedex; email: firstname.lastname@cea.fr

Alessandra Bagnato

Softeam, France; email: alessandra.bagnato@softeam.fr

Abstract

Transport of the future embraces several and different products. Among them, autonomous vehicles are considerably increasing their leading role in advanced industrial research. For autonomous vehicles, we target products such as cars, trains and drones. These products have different, albeit broadly similar requirements in the system engineering field. In this position paper, we argue that traditional paradigms, devoted to the design of critical systems and their relationships, are not fully adequate to the challenges involved in the transport of the future and new trends are emerging in the literature. We address the change of these paradigms. We devote a special focus to the specification and assessment of safety-related properties in the transportation of the future. Capital examples are provided throughout the paper in support of our analysis.

Keywords: transport of the future, methodology, safety, critical cyber-physical systems, CPS-IoT, social impact of the transport of the future.

1 Introduction

The transport of the future embraces several products. Among the most influential ones is - without any doubt - the autonomous car. Nowadays, prototypes are already in place, for example, in the United States, Uber, TESLA and Google are acquiring the leadership. The competition at industrial (research) level is so sharp that in February 2017 Waymo (belonging to the Alphabet society as Google) makes a complaint at the San Francisco tribunal about a violation of patents by Uber: Waymo states that ex-employees, now Uber employees, revealed some secrets related to ‘Lidar’, i.e. the combined use of laser technologies and sensors to localize a target [1]. This combined use is relevant because it is expected to avoid accidents like the one happened in 2016 with a TESLA car [2].

The key point in autonomous car addresses not only the embedded autonomous system, including physical networking, but also the vehicle-to-vehicle communication, i.e. the ability to send the information from one vehicle to another one, eventually exploiting the infrastructure. In Figure 1, the information related to collisions, obstacles, lines change is shared via two points in the monitoring and control infrastructure.

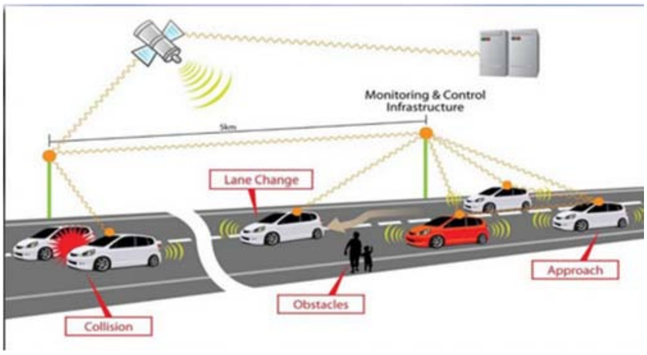


Fig. 1. Vehicle-To-Vehicle communication Image extracted from [3]

Like the automotive domain, the railways one is going to automatize their products further (such as trains, metros, tramway). In this regards, SNFC adopts drones to reduce the cost of maintenance and infrastructure [4] and Alstom envisages a train-to-train communication [5].

In spite of the role of safety issues, however, no completely satisfactory methodologies for dealing rigorously with the relationship between autonomous levels and safety-related properties has yet been put forward. Traditional methodologies addressing embedded critical systems are becoming inadequate to address the challenges conveyed by critical cyber-physical and Internet of Thinks systems (CPS-IoT). In this position paper, we discuss the change of the paradigms from embedded to cyber-physical critical systems. Moreover, we introduce the CPSwarm European project to support our analysis. Finally, we emphasize the importance of human persons in methodologies for CPS-IoT and we conclude with the social impact of the transportation of the future.

2 Embedded Critical Systems

Historically, critical embedded systems are based on (a variation of) the V-Schema. A V-schema includes the following phases: requirement, design, development (or implementation), integration and validation. The V-Schema is introduced in the safety standard IEC 61508 [6] to deal with software and system dependability. This standard is an umbrella of safety-related domain standards, such as the nuclear [7, 8, 9]; the railway [10, 11, 12] and, more recently, the automotive [13] application domains.

Finally, the variations of the methodology/process promoted by the literature and/or in place in the industry shall meet the certification process (including audit).

Among the main methodologies that have had a crescendo in their success in the scientific and industrial community, we recall the following ones:

correction-by-construction was firstly introduced to address correction for code [14], and then extended to the system's components [15, 16]. At system level, correction-by-construction means a design of a system's component such that we can automatically generate a code, which is correct with respect to a given specification. Correction-by-construction inherently involved *composability* and *compositionality* [15], i.e. the ability of a component to preserve properties during its integration in another system, and to extend properties from a set of components to the system which includes them.

(semi)-automatic generation of code is an issue, strictly related to correction-by-construction approach. It allows engineers to work with an abstraction level higher than code. It is expected to better manage complexity and therefore to reduce human errors.

separation of concerns entails not only the separation between software and hardware, required by safety norms, but it is adopted to address the separation between functional and non-functional properties (safety, real-time, performance, etc..).

modeling is more and more deployed on industries. Among the most adopted standards we have: UML and its profiles (for example, SysML) promoted by the OMG, AADL [17] and Simulink models.

modular pre-certification allows engineers to anticipate and structured the argumentation needed for the certification process. The underlying idea is to decrease the (high) cost linked to the certification of the whole system whenever a single component is modified. GSN (Goal Structural Notation) is a standard which aims to improve the certification process [18]. GSN exploits graphical notations and models (1) by specifying *safety objectives* of a system and (2) structuring the strategy in blocks to achieve the safety objectives.

In summary, most of the critical embedded system methodologies are devoted to the control of instructions, where real-time parameters play a crucial role to ensure safety. Designers and safety engineers work under the hypothesis that the environment is known.

3 Cyber-Physical Systems and IoT

Cyber-physiques systems and IoT consider the physical and the networking systems as an integral part during the design and the analysis of the system under development. Safety issues plays an important role in the CPS-IoT context as following example shows.

The importance to being safety In 2011, Iranian engineers capture a drone from the United States by exploiting the weakness in the GPS networking and combining them with the safe mode procedure embedded in the drone. More in detail, Iranian engineers change the GPS parameters, which provides the position to the drone, and the behavior of the drone from 'nominal' to 'degraded'. Therefore, the drone exploits the automatic safe mode procedure to come back to its base, which has also been changed. The drone lands to the Iranian base, without crash and with catastrophic consequences on the military plan. This example highlights how networking weakness influence the (nominal, degraded, safe) mode of an autonomous vehicle, and emphasizes the importance of safety issues.

In CPS-IoT design and analysis, we have witnessed in an increase of methodologies that address: the inclusion of networking parameters, to deal with e.g. latency, performance and end-to-end properties; a change in the measures of realtime parameters, that embrace physical systems (sensors and actuators) and the virtualization; the consideration of the 'unknow' parameter in the design phases [19]; the importance of the human person role.

In this regards, two different, albeit similar, communities, drones and data fusion, have highlighted the role of human and its impact on the design, development and analysis.

In 'What Drones inherits from their ancestors' [20], Clarke analyzes the human role and the autonomy levels of a drone to assess quality assurance measures. The author emphasizes that 'Even where the decisions delegated to drones are structured, the reliability of drone behaviour may not be high, because of inadequate quality assurance and inadequate audit'. Moreover - he continues- we have witnessed 'an irrational preference by humans to submit to the judgments of their peers rather than of machines: If someone is going to make a mistake costly to a human, better for it to be an understandably incompetent human than a mysteriously incompetent machine' [20]. Although the Clarks' works concern drones, the results can be easy applied to the fully autonomous car. Are-we ready to provide our total confidence and delegate the full control to the software of an autonomous car when we travel with our children in that car? In other words, even if the literature and real-case analysis have proved that automation can better assist a human person to do a job and decrease the number of human errors, humans have still a distrust to rely their safety on a full automatic robot.

In data fusion, D. Hall & al clearly introduces a human behavior has a relevant step in the 'Level-5 Information Fusion' [21]. Intuitively, level-5 Information Fusion calculates the position and kinematics parameters of an object, addresses the relationship with other objects and the environment, looks to prediction and multi-perspective assessment, deals with performance parameter and process control, and finally addresses the human computer interaction. This model was firstly introduced to deal with military applications by suitably transforming multi-sensors data into information. Then, it has been applied in different application domains (e.g. environmental, cyber-security and medical one). In the first versions, the human phase was not directly integrated (only 4-levels) or adequately developed. The focus mainly addressed a 'centralized' architecture. The last Level-5 Information Fusion version capitalizes on the change in the behaviors of the millennium and X generations in the use of digital technologies and their expectation on the social impacts. As a result, human person acquires relevance and Level-5 Information Fusion opens (new) research areas that are devoted to increase the quality of services for human [21].

Finally, the role of human is central in Design Thinking methodologies, today considered as a means to increase social innovations and CPS-IoT applications. Historically, Design Thinking was introduced in the 60's of the last century and it is evolved throughout time. It puts human persons at the center of the creation and development of technologies and ideas, as well as the target. In other words, thanks to design thinking we are moving from 'system's functionality' to 'services for human persons', where for example the comfort of a service is taken under consideration. In 2008, the Bill & Melinda Gates Foundation adopts design thinking of 'grassroots nongovernmental organizations working with small farmers in the developing world' [22].

Unlike embedded critical systems, critical CPS-IoT emphasize the human component in the innovation of technologies and methods, and in the development and use of products. Moreover, awareness of accessing to only a partial information involves an element of novelty in the methodologies, which aren't anymore based on the hypothesis that the environment and the system is perfectly controllable because we know everything about it. The CPSwarm European project provides us a capital example of the trend we have analyzed in this paper. Let us briefly introduce it.

3.1 CPSwarm

The CPSwarm, funded by the European Commission, started on 1st January 2017 project and could provide first help in this direction. As CPS find applications in a number of large-scale, safety-critical domains as. transportation and the increased CPS adoption has resulted in the maturation of solutions for CPS development, CPSwarm project aims at studying interactions amongst CPS might lead to new behaviors and emerging properties, often with unpredictable results [23]. CPSwarm aims to manage these interactions since early design stages and tackled and will propose a new science of system integration and tools to support engineering of CPS swarms. In this context, the tools will ease development and integration of complex herds of heterogeneous CPS that collaborate based on local policies and that exhibit a collective behavior capable of solving complex, industrial-driven, real-world problems. Model-centric design and predictive engineering are the pillars of the project, enabling definition, composition, verification and simulation of collaborative, autonomous CPS while accounting for various dynamics, constraints and for safety, performance and cost efficiency issues. CPSwarm pushes forward CPS engineering at a larger scale, with an expected significant reduction of development time and costs. Project results will be tested in real-world use cases in 3 different domains: swarms of Unmanned Aerial Vehicles and Rovers for safety and security purposes; autonomous driving for freight vehicles; and swarm of collaborating robots in logistics.

Subheadings are set in 11pt Times bold with 3pt before and 3pt after; the style is SH Sub heading. Further levels of heading are left to the author. It is suggested that they use the subheading style but with italic rather than bold.

4 Social Impact

The industrial and European commission are devoting their effort to the study and the realization of autonomous vehicles. However, their expected massive adoption from citizen, the municipality and police will potentially have a strong and direct impact on our life and, more in general, on our (future) society. In this regards, humanist disciplines (such as Sociology, Philosophy, Jurisprudence) are regarding with interest to the impact of the transportation of the future in our society.

For example, the use of (full autonomous) drones for public services (urgency, police, mountain accidents, etc..) is in general welcome. However, the private use of drones (sports, journalism) is less accepted, as discussed in [24, 25].

The autonomy of vehicles ought to change the regularization of roads and involves country agreements. At the infrastructure level, it is still unclear how put together autonomous and non-fully autonomous vehicles (included motorcars and bikes) in a road. The municipality of Paris, RATP (railway) and the SME EasyMile are testing an autonomous bus from two railway stations in Paris (Lyon et Austerlitz). To guarantee safety properties, the autonomous bus has a limited velocity (about 30 km/h) and the road (Charles de Gaulle bridge) is only devoted to it, i.e. other vehicles cannot access to the road. This solution is similar to that applied for automatic metros (for example the Antony/Orly connection) and avoids potential collisions with other type of vehicles.

5 Conclusion

This position paper provides an overview of methodologies to deal with safety from embedded to cyber-physical and IoT critical systems. Among the main novelties, the human component intervenes in the design and analysis phases, and highlights the importance of both the use and the comfort of a product. This novelty impacts safety and our perception of a CPS-IoT product. In this regards, the European commission is devoting effort to study and legislate on the relationship between the human component and the level of autonomy [26].

Acknowledgement

The authors thank Laurent Soulier (CEA) for the fruitful discussions on the transportation of the future.

References

[16] Matt McFarland, "Google's Waymo sues Uber over self-driving car technology". http://money.cnn. com/, February 2017.

[17] Danny Yadron and Dan Tynan, "Tesla driver dies in first fatal crash while using autopilot mode". https: //www.theguardian.com/, July 2016.

[18] http://www.centertechnews.com/2015/06/vehicle-to-vehicle-communication-to\ -save\-lives\-on\-the\-road\-driving\-cars.html.

[19] SNFC. http://www.sncf-reseau.fr/fr/ les-drones-au-service-des-entreprises, 2015.

[20] E. Soubiran, F. Guenab, D. Cancila, A. Koudri, and L. Wouters, *Ensuring dependability and performance for cps design: Application to a signaling system*, I Cyber-Physical Systems: Foundations, Principles and Applications (H. Song, D. B. Rawat, S. Jeschke, and C. Brecher, eds.), ch. 23, pp. 363–375, Elsevier, 2016.

[21] IEC. 61508, Functional Safety of Electrical, Electronic and Programmable Electronic Systems., 2000.

[22] IEC 60880, "Nuclear power plants – Instrumentation and control systems important to safety – Software aspects for computer-based systems performing category A functions ." IEC standard.

[23] IEC 61513, "Nuclear power plants – Instrumentation and control for systems important to safety – General requirements for systems." IEC standard.

[24] IEC 62138, "Nuclear power plants Instrumentation and control important for safety – Software aspects for computer-based systems performing category B or C functions." IEC standard.

[25] CENELEC 50126, "Railway applications - The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS) - Part 2: Systems approach to safety." CENELEC standard http: //www.cenelec.eu/, 2012.

[26] CENELEC 50128, "Railway applications - Communications, signaling and processing systems – Software for railway control and protection systems." CENELEC standard.

[27] CENELEC 50129, "Railway applications - Communications, signaling and processing systems - safety related electronic systems." CENELEC standard.

[28] ISO 26262, "Road vehicles—Functional safety." ISO standard.

[29] R. Chapman, "Correctness by construction: a manifesto for high integrity software," in SCS '05: Proceedings of the 10th Australian workshop on Safety critical systems and software, Australian Computer Society, Inc., 2006.

[30] J. Sifakis, "Embedded Systems - Challenges and Work Directions," in Principles of Distributed Systems (LNCS, ed.), vol. 3544, 2005.

[31] D. Cancila, R. Passerone, T. Vardanega, and M. Panunzio, "Toward Correctness in the Specification and Handling of Non-Functional Attributes of High-Integrity Real-Time Embedded Systems," IEEETransactions on Industrial Informatics, May 2010.

[32] SAE, "Architecture Analysis and Design Language (AADL)." www.aadl.info/aadl/currentsite/.

[33] The GSN Working Group Online, "Goal Structuring Notation (GSN)." https://esstatic. fbk.eu/tools/ocra/

[34] U-TEST H2020 Project, "Testing Cyber-Physical Systems under Uncertainty: Systematic, Extensible, and Configurable Model-based and Search-based Testing Methodologies." http://www.u-test.eu/.

[35] R. Clarke, "What drones inherit from their ancestors," Elsevier Computer Law and Security Review, pp. 247–262, 2014.

[36] D. L. Hall, S. A. H. McMullen, and C. M. Hall, "New Perspectives on Level-5 Information Fusion: The Impact of Advances in Information Technology and User Behavior," in 2015 IEEE International Conference on Multisensor Fusion and lntegration for Intelligent Systems (MFI), 2015.

[37] T. Brown and J. Wyatt, "Design Thinking for Social Innovation," tech. rep., Stanford Social Innovation Review, 2010.

[38] "CPSwarm Project." http://www.cpswarm.eu/.

[39] F. Klauser and S. Pedrozo, "Power and space in the drone age: a literature review and politico-geographical research agenda," Geographica Elevita, pp. 285–293, 2015.

[40] F. Klauser and S. Pedrozo.

http://www.rts.ch/info/sciences-tech, 2016.

[41] D.-G. for External Policies of the Union DIRECTIRATE B Policy Department, "Human rights implications of the usage of drones and unmanned robots in warfare," tech. rep., European Parlament, 2013.

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
Email: Info@Ada-DK.org
URL: Ada-DK.org

Ada-Deutschland

Dr. Hubert B. Keller
Karlsruher Institut für Technologie (KIT)
Institut für Angewandte Informatik (IAI)
Campus Nord, Gebäude 445, Raum 243
Postfach 3640
76021 Karlsruhe
Germany
Email: Hubert.Keller@kit.edu
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