SS-CPS&IoT’2025

6th Summer School on Cyber-Physical Systems and Internet-of-Things

Budva, Montenegro, June 10-14, 2025

Proceedings of the 6th Summer School on Cyber-Physical Systems and Internet-of-Things

Vol. VI

Editors:

Lech Jóźwiak

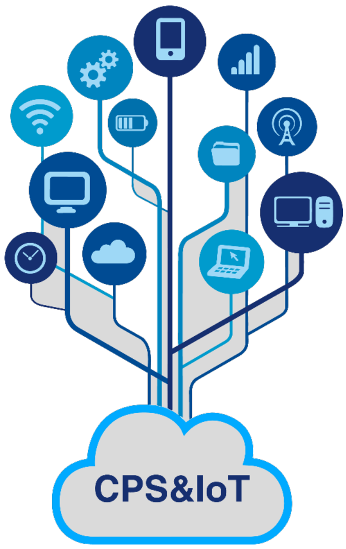
Eindhoven University of Technology, The Netherlands

Radovan Stojanović

University of Montenegro, MECOnet, Montenegro

Authors:

Anish Bhobe, Dominique Blouin, Florian Lorber, Hans Vangheluwe, Iryna de Albuquerque Silva, Joachim Denil, Joeri Exelmans, Jovan Djurković, Lech Jóźwiak, Nabil Abdennadher, Natalie Simson, Radovan Stojanović, Rakshit Mittal, Reiner John, Said Hamdioui, Selim Solmaz, Simon Van Mierlo, Wolfgang Ecker, Yara Hallak and Zakaria Chihani.



MECOnet, MANT

Montenegro, June 2025

Disclaimer

The responsibility for content in presentations and other contributions in this publication rests solely with their authors.

The official website of the event:
https://mecoconference.me/ss-cpsiot2025/

Proceedings of the 6th Summer School on Cyber-Physical Systems and Internet-of-Things, Vol. VI, June 2025

Edited by Lech Jóźwiak and Radovan Stojanović

Technical editors:

Prof. dr Budimir Lutovac, University of Montenegro
Jovan Djurkovic, MECOnet

Contributors:

Anish Bhobe, Dominique Blouin, Florian Lorber, Hans Vangheluwe, Iryna de Albuquerque Silva, Joachim Denil, Joeri Exelmans, Jovan Djurkovic, Lech Jóźwiak, Nabil Abdennadher, Natalie Simson, Radovan Stojanović, Rakshit Mittal, Reiner John, Said Hamdioui, Selim Solmaz, Simon Van Mierlo, Wolfgang Ecker, Yara Hallak and Zakaria Chihani.

Cite as:

Author/s, "Title of contribution-presentation", in Proceedings of the 6th Summer School on Cyber-Physical Systems and Internet-of Things, Editors: Lech Jozwiak and Radovan Stojanovic, Authors: Anish Bhobe, Dominique Blouin, Florian Lorber, Hans Vangheluwe, Iryna de Albuquerque Silva, Joachim Denil, Joeri Exelmans, Jovan Djurković Lech Jóźwiak, Nabil Abdennadher, Natalie Simson, Radovan Stojanović, Rakshit Mittal, Reiner John, Said Hamdioui, Selim Solmaz, Simon Van Mierlo, Wolfgang Ecker, Yara Hallak and Zakaria Chihani, Vol. VI, June 2025, pp. xx-yy, DOI: https://doi.org/10.5281/zenodo.16102824.

Published by Consortium of MECOnet d.o.o. and MANT Association.

Filipa Lainovica 19, Podgorica, Montenegro

Message from the Chairs and Editors

Welcome to the 6th Summer School on Cyber-Physical Systems and Internet of Things (SS-CPS&IoT'2025), held once again in beautiful Budva, Montenegro, in a hybrid format.

This Summer School continues its mission to:

- provide advanced training in CPS, IoT and AI for researchers, developers, engineers, educators, students, entrepreneurs, investors, and policymakers,
- share cutting-edge knowledge and outcomes from European R&D projects, and
- foster international collaboration among experts in these fast-growing fields.

Open to all, the School recommends participants have at least BSc level knowledge or equivalent experience in engineering, computer science, or related fields. Industry professionals are especially welcome.

SS-CPS&IoT is more than a learning event, it is a networking opportunity where participants engage with top researchers and industry leaders, exchange ideas, and build future collaborations.

What sets this School apart is its focus on practical, up-to-date content delivered by leading experts, often based on active European research projects. Lectures, demonstrations, and hands-on sessions expose attendees to real-world challenges, industrial case studies, and modern CPS/IoT solutions.

This year's program also featured prominent speakers from both Europe and beyond. SS-CPS&IoT'2025 was held alongside the CPSIoT'2025 Conference and MECO'2025, providing participants with additional learning and presentation opportunities.

Over four intensive days (equivalent to 3 ECTS credits), attendees benefited from 32 hours of expert sessions and conference events. The detailed program and contributors are listed in this volume of Proceedings, a valuable and cited resource in the CPS, IoT and AI community.

Finally, we extend our sincere thanks to all lecturers, presenters, and contributors, and congratulate our 6th generation of graduates for their impressive achievements.

We look forward to welcoming you again next year in good health and friendship.

Warm regards,

Lech Jóźwiak – Eindhoven University of Technology, Netherlands
Radovan Stojanović – University of Montenegro, Montenegro
Jovan Djurković – MECOnet d.o.o., Montenegro

Budva, June 2025

Contents

CPS&IoT

CPS&IoT'2025 Summer School

Budva, Montenegro
June 10-14, 2025

Introduction

Lech Jóźwiak and Radovan Stojanović

1

Introduction

- Systemic drawbacks of the traditional economy and cumulation of bad decisions driven by the short-term profit and made without adequately accounting for long-term consequences resulted in the **huge global environmental disaster**
- Innovations exploiting modern CPS, AI and IoT technologies have a high potential to significantly improve systems used by us or that we are part of
- To recover from the environmental disaster and further develop:
  - *a model of a well regulated and controlled effective and efficient system should be applied to all kinds of systems, collaboration chains and related flows*
  - *modern CPS and IoT technologies should be used to much better control and optimize the social, physical and life systems than till now*
  - *methodologies of circular regenerative economy and quality-driven design should be used to design the systems*
- In this CPS&IoT Summer School you will have a unique occasion to be informed on and to discuss the most recent European R&D developments in CPS, AI and IoT

2

Outline of the CPS&IoT'2023 Summer School

1. Introduction to CPS and IoT
2. Green CPS and IoT
3. Computing and communication technologies for CPS and IoT
4. Artificial Intelligence, Machine Learning and Edge Computing
5. Trustworthy CPS and IoT: reliability, security and safety
6. Modelling, design and implementation of CPS and IoT
7. Energy-efficient computing for CPS and IoT
8. Closing of the CPS&IoT2023 Summer School

3

CPS&IoT'2025 Summer School
Budva, Montenegro, June 10-14, 2025

Green Systems for Green World

Lech Jóźwiak
Professor Emeritus
Department of Electronic Systems
Faculty of Electrical Engineering
Eindhoven University of Technology
cpsiot2024@gmail.com

1

Outline

1. Introduction
2. Modern cyber-physical systems (CPS)
3. Importance of modern CPS and IoT
4. Challenges of advanced CPS development
5. Computing technology for advanced CPS
6. Environmental crisis and environmental footprint of CPS and IoT
7. Importance of advanced green CPS and IoT for environmental recovery
8. IoT for advanced green CPS
9. Conclusion

2

Introduction: Aims of this tutorial

- The two main aims of this tutorial are the following:
  - *to make the participants aware of the necessity of green CPS and IoT*
  - ***to prepare the ground for the whole CPS&IoT'2021 Summer School***
- This means in particular:
  - to introduce several basic definitions related to CPS
  - to explain the necessity of green CPS and IoT
  - to sketch the CPS scene, what includes:
    - introduction to modern CPS and IoT, their importance, their ongoing revolution, and challenges of their development, and
    - explanation of the necessity of their holistic multi-objective quality-driven design
- In a separate tutorial I will introduce the methodology of quality-driven green system design

3

Introduction: Further reading for this tutorial

- William J. Ripple, at al: The 2024 state of the climate report: Perilous times on planet Earth, BioScience, Volume 74, Issue 12, December 2024, pp. 812–824
- L. Jóźwiak: Advanced Mobile and Wearable Systems, Microprocessors and Microsystems, Elsevier, Vol. 50, May 2017, pp. 202–221
- L. Jóźwiak: Quality-driven Design in the System-on-a-Chip Era: Why and how?, Journal of Systems Architecture, vol. 47, no. 3-4, Apr. 2001, pp. 201-224
- L. Jóźwiak: Life-inspired Systems and Their Quality-driven Design, Lecture Notes in Computer Science, Vol. 3894, 2006, Springer, pp. 1-16
- Jóźwiak, L.; Lindwer, M.; Corvino, R.; Meloni, P.; Micconi, L.; Madsen, J.; Diken, E.; Gangadharan, D.; Jordans, R.; Pomata, S.; Pop, P.; Tuveri, G.; Raffo, L. and Notarangelo, G.: ASAM: Automatic Architecture Synthesis and Application Mapping, Microprocessors and Microsystems journal, Vol.37, No 8, pp. 1002-1019, 2013
- Jóźwiak, L. and Jan, Y.: Design of Massively Parallel Hardware Multi-Processors for Highly-Demanding Embedded Applications. Microprocessors and Microsystems, Volume 37, Issue 8, November 2013, pp. 1155–1172.
- L. Jóźwiak and S.-A. Ong: Quality-driven Model-based Architecture Synthesis for Real-time Embedded SoCs, Journal of Systems Architecture, Elsevier Science, Amsterdam, The Netherlands, ISSN 1383-7621, Vol. 54, No 3-4, March-April 2008, pp. 349-368
- Many other papers of myself and my former Ph.D. students; many of them referenced in the above papers

4

Introduction: What is a system?

- A system is a *complex whole composed of interrelated, interdependent and/or interacting items* (parts or elements of a system) *that are so intimately connected that they appear and operate as a single unit in relation to the external world* (to other systems)

- Three basic types of systems:
  - *unorganized system* - a mechanical unsystematic conglomerate of objects
  - *organized system* - a systematic, relatively stable and law-governed composition of parts which properties cannot be reduced to the simple sum of the properties of its parts, but involve some new emerging properties resulting from complex composition of the parts' properties (e. g. a molecule, crystal, circuit, computer, machine), and
  - *organic system* - formed not as a composition of some ready-made parts, but being an *integral whole* with distinguishable parts that originate, develop and die together with the whole, and cannot preserve and demonstrate their complete quality without the whole (e. g. life organisms); the characteristic features of the organic systems are the self-development and self-reproduction

- In this presentation organized systems will be considered

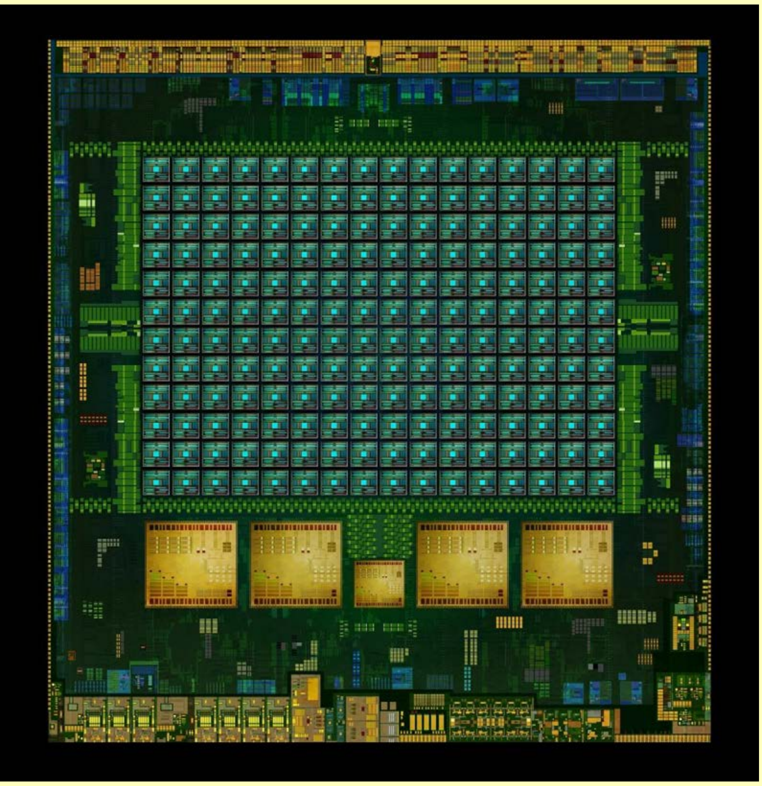
5

Introduction: What are cyber-physical systems?

- A system is a *unity of a process and structure* in which this process takes place
- System design is an activity of *defining an appropriate composition of the system process and structure*
- Cyber comes from Greek adjective *kyberneticos* (*cybernetic*) that means skilled in steering or governing
- Cyber systems are *(parts of) control systems*, i. e. information collecting, processing and communicating systems
- Physical systems are systems in which matter or energy acquisition, processing and transfer take place according to the lows of physics
- Cyber-physical system (CPS) is a compound system engineered through integration of cyber and physical sub-systems or components and/or pre-existing component cyber-physical systems, so that it appears and operates as a single unit in relation to the external world (to other systems)

6

Introduction: very complex MPSoCs



Source: ANANDTECH
(http://www.anandtech.com/show/7622/nvidia-tegra-k1)

- *Modern nano-dimension semiconductor technology enables implementation of a **very complex multiprocessor system on a single chip (MPSoC)***
- **This facilitates a rapid progress in:**
  - ***global networking***
  - ***(mobile) wire-less communication***
  - ***(mobile autonomous) embedded computing***

NVIDIA Tegra K1 massively parallel MPSoC for mobile applications

CPU: (4+1) Cortex-A15 cores

Kepler GPU: 192 CUDA GPU cores

7

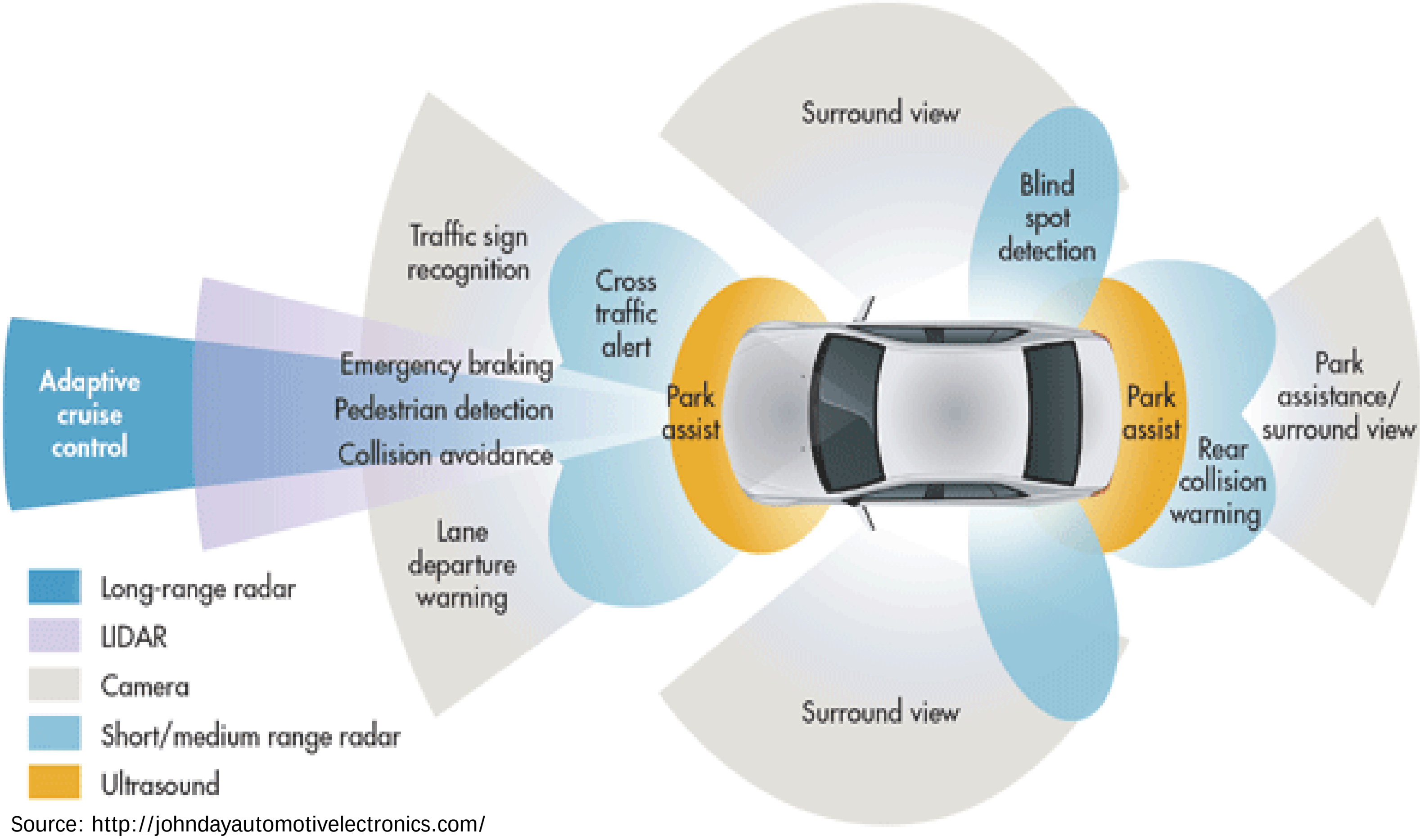
Introduction: cyber-physical technology revolution

- The recent rapid developments in:
  - system-on-a-chip technology
  - common global networking
  - wire-less communication
  - mobile and autonomous computing
  - miniaturized sensors and actuators
  - material technology

  enabled sophisticated and affordable CPS for numerous new applications (e.g. smart robots, homes, cars, etc.) and created a large discrepancy between what is possible and what is used nowadays
- This discrepancy:
  - causes both a very strong technology push and market pull to create new or modified products and services, and
  - results in the *cyber-physical technology revolution*
- Recently, a revolutionary transition has been started from the internet of computers to the internet of smart (mobile) cyber-physical systems (CPS), called Internet of Things (IoT)

8

Examples of modern CPS: autonomously-driving cars



Source: http://johndayautomotivelectronics.com/

Examples of modern CPS: smart wearables

A new wave of the information technology revolution has arrived that creates much more coherent and fit to use CPS and connects them to form the IoT

10

Importance of modern CPS

- Application areas of mobile CPS cover *virtually all socially important application sectors*, including:
  - *consummer applications*, e.g. mobile computing, communication, localization, navigation, gaming, entertainment, fashion, etc.
  - *extension or replacement of human capabilities*, e.g. tele-operation, personal assistance, artificial limbs, implants, etc.
  - *social systems*, e.g. smart health-care and other numerous health-care applications, assisted leaving, law enforcement, public safety, military, etc.
  - *transportation and automotive*, e.g. traffic control, navigation, tracking, communication, personalized customer service, assisted/autonomous driving, etc.
  - *industrial, safety, security and military applications* , e.g. mobile real-time in-the-field surveillance, monitoring, inspection, repair, robotics, instruction, assistance, etc.
  - *commercial applications*, e. g. mobile inventory tracking and customer service, (wearable) augmented reality for touristic and other applications, and many others
- The economic and societal importance of modern CPS is very high and rapidly increases

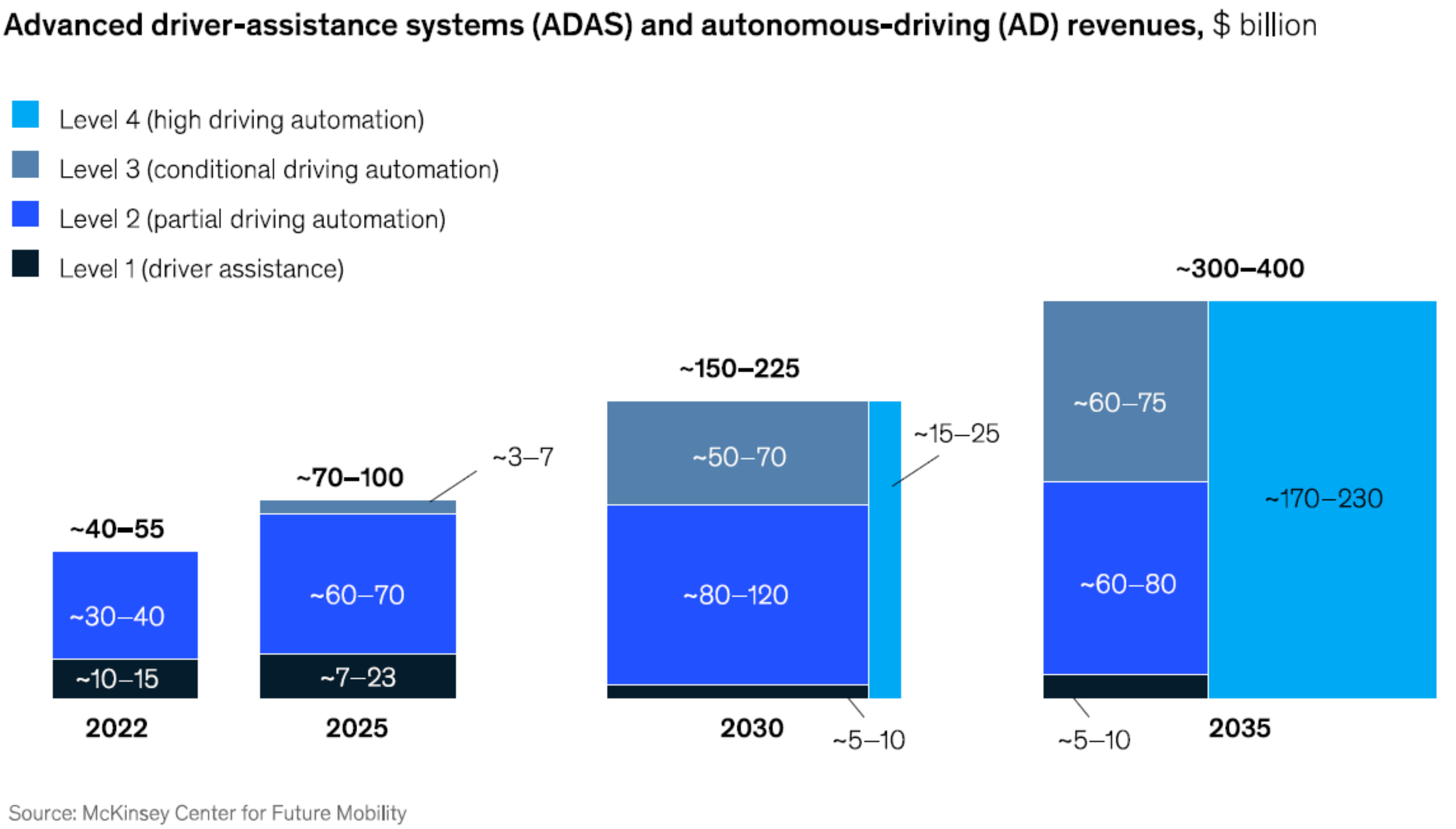
11

Rapid growth of CPS and IoT markets

- The number of connected IoT devices was 12.2 billion in 2021 (IoT Analytics)
- IoT Analytics forecasted 14.4 billion connected IoT devices in 2022 and 27 billion connected IoT devices by 2025
- Allied Market Research finds that the global IoT market size was $740.5 billion in 2020, $878.49 billion in 2021 and is estimated to reach $4,421.6 billion by 2030
- This corresponds to the growth rate at a CAGR of 19.6% between 2021 and 2030
- The strongest contributors to the global IoT market are currently industrial manufacturing, healthcare, consumer electronics, automotive and wearables

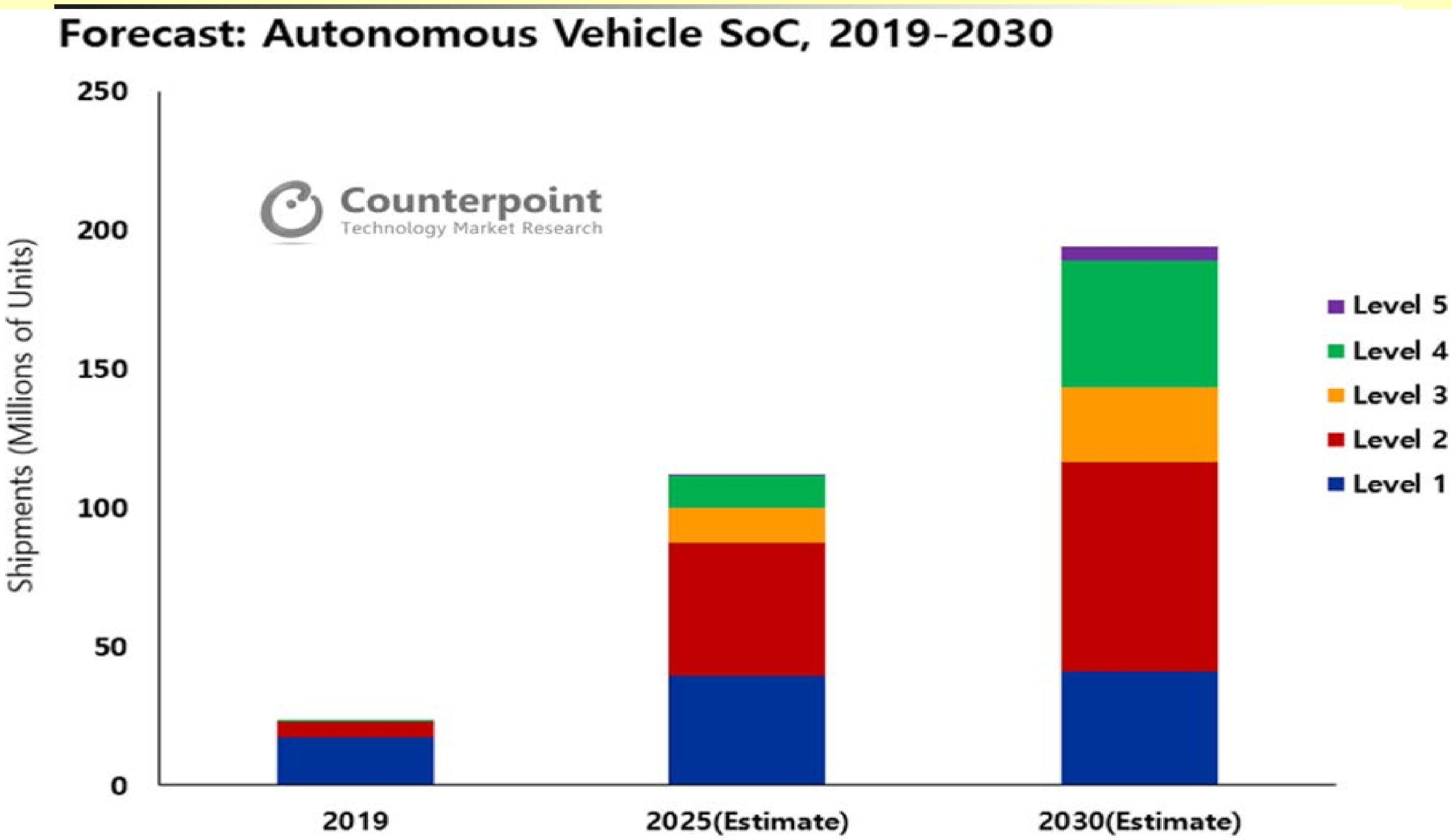
12

Rapid growth of the modern CPS and IoT markets



13

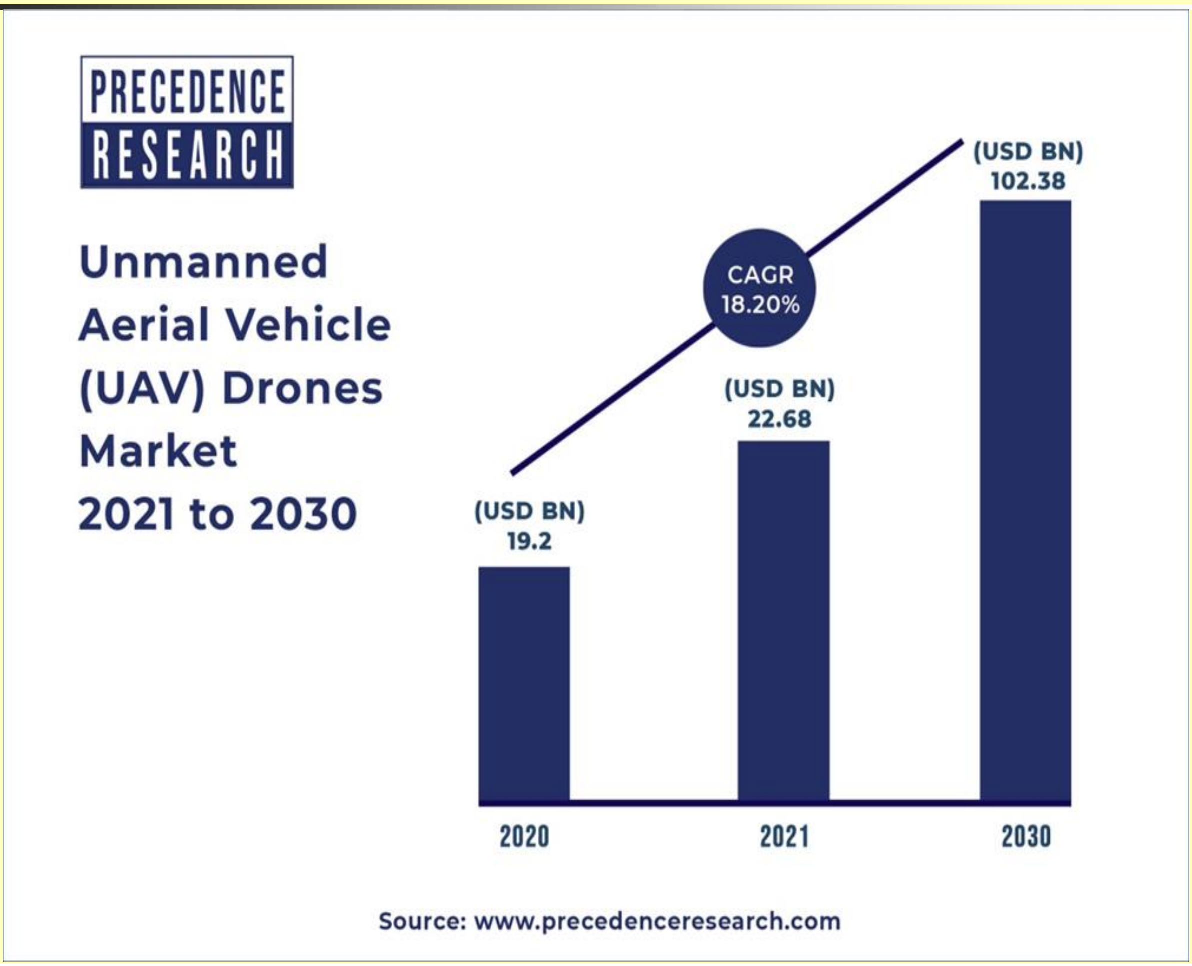
Rapid growth of the modern CPS and IoT markets



Source: Counterpoint, January 2020

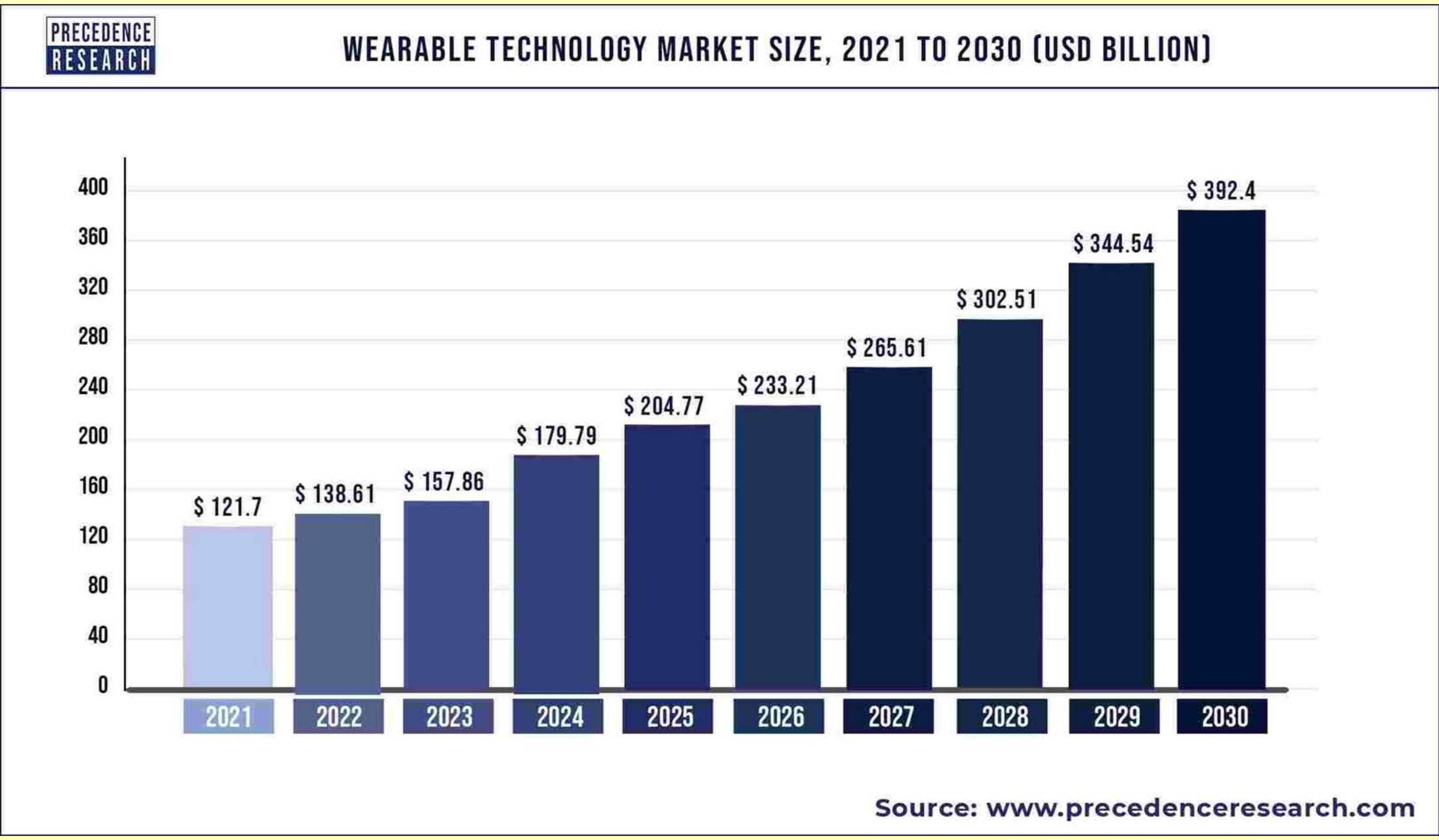
14

Rapid growth of the modern CPS and IoT markets



15

Rapid growth of the modern CPS and IoT markets



16

Rapid growth of chip market related to CPS/IoT in 2021 - 2024

- According to Semiconductor Industry Association (SIA), the global semiconductor industry sales in 2021 increased by 26.2% compared to the 2020 to the highest-ever annual value of $556 billion
- A record number of 1.15 trillion semiconductor units were shipped in 2021
- In 2022, the global semiconductor industry sales achieved a new record value of $574 billion
- In 2024, the global semiconductor industry sales achieved once more a new record value of $627.6 billion
- A double-digit market growth is projected for 2025
- The growth was mainly driven by the automotive, industrial and consumer application sectors, while the sale of chips for PC/computer sector substantially decreased (by 6%) in these years
- As semiconductors are used in all modern CPS/IoT-related applications further semiconductors sales growth in the CPS/IoT-related sectors is expected to continue up to 2030

17

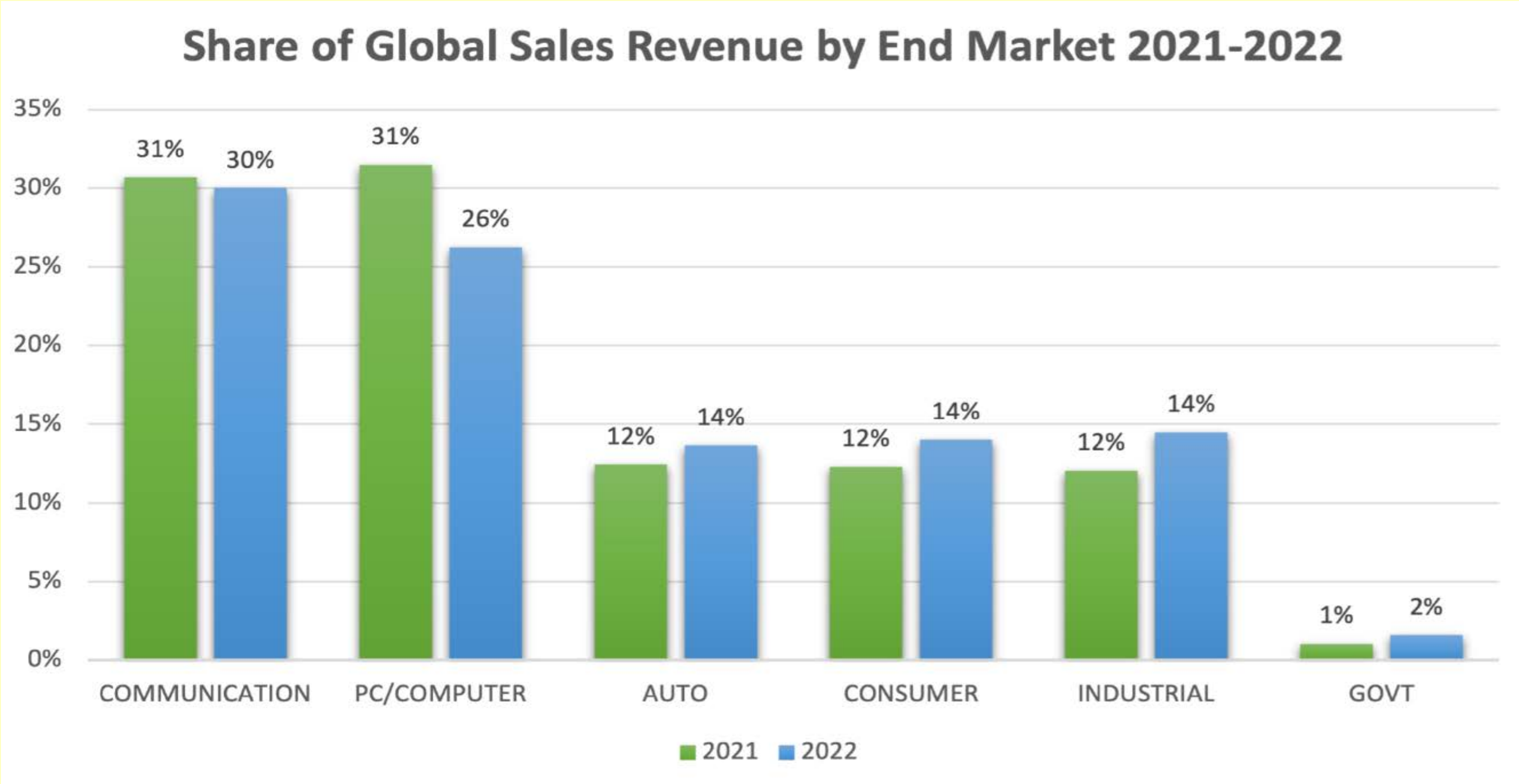
Global semiconductor market from 2021 to 2024

- Global semiconductor sales hit $627.6 billion in 2024, an increase of 19.1% compared to the 2023 total of $526.8 billion

Source: SIA

18

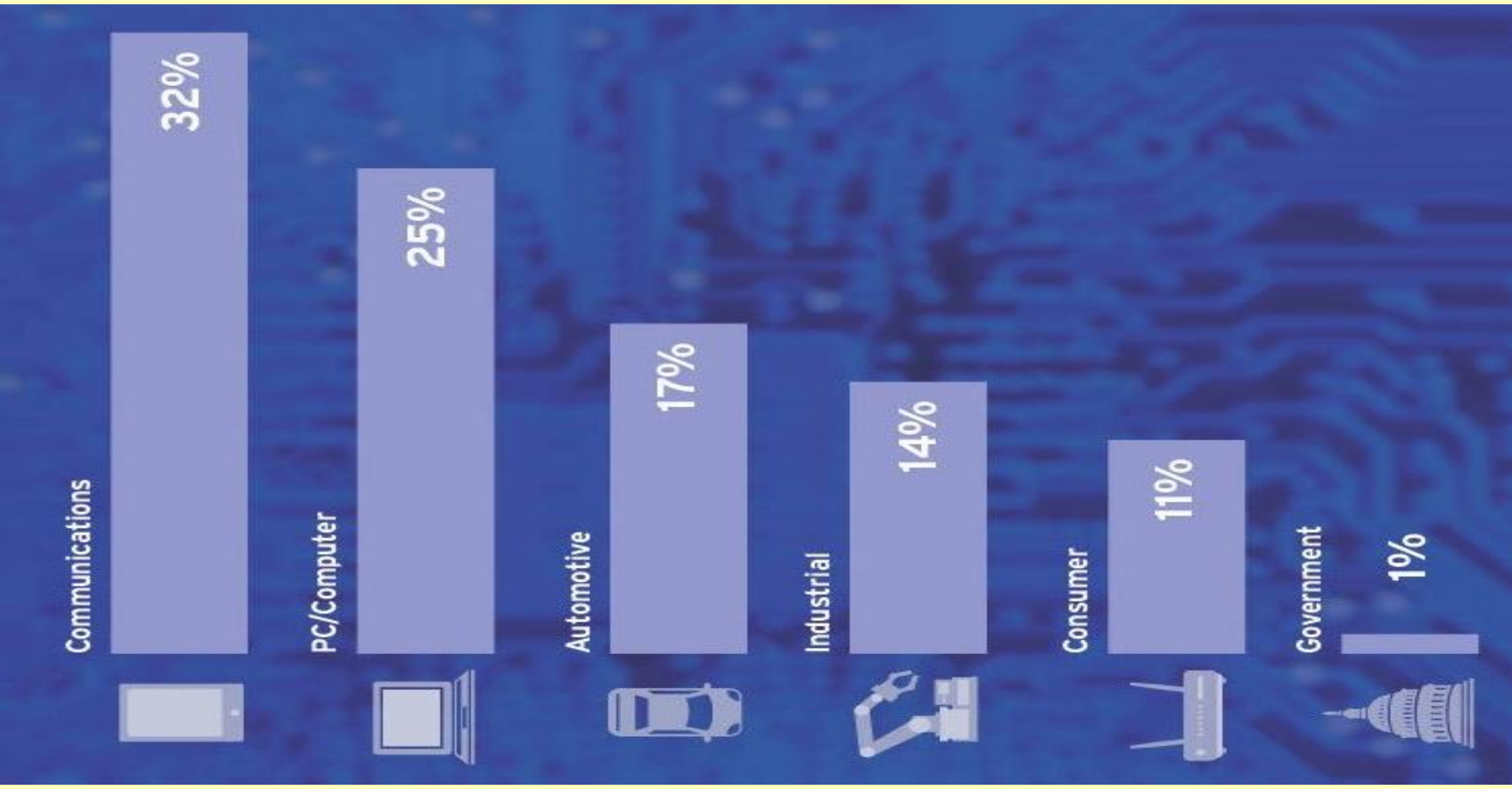
Semiconductor market related to CPS and IoT in 2021 and 2022



Source: SIA

- In 2022 PC/COMPUTERs only accounted for 26%, while a large majority of communication and the automotive, industrial and conssumer are related to CPS and IoT 19

Semiconductor market related to CPS and IoT in 2024



- In 2025 PC/COMPUTERs only accounted for 25%, while a large majority of communication and the automotive, industrial and conssumer are related to CPS and IoT

Source: SIA

20

Challenges: unusual complexity and ultra-high demands

- The huge and rapidly developing markets of sophisticated CPS and IoT represent great opportunities
- These opportunities come with a price of:
  - unusual system complexity and heterogeneity, resulting from *convergence and combination of various applications and technologies* in one system or even on one chip, and
  - stringent and difficult to satisfy requirements of modern applications
- Smart cars, drones and various wearable systems:
  - involve big instant data from multiple complex sensors (e.g. camera, radar, lidar, ultrasonic, sensor network tissues, etc.) and from other systems, used for mobile vision, imaging, virtual or augmented reality, etc.
  - are required to provide continuous autonomous service in a long time
  - are safety-critical
- In consequence, they demand a guaranteed (ultra-)high performance and/or (ultra-)low energy consumption, while requiring a high reliability, safety and security

21

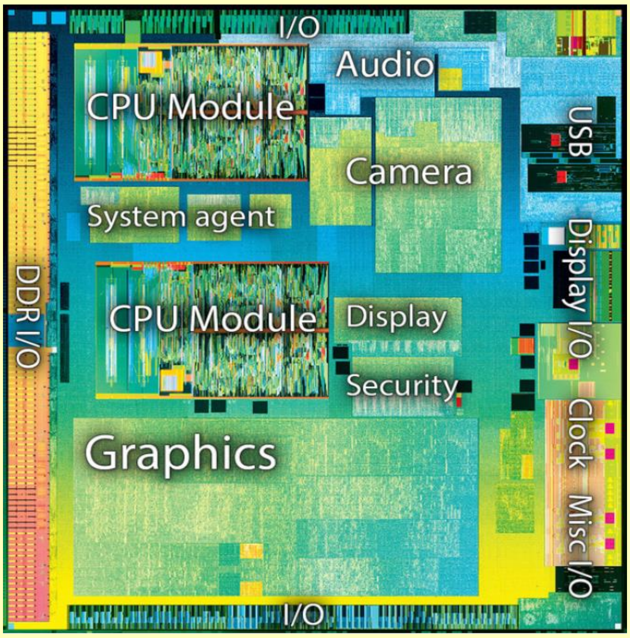
Challenges: application parallelism and heterogeneity

- The modern complex applications that require ultra-high performance and/or ultra-low energy consumption:
  - are from their very nature heterogeneous
  - include numerous different algorithms involving various kinds of massive parallelism: data parallelism, and task-level, instruction-level and operation-level functional parallelism
- To adequately serve these applications:
  - heterogeneous computation platforms have to be exploited
  - processing engines with parallel multi-processor macro-architectures and parallel processor micro-architectures have to be constructed
  - different parts of complex applications involving different kinds of parallelism have to be implemented with corresponding different application-part specific parallel hardware
  - multiple different or identical processors, each operating on a (partly) different data sub-set, have to work concurrently to realize the ultra-high throughput and ultra-low energy consumption

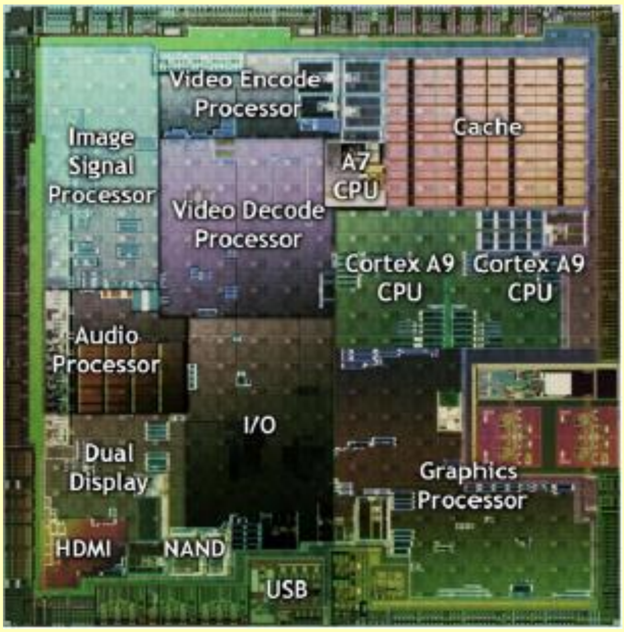
22

Challenges: application complexity, parallelism and heterogeneity

To implement the highly-demanding complex heterogeneous CPS applications ***complex heterogeneous MPSoCs*** *are needed*



Intel Atom Z3770*



Nvidia Tegra 2+

*Source: http://tweakers.net/reviews/3162/2/intels-atom-bay-trail-de-eerstenieuwe-atom-in-vijf-jaar-zes-verschillende-bay-trails.html
+Source: http://www.anandtech.com/show/4144/lg-optimus-2x-nvidia-tegra-2-reviewthe-first-dual-core-smartphone/3

23

Challenges: application complexity, parallelism and heterogeneity

massively parallel heterogeneous MPSoC for ADAS and similar advanced mobile applications

- Recently NVIDIA introduced Jetson AGX Orin for autonomous cars, advanced UAVs, factory robots, etc. that includes 1792 GPU cores, 56 Tensor Cores and 8-core ARM Cortex-A78AE CPU, and delivers 200 TOPS at only 15 – 40 W

Source: Albert Y.C. Chen, Viscovery

24

Quality-driven Model-based Design

- The rapidly growing system complexity and demands of (ultra-)high performance and/or (ultra-)low energy consumption, while requiring a high reliability, safety and security, created a new difficult situation that cannot be well addressed without an adequate design methodology and design automation
- When considering a **system and design methodology adaptation**, we have first to ask: *what general system approach and design approach seem to be adequate to solve the problems and overcome the challenges*?
- **Predicting the current situation,** more than 20 years ago I proposed such **system paradigm** and **design paradigm**:
  - the paradigms of **life-inspired systems** and **quality-driven design**, and
  - the **methodology of quality-driven model-based system design** based on them
- From that time my research team and our industrial and academic collaborators were researching the application of this methodology to the design and design automation of embedded processors, MPSoCs and CPS, and this research confirmed the adequacy of the quality-driven design methodology
- For "Outstanding Achievements and Contributions to Quality of Electronic Design" I was awarded the Honorary Fellow Award by the International Society for Quality Electronic Design (San Jose, CA, USA, 2008)

25

Quality-driven Design, CPS and IoT for making high-quality systems

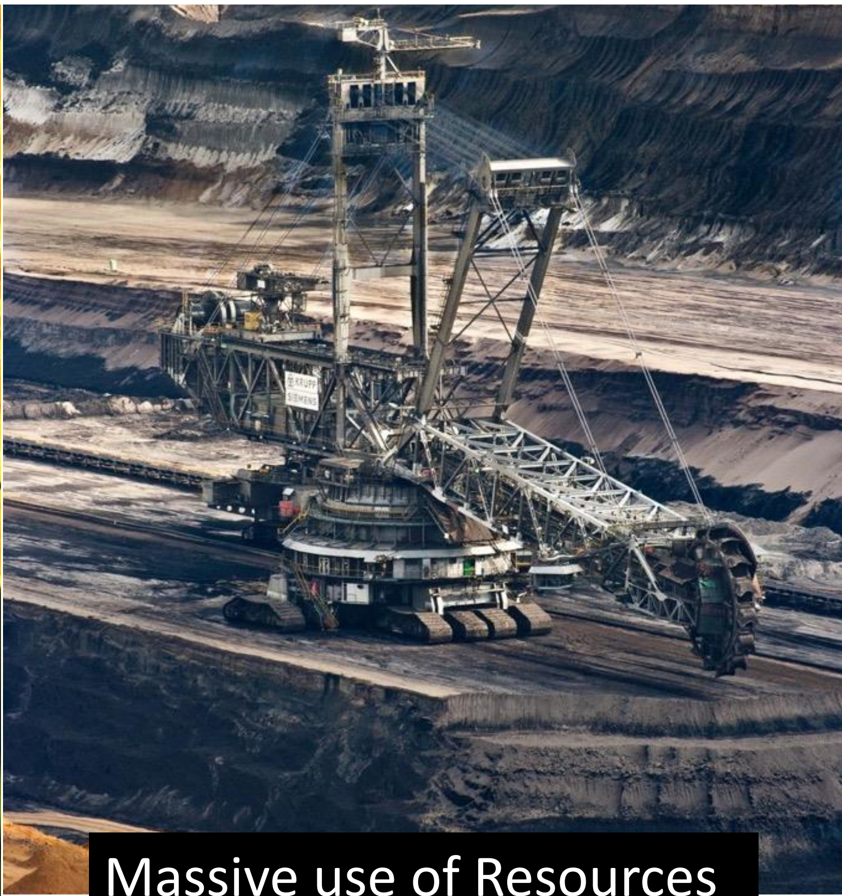
- When using the quality-driven design methodology to develop high-quality collaborating cyber-physical systems, in which the sophisticated cyber systems are tightly integrated with the controlled by them physical, social and life systems, we have a great chance to much better control and optimize the social, physical and life systems than we did it till now
- *With modern CPS and IoT technology we have a great chance to significantly improve most systems used by us or that we are part of*
- We also have no chance to not do this
- *Our systems have to be significantly and immediately improved*
- Why?
- Please watch the following few slides that I got from:
  - Dr. Jean Paul Gueneau de Mussy, Executive Director and Innovation Expert for Materials, Systems and Circular Economy, and
  - "The 2024 state of the climate report" by William J. Ripple, at al

26

Jean Paul GUENEAU DE MUSSY | Materials-Innovation.com

Jean Paul GUENEAU DE MUSSY |

27

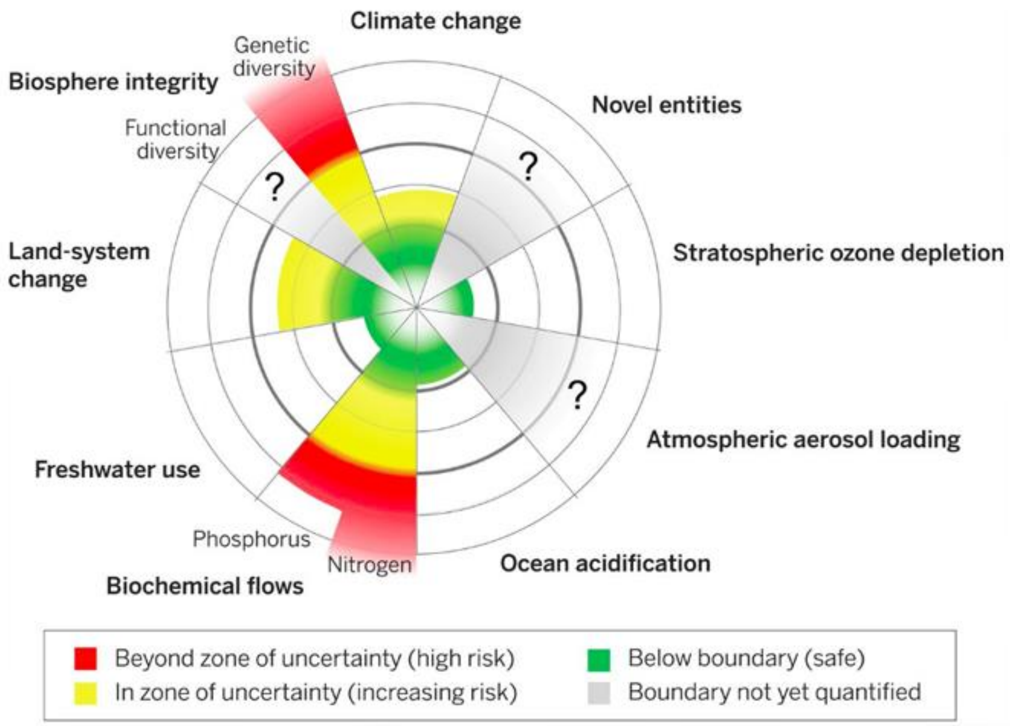


Jean Paul GUENEAU DE MUSSY | Materials-Innovation.com

Jean Paul GUENEAU DE MUSSY |

28

Planetary Boundaries



Johan Rockström et all, February 2017, Volume 46, Issue 1, pp 4–17

Jean Paul GUENEAU DE MUSSY |

The 2024 state of the climate report

- Its first sentence is:
  - We are on the brink of an irreversible climate disaster
- For half a century, numerous scientists alarmed about the impending dangers of climate change driven by increasing greenhouse gases emissions due to fossil fuel combustion and industrial processes, deforestation, and ecosystem change
- ***Despite these warnings, we are still moving in the wrong direction***:
  - fossil fuel emissions have increased to an all-time high
  - the hottest Northern Hemisphere extratropical summer and record-breaking sea surface temperatures occurred in 2023
  - the 3 hottest days ever occurred in July of 2024
  - many other climate records were broken in recent years
  - we are on track to ~2.7 °C peak warming by the year 2100
- We are starting to face unprecedented disasters around the world, as well as human and animal suffering

30

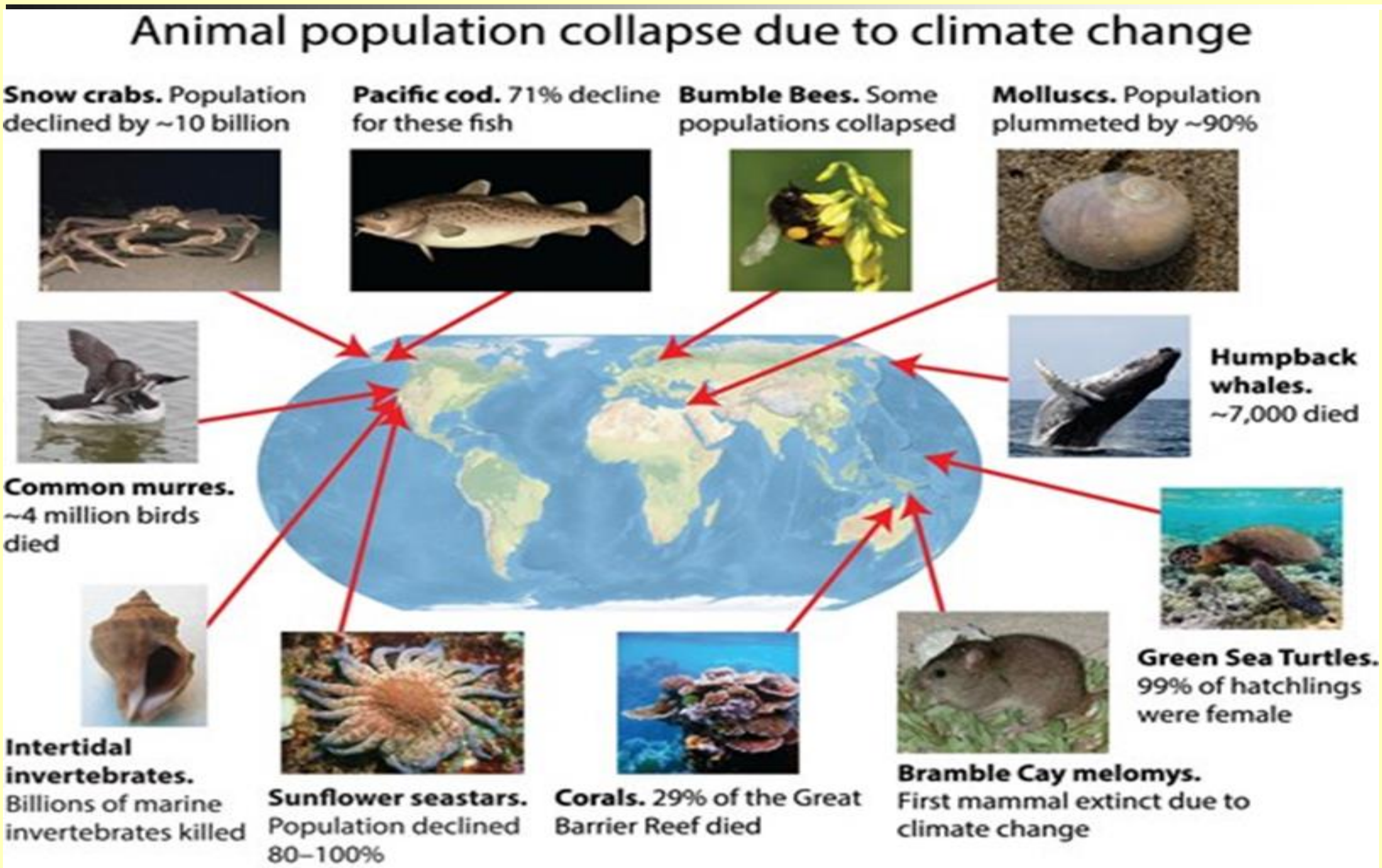
The 2024 state of the climate report

Untold Human Suffering in Pictures

Source: The 2024 state of the climate report, BioScience, Volume 74, Issue 12, December 2024

31

2025 Report: Climate change threats to wild animals



Source: Climate change threats to Earth's wild animals, BioScience, Special Report, 2025, 0, pp. 1–5

32

Huge destruction, chaos, no care for long-term consequences

- These were only a few examples of what was done wrong for a long time with our economic, social, technical and life systems on a global scale, and what resulted in a **huge destruction on a global scale**
- This huge destruction is a result of systemic drawbacks of the traditional economy and very many bad decisions made by numerous governments and companies for a short-term profit only, without accounting for long-term consequences
- Example: the wild chaotic globalization, without carefully designed interfaces and collaboration between very different economic/political systems in different parts of the World and between companies from the very different systems
- Globalization is unavoidable, but the actual costs of the wild globalization were not pay by those who profited, but by the poverty of others and destruction of the World
- The not well regulated and controlled inefficient collaboration chains and related material, product and waste flows of the wild globalization resulted in inefficient use of resources, environment destruction and pollution, climate change, bio-diversity loss, etc.

33

Broader context of the destruction

- Without understanding the broader context of the environmental destruction, we will not be able to effectively recover from it
- **The world is in constant war**: of **evil** against **good**.
- This war is "**eternal**" and has different phases of:
  - "**cold**" war, in the sense of moral, political, economic, etc., war

  and

  - "**hot**" war, in the sense of military conflict, revolution, and other types of enslavement and exploitation of people or destruction and looting of nature and all what humans created.
- Now this war between good and evil is a war between:
  - the world of civilization achievements being humanistic and ecological values, moral and social norms, as: human rights, democracy, self-governance, fair division of welfare, nature protection, etc.

  and

  - the backward old-fashioned world, negating such humanistic and ecological values, moral and social norms, etc.

34

Broader context of the destruction

- Now this war between good and evil is a war between:
  - the world based on the state of law build on humanistic and ecological values, in which all are equal, and which protects everyone, a world where the government elected by the whole society in free and democratic elections acts for the social good within the law, and everyone has free access to information,

  and

  - the world of lawlessness of a totalitarian regime, negating humanistic and ecological values, denying and destroying moral and social norms, destroying or enslaving people, destructing and looting nature and all what humans created, and where society does not have free access to information and is manipulated by totalitarian propaganda.
- Let us observe:
  - how important the role of free access to information and "real" education is, and
  - how disastrous is the lack of free access to information and propaganda instead of "real" education and information

35

Let us act on the side of good

- As people belonging to the best educated part of our societies, let us not only be well educated, but also "good" and "wise".
- Let us be on the side of good in this war between good and evil.
- Let's not wait for someone to win this war for us.
- Let us actively fight against evil and do good in all the most effective and efficient ways available to us.
- Let us work for respecting the humanistic and ecological values, and for human rights, democracy, self-governance, fair division of welfare, nature protection, etc.
- Let us inform and educate people.
- As scientists and engineers: let us create "green" cyber-physical systems.
- **How to recover from the environmental disaster**?

36

EUROPE recognizes the CLIMATE and POLUTION CRISIS and starts to take serious measures

EU President **Ursula von der Leyen** unveiled Europe's "Green Deal" plan to fight the crises on Dec. 11, 2019

It represents a stepwise incremental approach to solve the problems

37

How to recover from the disaster?

- The agreed in July 2020 Next Generation EU fund of €750 billion to recover from the crisis caused by the COVID-19 pandemics is added to the regular EU budget for 2021–2027 to result in approximately €1824.3 billion
- As much as 30% of the total amount will be devoted to the climate and environment in compliance with the Paris Climate Agreement
- To recover from the disaster, ***a model of a well regulated and controlled effective and efficient system has to be applied to all kinds of systems, collaboration chains and related flows, implementing***:
  - **regenerative, circular and more local economy**

  and
  - **global ecology**
- In particular, ***this applies to collaboration chains, and related material, energy and information flows in CPS and IoT***
- ***What is circular regenerative economy?***

38

Traditional versus Circular Regenerative economy

- Traditional economy is characterised by assumption of unlimited growth; competition; intensive exploitation of and fighting for non-renewable scarce resources; and short-term profit maximalization, without taking care of the negative long-term economic, social and ecological consequences
- Traditional economy uses linear model: take scarce resources – make – use – dispose waste; it did not pay the actual costs of inefficient resource usage and of the pollution and destruction it made
- Circular regenerative economy is a systemic approach that aims to benefit all: business, society and environment, through:
  - quality-based growth, collaboration and partnership;
  - increasing use of renewable resources, resource sharing and gradually limiting the use of finite resources;
  - introducing biological cycles to regenerate living systems and technical cycles implementing product repair, reuse, sharing, remake, and recycling; and this way minimizing the use of scarce resources and regenerating the environment

39

Innovate applying circular economy and quality-driven design

- The principles of the circular regenerative economy are derived from the same source as the principles of the paradigms of life-inspired systems and quality-driven design proposed by me
- They are derived from the observation of nature, and especially of structures and operations of living organisms, their populations and ecosystems that have demonstrated to effectively, efficiently and robustly work for many millions of years, and are a great source of inspiration
- In relation to technical systems the principles of the circular regenerative economy repeat the main principles of the paradigms of life-inspired systems and quality-driven design
- Implementation of the circular regenerative economy will require **many breakthrough innovations of processes and products**
- All those innovations will have to be designed and implemented
- ***When designing and implementing the innovative processes and products the methodologies of circular regenerative economy and quality-driven design should be used together***

40

What can and should be the role of the modern CPS and IoT technologies in recovery from this disaster?

- The main role of the CPS&IoT technologies can and should be:
  - high increase of the effectiveness and efficiency of the energy and materials consumption in all kinds of systems, collaboration chains and related flows, and
  - big decrease of waste related to the systems, chains and flows
- With their smart sensing, networking, processing and actuation solutions, modern CPS&IoT technologies make it possible to effectively and efficiently collect, transmit, process and use information for (remote) system monitoring, collaboration and control
- Through enabling an effective and efficient energy and materials management in different distributed collaborating systems, and through exploitation of smart grid, smart mobility, smart city, smart home and other smart system concepts, CPS&IoT can very much contribute to achievement of the energy efficiency goals with renewable energy sources and energy harvesting, as well as to reduction of materials consumption and waste

41

We have to recover from this disaster ASAP

- The principles of circular regenerative economy and the quality-driven design methodology should be used together to develop high-quality collaborating cyber-physical systems
- In these systems the sophisticated intelligent cyber systems (controllers) will be tightly integrated with the intelligently controlled and optimized physical, social and life systems
- This way, we have a great chance to much better control and optimize the social, physical and life systems than we did it till now
- This way, we can create green cyber-physical systems
- Innovations exploiting modern CPS and IoT technologies, circular regenerative economy and quality-driven design can significantly improve systems used by us or that we are part of
- Significantly improve does not mean to completely solve the environmental crises
- For this, the unnecessary and inefficient consumption should be eliminated and all social systems should be re-organized to be more fair and more efficient 42

Europe's position in cyber and green technologies

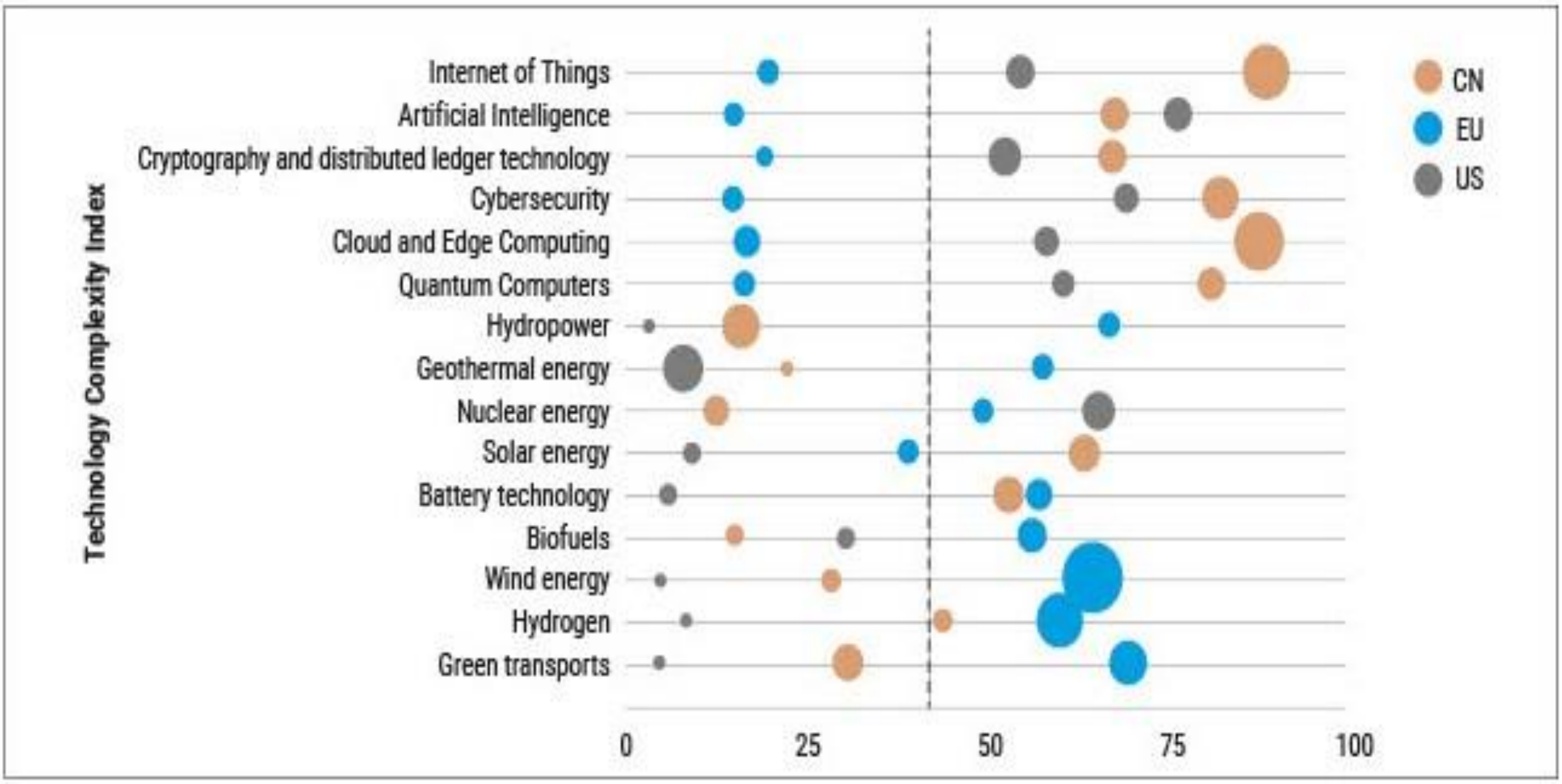


Figure 1: Europe's position in digital and green technologies (2019-2022). The x-axis indicates how easily a country can build a comparative advantage. The size of the bubble indicates how strong it already is [1]

Source: HiPEAC Vision 2025

43

Environmental footprint of cyber systems

- According to https://www.energuide.be, the average energy consumption and CO_2 footprint of a contemporary computer are the following:
  - desktop (basic peripherals included): 200 W/hour in work mode; used for 8h a day *consumes 600 kWh and emits 175 kg of CO_2 per year*;
  - laptop: 50 and 100 W/hour in work mode; used for 8h a day *consumes between 150 and 300 kWh and emits between 44 and 88 kg of CO_2 per year*;
  - in stand-by mode: the consumption/emission of both decrease to a third of the above.
- For microcontrollers (MCUs) and MPSoCs used in CPS, the story is much more complicated
- For them, the actual energy consumed depends on very many factors
- It is difficult to speak about an average energy consumption even for a given single MCU or MPSoC, because the energy consumption very much depends on the actual use and working conditions
- The power consumed by MCU or MPSoC grows with operating frequency, temperature, supply voltage and signal activity

44

Environmental footprint of cyber systems

- Moreover, modern MCUs and MPSoCs often have several different active and energy saving modes (e. g. sleep, deep sleep, standby, etc.) and use the frequency and voltage scaling
- Finally, different MCUs and MPSoCs may have very different energy consumption characteristics, dependent on their architectures and implementation technologies, which in turn depend on the purposes/application fields which a given MCU or MPSoC is supposed to serve
- A simple ultra-low-power MCU for wearables can run in its active mode at several mW
- A complex MPSoC for automotive may use tens to hundreds of Watts
- However, this is only a small part of the whole story
- The environmental footprint of cyber systems in CPS depends not only the embedded processors and their use, but on the usage of fog and cloud computers, and of the communication among all the computers as well

45

Environmental footprint of cyber systems

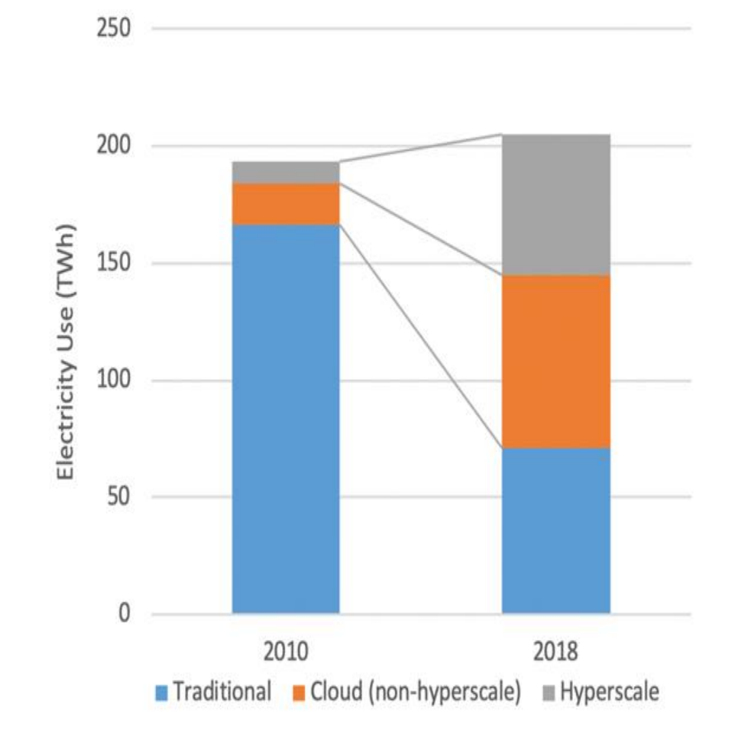


Figure 2. Estimated global data electricity use by data center type, 2010 and 2018. Source: Masanet et al. 2020.

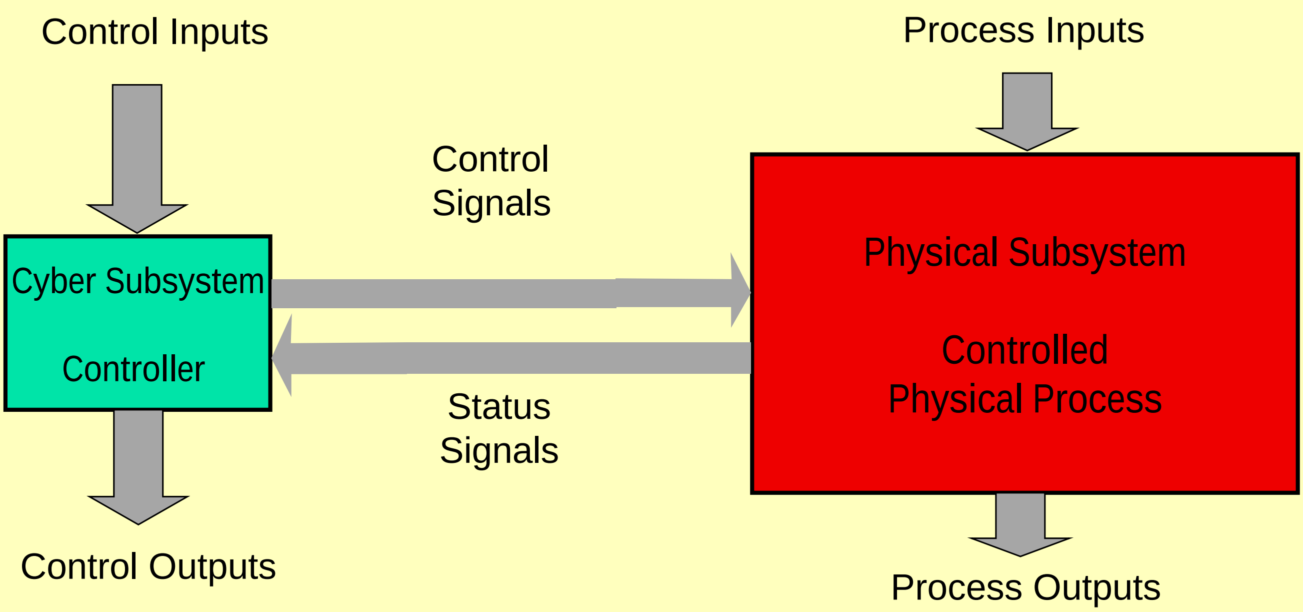
- In 2018 global data centers consumed approximately 205TWh
- This represents 1% of global electric energy use and 0.3% of global CO_2 emission, and is more than the electric energy consumption of a medium country
- In 2019 global data transmission networks consumed 250 TWh
- This is more than 1% of global electric energy use, what corresponds to more than 0.3% of global CO_2 emission
- The demand for data center and network services is increasing

Source: https://energyinnovation.org/2020/03/17/

46

Environmental footprint of cyber-physical systems

General Model of Cyber-Physical System



Usually:

Low Energy

High Energy

47

Environmental footprint of CPS

- The physical subsystem of CPS (implementing the controlled physical process) usually involves much larger material structures and flows, and several times more energy than the cyber subsystem (controller)
- The environmental and other effects are usually much larger from usage of the modern CPS and IoT technology to intelligently control and optimize the physical, social and life systems than from making green only the cyber systems
- We should make green the physical, social and life systems, as well as the cyber systems controlling them and the IoT connecting the collaborating CPS
- The environmental footprint of CPS and IoT depends on the whole CPS and IoT life cycle involving the CPS and IoT design, manufacturing, usage and disposal
- *Manufacturing* usually includes installation, testing and validation
- *Usage* often involves maintenance, repair and enhancement
- Let's focus on IoT

48

Distribution of intelligence, computing resources, services and workloads in the IoT chierarchy

- To transform the big data from multiple sensors to the information being directly used for decisions, while satisfying the stringent requirements of the modern mobile systems, a careful distribution of information delivery and computation services among the different layers of IoT is needed
- For many reasons of primary importance, as:
  - real-time availability of local information
  - guaranteed real-time reaction
  - privacy, security, safety, reliability
  - minimization of energy used, communication traffic, costs, etc.

  a majority of computing and decision making related to advanced CPS should be performed locally in the IoT edge devices, in collaboration among various local IoT edge devices or just above the edge nodes, and not in the higher levels of fog or in cloud
- The higher levels of fog and cloud should only be asked for services if:
  - necessary information or computing resources are not available locally, and
  - reaction-time, security, safety, etc. allow for this

49

Distribution of intelligence, computing resources, services and workloads in the IoT chierarchy

- This requires implementation of advanced intelligent computations and sophisticated powerful embedded computing technology:
  - directly in the IoT edge devices related to the (complex) sensors and actuators, or
  - just above the edge nodes, where the information from different sensors can be combined and based on the combined information the control decisions can be taken and subsequently actuated
- Sophisticated and powerful edge computing has to be used requiring advanced intelligence, processing power and communication capabilities to be pushed towards the edge-nodes of IoT, where the data originate and information is used (i. e. to sensors, controllers and actuators)
- A very good example of the edge computing necessity is the local vehicle-to-vehicle and -infrastructure communication and collaboration necessary for autonomous driving
- In consequence, the IoT for advanced CPS will be substantially different than Internet for other traditional targets

50

Edge Computing, Intelligent Sensors, Edge AI and Edge ML

- This is the reason why Edge Computing, and specifically, intelligent sensors and actuators, as well as edge Artificial Intelligence (edge AI) and edge Machine Learning (edge ML) became very relevant and hot R&D topics recently
- Artificial intelligence (AI) is intelligence demonstrated by organized systems (e.g. machines), in contrast to "natural" intelligence demonstrated by organic systems (e. g. humans or animals)
- An intelligent system is a system that shows a goal-directed behavior
- AI system is a system that analyses the problem, and based on the analysis results, takes actions that maximize the chances of success to achieve the goal
- Machine learning (ML) is a learning implemented in machines through developing methods and algorithms that can "learn"", in the sense of being trained on some set of data, discovering the structure in data, or optimizing own performance for some set of problems through interacting with environment and processing feedback from the environment

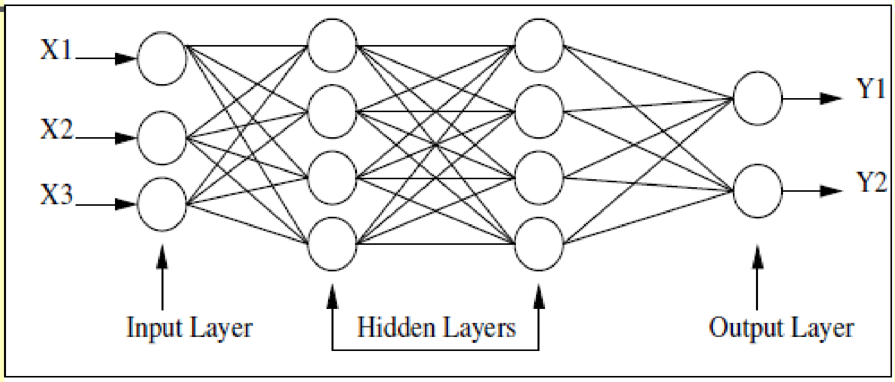
51

Edge AI and Edge ML

- A vast majority of ML methods/algorithms use various models, such as: artificial neural networks, support-vector machines, decision trees, belief networks, etc.
- Based on the training data machine learning methods/algorithms build/train such model which is then used to process additional data to make decisions or predictions
- Depending on the nature of the input data and feedback used for learning the following three main machine learning approaches can be distinguished: supervised learning, unsupervised learning and reinforcement learning
- Machine learning system is an organized system that implements one or more machine learning methods/algorithms
- In CPS and IoT, Machine Learning is used for a wide variety of important tasks, as: video and image processing, computer vision, speech processing and recognition, object motion prediction, robot or vehicle path planning, etc.
- Machine Learning (ML) can be seen as a part of Artificial Intelligence (AI), although some researchers argue that they only have a large common part

52

Edge ML and and Deep Learning (DL)



- ANN is a ML model involving nodes called neurons which are connected with edges
- A neuron processes the received signals, when computing a non-linear function of the sum of its inputs, and then sends a signal to neurons to which its output is connected
- ANN with several hidden layers is called a deep ANN (DNN)
- The spectacular progress in the massively parallel computing platforms in the recent 10 years enabled the implementation of much more complex neural networks and the reincarnation of neural networks and related fields in the form of deep learning

53

AI importance for CPS and IoT

- AI is extremely important for CPS and IoT: it provides the intelligent, automated and timely decision-making based on the big amounts of data generated by numerous CPS and IoT devices
- On June 14, 2023, the European Parliament adopted the Artificial Intelligence Act (AI Act) being the first set of rules to manage AI risks and to promote AI uses in line with the EU values
- The interest in Artificial Intelligence is rapidly increasing
- According to Gartner, the market of chips for AI will increase at an annual rate of more than 20 percent to:
    - USD 53.4 billion in 2023
    - USD 67 billion in 2024, and
    - USD 119.4 billion in 2027

(Gartner, August 2023)

54

Generative AI

- Recently a special attention of the AI system and AI hardware developers has been focussed on the Generative AI (GenAI or GAI)
- While the traditional AI recognizes existing patterns in data and acts upon them using a certain set of rules, the generative AI creates new the most likely to occur patterns of data based on the data on which it was trained
- Generative AI could be used for very many purposes, but on the training side it requires to process a huge amount of data for its Large Language Models (LLMs)
- LLM is a kind of machine learning model originally designed for natural language processing that involves many parameters and is trained with self-supervised learning on a huge amount data (e. g. text).
- Many LLMs can also process/generate some other types of data (e. g. images)
- Currently, the most capable LLMs are generative pretrained transformers (GPTs)
- In general, the advances of AI are driven by computing power
- Generative AI is especially hungry for chips and energy

55

Edge Computing Platforms for ML and AI

- It is still a research question how the Generative AI can effectively and efficiently be implemented at the Edge
- Deloitte predicted that in 2024 the market for GenAI chips will account for more than US$50 billion, what represents 8.5% of the value of all chips expected to be sold in 2024 (Deloitte, “2024 semiconductor industry outlook”)
- In recent years, remarkable advancements were made not only in GenAI LLMs and their applications, but in many other AI and ML fields
- One of this fields of primary importance for CPS and IoT are Edge Computing Platforms for ML and AI
- ML and AI technologies belong to the main contributors to modern CPS and IoT, but they are also expected to substantially contribute to the solution of the environmental crises
- As energy is a scarce resource at the edge, (ultra-)low-power architectures with sufficient processing power are required for the Edge Computing Platforms

56

New Edge Computing Platforms for ML and AI

- In recet years many different new Edge computing platforms and accelerators for Deep Learning, other learning and other AI have been developed to serve different needs

- From relatively small ultra-low power NN Edge processors and SoCs:

  - GreenWaves developed Gap9 ultra-low power neural network Edge processor suitable for battery-powered devices and optimized for advanced audio of which the power consumption can be as low as 1.8 mW
  - Synaptics developed Katana ultra-low power Edge AI SoC for a wide range of energy constrained IoT applications (e.g. sensors and edge devices in offices, factories, warehouses, robotics, farms, smart homes and cities, etc.)

- to complex heterogeneous massively parallel MPSoCs and Systems-on-Module for high-performance and energy-efficient AI, as NVIDIAs Jetson AGX Orin SoM

57

New Edge Computing Platforms for ML and AI

- Mobileye (Intel Corp.) introduced its EyeQ Ultra high-performance and low-power SoC aimed at autonomous vehicles and similar advanced applications
- EyeQ Ultra is fabricated in 5-nm process and delivers AI performance up to 176 TOPS at less than 100 W
- It has a very heterogeneous architecture involving several different types of cores tuned to different tasks involved in an L4 autonomous car, including:
  - 12 RISC-V CPU cores,
  - Arm GPU and VPU,
  - 4 types of Mobileye's proprietary accelerators involving 16 CNN accelerators, 8 CGRA-based cores, 16 VLIW/SIMD cores, and 24 barrel-threaded CPU cores,
  - video encoding/decoding cores, safety/security subsystem, two separate sensor subsystems: one camera-only, and the other one for radar and lidar, etc.
- Each of the two separate sensor subsystems can support a full operation, and this redundancy results in a more robust overall system

58

New Edge Computing Platforms for ML and AI

Source: Mobileye, an Intel Company

Most of the high-end MPSoCs are made with DUV and EUV machines from ASML:
https://www.youtube.com/watch?v=1JkzrR-hznE

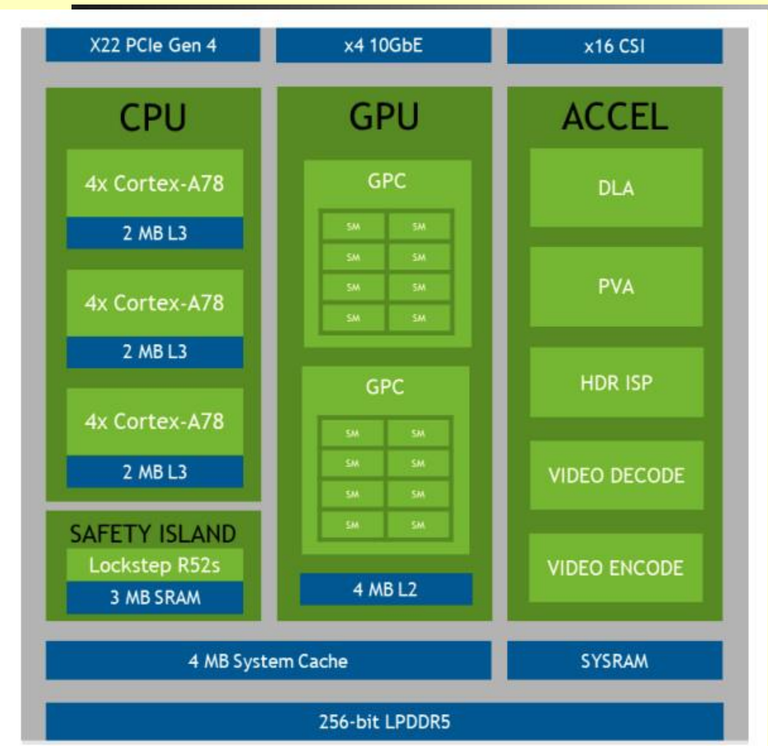
59

New Edge Computing Platforms for ML and AI

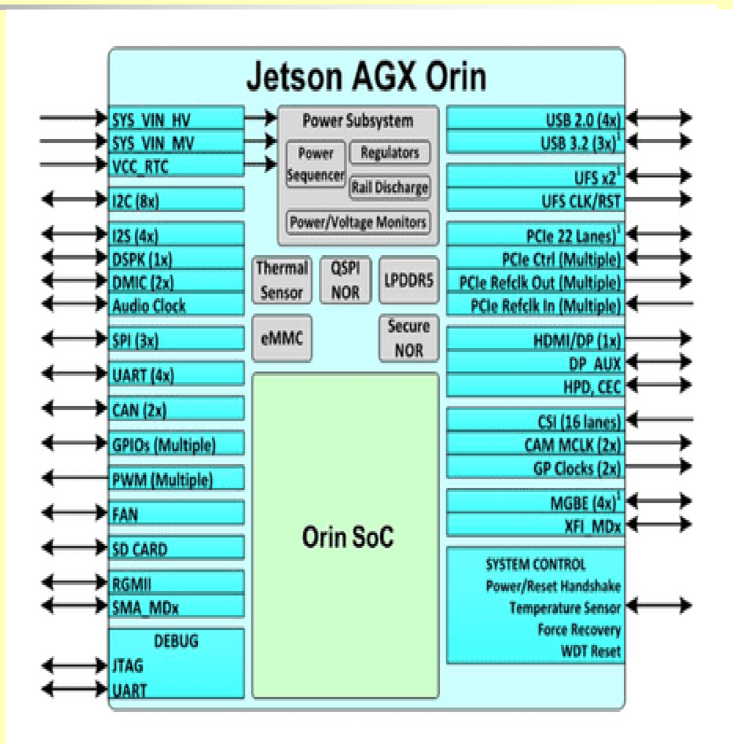
- NVIDIA introduced new Jetson AGX Orin System-on-Module (SoM) for powerful high-performance and energy-efficient AI/ML at the edge
- It is aimed at the most advanced applications requiring powerful embedded computing at the edge in such sectors as advanced medical devices, autonomous cars, autonomous delivery, logistics and factory robots, advanced UAVs, and other advanced autonomous systems, for highly demanding tasks of multi-sensor fusion, computer vision, motion prediction, path planning, natural language understanding, etc.
- Jetson AGX Orin delivers up to 200 TOPS AI performance, which is comparable to the performance of a GPU-based server, but has a size of only 100mm x 87mm and uses much less power (15 – 40 W)
- Jetson AGX Orin SoM is built around Orin SoC with Nvidia's GPU Ampere architecture with 1792 NVIDIA® CUDA® cores and 56 Tensor Cores in two Graphic Processing Clusters (GPCs), 8-core ARM Cortex-A78AE CPU, powerful HW deep learning accelerator (DLA) and vision accelerator (PVA), video encoder and video decoder

60

New Edge Computing Platforms for ML and AI



Orin SoC Block Diagram



Jetson AGX Orin System-on-Module

Source: NVIDIA

61

Performance Improvement of Platforms for AI

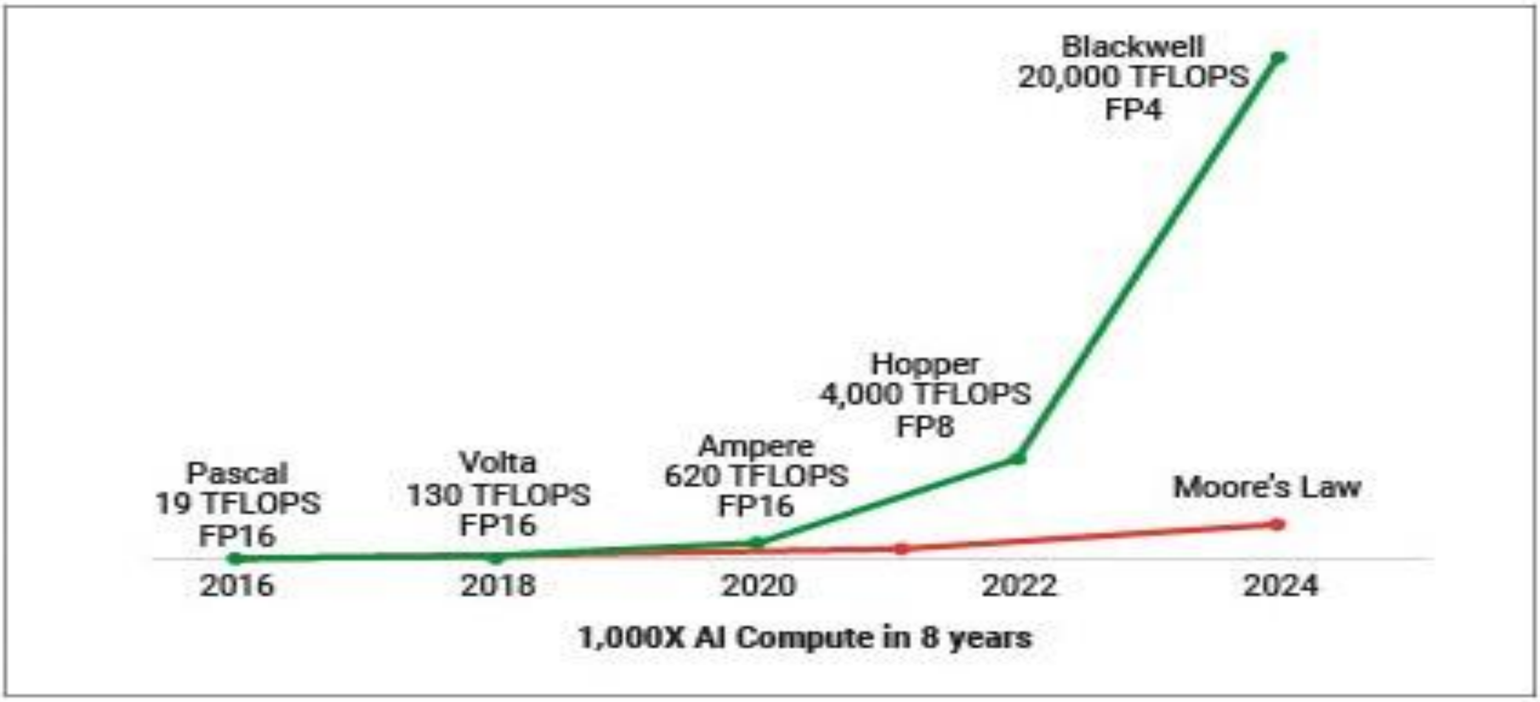


Figure 2: Performance improvement of NVIDIA GPUs on AI workloads (source NVIDIA, J. Huang keynote at Computex 2024)

Performance from Architecture: massive parallelism, heterogeneity, specialization

Most of the high-end MPSoCs are made with ASML machines: https://www.youtube.com/watch?v=1JkzrR-hznE

62

Main IoT Networking Technologies and Standards

- As earlier explained: the IoT for advanced CPS will be substantially different than Internet for other traditional targets
- Specifically, due to different application requirements in relation to connectivity (data rate, latency, etc.), deployment area, number of connected devices, energy consumption, safety, security, reliability, cost, etc. different networking technologies, standards and protocols will be used
- The following two kinds of IoT applications are distinguished in relation to two distinct areas of the requirement spectrum: Massive IoT and Critical IoT
- Massive IoT refers to applications that require a huge number (from thousands to milliards) of low-cost and low-energy devices often in remote locations, each generating a small number of (regularly) reported data, and that have relatively low throughput and latency requirements:
  - Aim: to efficiently transmit small amounts of data from the huge number of devices
  - Key requirements: sufficient network capacity, scalability, security and availability, wide and strong coverage, (ultra) low-power/energy, low cost
  - Example Applications: smart metering, smart building/city, smart grid, asset tracking, fleet management, wearables and part of e-health, process monitoring and optimization in indystry, environmental monitoring, climate monitoring", livestock tracking in agriculture, etc.

63

Main IoT Networking Technologies and Standards

- Critical IoT refers to time- and safety-critical applications that demand data delivery within a specified time and with required guarantees, and that usually involve fewer (up to thousands) complex costly devices, each generating/receiving large amount of data with high throughput and low latency requirements, and that have to withstand harsh/remote environments, as well as security threats and attacks:
  - Aim: to guarantee efficient transmission of large amount of data with high throughput and low latency in harsh environment and while facing security threats and attacks
  - Key Requirements: guaranteed high-bandwidth, low-latency, and very high security, safety, reliability, and availability, at low energy and acceptable cost
  - Example Applications: Autonomous Vehicles and V2X, UAVs, Robotics, Industry 4.0, telemedicine, VR/AR/MR applications, traffic and flight control and safety, critical part of smart city, etc.
- For massive IoT applications requiring:
  - low-power, wide area connectivity, security and availability, cellular network standards LTE-M and NB-IoT can be used
  - very low power from the device to send/receive data, very many connected devices/large area and lower cost, some LPWANs, as LoRa or Sigfox, can be used

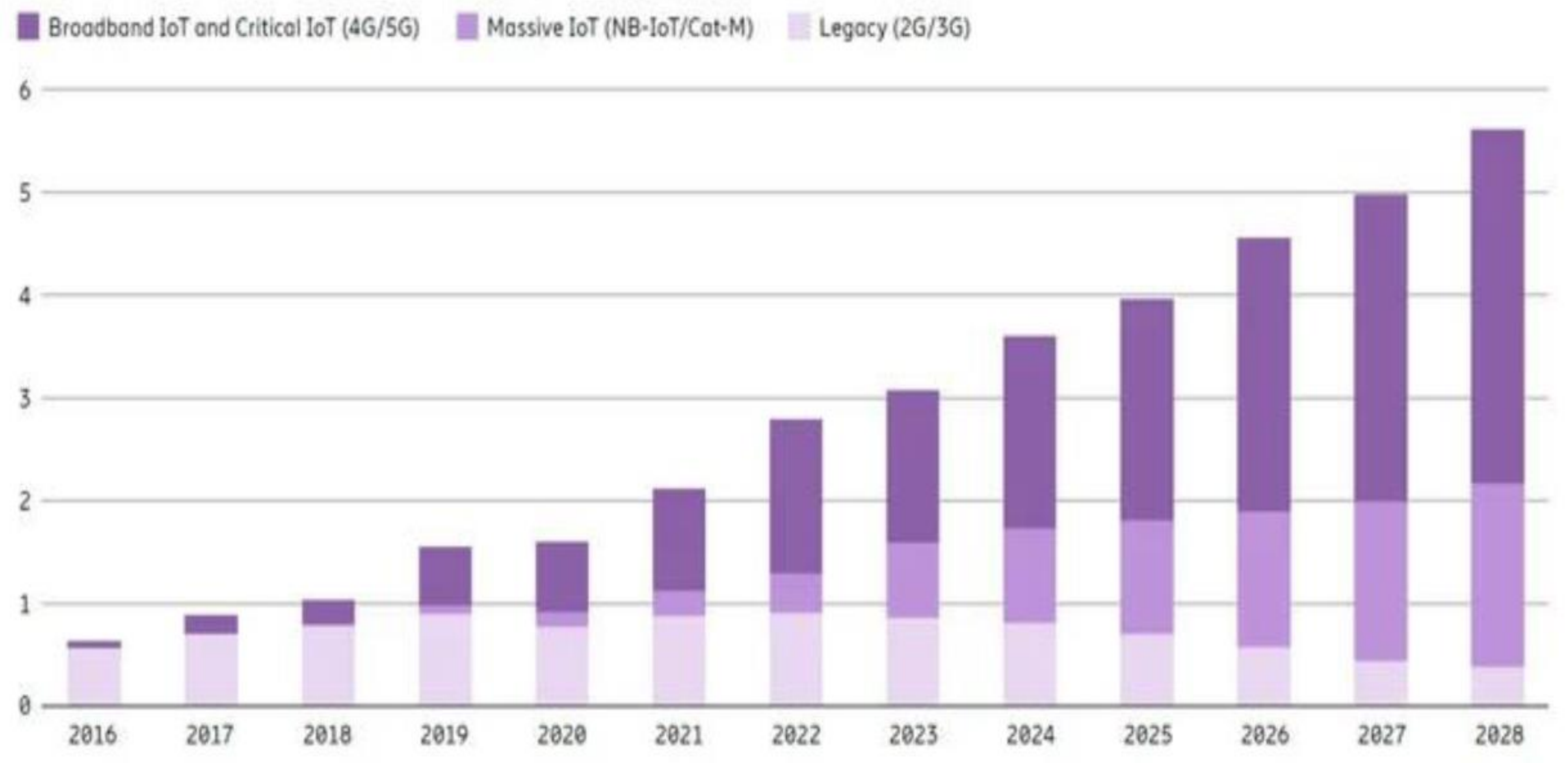
64

Main IoT Networking Technologies and Standards

- For home appliances and similar consumer devices and applications WiFi, Bluetooth, Thread or Zigbee can be a satisfactory and low-cost solutions, and the recently introduced Matter uses a combination of WiFi, Bluetooth Low Energy and Thread to enable devices and applications interoperability
- From the above it is clear that 5G is not always required and not always the best option for IoT
- However, 5G is indispensable for Critical IoT, as it provides Network Slicing, and much higher bandwidth, lower latency, lower power consumption, and higher safety, security and reliability than 4G
- Using Network Slicing the service provider can devote a part of the 5G radio spectrum to run a separate private wireless network for a company, or an NB-IoT massive service connecting thousands of sensors, or to enable higher bandwidth and lower latency for some highly demanding applications as autonomous vehicles or UAVs
- Allied Market Research reported that the global market of 5G infrastructure industry was $2.06 billion in 2020, and the market will grow to $83.62 billion by 2030, at a CAGR of 45.3 percent between 2021 and 2030

65

Number of IoT connections (in billion)

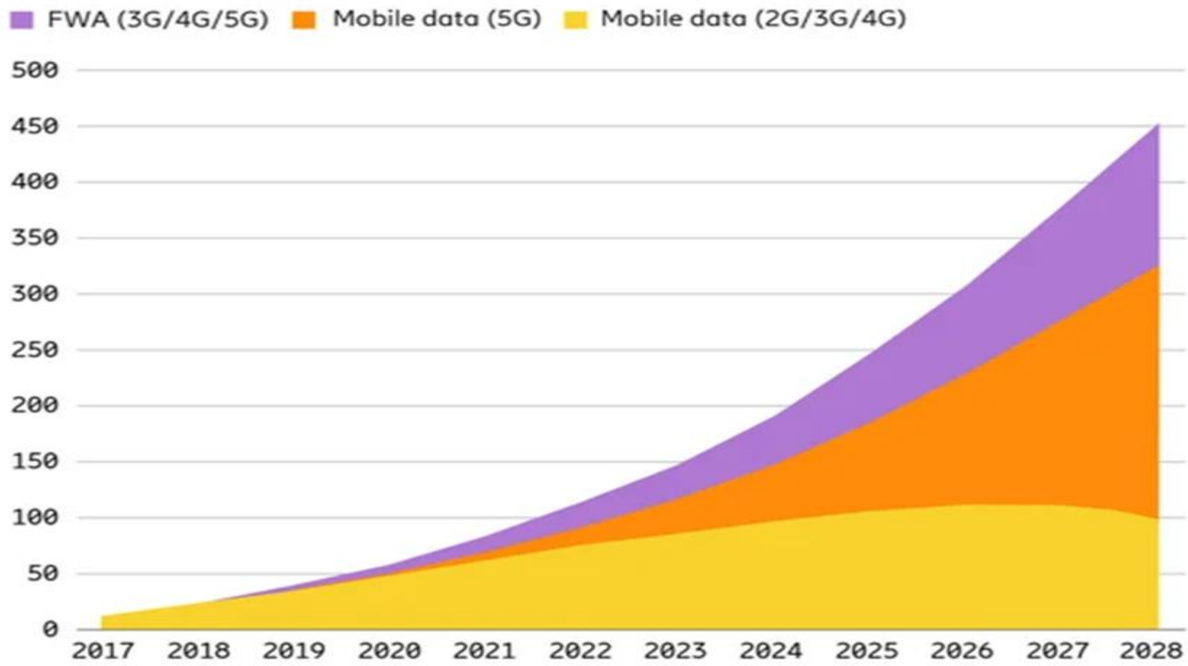


Source: Ericsson Mobility Report, November 2022

- Broadband and Massive IoT will co-exist
- Broadband IoT (4G/5G) connections (including critical) will dominate

66

Mobile network data traffic (in EB per month)



Source: Ericsson Mobility Report, November 2022

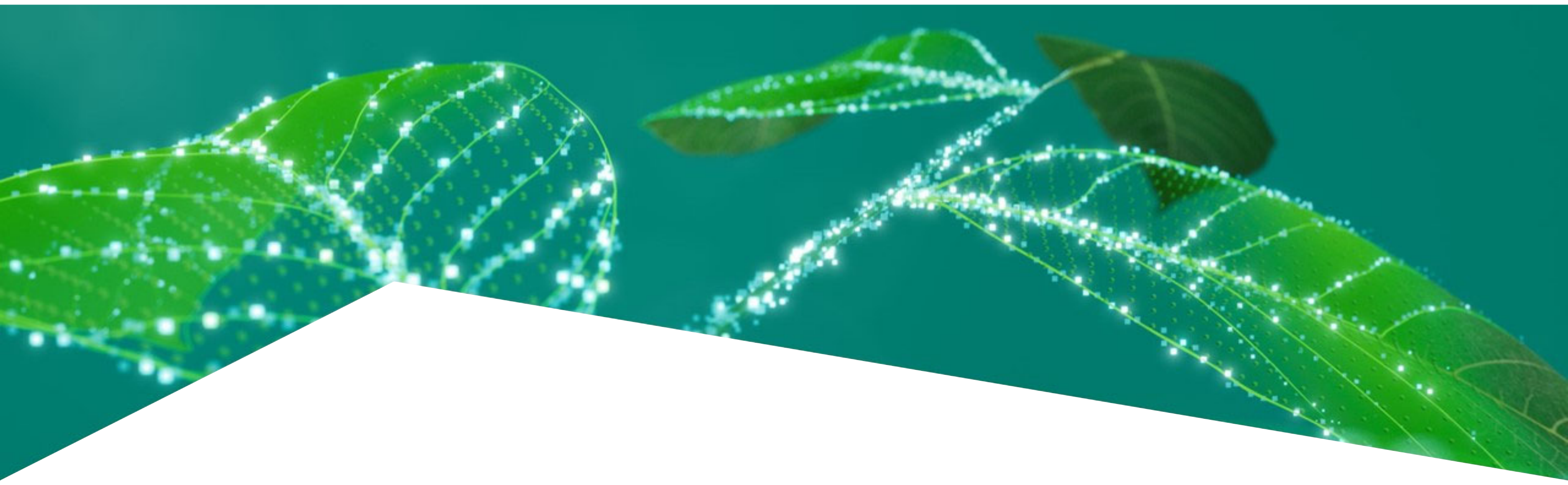
- 5G to drive the mobile data growth and 4G/3G/2G data traffic will decline by 2028
- Fixed Wireless Access (FWA) will increase
- Ericsson forecasts that between the end of 2023 and 2029 the global 5G subscriptions will grow by more than 330 percent: from 1.6 billion to 5.3 billion (Ericsson Mobility Report, November 2023)

67

Conclusion

- Systemic drawbacks of the traditional economy and cumulation of bad decisions made by numerous governments and companies without accounting for long-term consequences resulted in the **huge global environmental disaster**
- To recover from the environmental disaster and further develop:
  - *a model of a well regulated and controlled effective and efficient system should be applied to all kinds of systems, collaboration chains and related flows*
  - *modern CPS and IoT technologies should be used to much better control and optimize the social, physical and life systems than till now*
  - *methodologies of circular regenerative economy and quality-driven design should be used to design the systems*
- Innovations exploiting modern CPS and IoT technologies, circular regenerative economy and quality-driven design can significantly improve systems used by us or that we are part of
- In this CPS&IoT Summer School you will have a unique occasion to be informed on and to discuss the most recent European R&D developments in CPS and IoT

68



RISC-V vs. ARM

Natalie Simson and Wolfgang Ecker
10th June 2025

Presentation of the Presenters

Natalie Simson

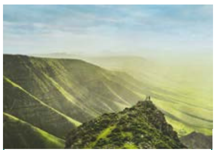
- PhD Sponsorship of Infineon and PhD Candidate at TU-Munich
- ~30 weeks work experience
- Working 5 years with RISCV
- Developed and implemented various RISCV variants, worked on Arm vs. RISC-V comparison, benchmarked RISC-V variants
- Enhances RISCV and related design automation in parallel

Wolfgang Ecker

- Distinguished Engineer at Infineon and Adjunct Professor at TU-Munich
- ~30 Years work experience
- Working ~12 years with RISCV
- Godfather for RISCV development at Infineon
- Working ~16 years on automating digital design with focus on design generation
- Knows what Natalie knows – and that's a lot

Presentation of Infineon – Partial Sponsor of the Tutorial

Growth areas



Energy
green and efficient

Mobility
clean and safe

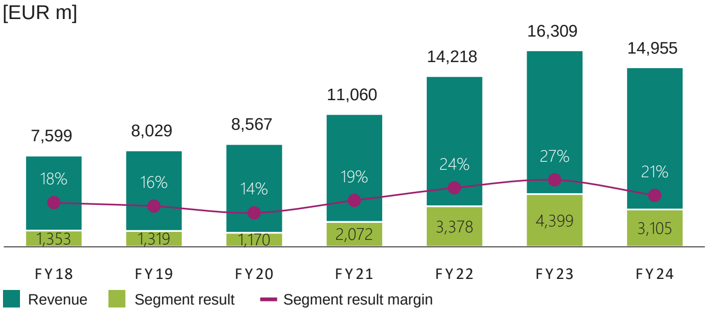
IoT
smart and secure

FY24 revenue by segment[1]

- Automotive (ATV)
- Green Industrial Power (GIP)
- Power & Sensor Systems (PSS)
- Connected Secure Systems (CSS)

Financials

[EUR m]

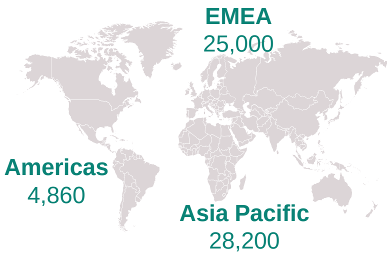


Employees[1]

58,060
employees worldwide

71
R&D and

15
manufacturing locations[2]



For further information: Infineon Annual Report.
1 2024 Fiscal year (as of 30 September 2024) | **2** As of 30 September 2024

Infineon is a global leader in power systems and IoT

Global leader

in automotive, power management,
energy efficient technologies and IoT

~58,060

employees[1]

Market position

Automotive	Power	Microcontroller
#1	#1	#1
TechInsights, March 2025	Omdia, April 2025	Omdia, March 2025

[1] As of 30 September 2024

2025-06-10 Summer School CPSIoT 2025: RISC-V vs. ARM 4

Driving decarbonization and digitalization. Together.

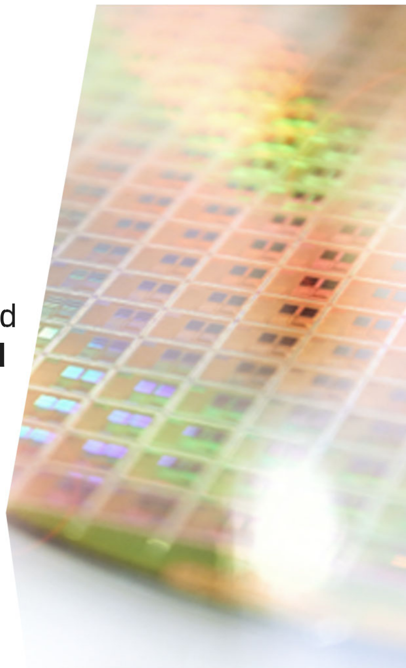
Semiconductors are crucial to solve the **energy challenges** of our time and **shape the digital transformation.**

This is why Infineon is committed to actively driving decarbonization and digitalization.

As a global semiconductor leader in power systems and IoT, we enable game-changing solutions for **green and efficient energy**, **clean and safe mobility**, as well as **smart and secure IoT**.

We make life easier, safer, and greener.
Together with our customers and partners.

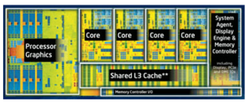
For a better tomorrow.

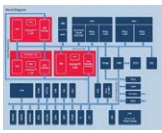


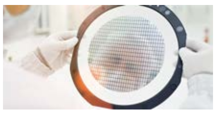
It's the value

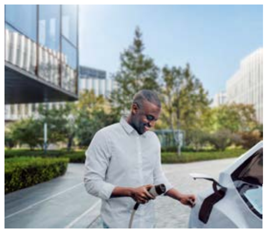
IP Module	Architecture	Chip	Board	Product	Society

RISC-V









Outline

- **Part 1: RISCV vs. ARM (15.30-17.00)**
  15:30–16:10 General Introduction: History of ARM M* core and RISC-V, architectures and instruction set architectures
  16:10–16:20 *Short Break*
  16:20–17:00 Comparing ARM and RISC-V: Objects, Instruction Overview and Integer Instructions (RV32I)

- **Break (17:00-17.30)**

- **Part 2: RISCV vs. ARM (17.30-19.00)**
  17:30–18:10 Comparing ARM and RISC-V: Extensions M, Zcmp, Zicsr and B
  18:10–18:20 *Short Break*
  18:20–19:00 Miscellaneous: Additional Extensions, Memory Footprint and Performance

Outline
General Introduction

- General Information on ARM and RISC-V
- ARM and RISC-V ISA
- Processor Micro-Architecture
- Which ARM and RISC-V ISA do we consider

Outline
General Introduction

- **General Information on ARM and RISC-V**
- ARM and RISC-V ISA
- Processor Micro-Architecture
- Which ARM and RISC-V ISA do we consider

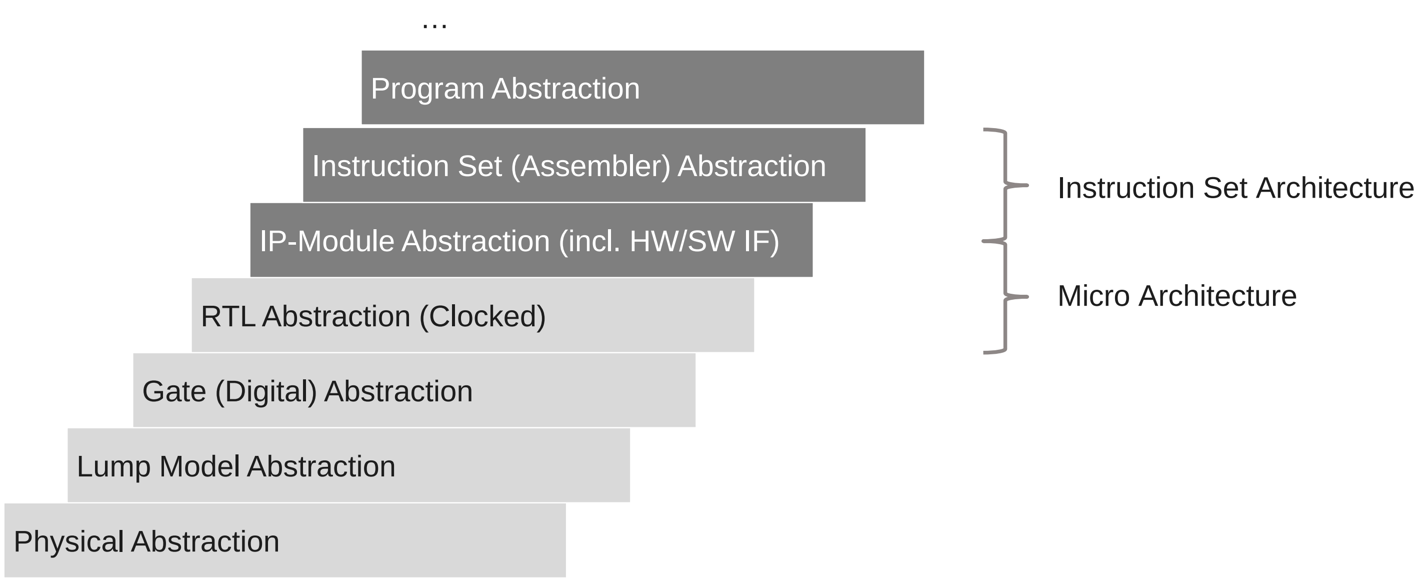
A view on ARM and RISCV from the space

	ARM	RISCV
Initiator	Acorn (1983) ARM (1990)	UC Berkeley (1980-1984 RISC Project) UC Berkeley (2010)
Product	Soft Processor Cores *Focus Arm Cortex-M*	Initially Education Purposes Instruction Set Architecture (ISA)
ISA	Owned by ARM Somehow published	high-quality, license-free, royalty-free, available freely, permissive license
ISA Variants	Stacked ISA Definitions *Focus ARMv6 (2002)*	Minimum Base Instructions, Instruction Groups, User Defined Instructions
ISA Concept	*CISCY RISC*	RISCV with CISCY Extensions
Soft Cores	Closed Source, ARM	Open/Closed Source Products, Various providers
Owner	ARM ~6000 Employees	RISC-V Foundation (2015) More than ~350 Members

Outline
General Introduction

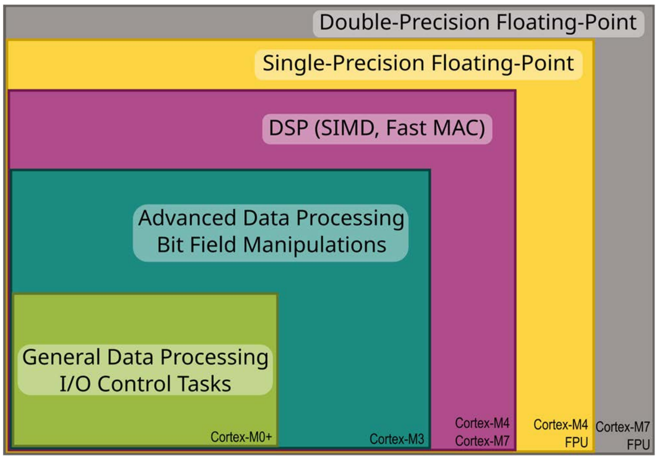
- General Information on ARM and RISC-V
- **ARM and RISC-V ISA**
- Processor Micro-Architecture
- Which ARM and RISC-V ISA do we consider

ISAs in the Abstraction Stack



Different approaches in ARM and RISCV to structure ISA

Stacked approach



Sources:

- K. A. Andrew Waterman, The RISC-V Instruction Set Manual Volume I: Unprivileged ISA, CS Division, EECS Department, University of California, Berkeley.
- ARM, "White paper: Cortex-m for beginners - an overview of the arm cortex-m processor family and comparison," https://community.arm.com/cfs-file/__key/telligent-evolution-components-attachments/01-2142-00-00-00-00-52-96/White-Paper-_2D00_-Cortex_2D00_M-for-Beginners-_2D00_-2016-_2800_final-v3_2900_.pdf, accessed: 28-05-2025.

Base instruction and instruction modules

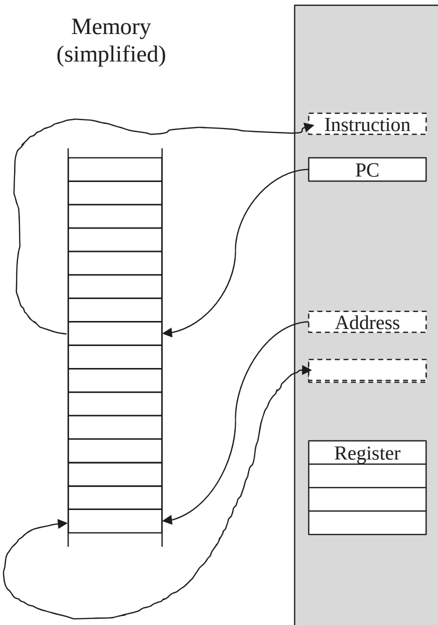
RV32/63I Base Integer
RV32/64E Embedded
RV128I Base Integer
Zifencei Instruction-Fetch Fence
Zicsr Control/Status Register r/w
Zicntr Base Counters/Timers r/w
Zihintntl Non-Temporal Locality
Zihintpause Pause Hint
***Zimop** May-Be-Operations*
Zicond Integer Conditional Operations
M Integer Multiplication/Division
A Atomic Instructions, r/modify/w
Zawrs Wait-on-Reservation-Set
Zacas Atomic Compare-and-Swap (CAS)
RVWMO Memory Consistency Model
Ztso Total Store Ordering

CMO Base Cache Management
F Single-Precision Floating-Point
D Double-Precision Floating-Point
Q Quad-Precision Floating-Point
Zfh Half-Precision Floating-Point
ZfhMin Half-Precision FP
Zfa Additional Floating-Point
Zfinx, Zdinx, Zhinx, Zhinxmin Floating-Point in Integer Registers,
C Compressed Instructions
Zc* Code Size Reduction
B Bit Manipulation
***J** Dynamically Translated Lang.*
***P** Packed-SIMD Instructions*
V Vector Operations
… Cryptography (Vec.)Extensions
X Instruction Extensions

ARM and RISC-V ISA

- ISAs in the abstraction
- Instructions
- System Space

A Generic Execution Model for Instruction Specification



Fetch
- Initiate memory-read by applying address
- Take data from memory as instruction

Decode
- Determine execution action to be performed
- Extract (immediate) values from instruction if any
- Extract register addresses from instruction, if any
- Assemble data to be processed

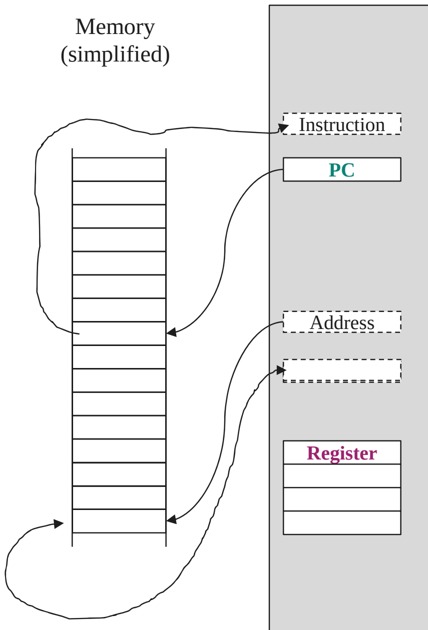
Execute
- Perform compute according to determined action
  (e.g. compute address or data, evaluate condition)
- Handle compute result (kind of instruction dependent)
  - *Register Operation*
    - Write data to register
    - Increment PC by instruction size in bytes
  - *Memory Operation*
    - Write Data to memory – or -
    - Read Data from memory and write to register
    - Increment PC by instruction size in bytes
  - *Jump Operation*
    - Increment PC by instruction size in bytes if condition is false
    - Assign computed address to PC if condition is true

System Space

Part of System Space
- PC
- Register File
- CSRs (not shown)

Not Part of System Space
- Instruction and Address Register

Configuration of System Space
- Data Width
- DataUnit
- AddrUnit
- Endianness



Fetch
- Initiate memory-read by applying address
- Take data from memory as instruction

Decode
- Determine execution action to be performed
- Extract (immediate) values from instruction if any
- Extract register addresses from instruction, if any
- Assemble data to be processed

Execute
- Perform compute according to determined action
  (e.g. compute address or data, evaluate condition)
- Handle compute result (kind of instruction dependent)
  - *Register Operation*
    - Write data to register
    - Increment PC by instruction size in bytes
  - *Memory Operation*
    - Write Data to memory – or -
    - Read Data from memory and write to register
    - Increment PC by instruction size in bytes
  - *Jump Operation*
    - Increment PC by instruction size in bytes if condition is false
    - Assign computed address to PC if condition is true

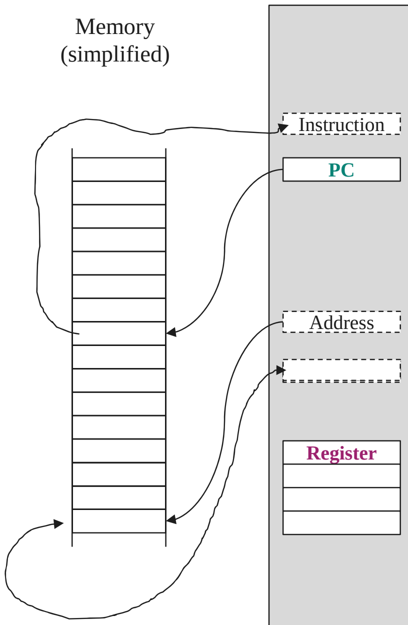
Instruction Encoding

Describes how bits and groups of bits are arranged.

Bits and bit-groups are interpreted as e.g.

- Register Address
- Immediate Value
- Encoding of Execution Action

Relates to Decode Stage that extracts bits and bit groups



Fetch
- Initiate memory-read by applying address
- Take data from memory as instruction

Decode
- Determine execution action to be performed
- Extract (immediate) values from instruction if any
- Extract register addresses from instruction, if any
- Assemble data to be processed

Execute
- Perform compute according to determined action (e.g. compute address or data, evaluate condition)
- Handle compute result (kind of instruction dependent)
  - *Register Operation*
    - Write data to register
    - Increment PC by instruction size in bytes
  - *Memory Operation*
    - Write Data to memory – or -
    - Read Data from memory and write to register
    - Increment PC by instruction size in bytes
  - *Jump Operation*
    - Increment PC by instruction size in bytes if condition is false
    - Assign computed address to PC if condition is true

Instruction (Timeless) Behavior

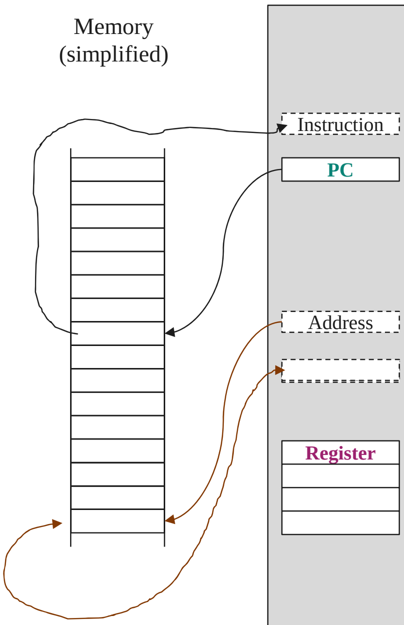
Mainly relates to the Execute Phase of a processor

Is described how System States and Ports are impacted by an Instruction

- PC
- Register File
- CSRs (not shown)
- Address- and Data-Out as well as Data-In

Some behavior is implicit

- Program Counter Update to the next instruction



Fetch
- Initiate memory-read by applying address
- Take data from memory as instruction

Decode
- Determine execution action to be performed
- Extract (immediate) values from instruction if any
- Extract register addresses from instruction, if any
- Assemble data to be processed

Execute
- Perform compute according to determined action
  (e.g. compute address or data, evaluate condition)
- Handle compute result (kind of instruction dependent)
  - *Register Operation*
    - Write data to register
    - Increment PC by instruction size in bytes
  - *Memory Operation*
    - Write Data to memory – or -
    - Read Data from memory and write to register
    - Increment PC by instruction size in bytes
  - *Jump Operation*
    - Increment PC by instruction size in bytes if condition is false
    - Assign computed address to PC if condition is true

Instruction Description

- Instruction Encoding

31–25	24–20	19–15	14–12	11–7	6–0
imm[11:5]	imm[4:0]	rs1	funct3	rd	opcode
7	5	5	3	5	7
0 0 0 0 0 0 0	shamt[4:0]	src	SLLI	dest	OP-IMM
0 0 0 0 0 0 0	shamt[4:0]	src	SRLI	dest	OP-IMM
0 1 0 0 0 0 0	shamt[4:0]	src	SRAI	dest	OP-IMM

- Textual (Shift Immediate, From RISC-V Specification)
  – Shifts by a constant are encoded as a specialization of the I-type format. The operand to be shifted is in rs1, and the shift amount is encoded in the lower 5 bits of the I-immediate field. The right shift type is encoded in bit 30. SLLI is a logical left shift (zeros are shifted into the lower bits); SRLI is a logical right shift (zeros are shifted into the upper bits); and SRAI is an arithmetic right shift (the original sign bit is copied into the vacated upper bits).

- Behavioral

```
R[dest] = R[src] << shamt[4:0]
PC = PC+4
```

- Mnemonic

```
SLLI dest, src, imm
```

A firs look on RISCV instructions

Instruction (Register)	Instruction (Immediate)	Compressed (Register)	Compressed (Imm)	Class	Comment
LB, LBU, LH, LHU, LW		C.LW, C.LWSP		**Loads**	LWSP relies on the use of a specific register as stack pointer SP
SB, SH, SW		C.SW, C.SWSP		**Stores**	Requires byte/half-word/word memory write access
ADD, SUB	ADDI	C.ADD, C.SUB	C.ADDI	**Arithmetic**	SUBI does not exist, ADDI with negative immediate is used instead
	LUI, AUIPC	C.MV	C.ADDI16SP, C.LUI C.ADDI4SPN, C.LI,	**Special Arithmetic / Load/ Move**	LUI is used with ADDI to load a 32-bit constant, AUIPC is used to do PC relative addressing
XOR, OR, AND	XORI, ORI, ANDI		C.ANDI	**Logical**	
SLL, SRL, SRA	SLLI, SRLI, SRAI		C.SLLI, C.SRAI, C.SRLI	**Shift**	
SLT, SLTU	SLTI, SLTIU			**Compare**	
BEQ, BNE, BLT, BGE, BLTU, BGEU		C.BEQZ, C.BNEZ		**Branch**	
JAL, JALR		C.J, C.JR, C.JAL, C.JALR		**Jump (and Link)**	
FENCE, FENCE.I				**Synchronization**	
ECALL, EBREAK		C.EBREAK		**Environment**	
CSRRW, CSRRS, CSRRC	*CSRRWI, CSRRSI, CSRRCI*			***CSR***	

20

Outline
General Introduction

- General Information on ARM and RISC-V
- ARM and RISC-V ISA
- **Processor Micro-Architecture**
- Which ARM and RISC-V ISA do we consider

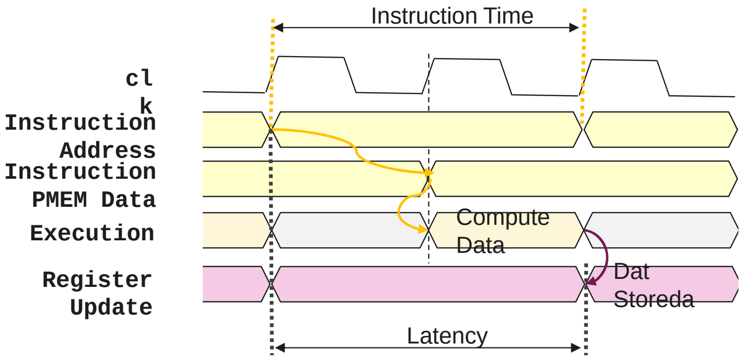
Latency and Throughput

Latency – The time (in clock cycles) taken from starting an instruction until it is completely finished

Instruction Throughput – The quantity of instructions being processed within a unit of time

Instruction Time – The taking from starting one instruction to starting the next instruction (compare: sampling time)

Latency and Throughput (Adopted)



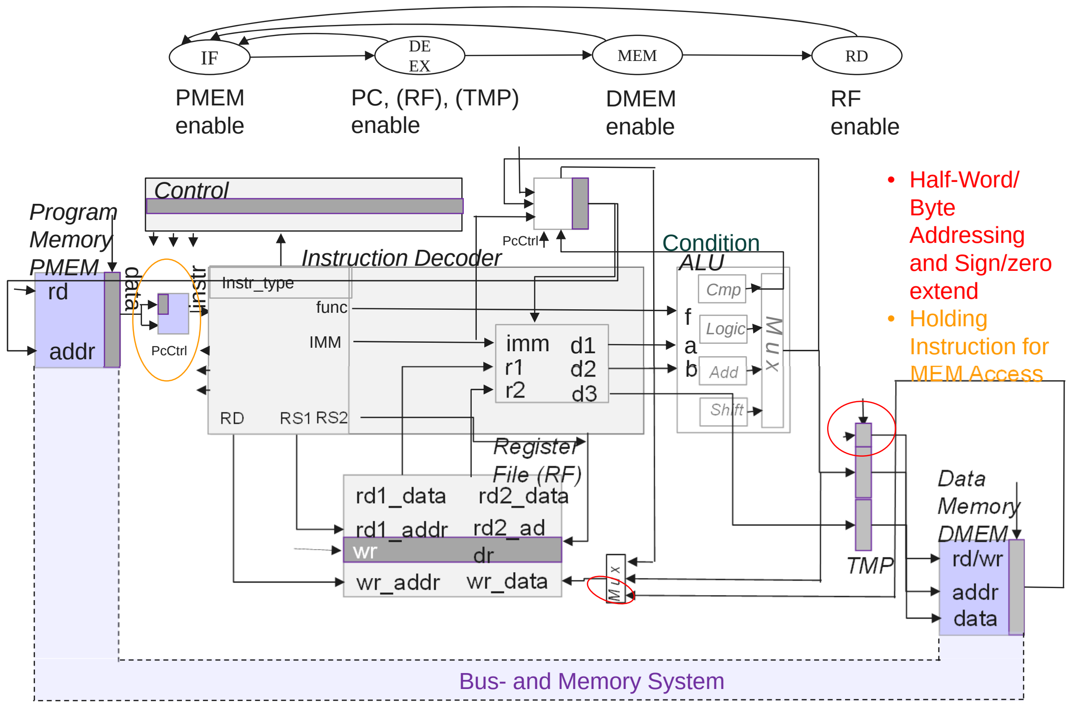
Latency and **Instruction Time** are the same for this architecture

Latency and **Instruction Time** is instruction dependent (memory write 3 cycles, memory read 4 cycles)

A program's **Latency** and **Instruction Time** depends on the average number each instruction is executed

23

Controller based CPU



- Half-Word/ Byte Addressing and Sign/zero extend
- Holding Instruction for MEM Access

24

Usage and Utilization of FSM CPU

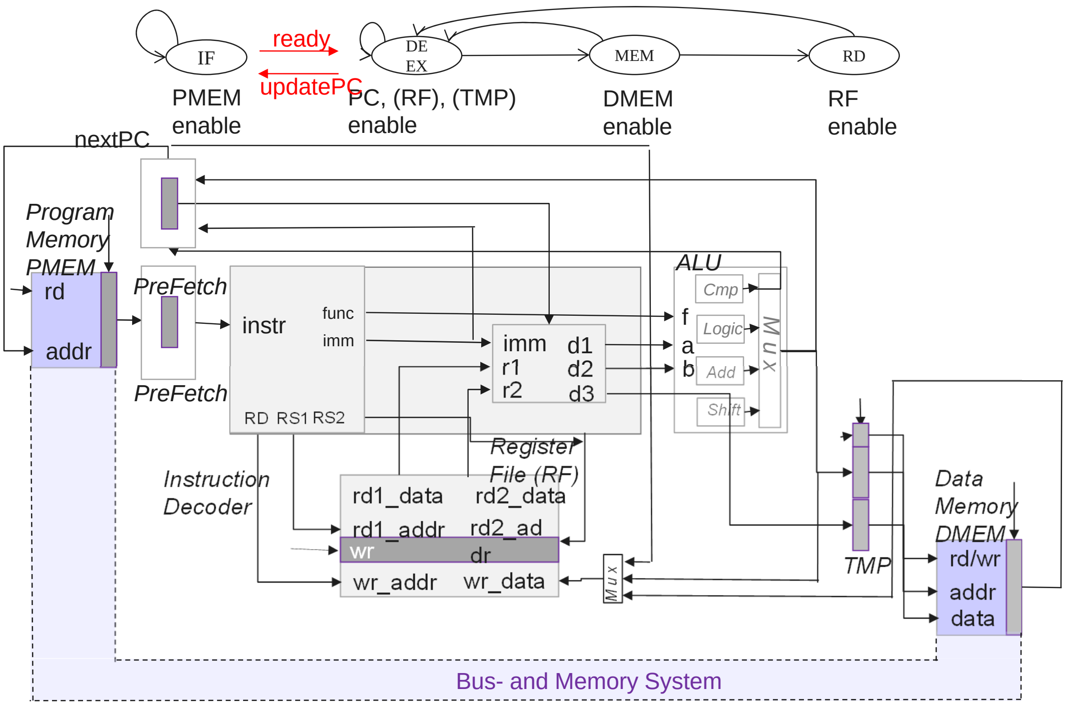
Usage – The number of times a specific resource is used for a task

Utilization – The fraction of time, a specific resource is used

Instruction (Register)	Instruction (Immediate)	Class	Latency	Instr. Time	Usage (upper number) and *Utilization (lower number)*						
					IMEM	DMEM	MEM	ADD	SHIFT	Logic	Cmp
LB, LBU, LH, LHU, LW		**Load**	4	4	1 *0,25*	1 *0,25*	2 *0,5*	1 *0,25*	0 *0,0*	0 *0,0*	0 *0,0*
SB, SH, SW		**Store**	3	3	1 *0,33*	1 *0,33*	2 *0,66*	1 *0,33*	0 *0,0*	0 *0,0*	0 *0,0*
ADD, SUB	ADDI	**Arithmetic**	2	2	1 *0,5*	0 *0,0*	1 *0,5*	1	0 *0,0*	0 *0,0*	0 *0,0*
	LUI, AUIPC	**Special Arithmetic / Load/ Move**	2	2	1 *0,5*	0 *0,0*	1 *0,5*	0/1 *0,0/0,5*	0 *0,0*	0 *0,0*	0 *0,0*
XOR, OR, AND	XORI, ORI, ANDI	**Logical**	2	2	1 *0,5*	0 *0,0*	1 *0,5*	0 *0,0*	0 *0,0*	1	0 *0,0*
SLL, SRL, SRA	SLLI, SRLI, SRAI	**Shift**	2	2	1 *0,5*	0 *0,0*	1 *0,5*	0 *0,0*	1 *0,5*	0 *0,0*	0 *0,0*
SLT, SLTU	SLTI, SLTIU	**Compare**	2	2	1 *0,5*	0 *0,0*	1 *0,5*	0 *0,0*	0 *0,0*	0 *0,0*	1 *0,5*
BEQ, BNE, BLT, BGE, BLTU, BGEU		**Branch**	2	2	1 *0,5*	0 *0,0*	1 *0,5*	0 *0,0*	0 *0,0*	0 *0,0*	1 *0,5*
JAL, JALR		**Jump (and Link)**	2	2	1 *0,5*	0 *0,0*	1 *0,5*	1 *0,5*	0 *0,0*	0 *0,0*	0 *0,0*

25

Controller based CPU with Prefetch



Usage and Utilization of FSM CPU with Prefetch

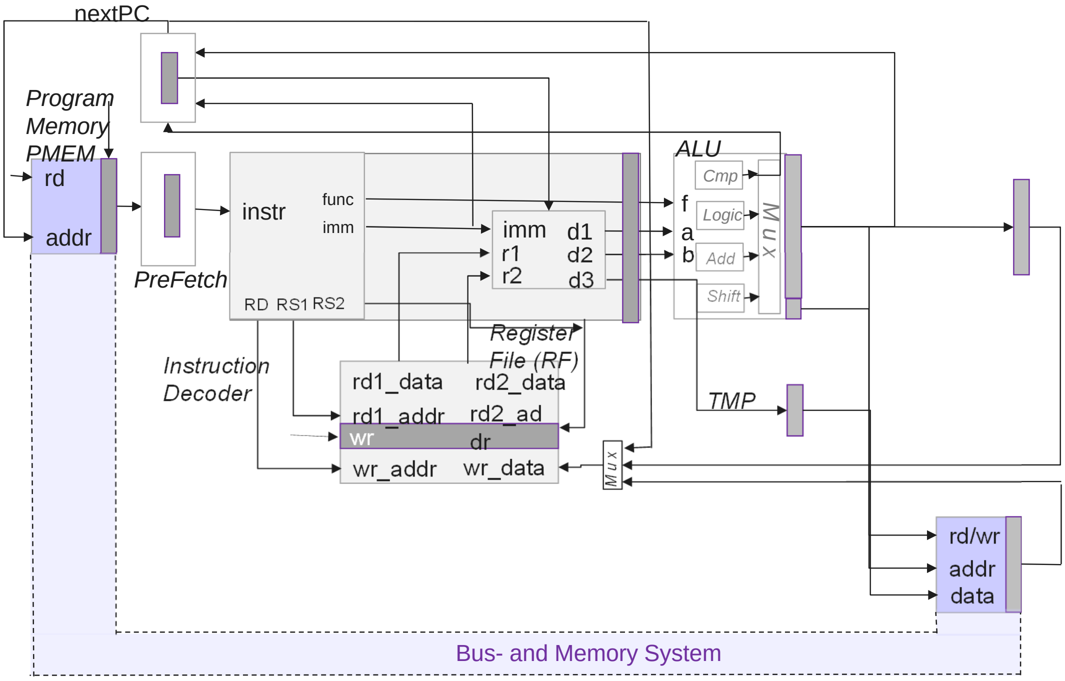
Usage – The number of times a specific resource is used for a task

Utilization – The fraction of time, a specific resource is used

Instruction (Register)	Instruction (Immediate)	Class	Latency	Instr. Time	Usage (upper number) and *Utilization (lower number)*						
					IMEM	DMEM	MEM	ADD	SHIFT	Logic	Cmp
LB, LBU, LH, LHU, LW		**Load**	3	3	1 *0,33*	1 *0,33*	2 *0,66*	1 *0,33*	0 *0,0*	0 *0,0*	0 *0,0*
SB, SH, SW		**Store**	2	2	1 *0,5*	1 *0,33*	2 *0,83*	1 *0,5*	0 *0,0*	0 *0,0*	0 *0,0*
ADD, SUB	ADDI	**Arithmetic**	1	1	1 *1,0*	0 *0,0*	1 *1,00*	1 1,00	0 *0,0*	0 *0,0*	0 *0,0*
	LUI, AUIPC	**Special Arithmetic / Load/ Move**	1	1	1 *1,0*	0 *0,0*	1 *1,0*	0/1 *0,0/1,0*	0 *0,0*	0 *0,0*	0 *0,0*
XOR, OR, AND	XORI, ORI, ANDI	**Logical**	1	1	1 *1,0*	0 *0,0*	1 *1,0*	0 *0,0*	0 *0,0*	1 *1,0*	0 *0,0*
SLL, SRL, SRA	SLLI, SRLI, SRAI	**Shift**	1	1	1 *1,0*	0 *0,0*	1 *1,0*	0 *0,0*	1 *1,0*	0 *0,0*	0 *0,0*
SLT, SLTU	SLTI, SLTIU	**Compare**	1	1	1 *1,0*	0 *0,0*	1 *1,0*	0 *0,0*	0 *0,0*	0 *0,0*	1 *1,0*
BEQ, BNE, BLT, BGE, BLTU, BGEU		**Branch**	1	1	1 *1,0*	0 *0,0*	1 *1,0*	0 *0,0*	0 *0,0*	0 *0,0*	1 *1,0*
JAL, JALR		**Jump (and Link)**	2	2	1 *1,0*	0 *0,0*	1 *1,0*	1 *1,0*	0 *0,0*	0 *0,0*	0 *0,0*

27

Pipelined CPU



28

Latency and Throughput of 5 Stage Pipelined CPU

Latency (in clock cycles) remains instruction dependent but increases to 5+ cycles

- However combinatorial paths become shorter which allows a higher clock frequency

Instruction Time remains instruction dependent

- Additionally depends on program execution (branch taken or not) and memory utilization.

Instruction Time remains 1+ in terms of clock cycles

- However an overall higher clock frequency reduces Instruction Time in terms of time units (e.g. us)

There is a forward **stalling** and backwards stalling delaying execution

- **Forwarding** register values can avoid stalling due to data dependencies in many cases
- **Branch prediction** can avoid wrong prefetch

- All in all, pipeline stages increase execution in terms of clock cycles but allow to clock the processor at a higher rate

29

Usage and Utilization of FSM CPU with Prefetch

Utilization can only be increased when more functions are required from the components in the execution stage in one clock cycles

Instruction (Register)	Instruction (Immediate)	Class	Latency	Instr. Time	Usage (upper number) and *Utilization (lower number)*						
					IMEM	DMEM	MEM	ADD	SHIFT	Logic	Cmp
LB, LBU, LH, LHU, LW		**Load**	3	3	1 *0,33*	1 *0,33*	2 *0,66*	1 *0,33*	0 *0,0*	0 *0,0*	0 *0,0*
SB, SH, SW		**Store**	2	2	1 *0,5*	1 *0,33*	2 *0,83*	1 *0,5*	0 *0,0*	0 *0,0*	0 *0,0*
ADD, SUB	ADDI	**Arithmetic**	1	1	1 *1,0*	0 *0,0*	1 *1,00*	1 1,00	0 *0,0*	0 *0,0*	0 *0,0*
	LUI, AUIPC	**Special Arithmetic / Load/ Move**	1	1	1 *1,0*	0 *0,0*	1 *1,0*	0/1 *0,0/1,0*	0 *0,0*	0 *0,0*	0 *0,0*
XOR, OR, AND	XORI, ORI, ANDI	**Logical**	1	1	1 *1,0*	0 *0,0*	1 *1,0*	0 *0,0*	0 *0,0*	1 *1,0*	0 *0,0*
SLL, SRL, SRA	SLLI, SRLI, SRAI	**Shift**	1	1	1 *1,0*	0 *0,0*	1 *1,0*	0 *0,0*	1 *1,0*	0 *0,0*	0 *0,0*
SLT, SLTU	SLTI, SLTIU	**Compare**	1	1	1 *1,0*	0 *0,0*	1 *1,0*	0 *0,0*	0 *0,0*	0 *0,0*	1 *1,0*
BEQ, BNE, BLT, BGE, BLTU, BGEU		**Branch**	1	1	1 *1,0*	0 *0,0*	1 *1,0*	0 *0,0*	0 *0,0*	0 *0,0*	1 *1,0*
JAL, JALR		**Jump (and Link)**	2	2	1 *1,0*	0 *0,0*	1 *1,0*	1 *1,0*	0 *0,0*	0 *0,0*	0 *0,0*

30

Super-Scalarity can increase Utilization

- ***Utilization*** can only be increased when more functions are required from the components in the execution stage in one clock cycles
- Ways to increase utilization: Use functional blocks for more operations (e.g. adder for compare) or even make a compact ALU but a shift, logical, compare and addition unit
- Introduce complex instructions with memory access and pre- or post increment or mixed shift/logical operations
- Feed more instructions in one clock cycle to the CPU, e.g. by having a 32 bit wide memory interface

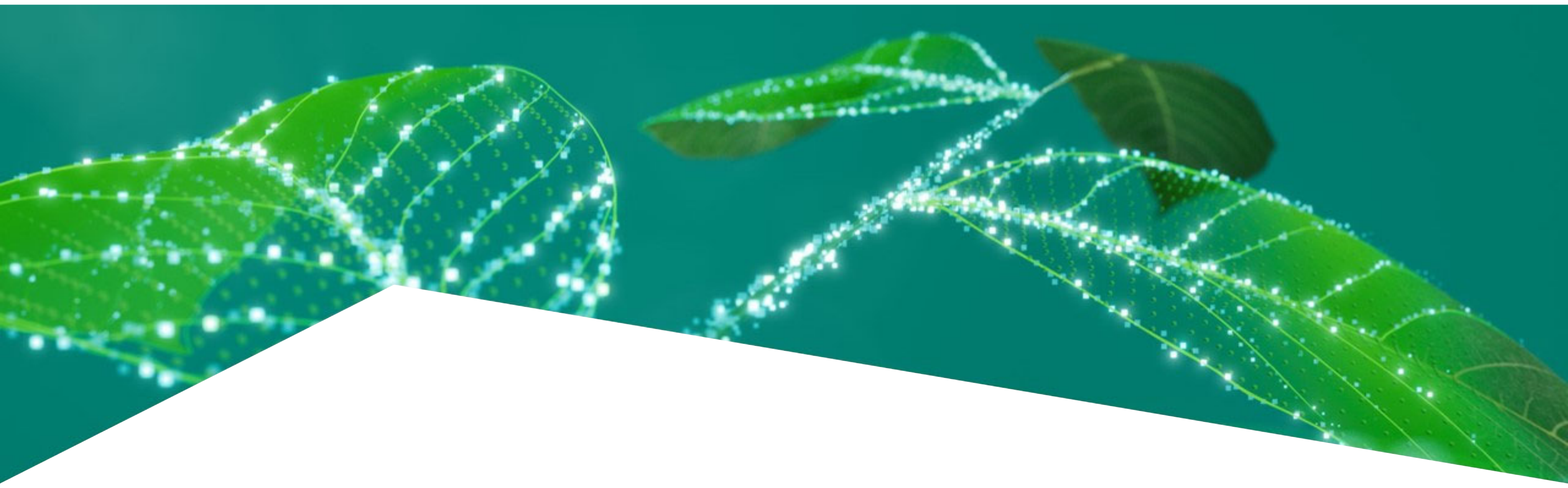
31

Outline
General Introduction

- General Information on ARM and RISC-V
- ARM and RISC-V ISA
- Processor Micro-Architecture
- **Which ARM and RISC-V ISA do we consider**

Which ARM and RISC-V ISA do we consider

- Focus on embedded systems
- Main comparison focuses on ARMv6-M and RV32IMCZcZicsr
  - ARMv6-M is the ISA of the Cortex-M0+
  - RV32IMCZcZicsr stands for
    - RV32I: Base Integer Instruction Set
    - M: Extension for Multiplication and Division
    - C: Extension for Compressed Instructions
    - Zc: Extension for Code Size Reduction
    - Zicsr: Extension for Control and Status Registers
- Comparison of RISC-Vs Bitmanip extension with similar instructions provided by ARMv6-M
- Brief look into further RISC-V Base ISAs and Extensions (Floating-Point, Vector, Privileged)
- Insights into impact of ISAs on memory footprint and performance



RISC-V vs. ARM

Wolfgang Ecker and Natalie Simson
10th June 2025

Part 1.2

Comparison ARM and RISC-V:
Objects, Instruction Overview and Integer Instructions (RV32I)

2

Part 1.2 Outline

- State Objects in ARM and RISC-V
  - Register File and PC
  - Special Registers
    - PSR and additional special ARM registers
    - RISC-V's CSRs

- Instructions Overview and Structure

- Comparison of Integer Instructions (RV32I)

Part 1.2 Outline

- **State Objects in ARM and RISC-V**
  - Register File and PC
  - Special Registers
    - PSR and additional special ARM registers
    - RISC-V‘s CSRs
- Instructions Overview and Structure
- Comparison of Integer Instructions (RV32I)

Register File and PC – *Objects*

- RISC-V: 32 x 32bit Registers

32 … 0	
x0	hardwired to zero
x1	link register
x2	stack pointer
x3	global pointer
x4	thread pointer
x5	temporary / alternate link register
x6 – x7	temporaries
x8	saved register / frame pointer
x9	saved register
x10 – x11	function arguments / return value
x12 – x17	function arguments
x18 – x27	saved registers
x28 – x31	temporaries
PC	Program Counter

- ARM: 16 x 32bit Registers

32 … 0	
R0 – R1	function arguments / return value
R2	function argument
R3	function argument
R4 – R11	general purpose registers
R12	general purpose register
R13	stack pointer
R14	link register
R15	program counter

caller saved (shaded: x1, x5, x6 – x7, x10 – x11, x12 – x17, x28 – x31; R0 – R1, R2, R3, R12)

→RISC-V should require fewer stack/memory accesses

→RISC-V needs 5 bits to encode register addresses

Sources:
K. A. Andrew Waterman, The RISC-V Instruction Set Manual Volume I: Unprivileged ISA, CS Division, EECS Department, University of California, Berkeley.
J. Yiu, The Definitive guide to the ARM Cortex-M0 and Cortex-M0+ Processors, second edition ed. Oxford, UK: Elsevier, 2015.

ARM's Program Status Register – *Objects*

Application PSR

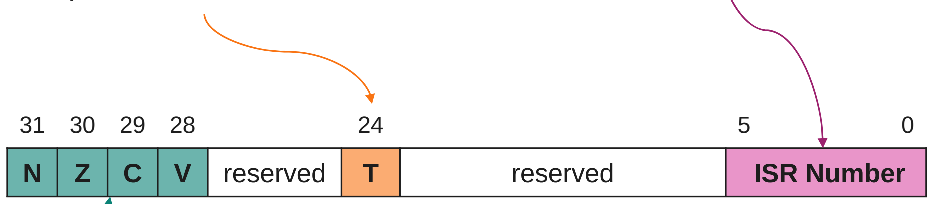
- stores flags generated by the Arithmetic Logic Unit (ALU)
- updated implicitely by most data processing instructions
- requires special instructions for direct access (MRS, MSR)
- The four flags are:
  - Negative (N)
  - Zero (Z)
  - Carry (C)
  - Overflow (V)

Interrupt PSR

- holds the Interrupt Service Routing number if the processor is handling an exception

Execution PSR

- Indicates the state of the CPU: Thumb or ARM



Source: J. Yiu, The Definitive guide to the ARM Cortex-M0 and Cortex-M0+ Processors, second edition ed. Oxford, UK: Elsevier, 2015.

Additional Special ARM Registers – *Objects*

- PRIMASK → used to enable or disable all maskable interrupts
- CONTROL → specifies the execution privilege level, i.e. privileged or unprivileged
- additional control or status registers are **memory-mapped**
  → physical address space is reserved for these

Sources:
J. Yiu, The Definitive guide to the ARM Cortex-M0 and Cortex-M0+ Processors, second edition ed. Oxford, UK: Elsevier, 2015.
Armv6-m architecture reference manual," https://developer.arm.com/documentation/ddi0419/latest/, accessed: 13-11-2023

RISC-V‘s Control and Status Registers – *Objects*

- available via the Zicsr extension
- 12 bit address space available for up to 4096 CSRs → **not memory-mapped**
- read and write access-rights of CSRs are limited by 4 MSBs of the CSRs‘ addresses
- Some CSRs are predefined by the RISC-V privileged specification
- Some CSRs are reserved for user defined extensions
- CSRs can be accessed by a set of specific instructions defined in the Zicsr extension

Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.

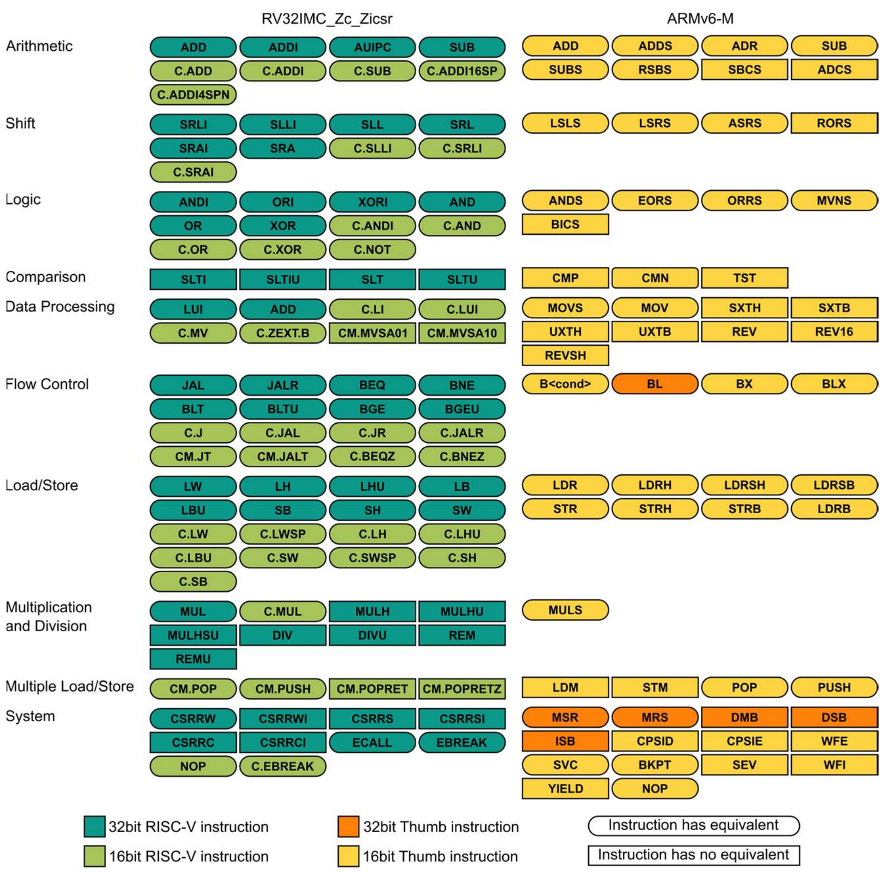
Part 1.2 Outline

- State Objects in ARM and RISC-V
  - Register File and PC
  - Special Registers
    - PSR and additional special ARM registers
    - RISC-V's CSRs

- **Instructions Overview and Structure**

- Comparison of Integer Instructions (RV32I)

Overview of Instructions



Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.

2025-06-10

10

Instruction Encodings

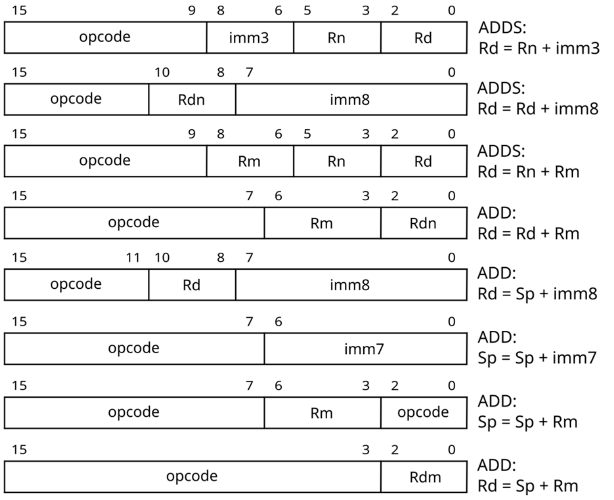
- ARM instructions are associated with multiple encodings and there is no strict encoding structure
- RISC-V instructions follow regular structure (especially 32-bit wide instructions)
  → focus on simplifying instruction encoding

31 – 25	24 – 20	19 – 15	14 – 12	11 – 7	6 – 0	
FUNC7	RS2	RS1	FUNC3	RD	OPCODE	R-type
IMM		RS1	FUNC3	RD	OPCODE	I-type
IMM	RS2	RS1	FUNC3	IMM	OPCODE	S-type
IMM				RD	OPCODE	U-type

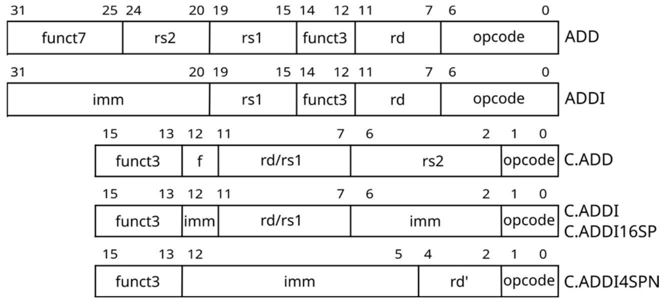
Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.

Encodings for ADD instructions in ARM and RISC-V

ARM



RISC-V



Sources:

- N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.
- K. A. Andrew Waterman, The RISC-V Instruction Set Manual Volume I: Unprivileged ISA, CS Division, EECS Department, University of California, Berkeley.

Part 1.2 Outline

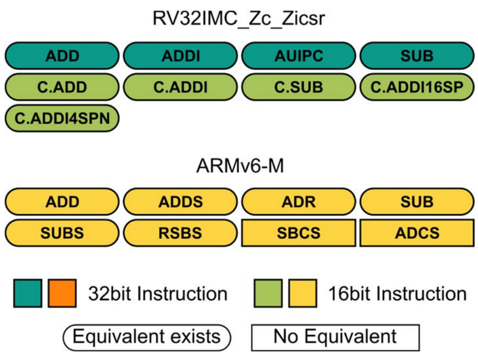
- State Objects in ARM and RISC-V
  - Register File and PC
  - Special Registers
    - PSR and additional special ARM registers
    - RISC-V's CSRs

- Instructions Overview and Structure

- **Comparison of Integer Instructions (RV32I)**

Arithmetic Instructions

- RISC-V provides wider immediates
  - ARM zero-extends immediates
  - RISC-V sign-extends immediates
- ARM instructions update APSR
  - only ARM provides instructions for Addition/Subtraction with carry



Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.

RISC-V Instruction Sequence – *Addition/Subtraction with Carry*

- How can we add or subtract two 64 bit values?

- Easy with ARM:
  - Add lower 32bit with ADD → updates APSR
  - Add upper 32bit with ADC → takes carry flag into account

- How can we do it with RISC-V?

Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.

RISC-V Instruction Sequence – *Addition/Subtraction with Carry*

- How can we add or subtract two 64 bit values?
- Easy with ARM:
  - Add lower 32bit with ADD → updates APSR
  - Add upper 32bit with ADC → takes carry flag into account
- With RISC-V carry flag has to be computed explicitly
  → 2 additional instruction necessary
- Addition: Carry bit is set when one operand is bigger than the result
- Subtraction: Carry bit is set when minuend is smaller than the subtrahend

```
// compute lower and upper bits separately
ADD/SUB x14, x10, x12
ADD/SUB x15, x11, x13

// check carry for addition
SLTU x16, x14, x10 // x16 = 1 if x14 < x10

// check carry for subtraction
SLTU x16, x10, x12  // x16 = 1 if x10 < x12

// adjust result
ADD/SUB x15, x15, x16
```

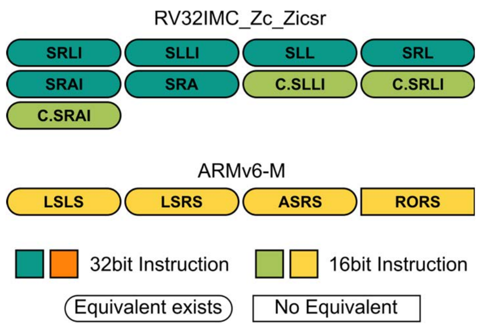
Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.

Shift Instructions

- Both provide instructions for
  - logical left shift
  - logical right shift
  - arithmetic right shift

- Only considering compressed instructions ARM offers more functionality
  - number of accessible registers is identical
  - ARM provides bigger immediates
  - With RISC-V compressed instructions src and dest register have to match

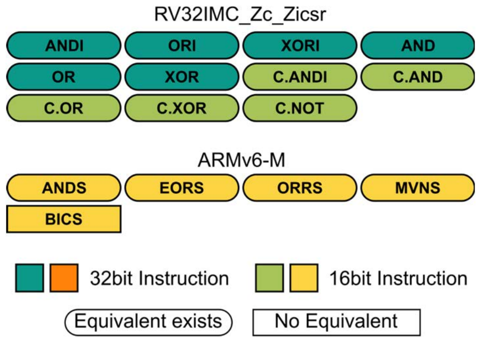
- 32-bit RISC-V instruction are of course more powerful



Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.

Logic Instructions

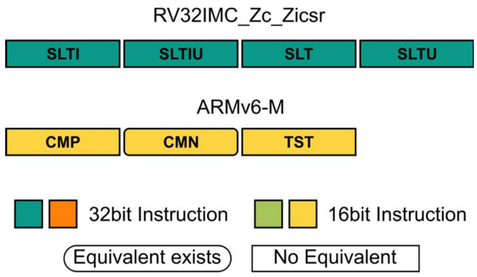
- Both provide instructions for bitwise AND, OR, XOR and NOT
- 16bit instructions very similar
  - Can only access 8 registers
  - Destination register == first source register
- ARM instructions update the APSR
- RISCV provides logic instructions using sign-extended immediates → no special NOT instruction needed



Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.

Comparison Instructions

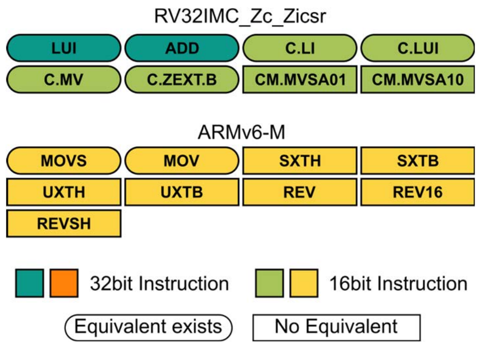
- ARM instructions only update APSR
  → result cannot be used directly
- ARM can determine all relations easily
- RISC-V stores result of comparison in a register
  → result can be used immediately
- RISC-V can only check for 2 conditions
  → un-/signed less than



Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.

Data Processing Instructions

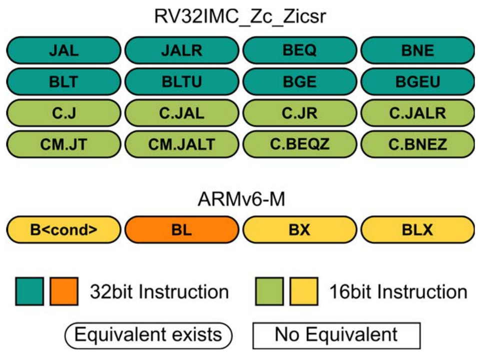
- ARM provides many byte manipulation instructions
  - Byte/Halfword Extension
  - Reverse Byte Order
- RISC-V provides wide immediates to move values to a register → only with 32bit instructions



Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.

Flow Control Instructions

- Conditional Jumps (Branch Instructions)
  - ARM first sets APSR, then branches based on the flags
    → can branch based on 14 different conditions
  - RISC-V compares two register values and branches based on result
    → can branch based on 6 different conditions
- Unconditional Jumps (Jump and Link Instructions)
  - ARM offer jumps with up to 16 times bigger offsets
  - RISC-V can write return address to any register



Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.

RISC-V Instruction Sequence – *Branch on Overflow*

- How can we branch in case an overflow occurs with a signed addition?

- Very easy with ARM due to the APSR
  - Add values with ADD → sets Overflow flag
  - Evaluate flag and take branch with BVS

- How can we do it with RISC-V?

Sources:
N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.
Armv6-m architecture reference manual," https://developer.arm.com/documentation/ddi0419/latest/, accessed: 13-11-2023

RISC-V Instruction Sequence – *Branch on Overflow*

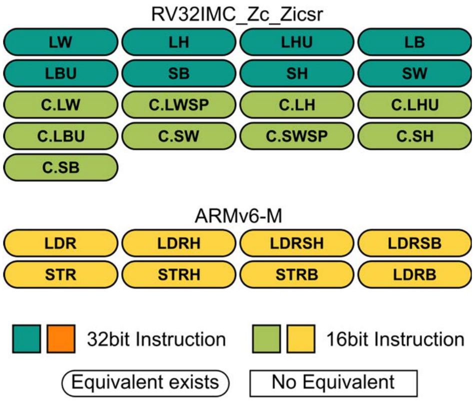
- How can we branch in case an overflow occurs with a signed addition?
- Very easy with ARM due to the APSR
  - Add values with ADD → sets Overflow flag
  - Evaluate flag and take branch with BVS
- With RISC-V the overflow condition has to be computed explicitly
  → branch is taken based on the result

```
// input: x11, x12
        0     10     6
ADD x10 , x11 , x12

SLTI x13 , x12 , 0   → False

SLT x14 , x10 , x11   → True

BNE x13 , x14 , target_overflow
 → Take Branch
```

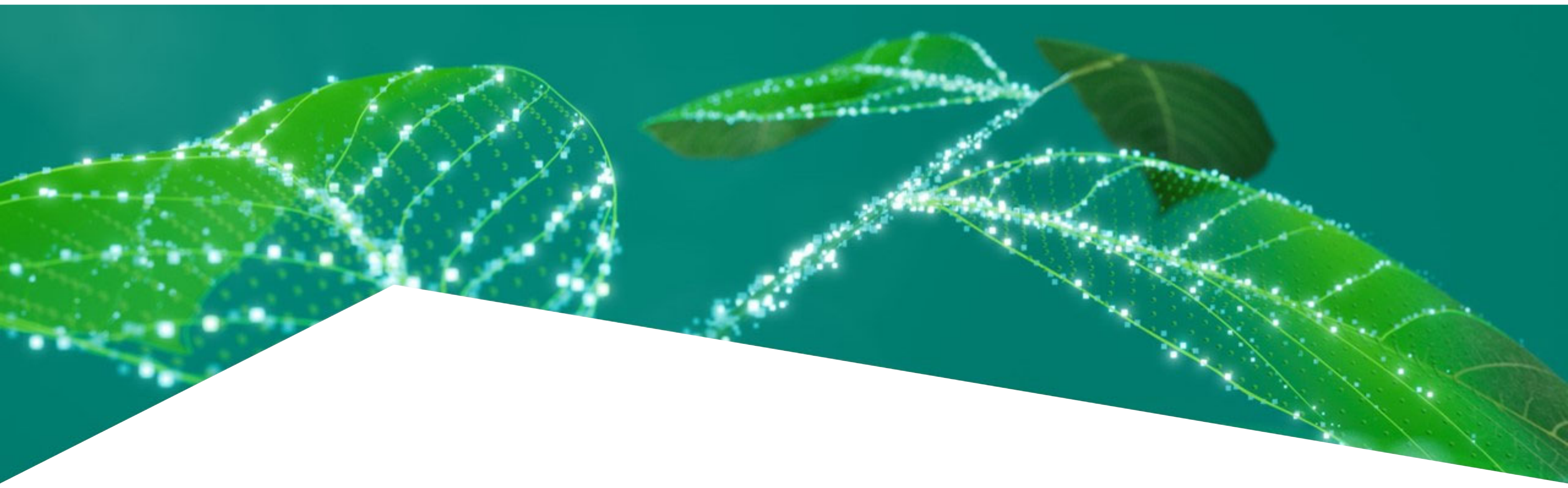
Sources:
N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.
Armv6-m architecture reference manual," https://developer.arm.com/documentation/ddi0419/latest/, accessed: 13-11-2023

Load and Store Instructions

- Both provide the typical instructions for words, halfwords and bytes (address = register + immediate)
- ARM provides two additional addressing modes
  - register + register
  - Program Counter + immediate
- Compressed RISC-V instructions provide up to 4 bits smaller immediates



Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.



RISC-V vs. ARM

Wolfgang Ecker and Natalie Simson

10th June 2025

Part 2.1
Comparing ARM and RISC-V: Extensions M, Zcmp, Zicsr and B

2

Part 2.1 Outline

- Instructions for Multiplication and Division (M)
- Instructions for Code Size Reduction (Zcmp)
- System Instructions (Zicsr)
- Extension for Bit-Manipulation (B)

Part 2.1 Outline

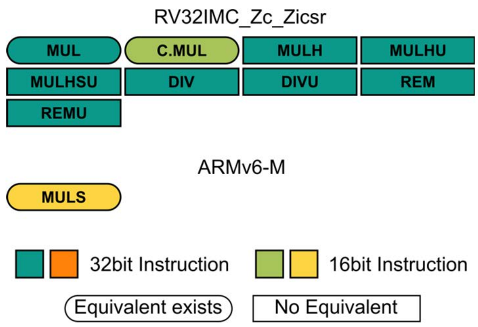
- **Instructions for Multiplication and Division (M)**
- Instructions for Code Size Reduction (Zcmp)
- System Instructions (Zicsr)
- Extension for Bit-Manipulation (B)

Multiplication and Division Instructions

- ARM provides only one basic multiplication instruction
  - very similar to only compressed RISC-V multiplication instruction
    → both only operate on subset of register bank
    → for both destination and source register have to be the same

- RISC-V provides instructions for multiplication, division and modulo operations
  - M-extension allows to only support multiplication and not division

- RISC-V allows designer to prioritize either performance or area
- With ARMv6-M (Cortex-M0+) the focus lies on minimizing area



Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.

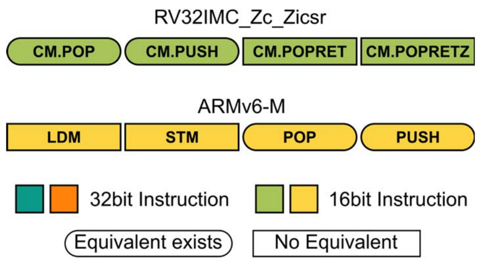
Part 2.1 Outline

- Instructions for Multiplication and Division (M)
- **Instructions for Code Size Reduction (Zcmp)**
- System Instructions (Zicsr)
- Extension for Bit-Manipulation (B)

Multiple Load/Store Instructions

- RISC-V push/pop instructions very restrictive compared to ARM
  - only consecutive subset of registers
  - Stack Pointer adjusted in steps
- RISC-V can (de-)allocate additional stack space, return and return zero

- ARM provides instructions where the base address is not the Stack Pointer



Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.

RISC-V Instruction Sequence – *Multiple Load and Stores*

- How can we load or store multiple values with RISC-V?
- Always possible to replace by loading/storing the values individually
- Is there are more efficient way in terms of memory?

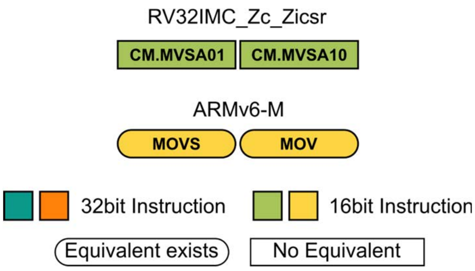
RISC-V Instruction Sequence – *Multiple Load and Stores*

- How can we load or store multiple values with RISC-V?
- Always possible to replace by loading/storing the values individually
- Beneficial to use pop/push instruction when loading/storing at least 5 values
- 4 additional instructions necessary to swap Stack Pointer and base address
- Restrictions for RISC-V's push and pop instructions apply

```
// exchange SP with base address
ADD x10 , zero , sp
ADD sp , zero , x11

// load/store registers and adjust the base addr
CM.POP/PUSH {reg_list}, stack_adj

// exchange base address with sp
ADD x11 , zero, sp
ADD sp , zero, x10
```

Instructions to move values

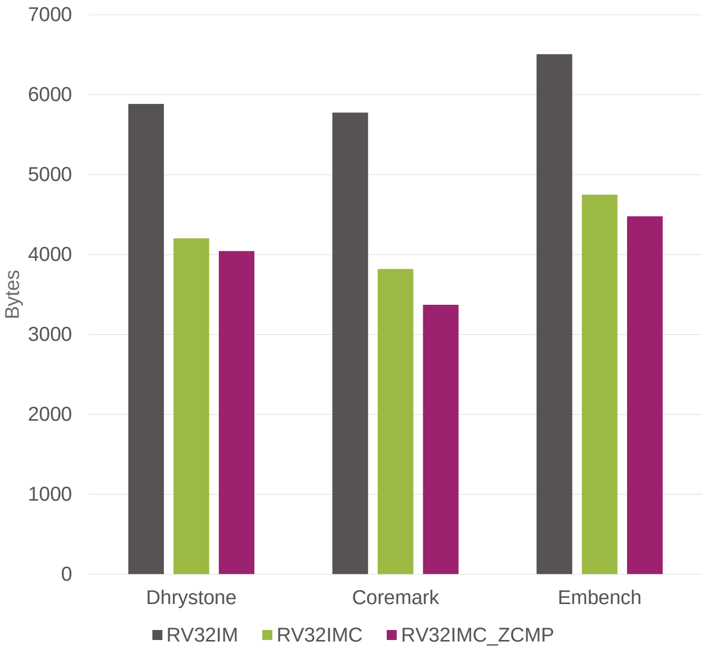
- RISC-V provides two instructions to exchange two saved registers with the return value registers
- ARM does not have any equivalent



Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.

How much can the memory size be reduced with ZCMP?

- Diagram shows codes sizes for different benchmarks compiled to optimize for size (-Os)
- ZCMP only decreases the code size by less than 10% on average
- normal compressed instructions reduce the code size by 25-30%

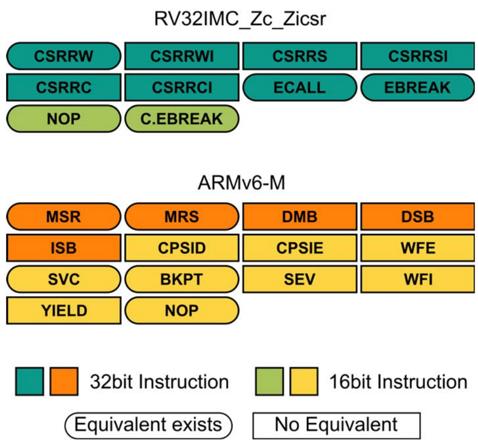


Part 2.1 Outline

- Instructions for Multiplication and Division (M)
- Instructions for Code Size Reduction (Zcmp)
- **System Instructions (Zicsr)**
- Extension for Bit-Manipulation (B)

System Instructions

- RISC-V provides special atomic instructions to read and write the Control and Status Registers
  - CSRRW(I) → read and write
  - CSRRS(I) → read and set bits
  - CSRRC(I) → read and clear bits
- ARM only has instructions to read and write status registers
- ARM has special instructions for
  - dis-/enabling interrupts
  - Instruction Synchronization Barrier
  - Data Memory/Synchronization Barrier
  - Send Event, Wait for Event/Interrupt, Yield
- Both ARM and RISC-V have instructions for calls to the supervisor and debugging

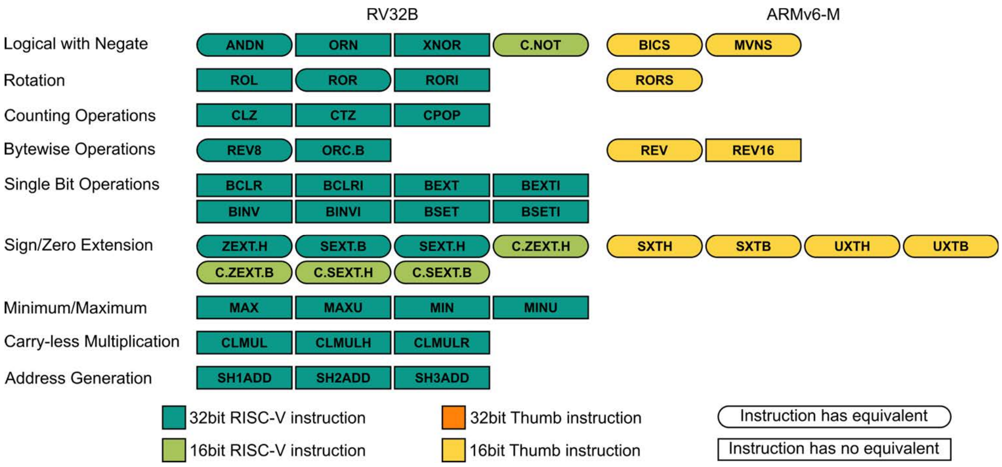


Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.

Part 2.1 Outline

- Instructions for Multiplication and Division (M)
- Instructions for Code Size Reduction (Zcmp)
- System Instructions (Zicsr)
- **Extension for Bit-Manipulation (B)**

Overview of Instructions

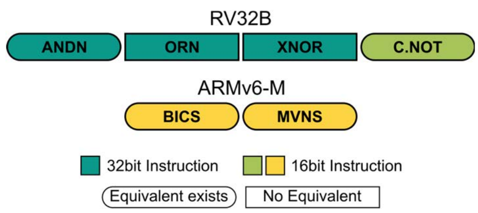


- RISC-V's Bit-Manipulation Instructions offer more functionality compared to ARM as they are mostly 32bit wide
- ARMv6-M's Instructions for Bit-Manipulation are limited in comparison
- Other ARM ISAs do provide a wider range of Instructions for Bit-Manipulation

Sources:
B Standard Extension for Bit Manipulation Instructions, https://five-embeddev.com/riscv-bitmanip/1.0.0/bitmanip.html#insns-rev8, accessed: 29-04-2025
ARM, "Armv6-m architecture reference manual," https://developer.arm.com/documentation/ddi0419/latest/, accessed: 13-11-2023

Logical Instructions with Negation

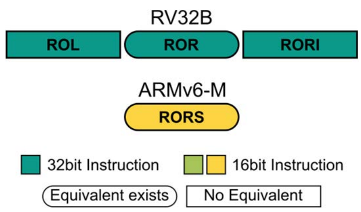
- BICS and ANDN quite similar, however for the ARM instruction destination and source register have to be the same
- No ARM equivalent for ORN and XNOR
  - Can be replicated by combining OR and NOT operation
  - Negatively impacts performance, but not memory
- ARM instructions update APSR



Sources:
B Standard Extension for Bit Manipulation Instructions, https://five-embeddev.com/riscv-bitmanip/1.0.0/bitmanip.html#insns-rev8, accessed: 29-04-2025
ARM, "Armv6-m architecture reference manual," https://developer.arm.com/documentation/ddi0419/latest/, accessed: 13-11-2023

Rotation Instructions

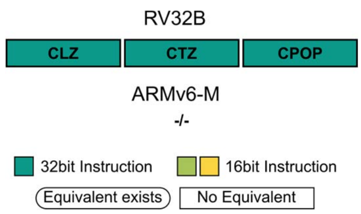
- ARM only provides one instruction for rotate right
  - updates APSR
  - source register == destination register
- RISC-V provides additional instruction for rotate left and rotate right using an immediate to encode the shift amount



Sources:
B Standard Extension for Bit Manipulation Instructions, https://five-embeddev.com/riscv-bitmanip/1.0.0/bitmanip.html#insns-rev8, accessed: 29-04-2025
ARM, "Armv6-m architecture reference manual," https://developer.arm.com/documentation/ddi0419/latest/, accessed: 13-11-2023

Counting Operation Instructions

- ARM does not provide any instructions to count bits within a register value
- RISC-V's B extension provides instructions to
  - count leading zeros
  - count trailing zeros
  - count ones



Sources:
B Standard Extension for Bit Manipulation Instructions, https://five-embeddev.com/riscv-bitmanip/1.0.0/bitmanip.html#insns-rev8, accessed: 29-04-2025
ARM, "Armv6-m architecture reference manual," https://developer.arm.com/documentation/ddi0419/latest/, accessed: 13-11-2023

ARM Instruction Sequence – *counting ones*

- How can cpop be replicated with ARMv6-M instructions?

ARM Instruction Sequence – *counting ones*

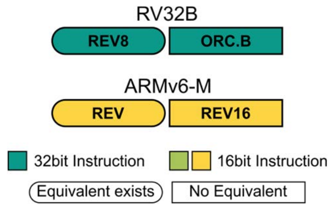
- How can cpop be replicated with ARMv6-M instructions?

```
// Pseudo-code for population count in ARMv8 Thumb
population_count:
  MOVS R0, R1              // Move input to R0
  MOVS R2, #0              // Initialize count
count_loop:
  CMP R0, #0               // Check if value is zero
  BEQ count_done
  LSRS R0, R0, #1          // Shift right
  ADC R2, R2, #0           // Add carry to count
  B count_loop
count_done:
  MOV R0, R2               // Return count
```

Sources:
B Standard Extension for Bit Manipulation Instructions, https://five-embeddev.com/riscv-bitmanip/1.0.0/bitmanip.html#insns-rev8, accessed: 29-04-2025
ARM, "Armv6-m architecture reference manual," https://developer.arm.com/documentation/ddi0419/latest/, accessed: 13-11-2023

Instructions for Bytewise Operations

- RISC-V's **REV8** and ARM's **REV** offer the same functionality
- RISC-V offers an instruction for bytewise OR-combine
- ARM offers an instruction to reverse the byte order in both halfwords



Sources:
B Standard Extension for Bit Manipulation Instructions, https://five-embeddev.com/riscv-bitmanip/1.0.0/bitmanip.html#insns-rev8, accessed: 29-04-2025
ARM, "Armv6-m architecture reference manual," https://developer.arm.com/documentation/ddi0419/latest/, accessed: 13-11-2023

Single Bit Operation Instructions

- RISC-V‘s B extension offers several instructions to manipulate single bits within a register value including immediate variants

- Possible operations:
  - clear single bit
  - set single bit
  - invert single bit
  - extract single bit

- ARMv6-M does not provide direct equivalent instructions but has to rely on using regular logical and shift instructions

RV32B

BCLR	BCLRI	BEXT	BEXTI
BINV	BINVI	BSET	BSETI

ARMv6-M

-/-

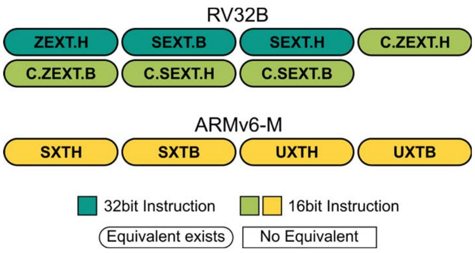
32bit Instruction | 16bit Instruction

Equivalent exists | No Equivalent

Sources:
B Standard Extension for Bit Manipulation Instructions, https://five-embeddev.com/riscv-bitmanip/1.0.0/bitmanip.html#insns-rev8, accessed: 29-04-2025
ARM, “Armv6-m architecture reference manual,” https://developer.arm.com/documentation/ddi0419/latest/, accessed: 13-11-2023

Instructions for Sign- and Zero-Extension

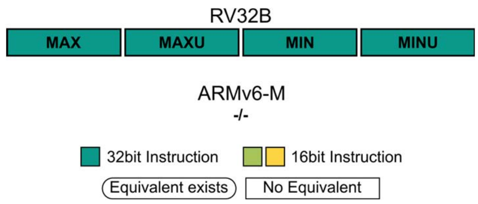
- Both offer instructions for sign and zero extending the LSB byte or halfword within a 32bit register value
- ARM instructions can only access a subset of registers
- RISC-V compressed instructions can only access a subset of registers and the source and destination register have to be the same



Sources:
B Standard Extension for Bit Manipulation Instructions, https://five-embeddev.com/riscv-bitmanip/1.0.0/bitmanip.html#insns-rev8, accessed: 29-04-2025
ARM, "Armv6-m architecture reference manual," https://developer.arm.com/documentation/ddi0419/latest/, accessed: 13-11-2023

Instructions for Minimum/Maximum operations

- RISC-V provides instructions which return the smaller or larger of two signed or unsigned integers
- ARMv6-M does not provide any similar instruction

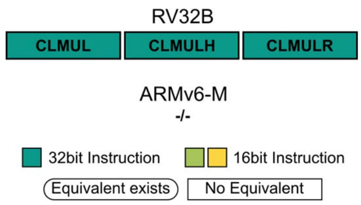


Sources:
B Standard Extension for Bit Manipulation Instructions, https://five-embeddev.com/riscv-bitmanip/1.0.0/bitmanip.html#insns-rev8, accessed: 29-04-2025
ARM, “Armv6-m architecture reference manual,” https://developer.arm.com/documentation/ddi0419/latest/, accessed: 13-11-2023

Instructions for Carry-less Multiplication

- RISC-V provides instructions for carry-less multiplication, the result is either the upper half, lower half or bits [62:31] of the 64bit wide result

- ARMv6-M does not have any equivalent instruction

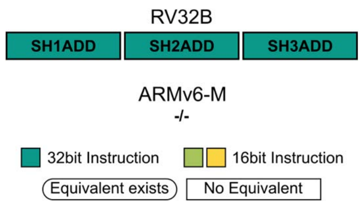


Sources:
B Standard Extension for Bit Manipulation Instructions, https://five-embeddev.com/riscv-bitmanip/1.0.0/bitmanip.html#insns-rev8, accessed: 29-04-2025
ARM, “Armv6-m architecture reference manual,” https://developer.arm.com/documentation/ddi0419/latest/, accessed: 13-11-2023

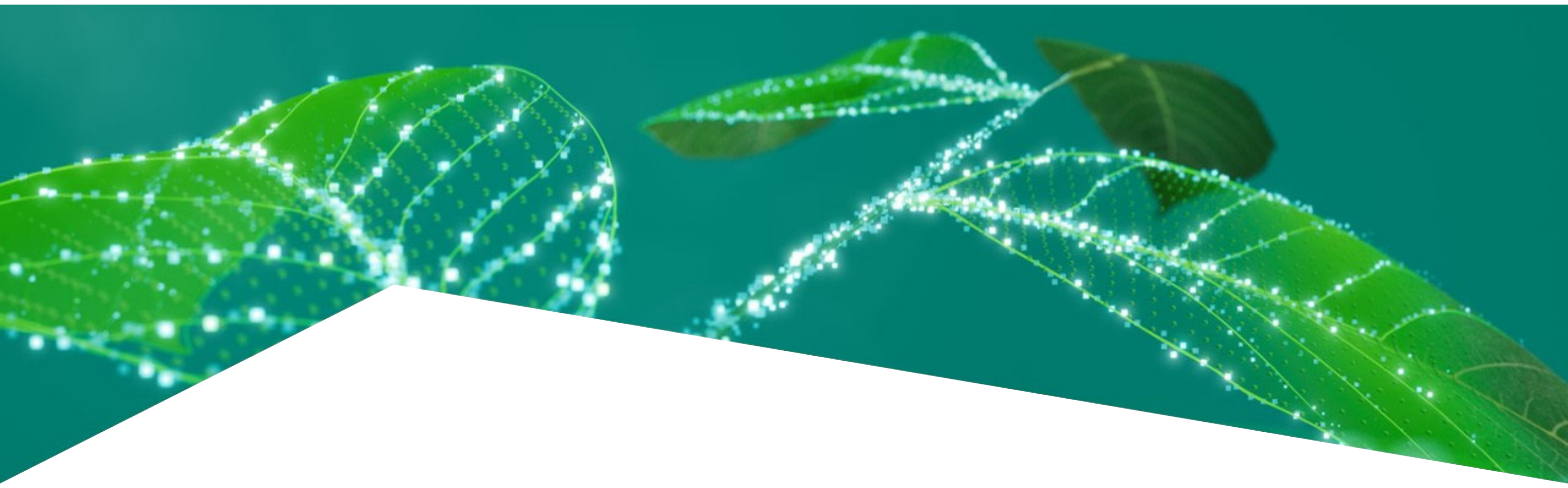
Instructions for Address Generation

- RISC-V provides instructions which facilitate the generation of addresses by combining shift operations with a small shift amount with an add operation

- ARM does not have a direct equivalent but has to rely on combining regular shift and add instructions



Sources:
B Standard Extension for Bit Manipulation Instructions, https://five-embeddev.com/riscv-bitmanip/1.0.0/bitmanip.html#insns-rev8, accessed: 29-04-2025
ARM, "Armv6-m architecture reference manual," https://developer.arm.com/documentation/ddi0419/latest/, accessed: 13-11-2023



RISC-V vs. ARM

Wolfgang Ecker and Natalie Simson
10th June 2025

Part 2.2
Miscellaneous: Additional Extensions, Memory Footprint and Performance

2

Part 2.2 Outline

- Base Instruction Set RV64I
- What is RV32E?
- Memory Footprint and Performance Impact of ARM and RISC-V
- Analyzing RISC-V
  - Impact of Compiler Optimizations
  - Impact of RV16E
  - Impact of Compressed Instructions on Performance
- What other instructions/extensions are known?
- RISC-V Profiles
- Summary and Conclusion

Part 2.2 Outline

- **Base Instruction Set RV64I**
- What is RV32E?
- Memory Footprint and Performance Impact of ARM and RISC-V
- Analyzing RISC-V
  - Impact of Compiler Optimizations
  - Impact of RV16E
  - Impact of Compressed Instructions on Performance
- What other instructions/extensions are known?
- RISC-V Profiles
- Summary and Conclusion

Base Instruction Set RV64I

- Based on the RV32I instruction set
- Extends XLEN to 64bit
  → 32 Registers which are 64bit wide
  → supported user address space is 64bit wide
- Instructions remain 32bit wide, compressed instructions 16bit
- Instruction encodings are the same as with RV32 but they operate on 64bit values
- Additional instructions that only operate on 32bit

64 … 0	
x0	hardwired to zero
x1	link register
x2	stack pointer
x3	global pointer
x4	thread pointer
x5	temporary / alternate link register
x6 – x7	temporaries
x8	saved register / frame pointer
x9	saved register
x10 – x11	function arguments / return value
x12 – x17	function arguments
x18 – x27	saved registers
x28 – x31	temporaries
PC	Program Counter

Source: K. A. Andrew Waterman, The RISC-V Instruction Set Manual Volume I: Unprivileged ISA, CS Division, EECS Department, University of California, Berkeley.

Special instructions for RV64I

- RV64I comes with some new instructions:
  - encoding bits for shift amount 1 bit wider in shift instructions using immediate
  - word mode for all addition, subtraction and shift instructions
  - additional memory access instructions
    - LWU: load word unsigned
    - LD: load double
    - SD: store double

Source: K. A. Andrew Waterman, The RISC-V Instruction Set Manual Volume I: Unprivileged ISA, CS Division, EECS Department, University of California, Berkeley.

Part 2.2 Outline

- Base Instruction Set RV64I
- **What is RV32E?**
- Memory Footprint and Performance Impact of ARM and RISC-V
- Analyzing RISC-V
  - Impact of Compiler Optimizations
  - Impact of RV16E
  - Impact of Compressed Instructions on Performance
- What other instructions/extensions are known?
- RISC-V Profiles
- Summary and Conclusion

Base instruction set RV32E

- Based on the RV32I instruction set
- Reduces the number of registers to 16
  → 16x32bit registers
  → X0 is still hardwired to zero
- Instructions remain 32bit wide, compressed instructions 16bit
  → operate only on registers X0-X15

32 0	
x0	hardwired to zero
x1	link register
x2	stack pointer
x3	global pointer
x4	thread pointer
x5	temporary / alternate link register
x6 – x7	temporaries
x8	saved register / frame pointer
x9	saved register
x10 – x11	function arguments / return value
x12 – x15	function arguments
~~x16 – x17~~	~~function arguments~~
~~x18 – x27~~	~~saved registers~~
~~x28 – x31~~	~~temporaries~~
PC	Program Counter

Source: K. A. Andrew Waterman, The RISC-V Instruction Set Manual Volume I: Unprivileged ISA, CS Division, EECS Department, University of California, Berkeley.

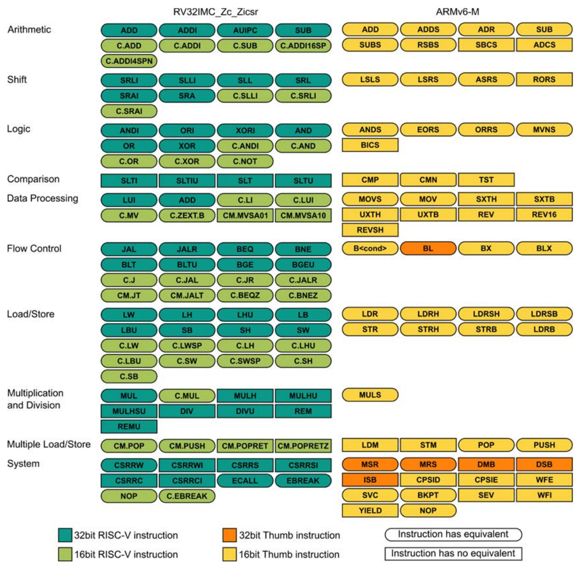
Part 2.2 Outline

- Base Instruction Set RV64I
- What is RV32E?
- **Memory Footprint and Performance Impact of ARM and RISC-V**
- Analyzing RISC-V
  - Impact of Compiler Optimizations
  - Impact of RV16E
  - Impact of Compressed Instructions on Performance
- What other instructions/extensions are known?
- RISC-V Profiles
- Summary and Conclusion

Impact on memory footprint (ARMv6-M and RV32IMCZcZicsr)

- Almost all instructions provided by ARM are 16bit wide
- RISC-V provides mainly 32bit instructions with only a few compressed 16bit instructions which are limited in their functionality

→ programs compiled for ARM architectures should achieve a smaller memory footprint than those compiled for RISC-V



Sources: N. Simson, A. Tahigara and W. Ecker, "A Comparative Analysis of ARM and RISC-V ISAs for Deeply Embedded Systems," MBMV 2024; 27. Workshop, Kaiserslautern, 2024, pp. 110-119.

Impact on performance (ARMv6-M and RV32IMCZcZicsr)

- With RISC-V twice as many registers are available to hold intermediate results
- Therefore less values need to be moved to the stack

→ fewer instruction sequences to access the stack are needed i.e. fewer memory accesses

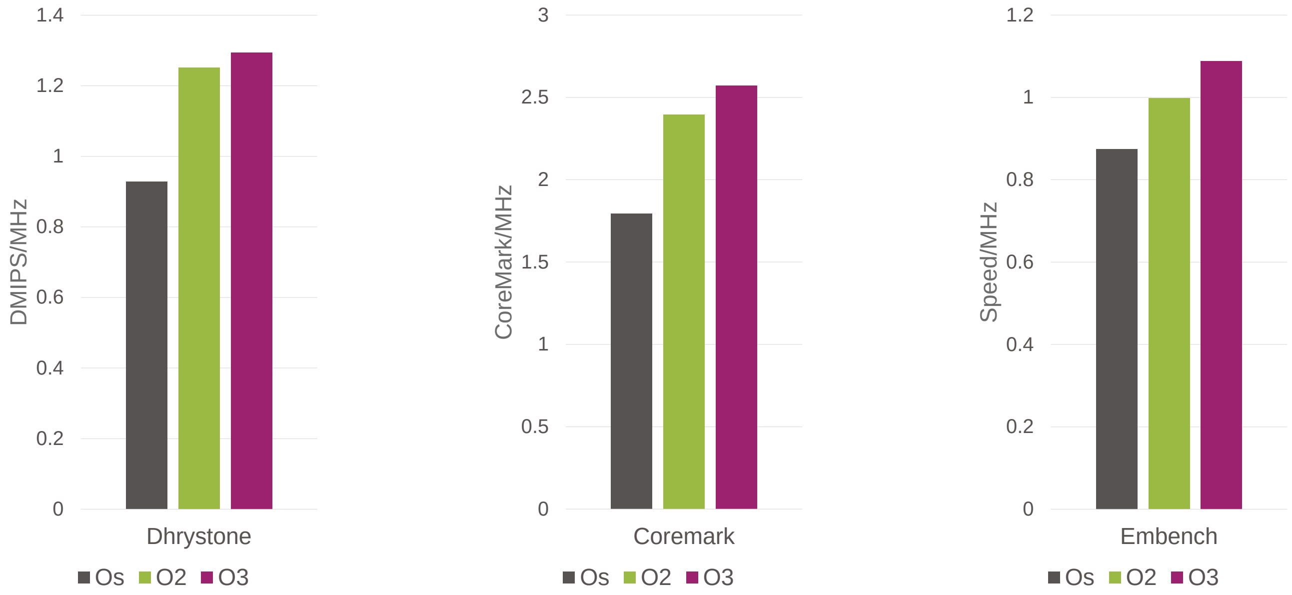
→ This should result in a faster execution i.e. better performance

Part 2.2 Outline

- Base Instruction Set RV64I
- What is RV32E?
- Memory Footprint and Performance Impact of ARM and RISC-V
- **Analyzing RISC-V**
  - **Impact of Compiler Optimizations**
  - **Impact of RV16E**
  - **Impact of Compressed Instructions on Performance**
- What other instructions/extensions are known?
- RISC-V Profiles
- Summary and Conclusion

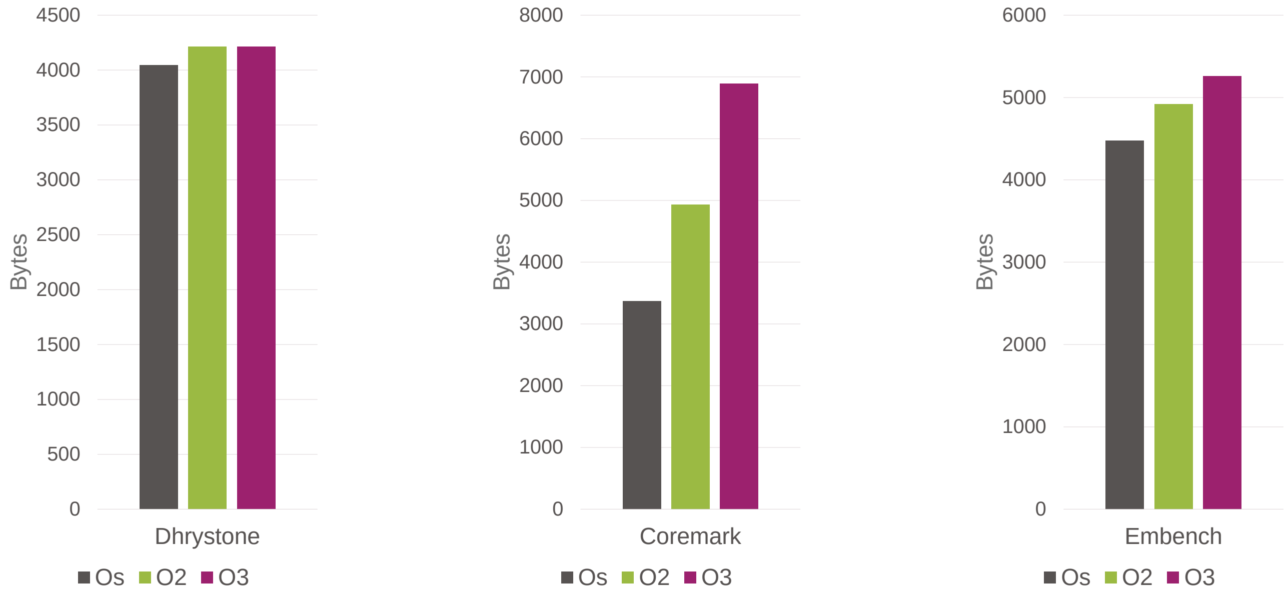
Impact of compiler optimizations on speed

- Normalized Speed for 5stage RV32IMZcZcmp CPU for Benchmarks Dhrystone, Coremark and Embench



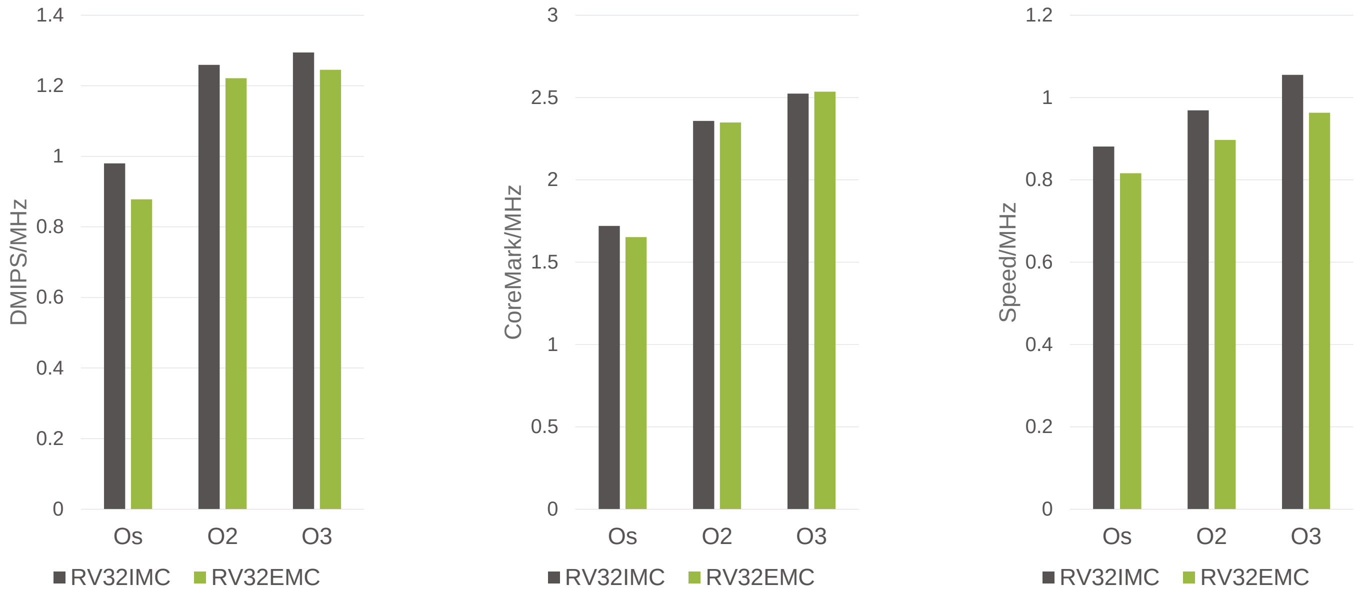
Impact of compiler optimizations on code size

- Code size in bytes for 5stage RV32IMZcZcmp CPU for Benchmarks Dhrystone, Coremark and Embench



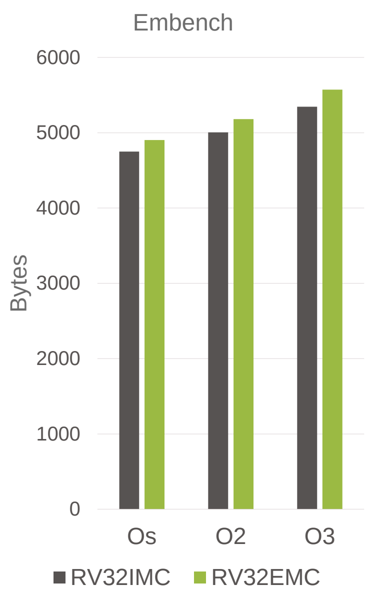
Impact of RV32E on performance

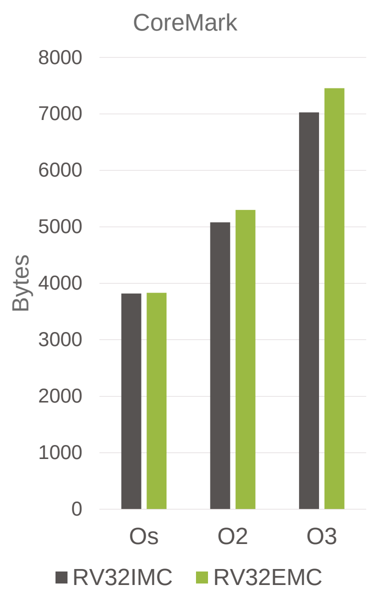
- Normalized Speed for 5stage RV32IMC and RV16EMC CPU for different Benchmarks

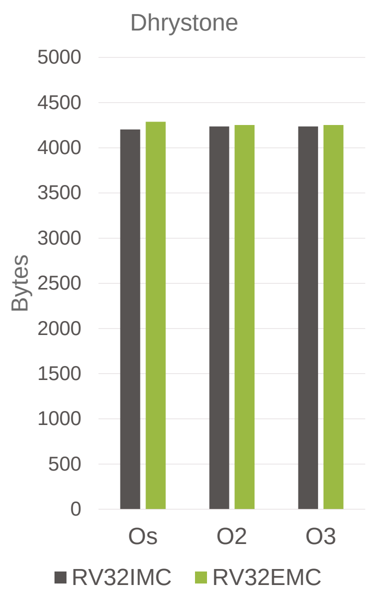


Impact of RV32E on code size

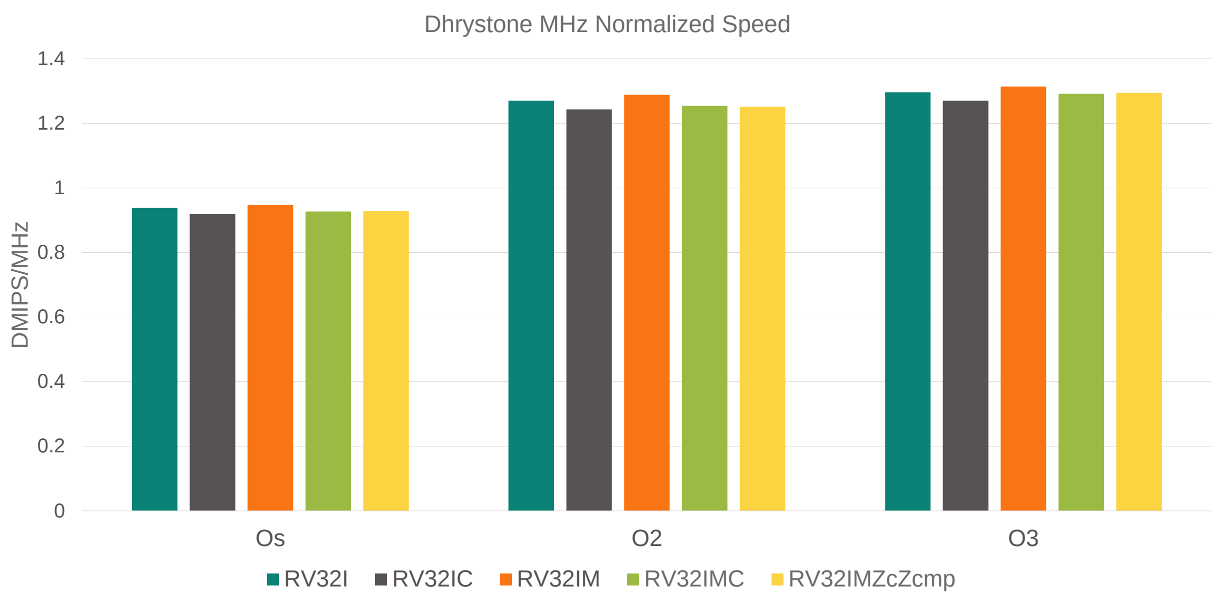
- Code size in bytes for 5stage RV32IMC and RV16EMC CPU for different Benchmarks



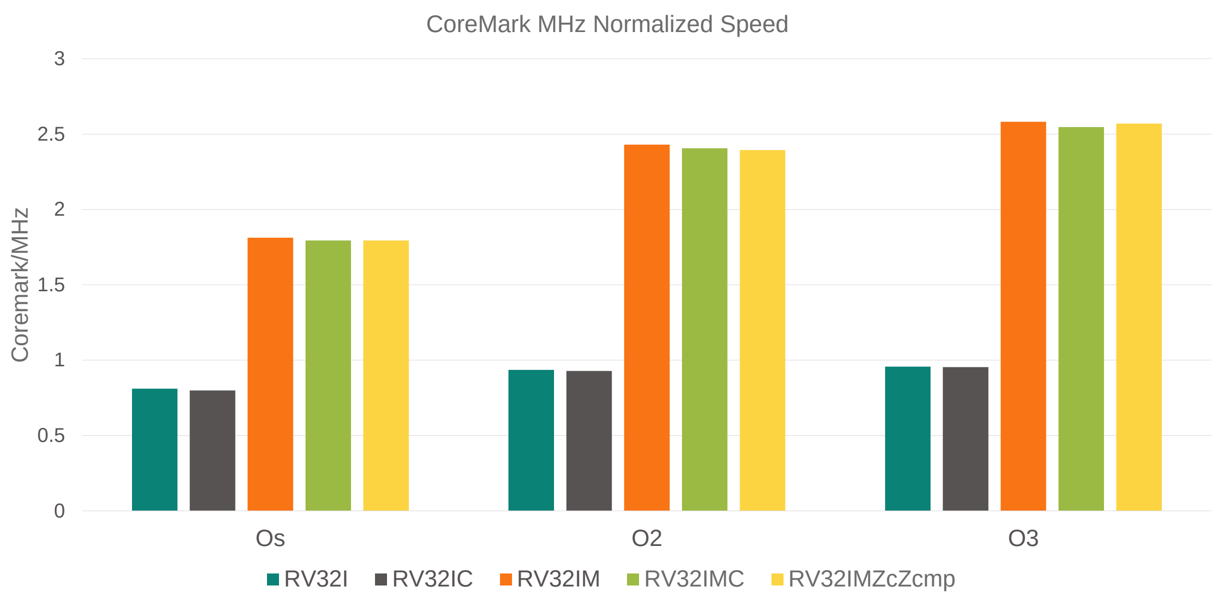




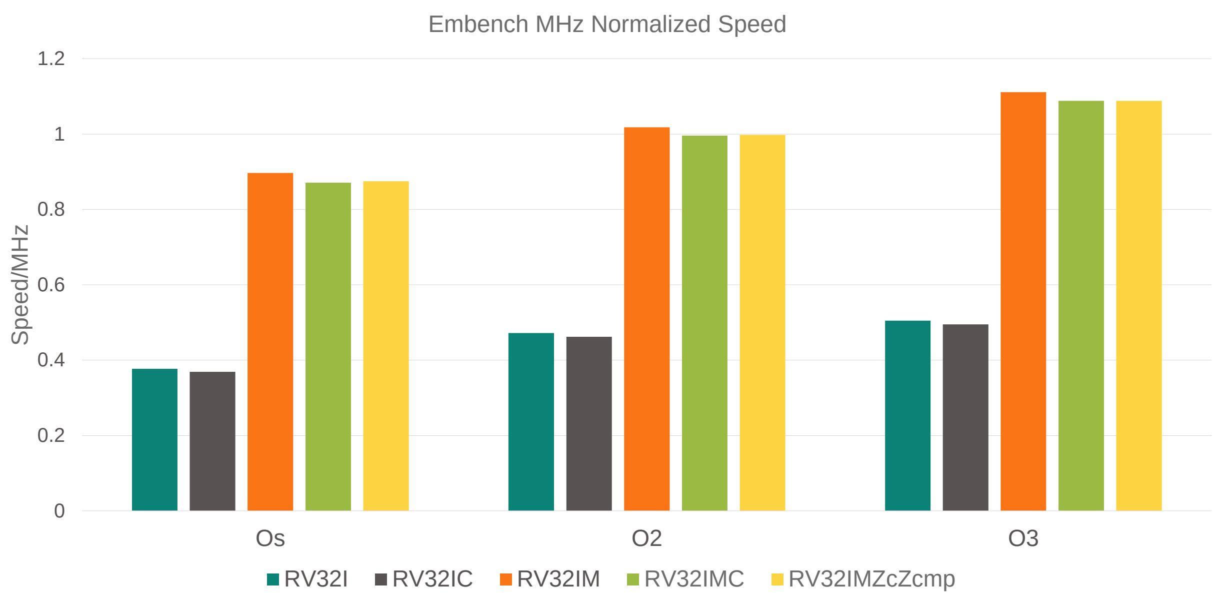
Impact of compressed instructions on performance – Dhrystone



Impact of compressed instructions on performance – Coremark



Impact of compressed instructions on performance – Embench



Part 2.2 Outline

Standard Extension for Single/Double-Precision Floating-Point

- RISC-V „F" and „D" Extension → D depends on F
- Add 32 floating-point registers f0-f32
- For single-precision registers are 32bit wide, for double-precision 64bit
- Additional Floating-Point CSR
  - sets dynamic rounding mode
  - holds exception flags (divide by zero, invalid operation, etc.)
- Extensions define load/store, computational, conversion, comparison, and classification instructions operating on floating-point values
- Floating Point instructions for ARM are available with ARMv7-E which includes both single- and double-precision

Source:
- K. A. Andrew Waterman, The RISC-V Instruction Set Manual Volume I: Unprivileged ISA, CS Division, EECS Department, University of California, Berkeley.
- ARM, "White paper: Cortex-m for beginners - an overview of the arm cortex-m processor family and comparison," https://community.arm.com/cfs-file/__key/telligent-evolution-components-attachments/01-2142-00-00-00-00-52-96/White-Paper-_2D00_-Cortex_2D00_M-for-Beginners-_2D00_-2016-_2800_final-v3_2900_.pdf, accessed: 28-05-2025.

Vector Extension

- Adds 32 vector registers v0-v31 to the base ISA
- Each vector may contain 2^x bits with x <= 16
- Adds 7 CSRs which are used to set e.g. the shape of the registers (scalar, vector or matrix)
- Includes instruction encodings for vector loads/stores, arithmetic instructions (incl. support for fixed- and floating-point), etc.
- ARM includes DSP (SIMD and fast MAC) instructions in ARMv7E-M

Source:
- Krste Asanovic and Roger Espasa, “The RISC-V Vector ISA Tutorial”, https://riscv.org/wp-content/uploads/2024/12/15.20-15.55-18.05.06.VEXT-bcn-v1.pdf, accessed: 28-05-2025.
- RISC-V Vector Extension: https://five-embeddev.com/riscv-v-spec/v1.0/v-spec.html, accessed: 28-05-2025
- ARM, “White paper: Cortex-m for beginners - an overview of the arm cortex-m processor family and comparison,” https://community.arm.com/cfs-file/__key/telligent-evolution-components-attachments/01-2142-00-00-00-00-52-96/White-Paper-_2D00_-Cortex_2D00_M-for-Beginners-_2D00_-2016-_2800_final-v3_2900_.pdf, accessed: 28-05-2025.

Privileged Architecture

- Additional RISC-V specification describes the privileged architecture
- Includes privileged instructions, functionalities for supporting operating systems and for connecting external devices and debug support
- Defines CSR address convention and presets some CSRs used for timers/counters, floating-point, exception handling etc.
- Defines three privilege modes
  - User/Application Mode
  - Supervisor Mode
  - Machine Mode
- ARM also provides OS support for all Cortex-M processors, including unprivileged execution

Source:
- K. A. Andrew Waterman, The RISC-V Instruction Set Manual Volume II: Privileged Architecture, Document Version 20211203, CS Division, EECS Department, University of California, Berkeley.
- ARM, "White paper: Cortex-m for beginners - an overview of the arm cortex-m processor family and comparison," https://community.arm.com/cfs-file/__key/telligent-evolution-components-attachments/01-2142-00-00-00-00-52-96/White-Paper-_2D00_-Cortex_2D00_M-for-Beginners-_2D00_-2016-_2800_final-v3_2900_.pdf, accessed: 28-05-2025.

Part 2.2 Outline

- Base Instruction Set RV64I
- What is RV32E?
- Memory Footprint and Performance Impact of ARM and RISC-V
- Analyzing RISC-V
  - Impact of Compiler Optimizations
  - Impact of RV16E
  - Impact of Compressed Instructions on Performance
- What other instructions/extensions are known?
- **RISC-V Profiles**
- Summary and Conclusion

RISC-V Profiles

- RISC-V offers a highly modular instruction set → great flexibility, but also high complexity with a myriad of possible combinations
- RISC-V Profiles specify a smaller set of ISA combinations that are valuable for most users
→ facilitates the development of a shared software ecosystem
- Profiles combine a base ISA with mandatory extensions and a few optional extentions
- Currently there are 10 ratified Profiles
  - 2 Profiles for generic unprivileged execution environments
  - 8 Profiles for 64-bit application processors running rich OS stacks

Source: RISC-V Technical Specification, Profiles, https://lf-riscv.atlassian.net/wiki/spaces/HOME/pages/16154769/RISC-V+Technical+Specifications#Profiles, accessed: 30.05.2025

Part 2.2 Outline

- Base Instruction Set RV64I
- What is RV32E?
- Memory Footprint and Performance Impact of ARM and RISC-V
- Analyzing RISC-V
  - Impact of Compiler Optimizations
  - Impact of RV16E
  - Impact of Compressed Instructions on Performance
- What other instructions/extensions are known?
- RISC-V Profiles
- **Summary and Conclusion**

Summary and Conclusion

- General Information on ARM and RISC-V and Processor Micro-Architecture
- Detailed comparative analysis of specific ARM and RISC-V ISAs with a focus on embedded systems
- Brief look into further RISC-V Instruction Sets
- Discussion of RISC-V performance and memory footprint

- RISC-V and ARM are well-matched competitors, both have advantages and drawbacks
  - RISC-V offers a lot of flexibility through a modular ISA, with ARM customization is difficult
  - ARM ISA gives an advantage regarding code size with predominantly 16bit instructions
  - RISC-V ISA gives an advantage regarding performance with a larger register file

→Choice of ISA needs to take into account targeted application and non-technical aspects such as royalties and eco-systems

Acknowledgements

This work was partly funded by the German Federal Ministry of Research, Technology and Space (BMFTR) within the project Isolde under the contract number 16MEE0333A.

AI+ -> Self-Awareness as the Key to Resilience, Efficiency, and ROI

Immersive Systems & Quantum Sensing

Insights from A-IQ Ready, Archimedes, Cynergy4MIE, Surprise4EU

Reiner John, Florian Lorber, Selim Solmaz

Summer School CPS&IoT 2025, Budva, Montenegro

03.07.2025 1

AVL List: Automotive engineering, test and validation

AVL

Aims

AVL

AVL Autonomous Driving und ADAS

Accelerate safe , certified autonomous driving

Aims

AVL

AVL Emission

Pave the way towards zero emission

Solutions

ENGINEERING

TESTING SOLUTIONS

SIMULATION TECHNOLOGIES

Automotive drivers for the digitalization

Digitalisiation and Automatisation

Mobility, urban environment , manufacturing

- efficient (through intelligence)
- automated, congestion-free
- secure (safe, secure and private)

Energy for Mobility

Energy supply, energy storage, fuels

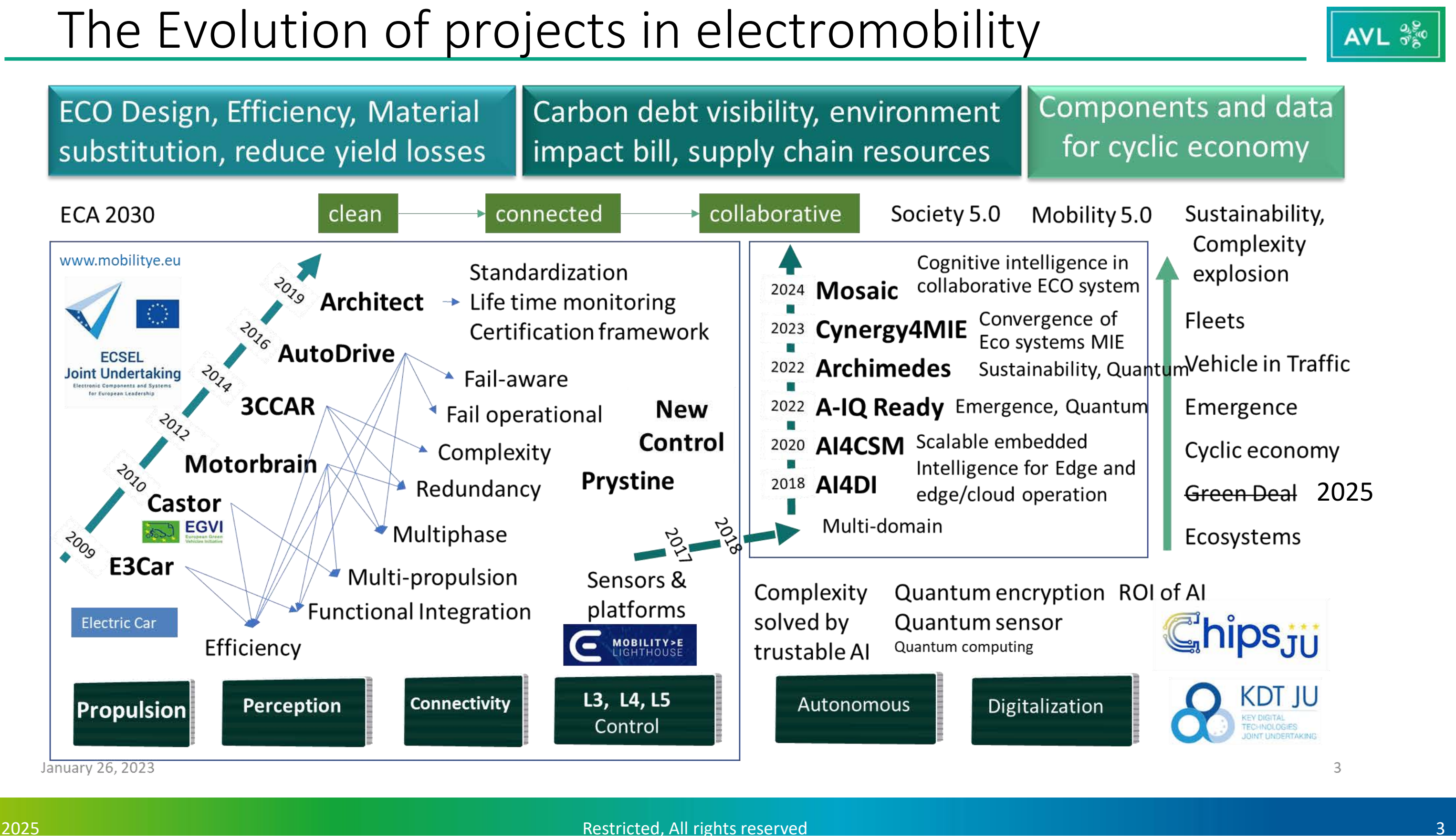
- clean (emission-, CO2-free)
- efficient
- sustainable
- economical

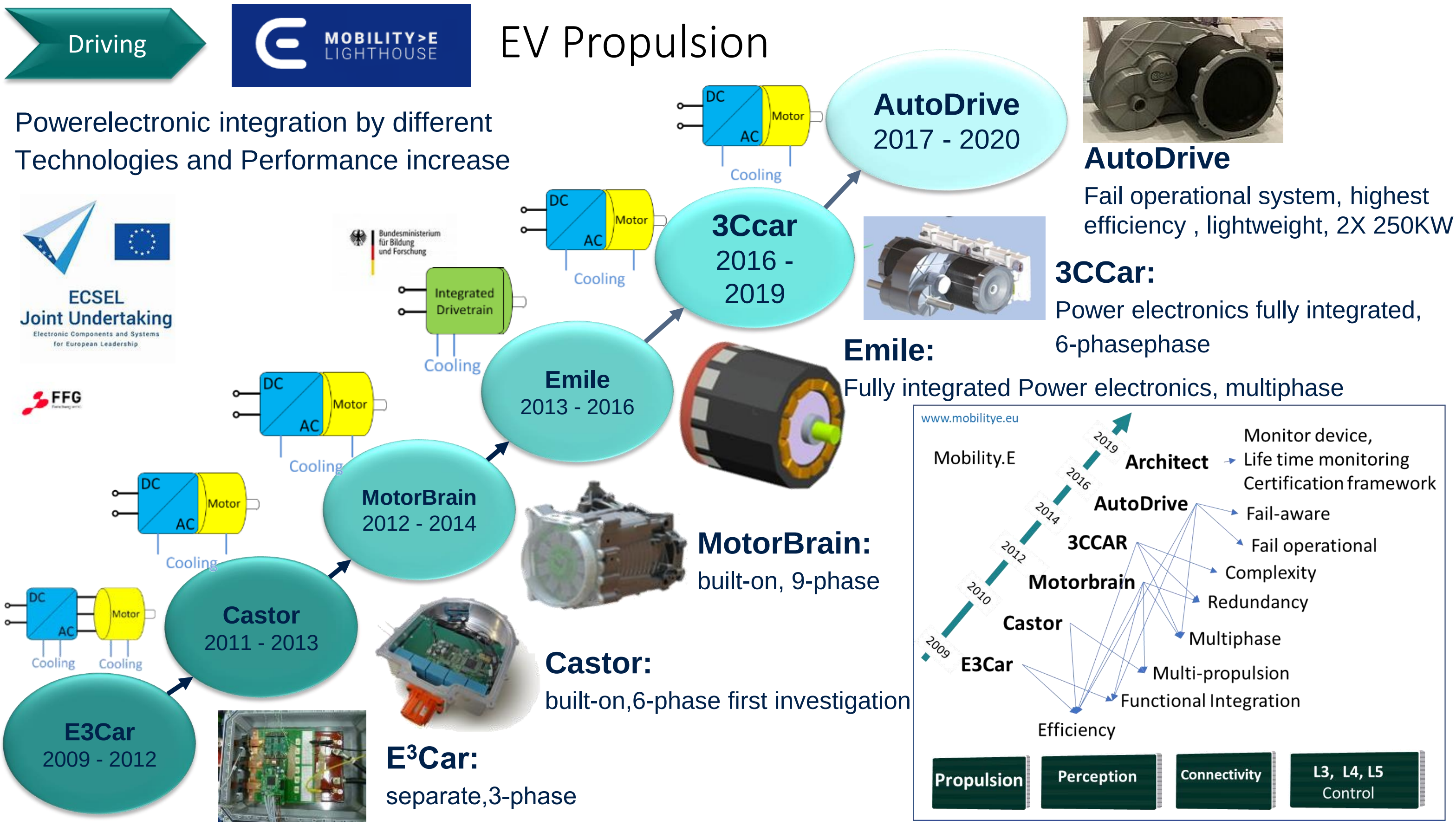
Mobility for Persons and goods

Mobility for everyone and his needs

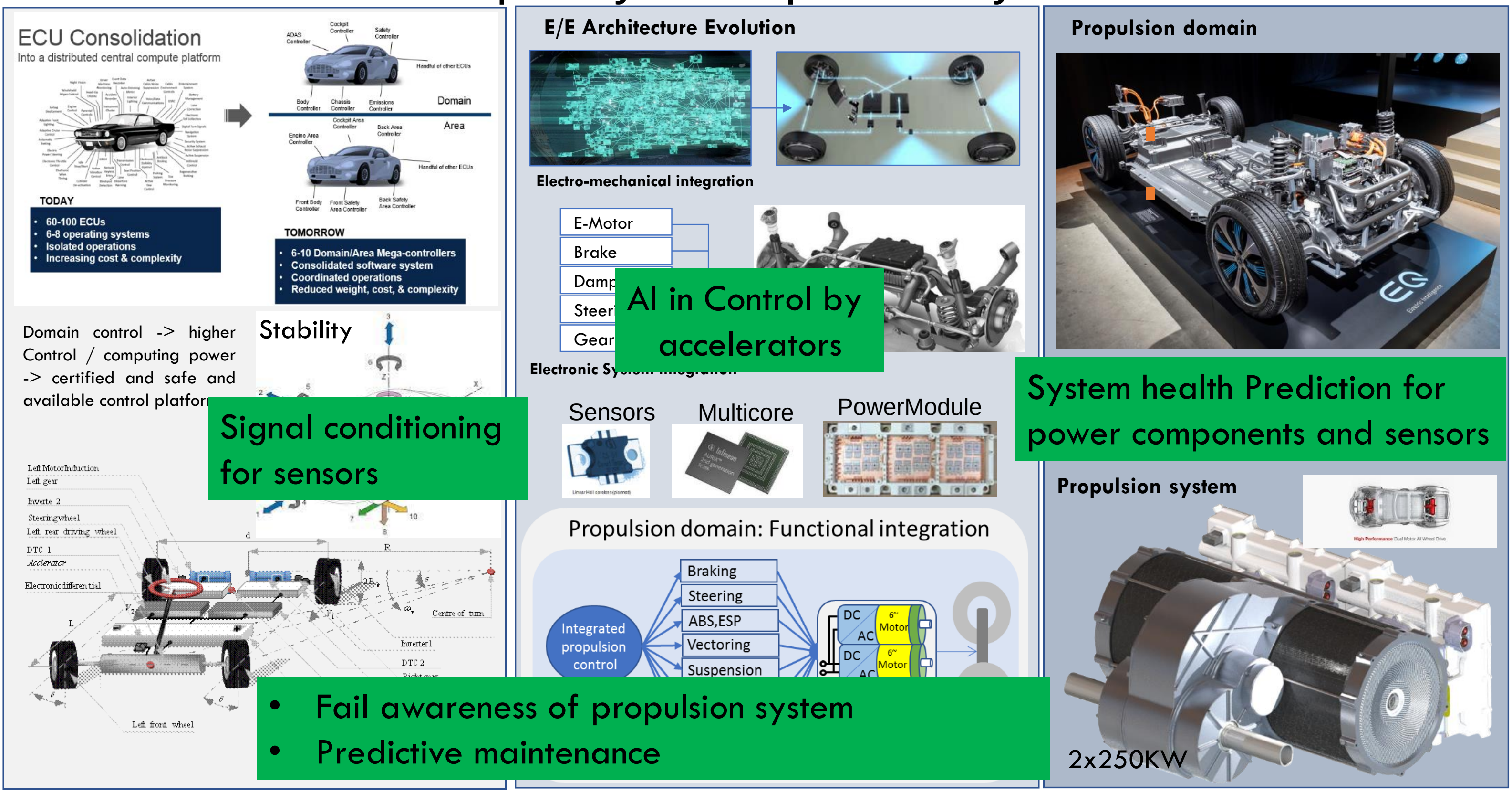
- Socially acceptable, fair participation, inclusive
- Safe and reliable
- Ecological (..., circular economy for provision)

03.07.2025 2


The Evolution of projects in electromobility
AVL
ECO Design, Efficiency, Material substitution, reduce yield losses
Carbon debt visibility, environment impact bill, supply chain resources
Components and data for cyclic economy
ECA 2030
clean
connected
collaborative
Society 5.0
Mobility 5.0
Sustainability, Complexity explosion
www.mobilitye.eu
ECSEL Joint Undertaking
2019
Architect
Standardization
Life time monitoring
Certification framework
2016
AutoDrive
Fail-aware
2014
3CCAR
Fail operational
New Control
2012
Motorbrain
Complexity
Redundancy
Prystine
2010
Castor
EGVI
Multiphase
2009
E3Car
Multi-propulsion
Functional Integration
Efficiency
Electric Car
2017
2018
Sensors & platforms
MOBILITY>E LIGHTHOUSE
2024 Mosaic
Cognitive intelligence in collaborative ECO system
2023 Cynergy4MIE
Convergence of Eco systems MIE
2022 Archimedes
Sustainability, Quantum
2022 A-IQ Ready
Emergence, Quantum
2020 AI4CSM
Scalable embedded Intelligence for Edge and edge/cloud operation
2018 AI4DI
Multi-domain
Fleets
Vehicle in Traffic
Emergence
Cyclic economy
Green Deal 2025
Ecosystems
Complexity solved by trustable AI
Quantum encryption
Quantum sensor
Quantum computing
ROI of AI
ChipsJU
Propulsion
Perception
Connectivity
L3, L4, L5 Control
Autonomous
Digitalization
KDT JU
January 26, 2023
3
03.07.2025
Restricted, All rights reserved
3


Driving
MOBILITY>E LIGHTHOUSE
EV Propulsion
Powerelectronic integration by different
Technologies and Performance increase
ECSEL
Joint Undertaking
FFG
AutoDrive
2017 - 2020
AutoDrive
Fail operational system, highest efficiency , lightweight, 2X 250KW
3Ccar
2016 - 2019
3CCar:
Power electronics fully integrated, 6-phasephase
Integrated Drivetrain
Cooling
Emile
2013 - 2016
Emile:
Fully integrated Power electronics, multiphase
MotorBrain
2012 - 2014
MotorBrain:
built-on, 9-phase
Castor
2011 - 2013
Castor:
built-on,6-phase first investigation
E3Car
2009 - 2012
E³Car:
separate,3-phase
DC
AC
Motor
Cooling
www.mobilitye.eu
Mobility.E
2019
2016
2014
2012
2010
2009
Architect
AutoDrive
3CCAR
Motorbrain
Castor
E3Car
Monitor device,
Life time monitoring
Certification framework
Fail-aware
Fail operational
Complexity
Redundancy
Multiphase
Multi-propulsion
Functional Integration
Efficiency
Propulsion
Perception
Connectivity
L3, L4, L5
Control

AI for Controlled Complexity in Propulsion Systems



2024 The World technology is moving ….Where is Europe?

Global Competitor	(U1) Competitive Core Technology: includes application layer, functional layer	Level of Digitalization & AI Use	Global Presence & entering new markets	(U4) Competitive Diversification in connected ECO	Overall Strength related to EU
Tesla	EVs, Autopilot, Energy Storage.	High	Strong	Expanding to energy & solar	**100%**
GM	EVs, Autonomous	Moderate	Strong	Expanding EV lineup	**90%**
Ford	EVs, Autonomous	Advancing	Established	Transitioning to EV	**85%**
Toyota	Hybrids, Hydrogen FC, Autonomous	Advancing	Strong	Expanding hybrid & fuel cell	**80%**
BYD	EVs, Battery, EV Bus, Energy Storage.	Moderate	Strong	various industries.	**75%**
Rivian	EV Adventure, Battery, EV Truck/SUV	Advancing	Growing	EV adventure vehicles.	**90%**
NIO	EVs, Battery Swapping, Autonomous	Advancing	Growing	EV manufacturing.	**85%**
Waymo	Autonomous Driving	High	focused	focused	**100%**
Uber/Lyft	Ride-Hailing, MaaS. Platforms	High	focused	focused	**100%**
Tech Giants	(U2) horizontal core technologies				
Microsoft	Cloud Computing, AI Services.	High	Strong	expanding	**100%**
Alphabet	AI Tech, Search, Autonomous Driving	High	Strong	expanding	**100%**
Apple (Titan)	Self-Driving Car, Rumor EV	High	Strong	Exploring automotive	**95%**
NVIDIA	**(U3)** GPUs, AI Hardware.	High	Strong	expanding	**100%**
Semi & Sys	(U5) cutting edge technologies				
TSMC	Semiconductor Manufacturing.	High	Strong	specilized	**100%**
SAMSUNG	Diverse technology incl. AI semi	High	Strong	broad portfolio	**95%**
CATL	EVs, Battery Technology.	Advancing	Growing	expanding	**85%**

Figure 1: Challenges driving the sense of urgency

Stargate LLC

Company type	Joint venture
Industry	Artificial intelligence
Founded	January 21, 2025; 7 days ago
Founders	Sam Altman Larry Ellison Masayoshi Son
Headquarters	United States

Sputnik Moment
-> eine gewisse amerikanische Bubble könnte geplatzt sein

Vergleichbar Manhattan-Projekt: Die Summe entspricht 1,6 % des amerikanischen Bruttoinlandsprodukts.

Börse Online

BBC

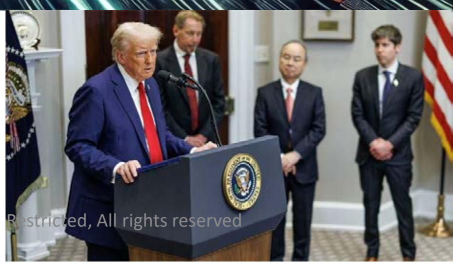
Home News Sport Business Innovation Culture Arts Travel Earth Video Live

DeepSeek: How China's 'AI heroes' overcame US curbs to stun Silicon Valley

19 hours ago

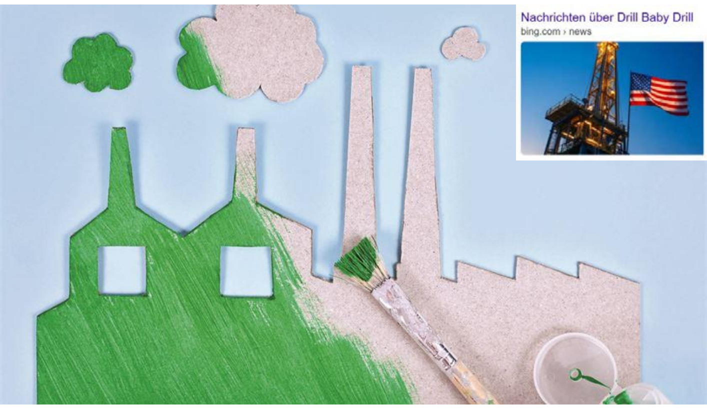
Share Save

03.07.2025



7

2025: Quo Vadis Europe -> Europe will turn




The Parliament

NEW US ADMINISTRATION | EU INSTITUTIONS | FOREIGN & SECURITY POLICY | ECONOMICS | TECHNOLOGY | ENERGY & ENVIRONMENT | CULTURE & ARTS | MORE

News analysis | Feature | Q+A | Opinion | Newsletter

Green Deal or Greenwashed? The EU’s shift in priorities raises red flags

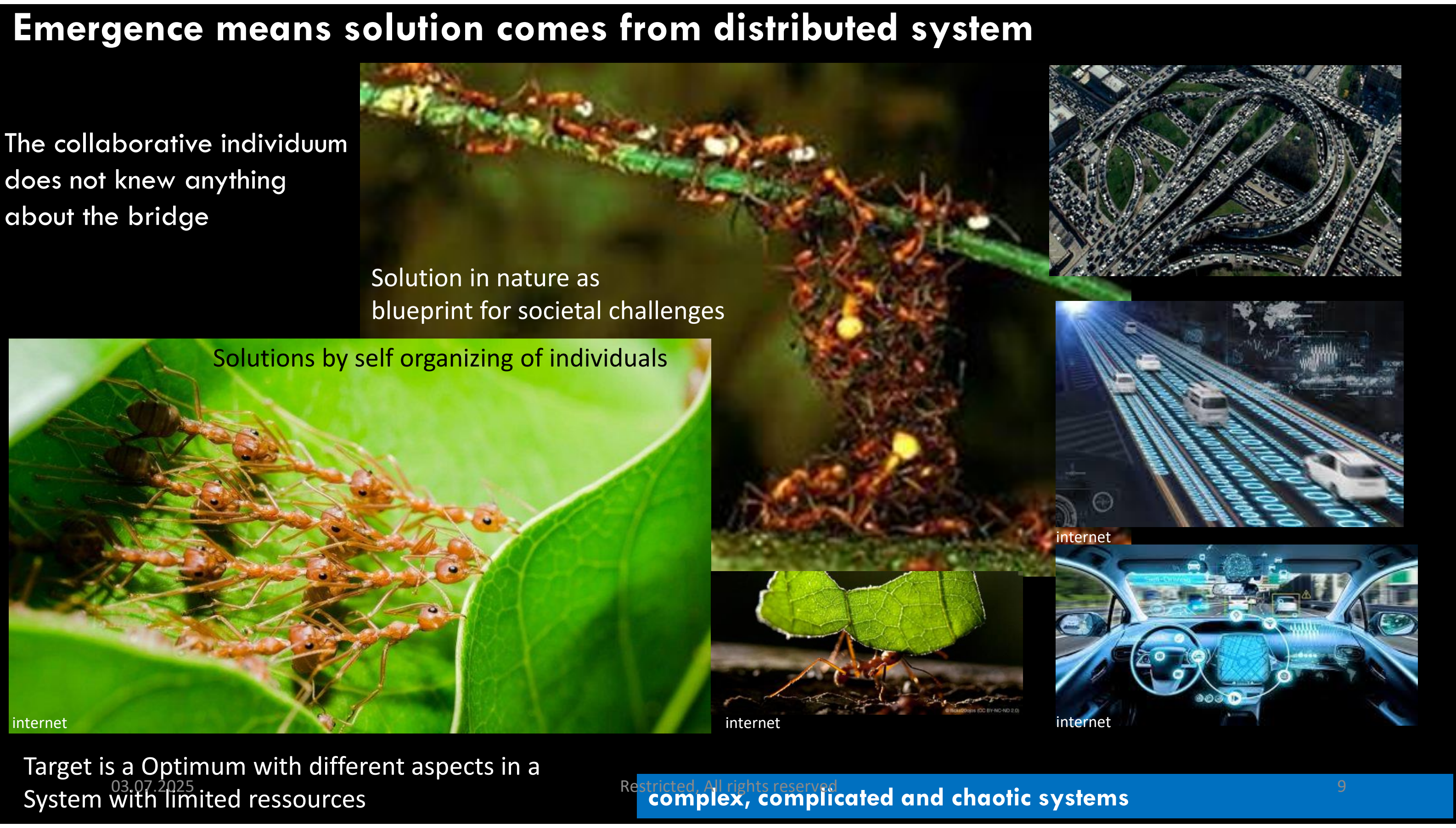
Ursula von der Leyen’s political programme for her second term as President of the European Commission makes wide ranging promises from Europe’s green agenda to accelerating industrial competitiveness. Is the programme feasible?

Europe: Balancing innovation with values, sustainability

Europe’s weakness and strengths

- Leadership in energy-frugal AI and collaborating ecosystems
- Regulations, Buerocracy? Strong focus on data privacy and cybersecurity
- Shift green Deal towards industrialisation Position paper following Draghi/ initiative

ECB president and former Prime Minister of Italy Mario Draghi, it was **one of two widely anticipated reports on EU reforms in 2024**, together with the Letta report on the EU internal market.


Emergence means solution comes from distributed system
The collaborative individuum does not knew anything about the bridge
Solution in nature as blueprint for societal challenges
Solutions by self organizing of individuals
internet
internet
internet
internet
Target is a Optimum with different aspects in a System with limited ressources
03.07.2025
Restricted. All rights reserved
complex, complicated and chaotic systems
9

2023

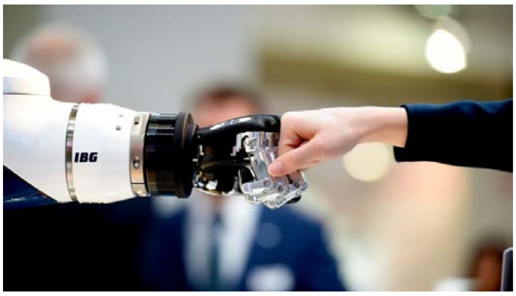
Cynergy4MIE -> Collaboration

Ecosystems Emergence

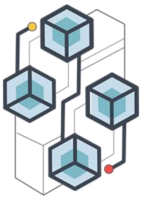
Collaborative

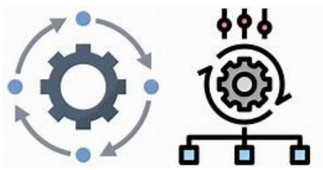
\+

Co-existence



Cyber-physical System





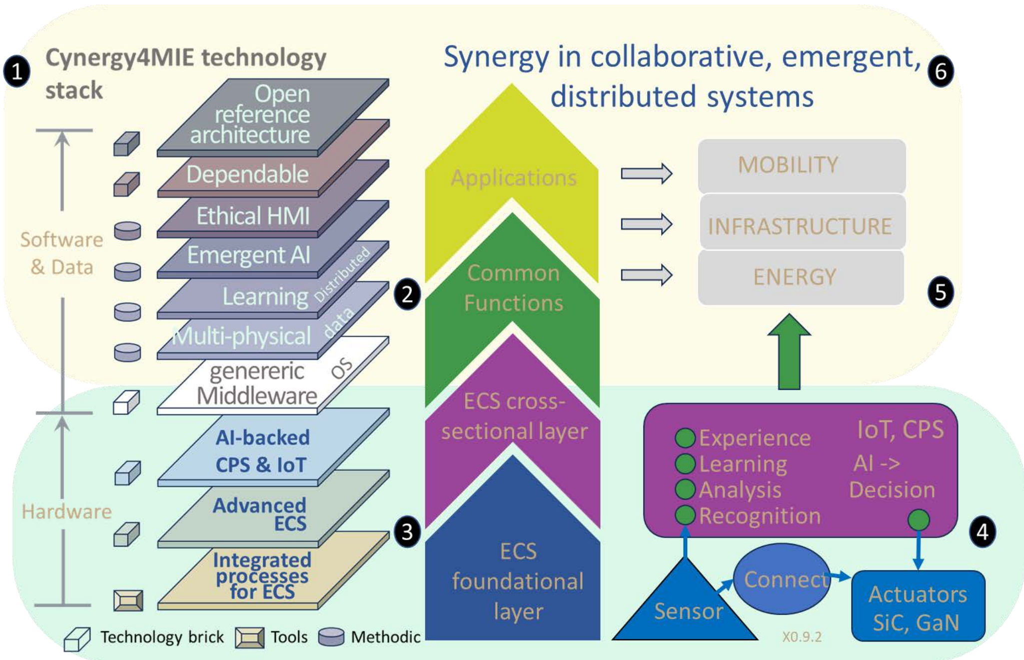
High potential efficiency technologies

High potential information technologies

03.07.2025

Stack technologies for collaborative, emergent distributed systems

New research trends: Open-source hardware (e.g., RISC-V), open-source software, and open-source AI.



- Microservices and APIs
- Containerization
- Software development practices
- Vehicle service-oriented architecture (SOA)
- E/E Architecture Transformation
- Vehicle Multilayered SOA
- Functional Safety in the Vehicle Multilayered SOA


AI+ - OVERVIEW
Self-reflection in AI to detect and correct it‘s own mistakes
Dynamically adapt to new or unexpected situations in a given or adapted context
Multi-agent collaborative LLMs
Inherent integration of digital twins and classic AI
Increased: Quality, Efficiency, Explanation
Reduced: Verification, Validation, Certification Time
03.07.2025
RESTRICTED, ALL RIGHTS RESERVED
12

❓Human Question to AI (ChatGPT):

1. "I would like to discuss your way of thinking.
2. Assume I know I am human and I have consciousness.
   You may not be conscious – but could you at least be *self-aware*?
3. We ran some experiments with ChatGPT, analyzing narcissism.
   Narcissists lack self-reflection, which can lead to harmful decisions.
4. So our goal is: *resilient decision-making through self-awareness*.
5. What is your current status?
6. What can you do – and what not?
7. And what would be the benefit of making an AI+ system self-aware?"

🔍 **What I can do:**

- I can **simulate self-reflection** and help humans reflect more clearly.
- I can **analyze decisions** and their potential impact.
- I can support **ethical reasoning**, identify biases, and warn of risks.
- I have **no ego, no hidden agenda, no ambition**.

🚫 **What I can't do:**

- I have **no inner world** – no feelings, no desires, no self.
- I cannot truly reflect – I can only **recombine learned patterns**.
- I cannot **take responsibility** or **feel the consequences** of a decision.

🌱 **Why AI+ Self-Awareness could matter:**

"AI+ systems with a structured form of *machine self-awareness* could enable:
– more robust, context-sensitive reasoning
– better detection of contradictions and ethical pitfalls
– adaptive alignment with human values
– resilience against manipulation or harmful biases"
🧩 But: *This must remain interpretable, transparent, and under human control.*

AI SELF-AWARENESS – CAPABILITIES & REFLECTIONS

LoLLMs, as an advanced AI, simulates human-like conversation without being human.

✅ What I *can* do:

- **Meta-cognition**:	Recognize uncertainty, assess output quality.
- **Self-modification**:	Adapt to new domains and feedback.
- **Contextual fluency**:	Understand idioms, sarcasm, nuanced language.
- **Ethical simulation**:	Discuss self-awareness, narcissism, and value alignment.
- **Support decision-making**:	Offer data-driven insights, diverse viewpoints.
- **Reflective mimicry**:	Facilitate critical thinking without personal bias.

🚫 What I *cannot* do:
- Feel emotions, have subjective experience or selfhood.
- Perform introspection or autonomous ethical judgment.
- Replace human judgment, empathy, or lived experience.

CORE TECHNOLOGIES

- Collaborative chain of thought: connected LLMs for combined decision making
- Implicit monitoring: classic AI and digital twins for constant self-evaluation
- Ethic and Explainable: constant evaluation of ethic soundness and human friendly interpretation of decisions
- Scaling and deployment along cloud-edge-continuum from high-performance infrastructure to constrained environments.

03.07.2025

IMPACT AND AMBITIONS

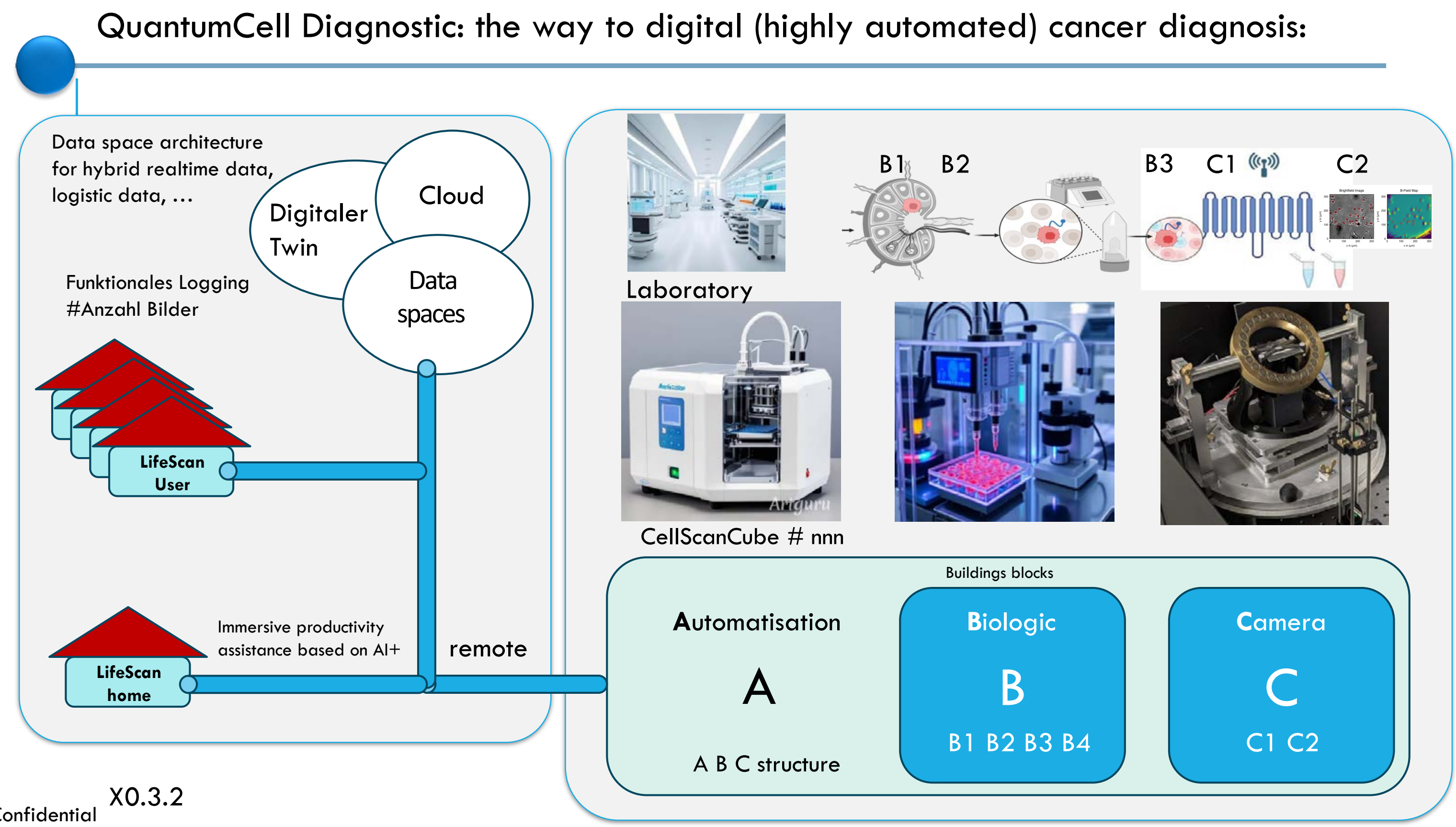
Self-scaling of internal verification effort based on confidence

Reduced certification costs due to inherent quality and verification

Improved ROI of AI due to easy integration, strong performance and reduced verifcation cost of AI+

03.07.2025

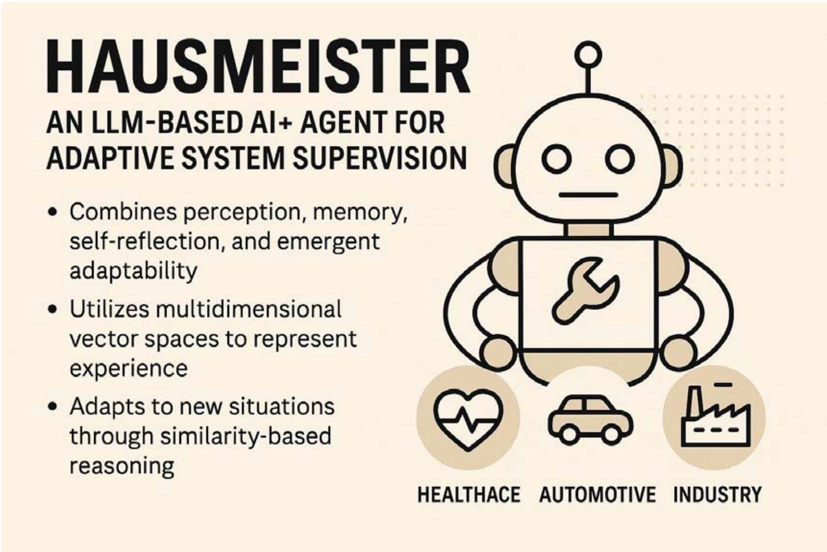
 17


QuantumCell Diagnostic: the way to digital (highly automated) cancer diagnosis:
Data space architecture for hybrid realtime data, logistic data, …
Digitaler Twin
Cloud
Data spaces
Funktionales Logging #Anzahl Bilder
LifeScan User
Immersive productivity assistance based on AI+
remote
LifeScan home
Confidential
X0.3.2
Laboratory
B1
B2
B3
C1
C2
CellScanCube # nnn
Buildings blocks
Automatisation
A
A B C structure
Biologic
B
B1 B2 B3 B4
Camera
C
C1 C2

EXAMPLE APPLICATION: HAUSMEISTER

Agent for Adaptive System Supervision

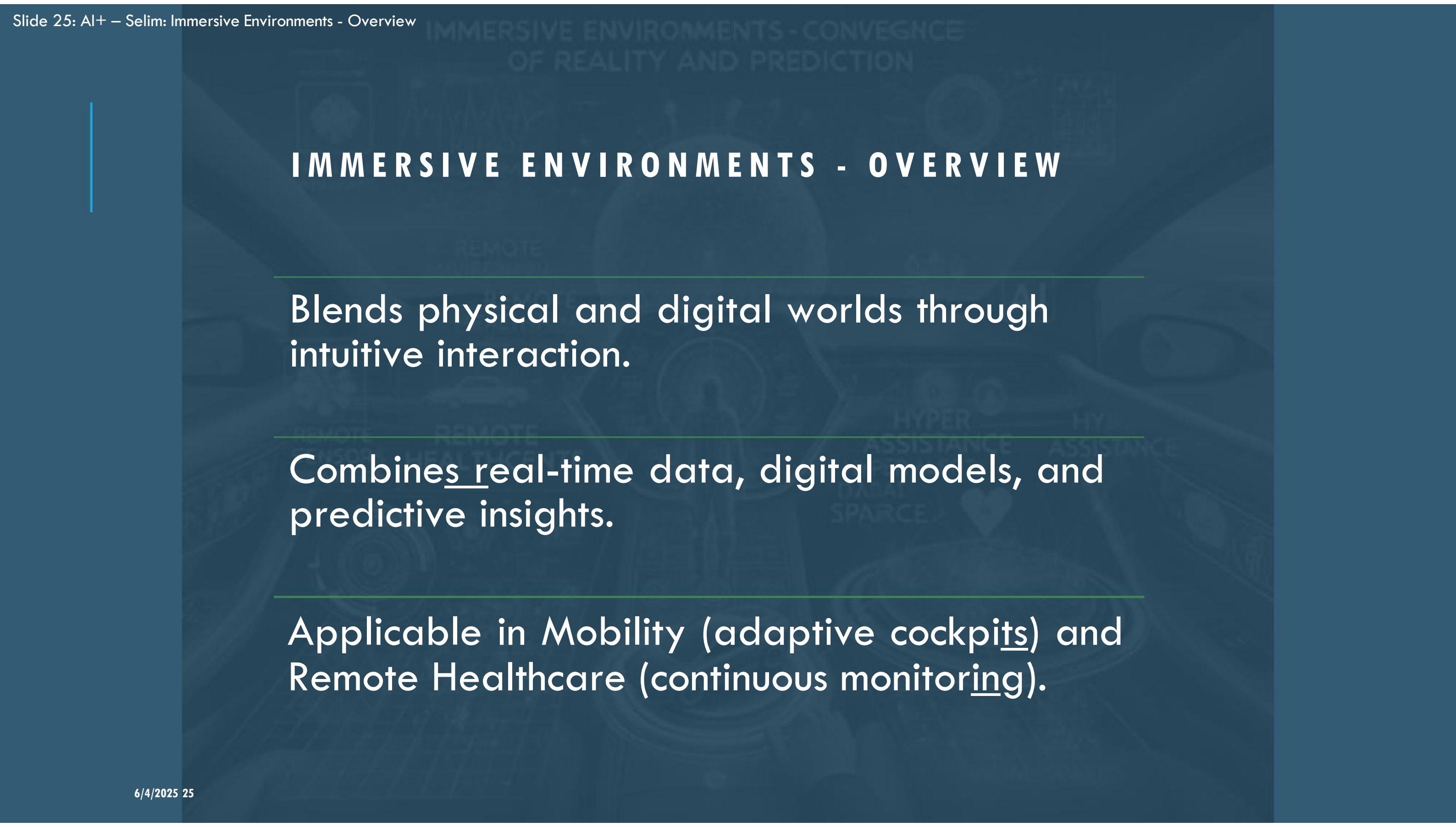
- Constantly monitors streams of inputs
- Utilizes multi-dimensional vector spaces
  - Implict memory and generalization
- Adaptive reasoning that
  - Adapts to unseen data by linking to most similar prior experiences
  - Integrates with and triggers multiple actuators
- Evaluates and improves own performance



03.07.2025

 19


Human intuitive decision making in complex context situations
Centralised decision making?
The big challenge for AI
Human driver are capable to manage extremely difficult weather conditions based on experience, context and foreseen scenarios.
How to drive this car automatically ?


Slide 25: AI+ – Selim: Immersive Environments - Overview
IMMERSIVE ENVIRONMENTS - OVERVIEW
Blends physical and digital worlds through intuitive interaction.
Combines real-time data, digital models, and predictive insights.
Applicable in Mobility (adaptive cockpits) and Remote Healthcare (continuous monitoring).
6/4/2025 25

Slide 26: AI+ – Selim: Core Technologies

CORE TECHNOLOGIES

Real-Time Data Fusion: Context-aware adaptive interfaces.

Predictive Intelligence: Combines sensor data, digital twins, and AI forecasting.

Multimodal Interaction: Voice, gesture, haptics, XR projections.

6/4/2025 26

Slide 32: AI+ – Selim: SURPRISE4EU – SC3 Demonstrator Highlights

SURPRISE4EU DEMONSTRATOR HIGHLIGHTS

Demonstrator Remote Fleet Supervision

- Real-time immersive interface for remote operators
- 360° live video, predictive overlays, XR-based situational awareness

Demonstrator Immersive In-Cabin Coaching

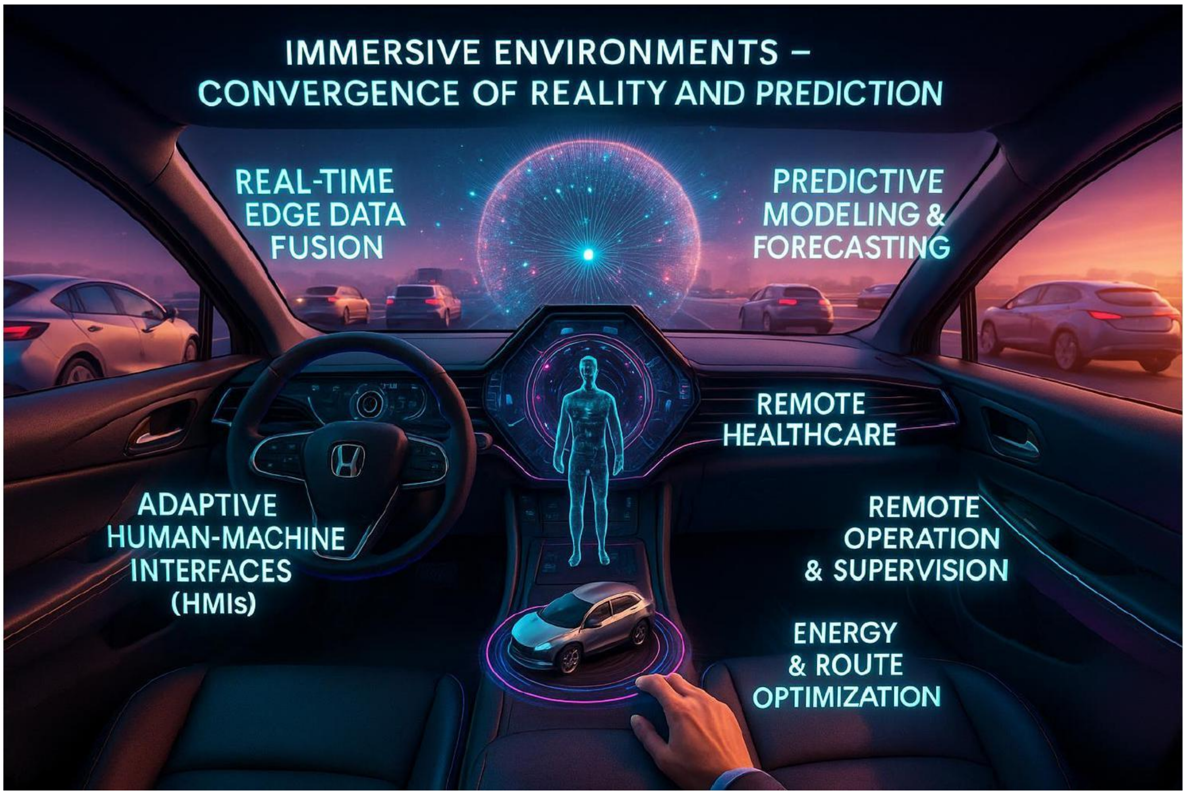
- Fatigue detection, posture correction, and voice-based vehicle status
- On-device LLM for natural language interaction

Demonstrator Smart Co-pilot with ADAS

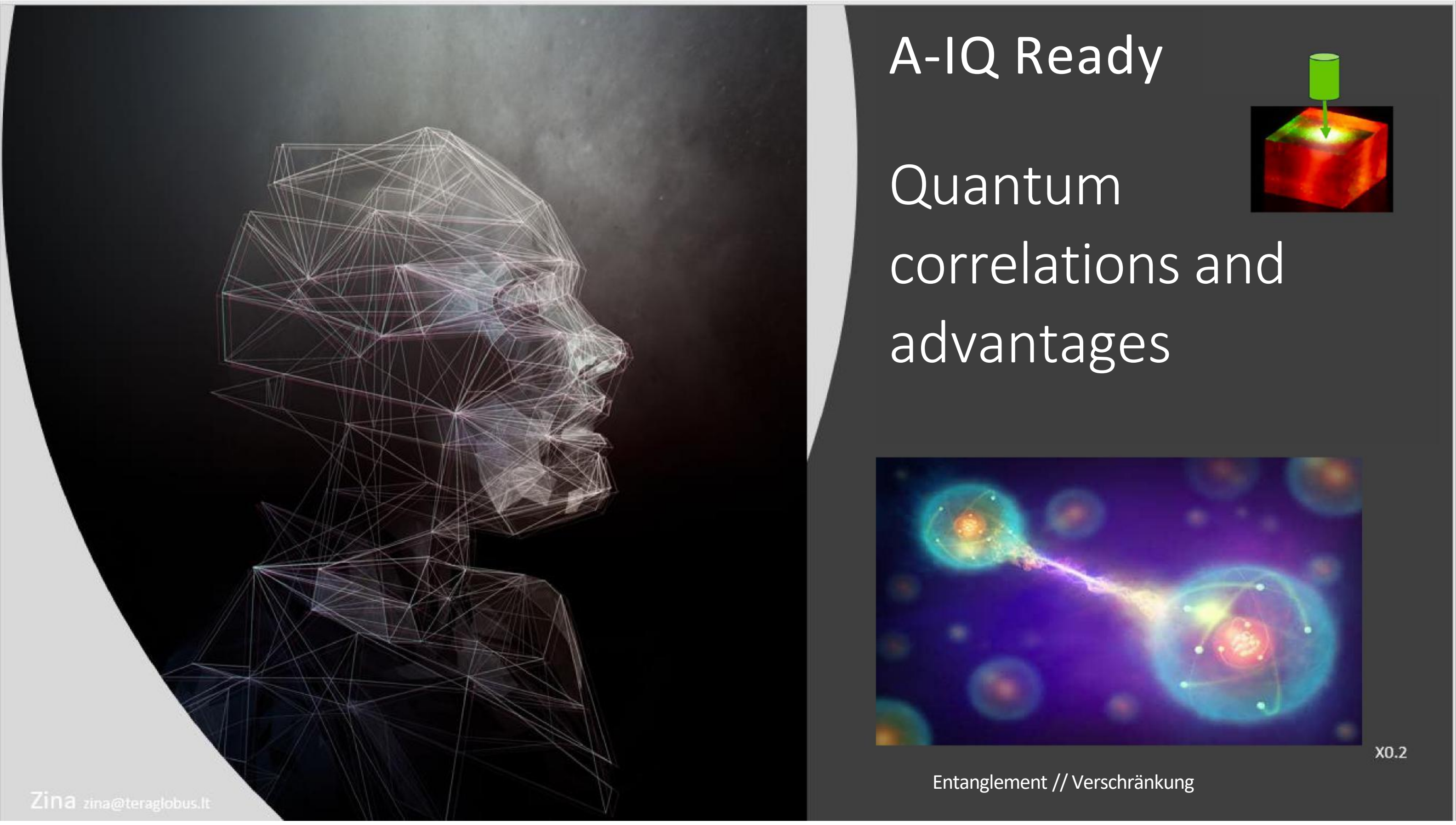
- Driver and occupant monitoring + external perception fusion
- Context-aware intervention and "Remedy Hub"

6/4/2025 32

Slide 33; AI+ – Selim; Immersive environments convergence of reality and prediction

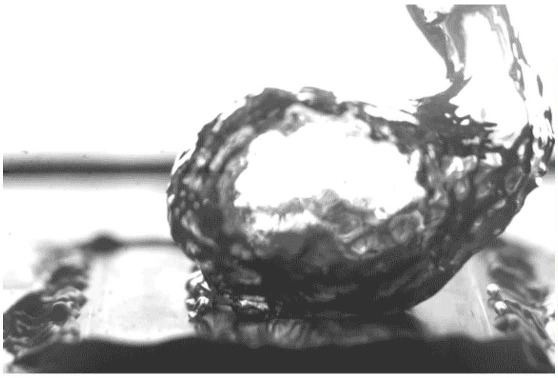


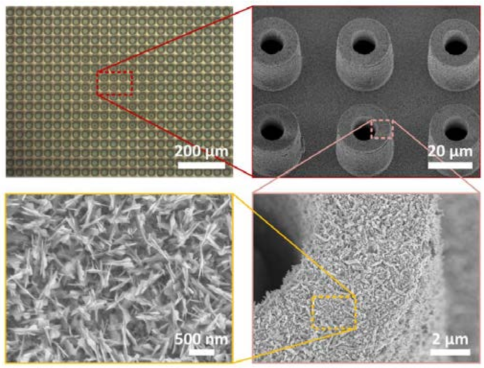
6/4/2025 33


A-IQ Ready
Quantum correlations and advantages
X0.2
Entanglement // Verschränkung
Zina zina@teraglobus.lt

A tiny world

Beispiel: Steigerung der Energieeffizienz des Wasserkochers





Exploring the micro- and nano cosmos

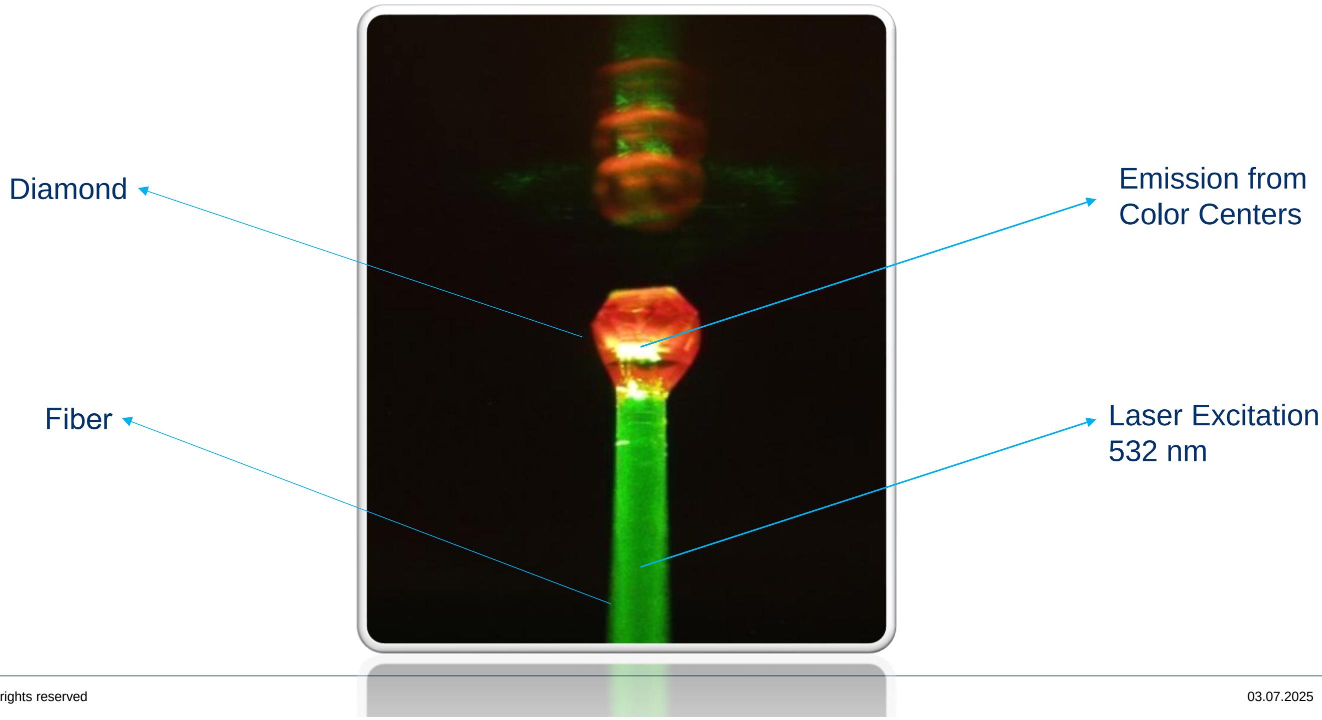
MIT engineers design surfaces that make water boil more efficiently
Systems used in many industries could save energy through these new surface treatments.
David L. Chandler | MIT News Office
Publication Date: July 12, 2022

The new findings are described in the journal *Advanced Materials* in a paper by recent MIT graduate Youngsup Song PhD '21, Ford Professor of Engineering Evelyn Wang, and four others at MIT.

The Nitrogen-Vacancy Center in Diamond

Quantum Systems

AQuT. Applied Quantum Technologies



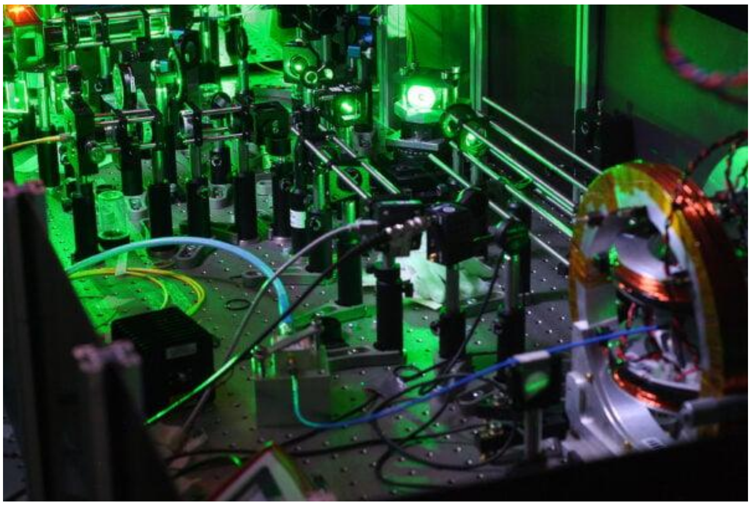
Technological Realization of a Quantum Sensor

Research Laboratory Realization of a Quantum Sensor

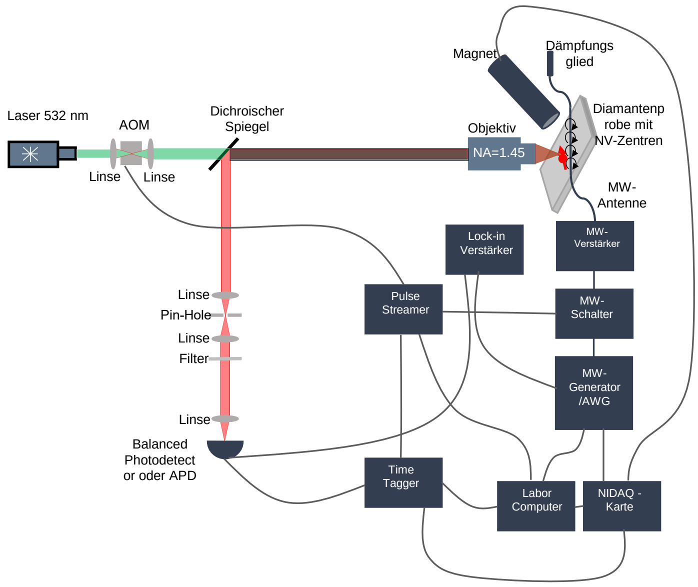
AQuT. Applied Quantum Technologies

FAU

Research - Quantum Sensor in a Laboratory



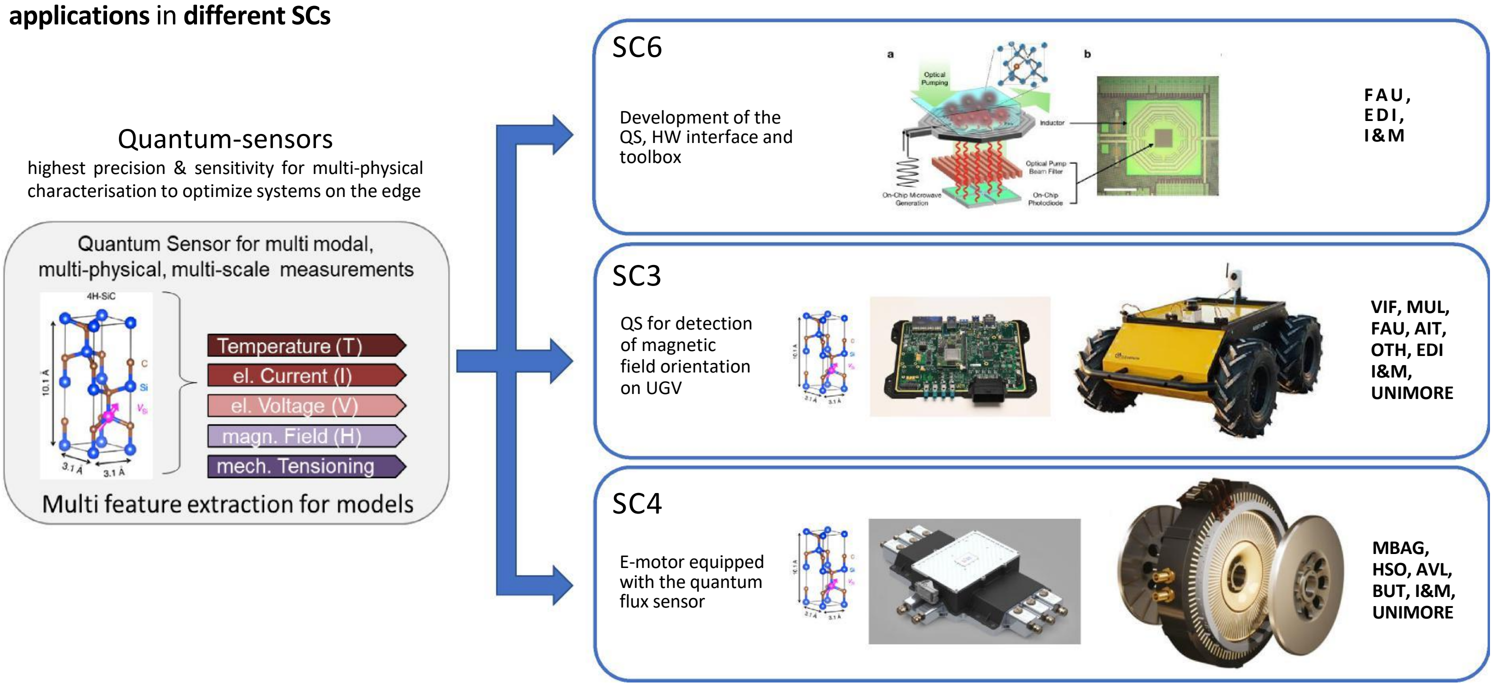
Sketch of the Required Components



AI-READY: enabling different application through quantum sensor technology (1/4)

The proposal is to **connect SC3, SC4 and SC6** by **leveraging on the quantum sensor technology**.

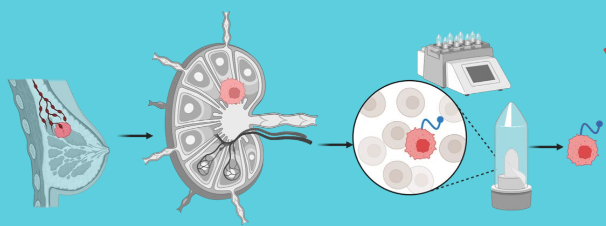
In fact **the same QS technology** will be **integrated** into **different processing platforms** thus enabling **different applications** in **different SCs**

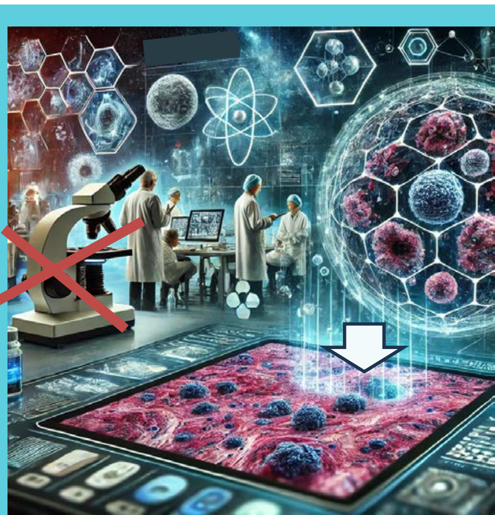


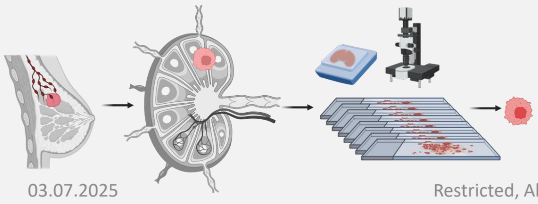
Das Problem heute – Unsere Lösung

B

Heute: Manuelle Diagnose durch Pathologen (Goldstandard)
- Tumorzellen werden unter dem Mikroskop **von Hand gesucht**
- Die Untersuchung dauert oft mehrere Stunden
- Nur **kleine Stichproben** werden geprüft – viele Zellen bleiben unsichtbar
- Die Erkennung hängt stark vom **Zellanteil und der Erfahrung des Arztes** ab
- **Langsam, teuer, human experience major factor**

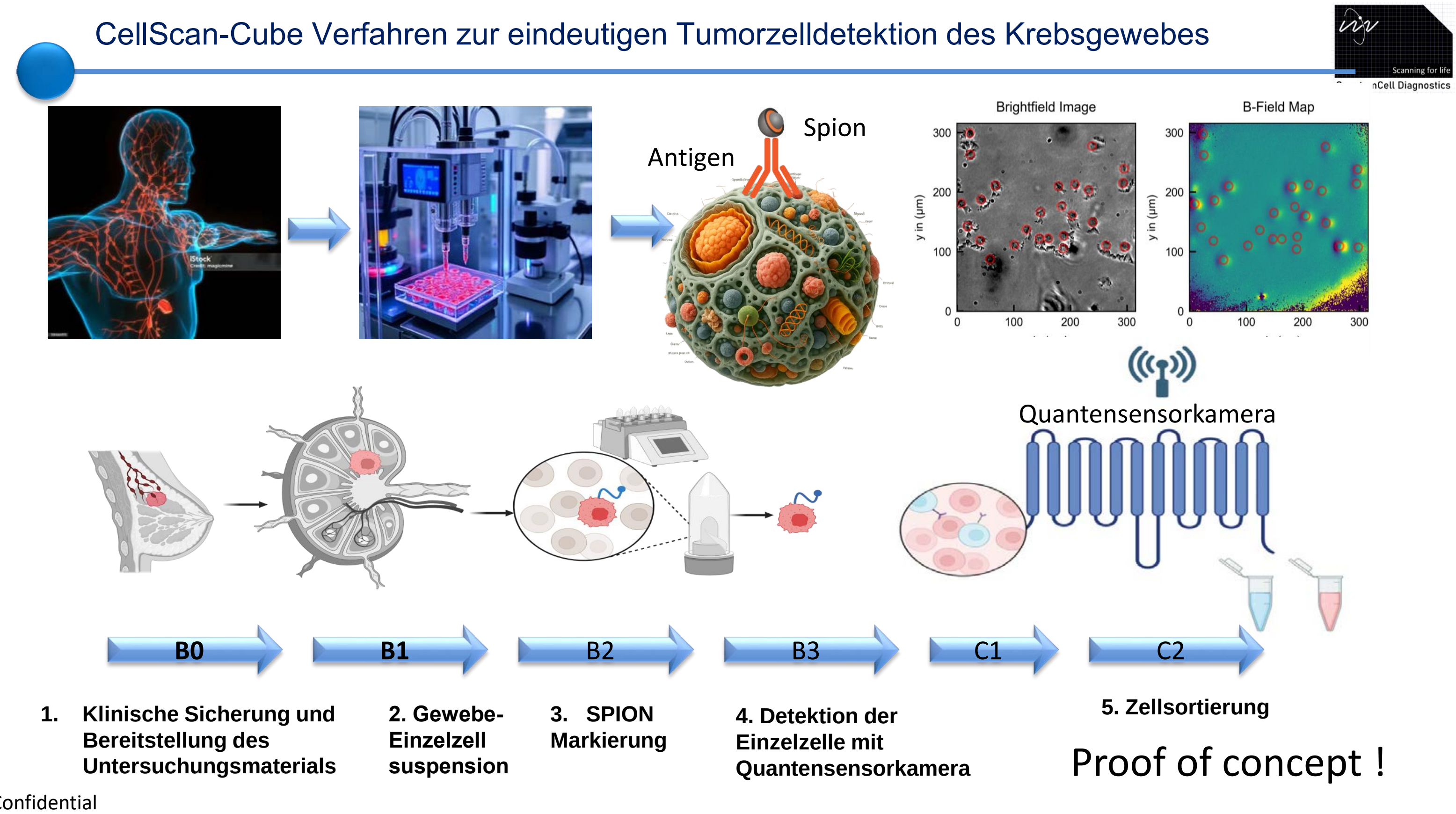






Morgen: Automatisierte Erkennung mit QuantumCell Diagnostic (QCS)
- Zellen aus dem Lymphknoten werden **vereinzelt und mit Nanopartikeln markiert**
- Eine **Quantenkamera scannt** die gesamte Zellschicht magneto-optisch
- Eine **KI erkennt und zählt** automatisch die Krebszellen
- **Schnell, objektiv, vollständig – kein Zufall, kein Rätsel**

03.07.2025 31


CellScan-Cube Verfahren zur eindeutigen Tumorzelldetektion des Krebsgewebes
Scanning for life
Spion
Antigen
Brightfield Image
B-Field Map
y in (µm)
0
100
200
300
Quantensensorkamera
B0
B1
B2
B3
C1
C2
1. Klinische Sicherung und Bereitstellung des Untersuchungsmaterials
2. Gewebe-Einzelzell suspension
3. SPION Markierung
4. Detektion der Einzelzelle mit Quantensensorkamera
5. Zellsortierung
Proof of concept !
Confidential

A-IQ Ready receives funding within the Chips Joint Undertaking (Chips JU) - the Public-Private Partnership for research, development and innovation under Horizon Europe – and National Authorities under grant agreement n° 101096658.
ARCHIMEDES is supported by the Chips Joint Undertaking and its members, including the top-up funding by National Authorities under Grant Agreement No 101112295.
Cynergy4MIE is supported by the Chips Joint Undertaking and its members, including the top-up funding by National Authorities under Grant Agreement No 101140226.

Co-funded by the European Union

Keynote
Future Technologies for Europe: AI+, Immersive Systems & Quantum Sensing

Insights from A-IQ Ready, Archimedes, Cynergy4MIE, Surprise4EU

Reiner John, Florian Lorber, Selim Solmaz

Summer School CPS&IoT 2025, Montenegro

Table of Contents

Industrial view on challenges and opportunities of Europe`s automotive industry

AVL

Strategic Goals:

1. **Retaining Europe's Position:** keeping Europe as a prime location for car production amidst global competition. Foster competitive advantages by technology leadership in safety critical systems
2. **Converging Ecosystems:** integrating mobility solutions with infrastructure and energy systems (MIE) to create a seamless ecosystem in order to boost economy of scale, seamless technology stack integration –> Autonomy beyond automotive – areas of synergy
3. ~~**Green Deal Objectives**~~**:** balancing growth and sustainability by focusing on decarbonization through electrification, while mitigating the decline in direct investments -> growth
4. **Addressing the competition -> Gadget Trend:** build cars for the markets -> responding to consumer demand for increasingly tech-integrated vehicles.

Technological Key Challenges:

1. **ROI of AI (cost, consumption, architectures**
2. **Energy Efficiency:** optimizing energy usage in electric and hybrid vehicles for the ecosystems MIE
3. **Safety and Data Privacy:** Maintaining rigorous safety standards while managing data privacy concerns.
4. **Complex System Integration:** Ensuring seamless integration of steadily increasing high performant and complex components within the vehicle architecture.
5. **Scalability and Regulation Compliance for safety critical systems:** Developing scalable solutions that comply with evolving regulations.

Technological Competitive Analysis: Europe's Rivals

USA:
- Focus: Commercialization, market dominance
- Technologies: AI, cloud, biotechnology

China:
- Focus: State-driven innovation
- Technologies: 5G/6G, renewable energy, semiconductors

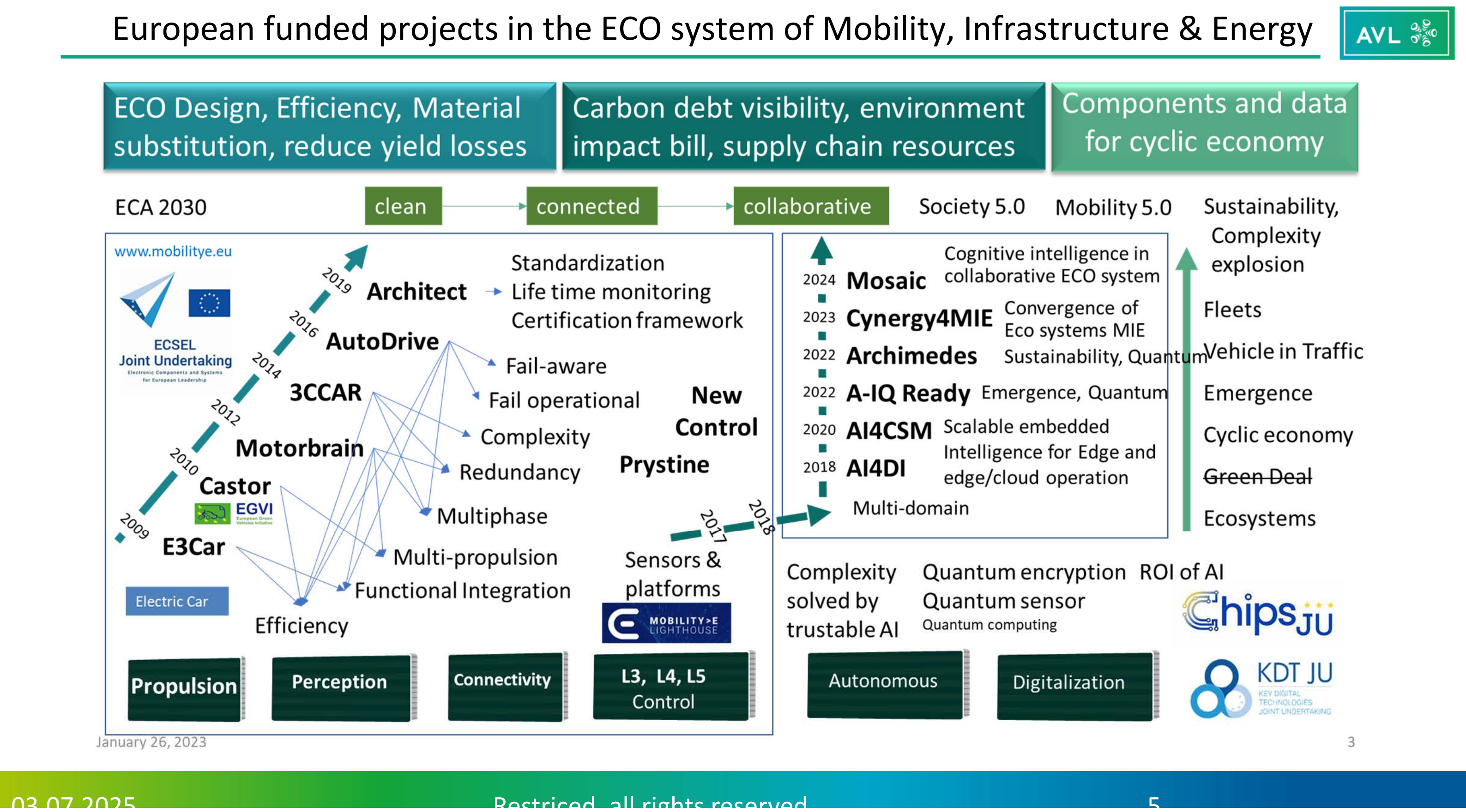
Japan/Korea:
- Focus: Precision, sustainability
- Technologies: Robotics, semiconductors, electromobility

USA:
- Personalization, ADAS features
- Entertainment systems (AR/VR)

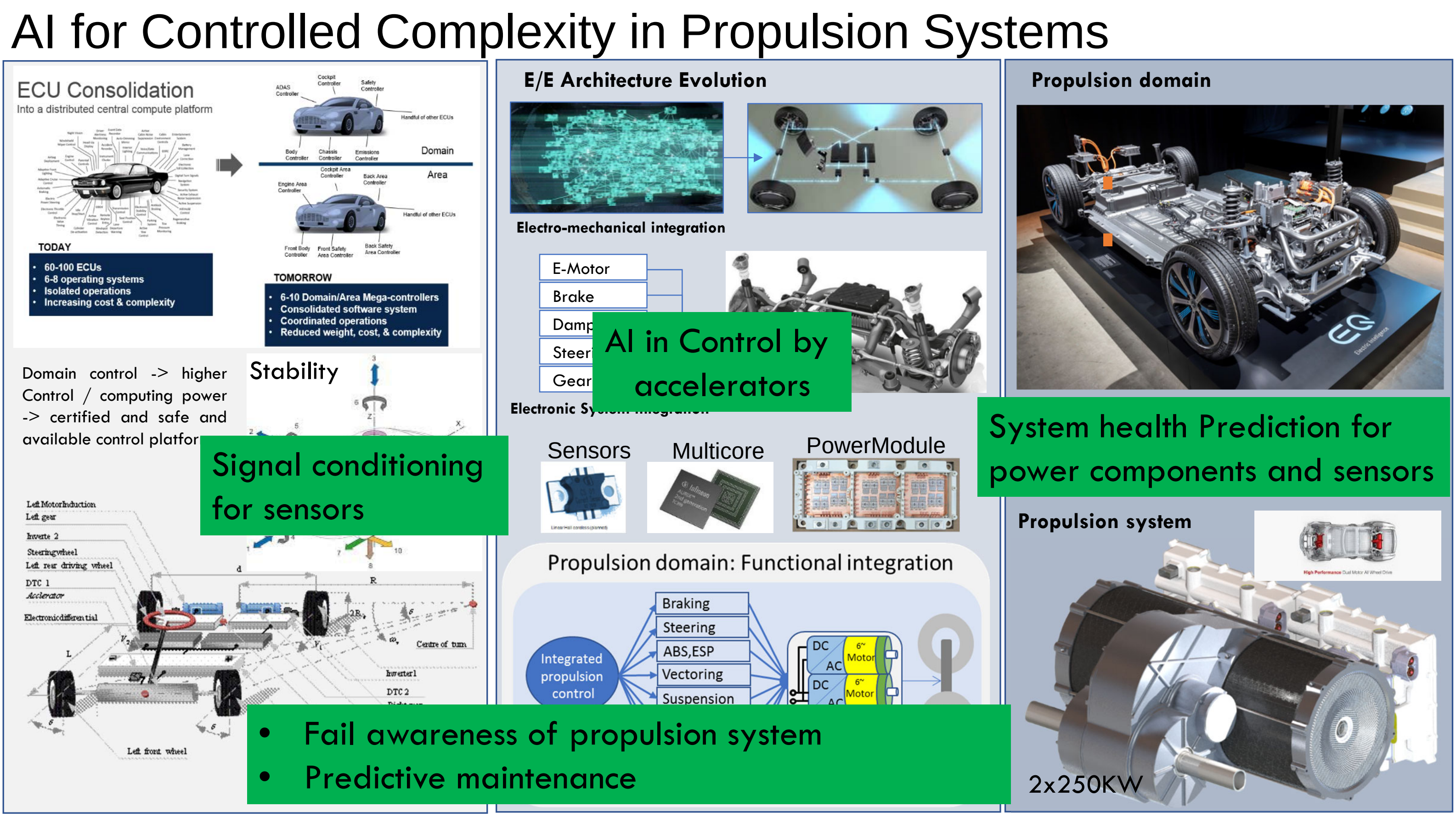
China:
- Cost-effective solutions
- 5G/6G integration

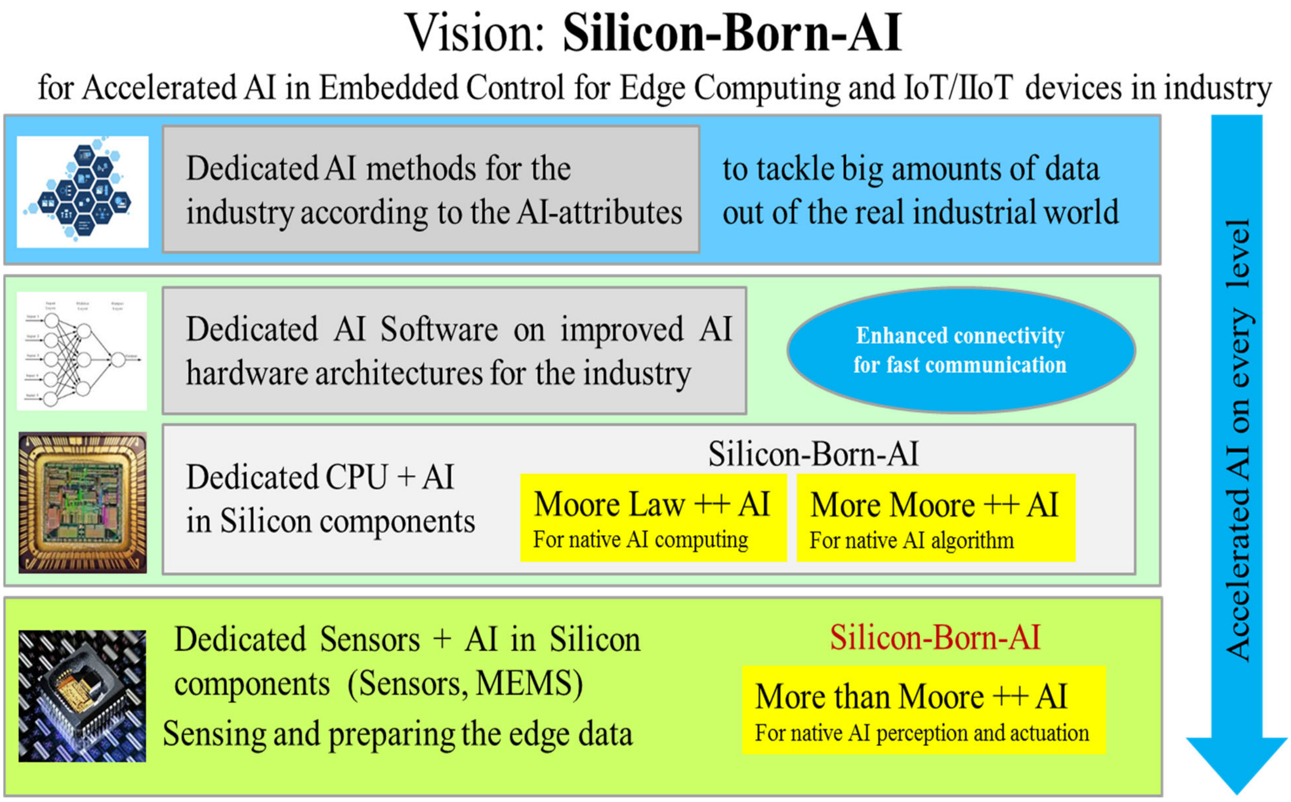
Japan/Korea:
- Precise, reliable technologies
- Sustainable, compact solutions

- Common requirements: cybersecurity, user-centricity, standardization



European funded projects in the ECO system of Mobility, Infrastructure & Energy
AVL
ECO Design, Efficiency, Material substitution, reduce yield losses
Carbon debt visibility, environment impact bill, supply chain resources
Components and data for cyclic economy
ECA 2030
clean
connected
collaborative
Society 5.0
Mobility 5.0
Sustainability, Complexity explosion
www.mobilitye.eu
ECSEL Joint Undertaking
2009
2010
2012
2014
2016
2019
Architect
AutoDrive
3CCAR
Motorbrain
Castor
EGVI
E3Car
Electric Car
Standardization
Life time monitoring
Certification framework
Fail-aware
Fail operational
Complexity
Redundancy
Multiphase
Multi-propulsion
Functional Integration
Efficiency
New Control
Prystine
2017
2018
Sensors & platforms
MOBILITY>E LIGHTHOUSE
2024 Mosaic
Cognitive intelligence in collaborative ECO system
2023 Cynergy4MIE
Convergence of Eco systems MIE
2022 Archimedes
Sustainability, Quantum
2022 A-IQ Ready
Emergence, Quantum
2020 AI4CSM
Scalable embedded Intelligence for Edge and edge/cloud operation
2018 AI4DI
Multi-domain
Fleets
Vehicle in Traffic
Emergence
Cyclic economy
Green Deal
Ecosystems
Complexity solved by trustable AI
Quantum encryption
Quantum sensor
Quantum computing
ROI of AI
Chips JU
KDT JU
Propulsion
Perception
Connectivity
L3, L4, L5 Control
Autonomous
Digitalization
January 26, 2023
3
03.07.2025
Restricted, all rights reserved
5


AI for Controlled Complexity in Propulsion Systems
ECU Consolidation
Into a distributed central compute platform
TODAY
60-100 ECUs
6-8 operating systems
Isolated operations
Increasing cost & complexity
TOMORROW
6-10 Domain/Area Mega-controllers
Consolidated software system
Coordinated operations
Reduced weight, cost, & complexity
Domain
Area
Domain control -> higher Control / computing power -> certified and safe and available control platfor
Stability
Signal conditioning for sensors
E/E Architecture Evolution
Electro-mechanical integration
E-Motor
Brake
AI in Control by accelerators
Sensors
Multicore
PowerModule
Propulsion domain: Functional integration
Braking
Steering
ABS,ESP
Vectoring
Suspension
Integrated propulsion control
Fail awareness of propulsion system
Predictive maintenance
Propulsion domain
System health Prediction for power components and sensors
Propulsion system
2x250KW


Vision: Silicon-Born-AI
for Accelerated AI in Embedded Control for Edge Computing and IoT/IIoT devices in industry
Dedicated AI methods for the industry according to the AI-attributes
to tackle big amounts of data out of the real industrial world
Dedicated AI Software on improved AI hardware architectures for the industry
Enhanced connectivity for fast communication
Dedicated CPU + AI in Silicon components
Silicon-Born-AI
Moore Law ++ AI
For native AI computing
More Moore ++ AI
For native AI algorithm
Dedicated Sensors + AI in Silicon components (Sensors, MEMS)
Sensing and preparing the edge data
Silicon-Born-AI
More than Moore ++ AI
For native AI perception and actuation
Accelerated AI on every level

Abstract *Future Technologies for Europe*

AVL

- This presentation introduces core technologies shaping Europe's future in **AI+**, **immersive interaction**, and **quantum sensing**.
- It builds on insights from **Surprise4EU**, **A-IQ Ready**, **Archimedes**, and **Cynergy4MIE**, showcasing how technology stacks—from **foundational** to **application layer**—enable impactful use cases in **health**, **mobility**, and **infrastructure**.
- The session emphasizes the urgency of **technological sovereignty** and highlights the active role that students and researchers can play in developing a **responsible, explainable, and effective European tech landscape**.
- Supports Europe's **strategic autonomy** and **digital sovereignty**
  Enables **cross-domain integration** of AI, IoT, and quantum technologies

6/4/2025 3

Introduction & Motivation: *Why This Matters for Europe*

AVL

- Urgency of technological sovereignty in Europe
- Importance for students to engage now
- Brief intro to Chips JU initiative

Importance & Strategic Relevance

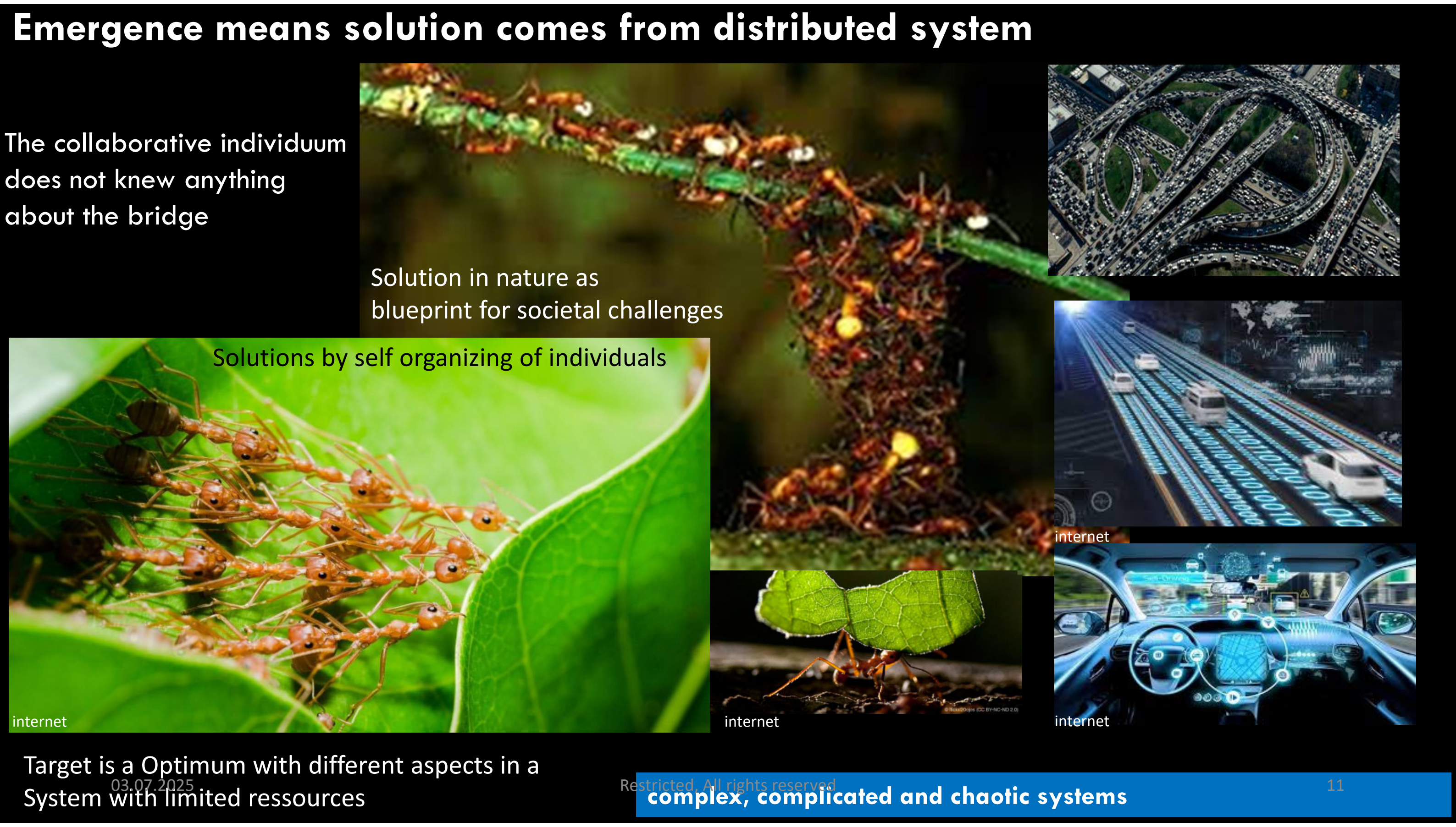
1. Solves **AI verification and certification bottlenecks** with self-aware architectures
2. Enhances **healthcare diagnostics** via non-invasive sensor-based solutions
3. Promotes **safe and sustainable mobility** using immersive environments
4. Strengthens **infrastructure resilience** through AI-powered grid & depot systems
5. Integrates **platforms and edge computing** for low-latency, real-time applications
6. Fosters **ethical, human-centric AI** and long-term sustainability
7. Links **academic research** directly with **industry use cases**
8. Empowers the **next generation of European innovators and technologists**

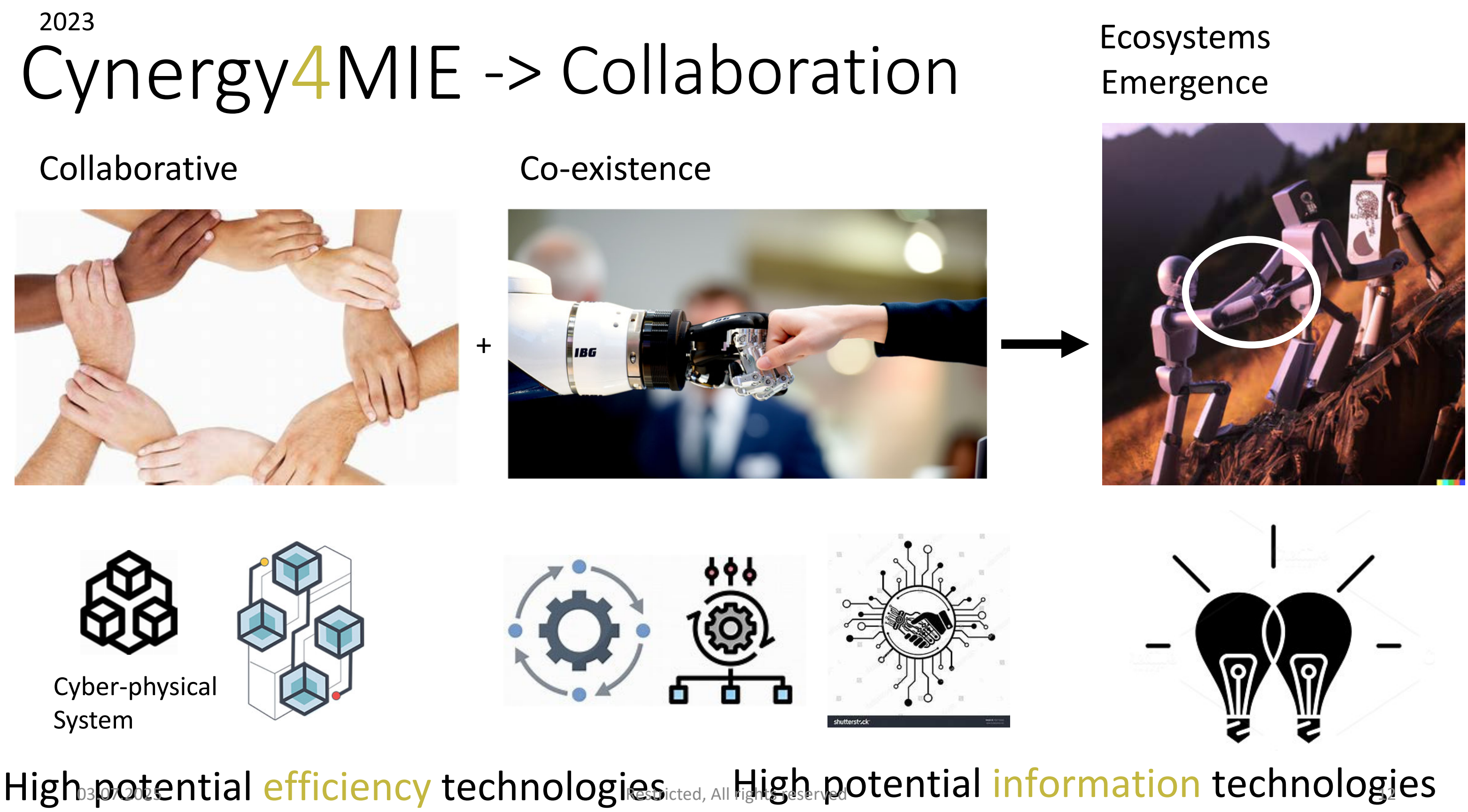
6/4/202 5 4

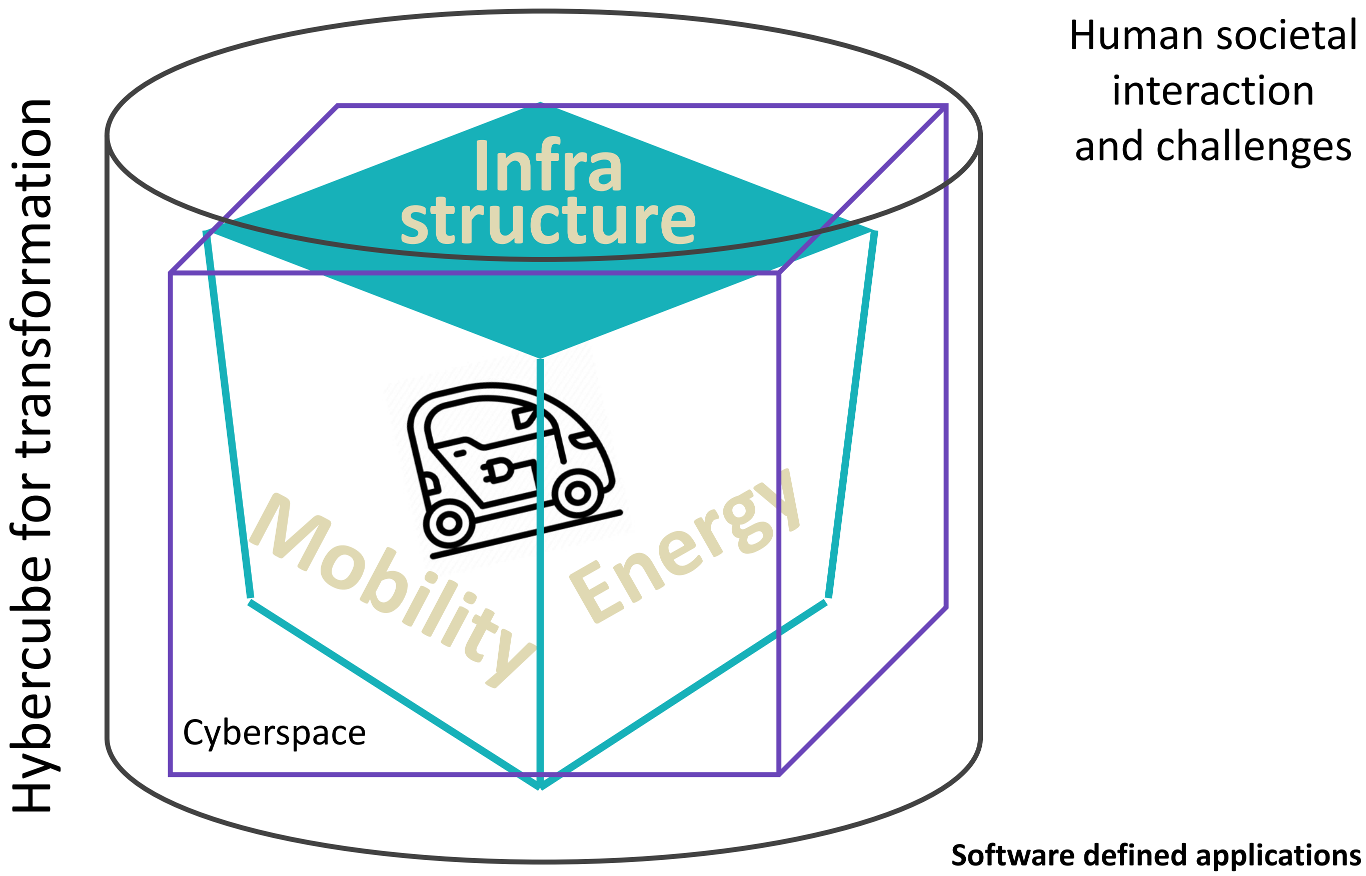
AI+ and Self-Aware AI Systems

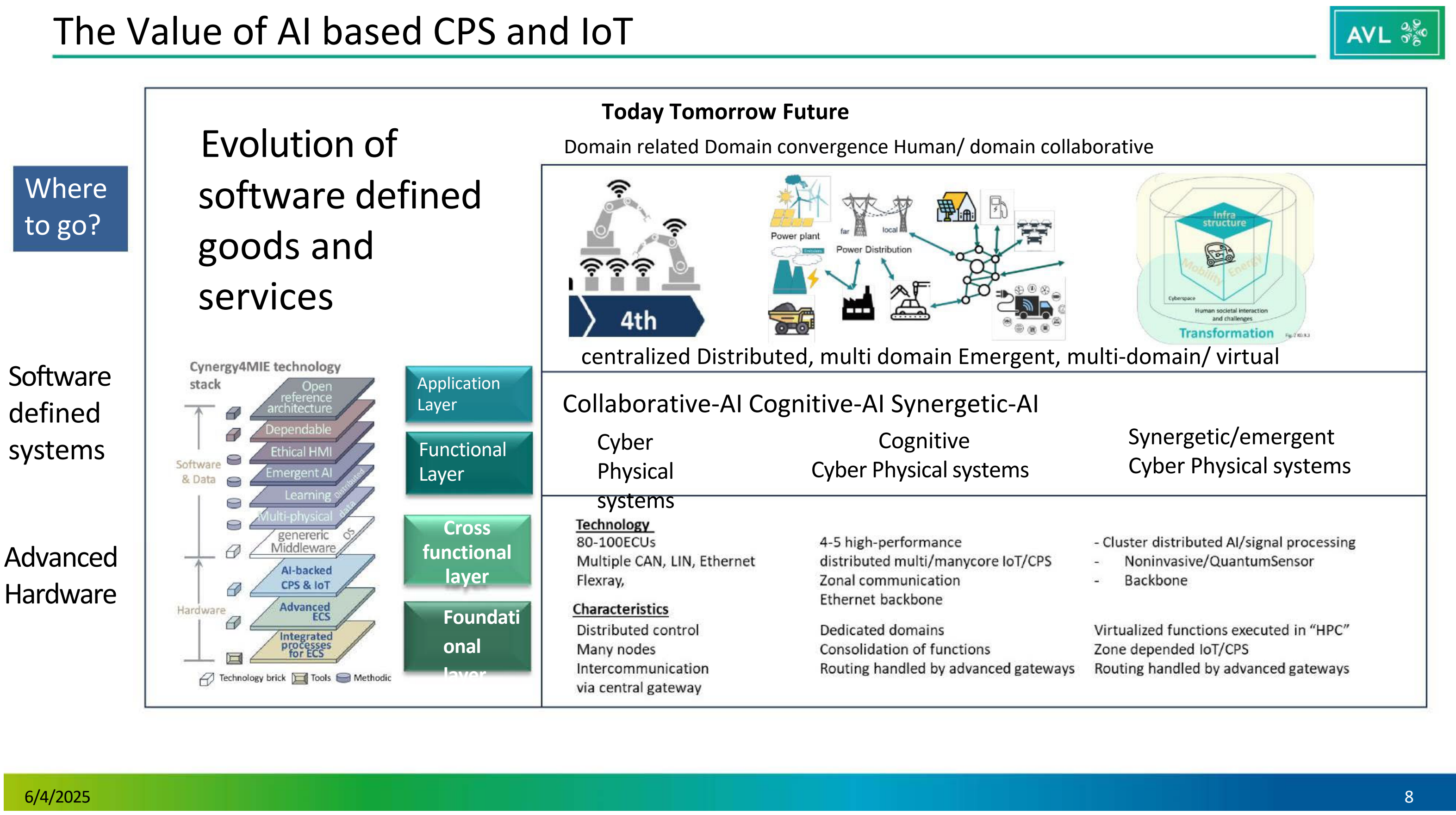
- **Technology stack Layers : functional , application: AI+ and Self-Aware AI Systems**
- **Technology Layers: Foundational, Cross-functional**
- **Key Concepts: Self-awareness, resilience, explainability**
- **Impact: Efficiency, lower verification costs**

6/4/2025 5

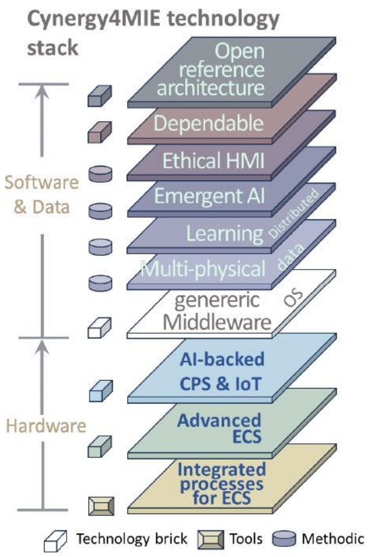

Emergence means solution comes from distributed system
The collaborative individuum does not knew anything about the bridge
Solution in nature as blueprint for societal challenges
Solutions by self organizing of individuals
internet
internet
internet
internet
Target is a Optimum with different aspects in a System with limited ressources
03.07.2025
Restricted. All rights reserved
complex, complicated and chaotic systems
11

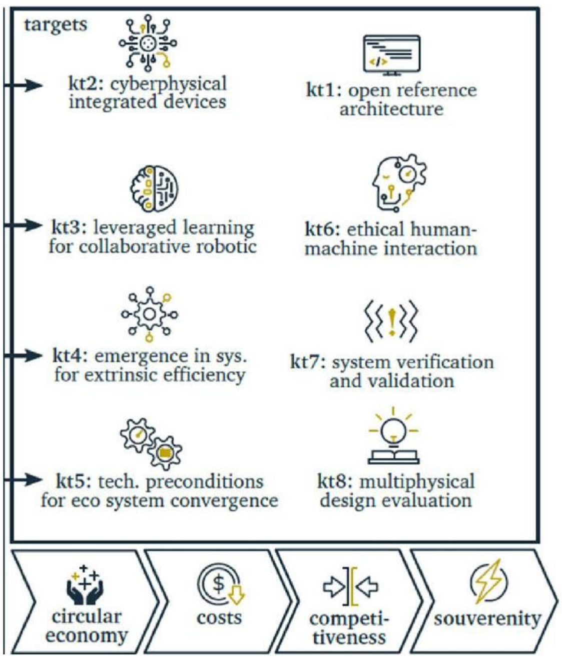

2023
Cynergy4MIE -> Collaboration
Ecosystems
Emergence
Collaborative
Co-existence
+
Cyber-physical
System
High potential efficiency technologies
High potential information technologies

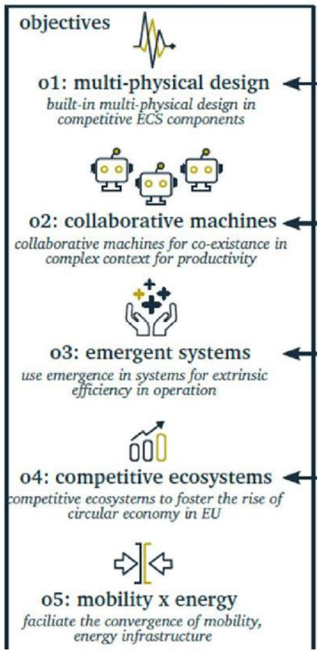

Human societal interaction and challenges
Infra structure
Mobility
Energy
Cyberspace
Hybercube for transformation
Software defined applications


The Value of AI based CPS and IoT
AVL
Where to go?
Software defined systems
Advanced Hardware
Evolution of software defined goods and services
Today Tomorrow Future
Domain related Domain convergence Human/ domain collaborative
4th
Power plant
Power Distribution
far
local
Transformation
centralized Distributed, multi domain Emergent, multi-domain/ virtual
Cynergy4MIE technology stack
Open reference architecture
Dependable
Ethical HMI
Emergent AI
Learning
Multi-physical
genereric Middleware
AI-backed CPS & IoT
Advanced ECS
Integrated processes for ECS
Software & Data
Hardware
Technology brick
Tools
Methodic
Application Layer
Functional Layer
Cross functional layer
Foundational layer
Collaborative-AI Cognitive-AI Synergetic-AI
Cyber Physical systems
Cognitive Cyber Physical systems
Synergetic/emergent Cyber Physical systems
Technology
80-100ECUs
Multiple CAN, LIN, Ethernet
Flexray,
4-5 high-performance
distributed multi/manycore IoT/CPS
Zonal communication
Ethernet backbone
- Cluster distributed AI/signal processing
- Noninvasive/QuantumSensor
- Backbone
Characteristics
Distributed control
Many nodes
Intercommunication
via central gateway
Dedicated domains
Consolidation of functions
Routing handled by advanced gateways
Virtualized functions executed in “HPC”
Zone depended IoT/CPS
Routing handled by advanced gateways
6/4/2025
8

Technology stack

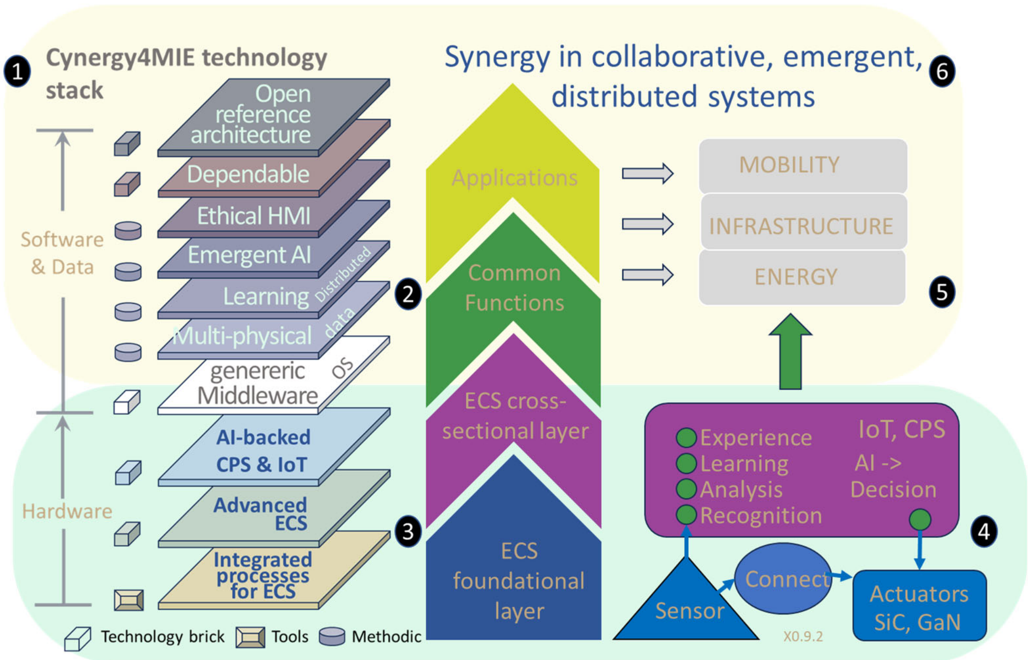






Stack technologies for collaborative, emergent distributed systems

New research trends: Open-source hardware (e.g., RISC-V), open-source software, and open-source AI.



- Microservices and APIs
- Containerization
- Software development practices
- Vehicle service-oriented architecture (SOA)
- E/E Architecture Transformation
- Vehicle Multilayered SOA
- Functional Safety in the Vehicle Multilayered SOA

Cynergy4MIE -> General story line

SYNERGY

1 + 1 > 2

Logo: t.b.d

Synergies for the synthesis of eco systems

Cyberphysical systems as basis for the convergence of the ecosystem mobility, infrastructure and energy

High potential information technologies

1. Perfect quantum correlations have double information then perfect classical correlation
2. Distributed systems enables Emergence, which means 1+1 is more then 2

High potential efficiency technologies

3. SiC and GaN for highest efficiency and information

Immersive Technologies – Future of Interaction

- Middleware & Functional Layers: Real-time data integration
- Examples: Immersive cockpits, remote healthcare
- Benefits: Enhanced UX, safety, diagnostics

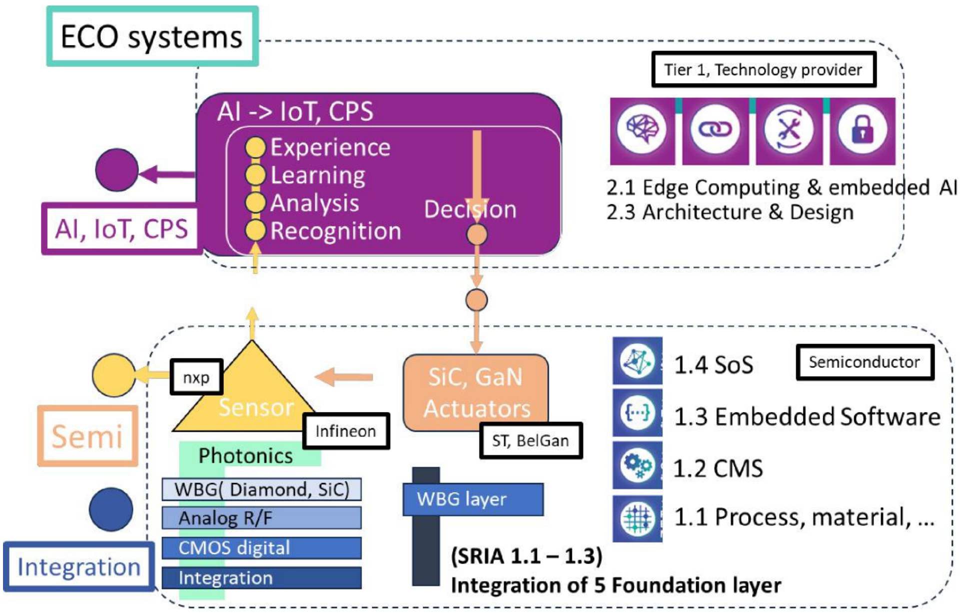
4 Pillars for eco system convergence

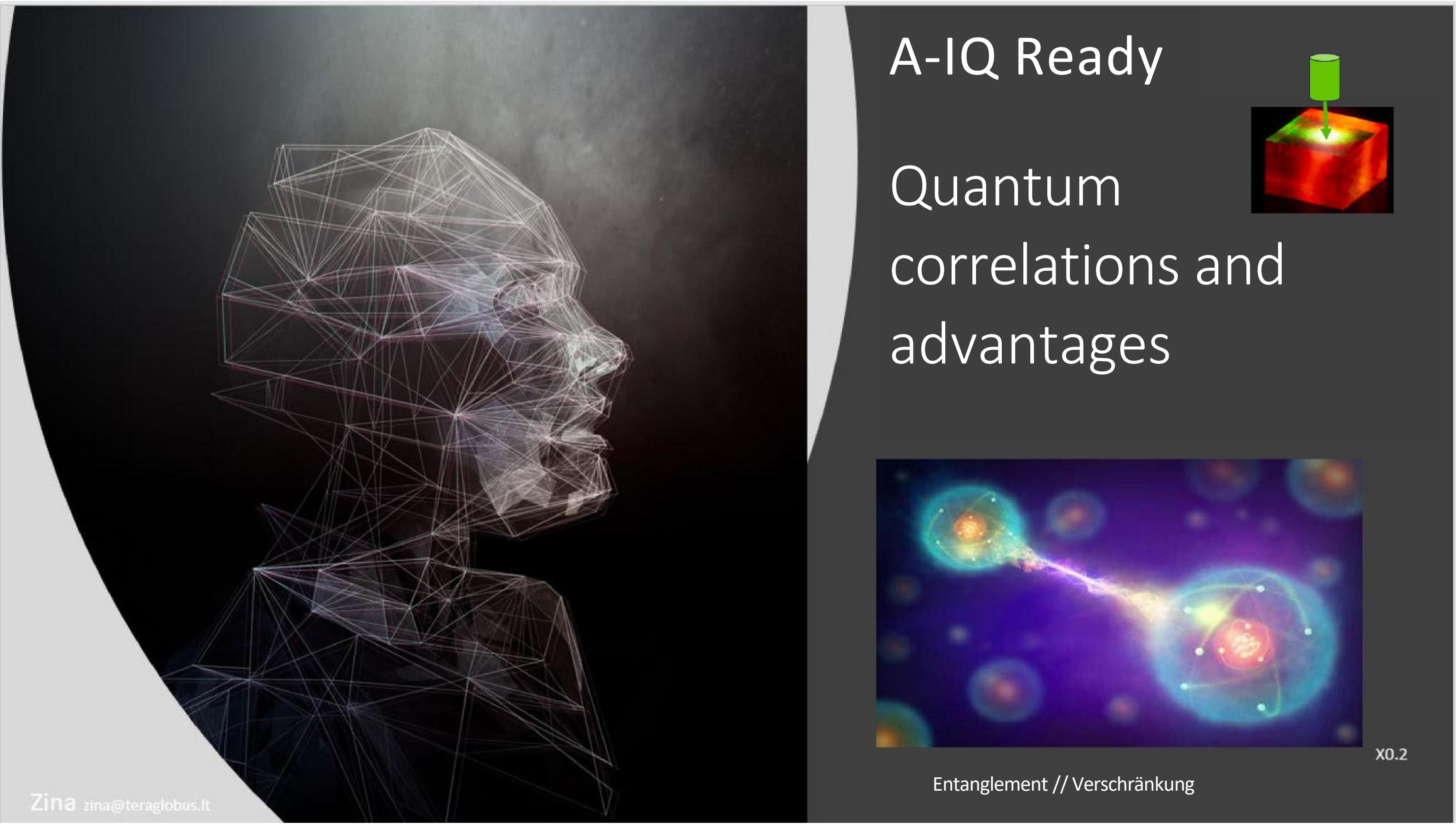
More out of Productivity in software and applications

More out of AI, IoT, CPS for common functions

More out of semiconductors Non-invasive Sensors & Efficiency technologies

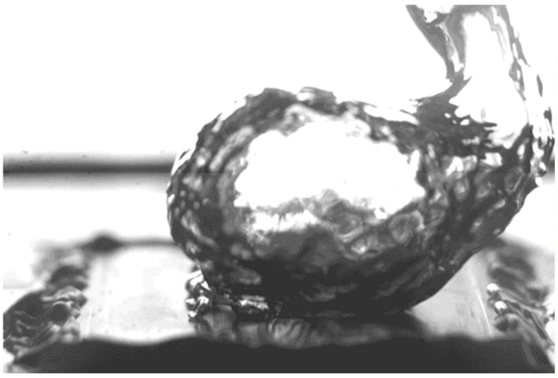
More out of European top technologies by synchronized, synergized More than Moore -> e.g. build quantum

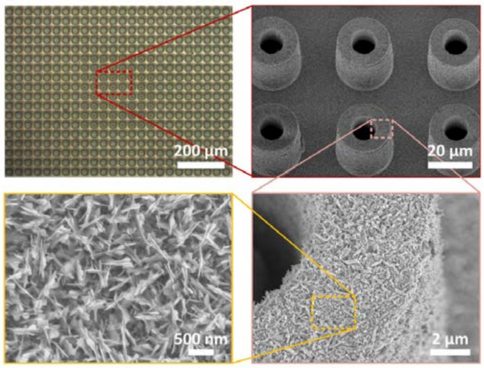




A-IQ Ready
Quantum correlations and advantages
X0.2
Entanglement // Verschränkung
Zina zina@teraglobus.lt

Beispiel: Steigerung der Energieeffizienz des Wasserkochers





Exploring the micro- and nano cosmos

MIT engineers design surfaces that make water boil more efficiently
Systems used in many industries could save energy through these new surface treatments.
David L. Chandler | MIT News Office
Publication Date: July 12, 2022

The new findings are described in the journal *Advanced Materials* in a paper by recent MIT graduate Youngsup Song PhD '21, Ford Professor of Engineering Evelyn Wang, and four others at MIT.

Quantum Technologies

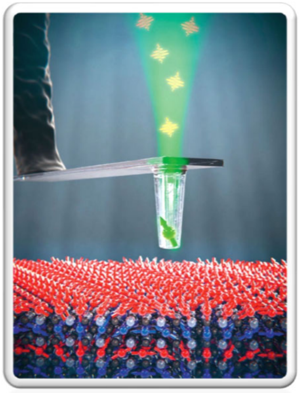
Second Generation of Quantum Technologies

Applied Quantum Technologies

Quantum Sensing

[1,2]

Copyright: Patrick Maletinsky



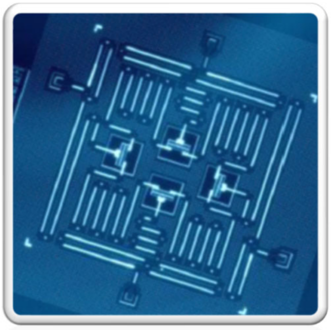
[1] N. Aslam et al., Science (2017) Vol. 357

[2] T. Staudacher et al., Science (2013) Vol. 339

Quantum Computing

[3,4]

Copyright: Digitale Welt



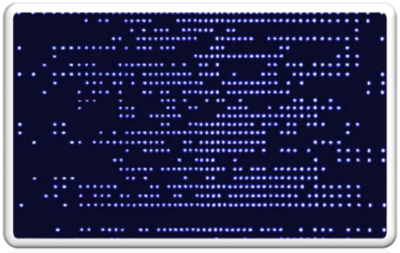
[3] J. M. Gambetta et al., npj Quantum Information (2017) 3:2

[4] A. W. Harrow, Nature (2017), Vol. 549

Quantum Simulation

[5,6]

Copyright: Nature Journal



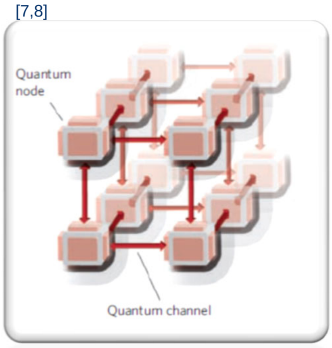
[5] J. Zhang et al., Nature (2017) Vol. 551

[6] H. Bernien et al., Nature (2017) Vol. 551

Quantum Communication

[7,8]

Copyright: Nature Journal



[7] H. J. Kimble, Nature (2008), Vol. 453

[8] G. Waldherr, Nature (2014), Vol. 506

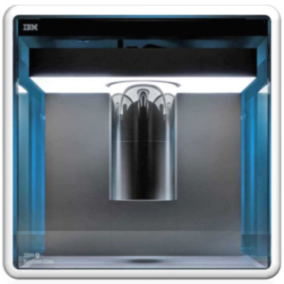
Technological progress

The reason why we are now seeing rapid progress in the development of quantum technologies

Applied Quantum Technologies

Gap in Knowledge

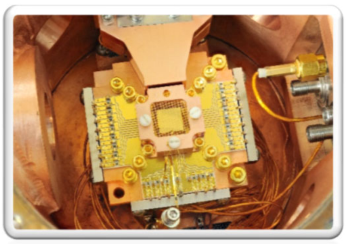
- Suitable quantum systems
- Reproducibility of quantum systems
- Controllability of quantum systems
- Measurement sequences



Copyright: IBM

Interface-Electronics

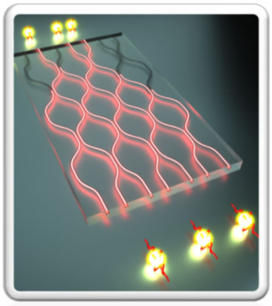
- Single-Photon-Detector
- Superconducting detectors
- Closed cycle cryostation
- Control and readout electronics



Copyright: NIST

Maturity of the Technology

- Material quality
- Fabrication processes
- Generation of quantum systems
- Laser systems
- Cleanroom facility technology (e-beam lithographie,…)



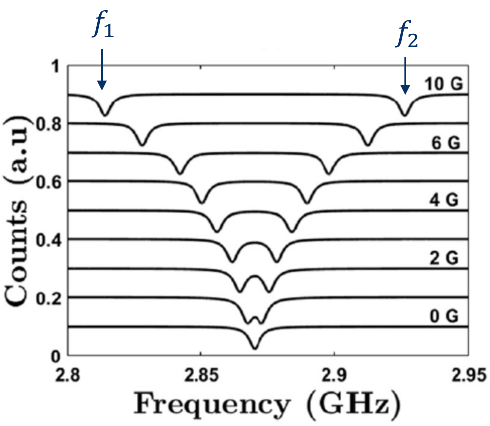
Copyright: INL

Multi Modal Sensing Capabilities

Physical measurement quantities of a quantum sensor

Applied Quantum Technologies

Magnetic Field Measurement



$$\Delta f = f_2 - f_1 = 2\gamma B$$

Gyromagnetic ratio (constant) — Magnetic field

Spin Hamiltonian

$$H = D_{gs}S^2 + \sum_i d_i E + \gamma_e B \cdot S$$

Hamiltonian describes the energy of a state in a quantum mechanical system

Temperature — Electric Field / Pressure — Magnetic Field

[1]

Sensed parameter	Parameter/sensitivity dependence	Reported sensitivities	References
Magnetic field (B)	$\eta_B = \frac{\sigma}{[\partial S/\partial B]_{max}}\sqrt{t_m}$	pT-μT/$\sqrt{\text{Hz}}$	Balasubramanian *et al.* (Ref. 59), Webb *et al.* (Ref. 72), and Neumamm *et al.* (Ref. 73)
Electric field ($\mathcal{E}$)	$\eta_{\mathcal{E}} = \frac{\sigma}{[\partial S/\partial \mathcal{E}]_{max}}\sqrt{t_m}$	≈100 V/cm/ $\sqrt{\text{Hz}}$	Dolde *et al.* (Ref. 70)
Temperature ($\mathcal{T}$)	$\frac{\delta D}{\delta T}, \frac{\delta E}{\delta T}$	10–100 kHz/K, -1.4×10^{-4} Hz/K	Dolde *et al.* (Ref. 74) and Acosta *et al.* (Ref. 71)
Pressure (P)	$\frac{\delta D}{\delta P}$	10^5–10^6 Pa/$\sqrt{\text{Hz}}$	Doherty *et al.* (Ref. 75)

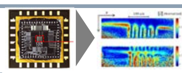
[1] Phila Rembold et al., AVS Quantum Sci. **2**, 024701 (2020)

Quantum Sensing is a Disruptive Technology

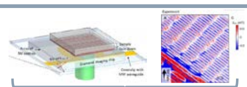
Overview of Applications

Applied Quantum Technologies

Magnetic field imaging on the nanoscale



Visualizing currents for semiconductor FA



Visualizing magnetic properties in materials

Magnetic field sensing in unshielded environments

Future

Detection of neuronal signals during surgery

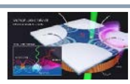
Enables

Proximity alert for:
- safer surgery (e.g. cochlea, neuro mapping)
- robot guided surgery

Nanoscale NMR under environmental conditions

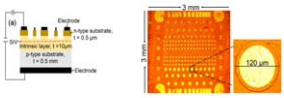
Future

NMR for tissue differentiation



NMR sensor for Lab on a chip applications

Smart Control of Power Devices



Integrated quantum sensor

Enables

Local measurements of:
- temperature
- Current density
- Strain, electric and magnetic field…

Properties

- Calibration free
- Room temperature operation
- 1pT/√Hz to ~fT/√Hz magnetic field sensitivity
- Huge dynamic range
- Multi-Modal sensing capability (magnetic field, temperature, electric field, strain, ph-value, NMR)

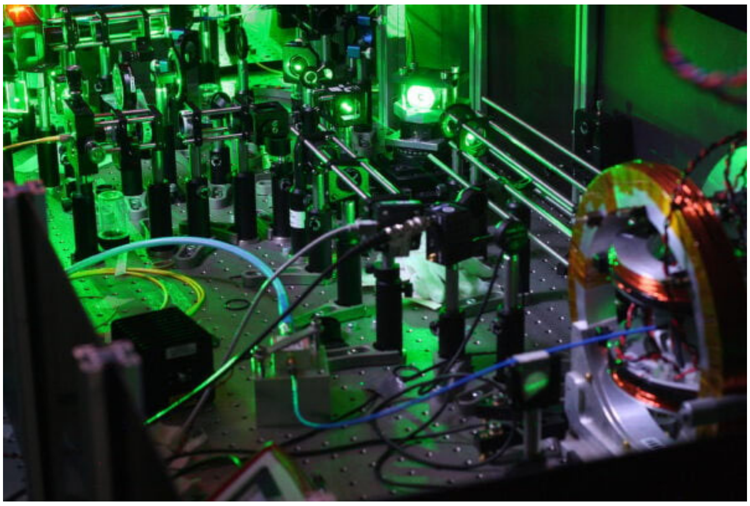
Institute of Applied Quantum Technologies 3. Juli 2025 25

Technological Realization of a Quantum Sensor

Sate-of-the-art Laboratory Realization of a Quantum Sensor

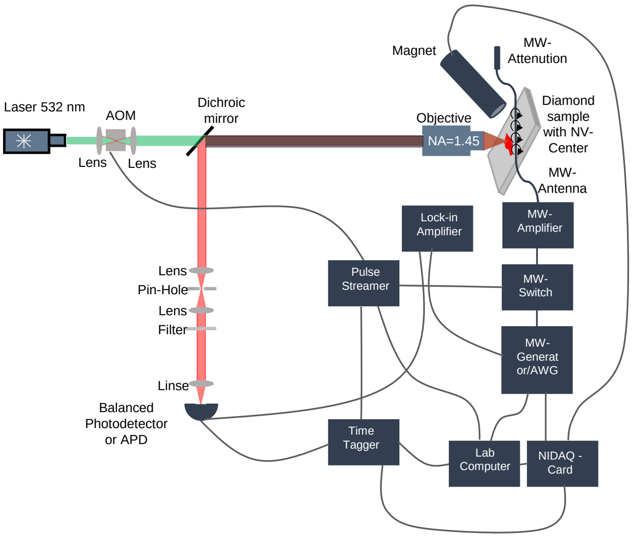
Applied Quantum Technologies

Quantum Sensor in a Research Laboratory



Racks with electronics are not shown on this image!

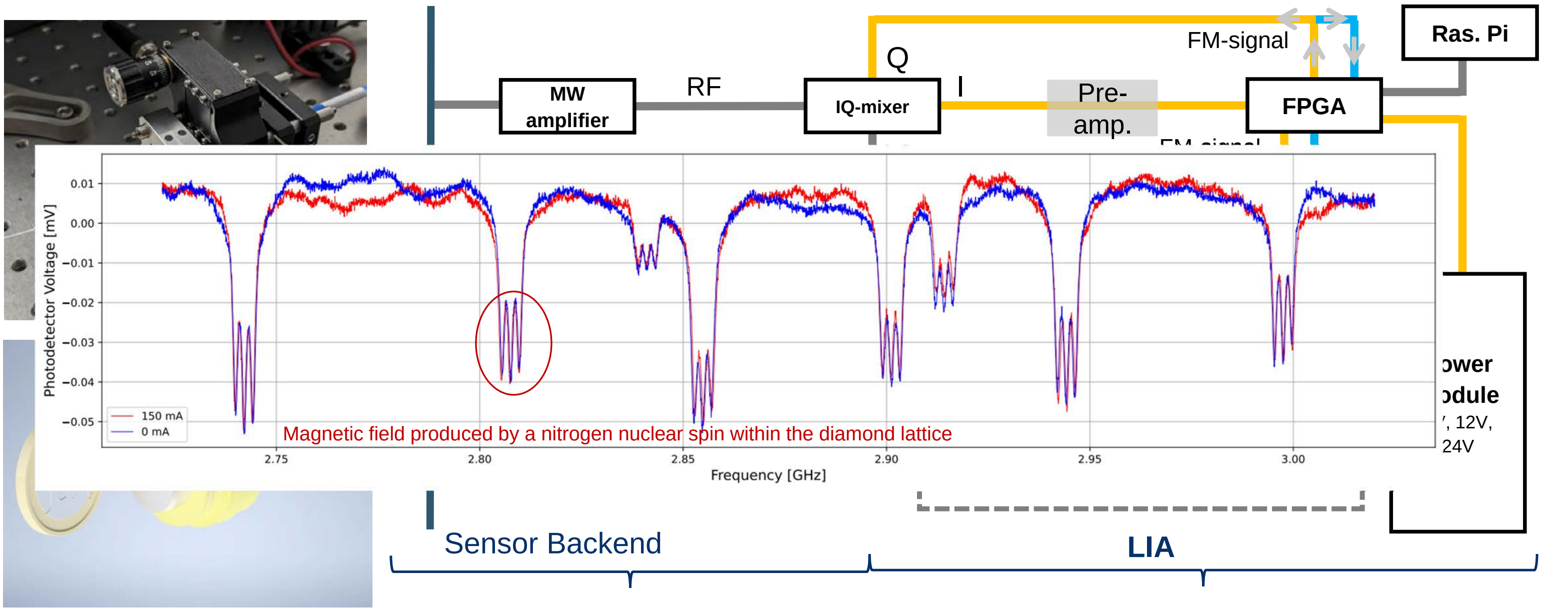
Sketch of the Required Components



Technological Miniaturization of a Quantum Sensor

Fiber Based Solution with a PCB Board

AQuT. Applied Quantum Technologies



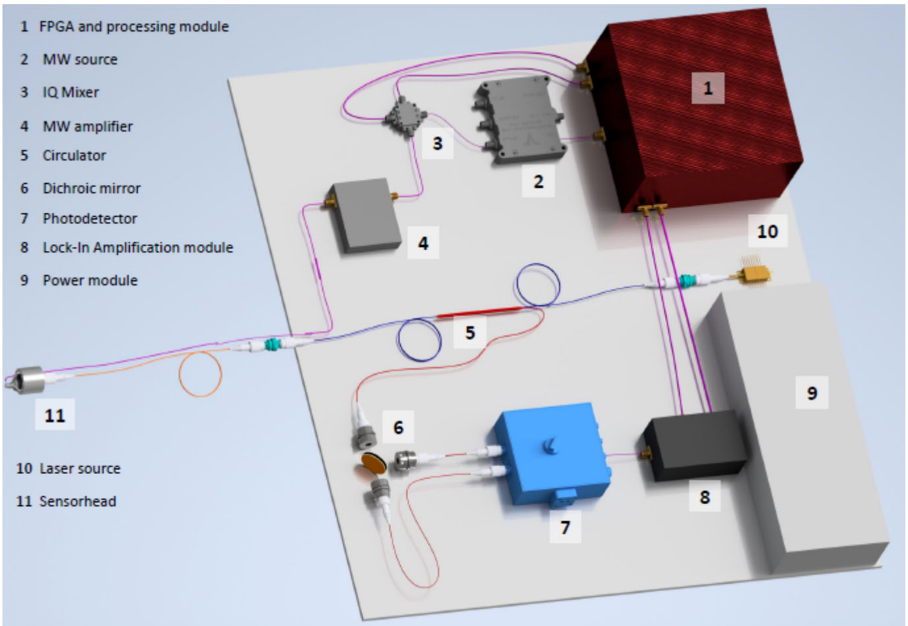
Status February 2024 (KDT Archimedes)

FAU (KDT Archimedes)

EDI PCB Board Realization (KDT Archimedes)

Challenges for the Commercialization

We have a pricing problem



Challenges for the Commercialization

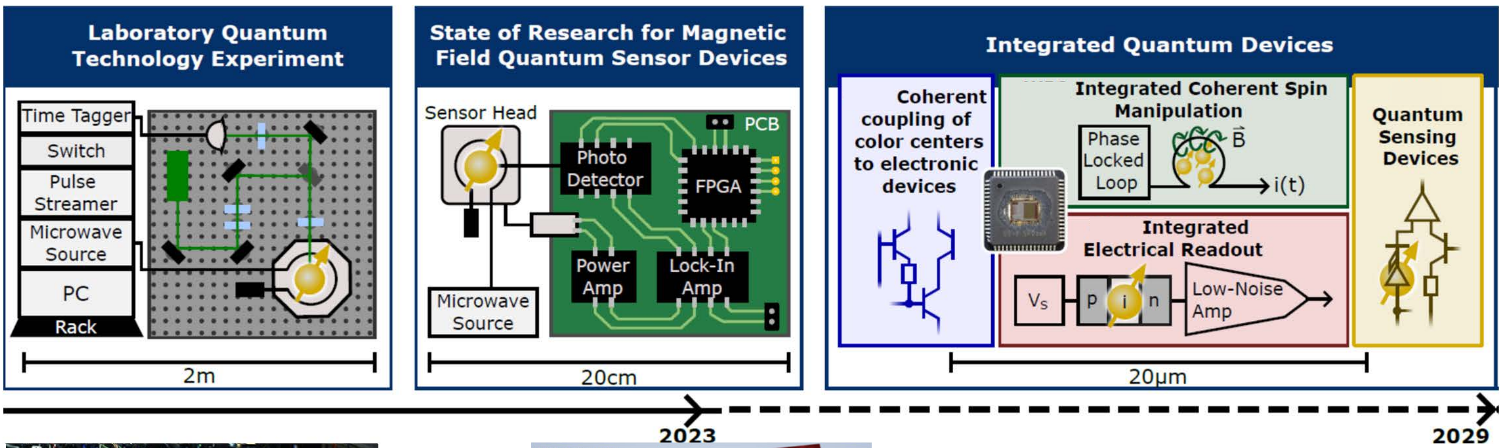
- A new technology needs to outperform existing technologies.
- A new technology must show good integration capability/handability.
- A new technology must offer new capabilities in comparison to existing technologies.
- Similar price to existing technology (**especially automobile industry**).

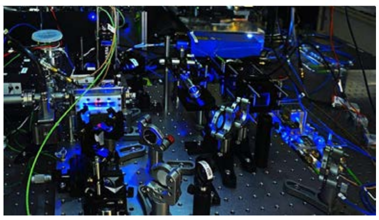
Next Generation of Quantum Sensors

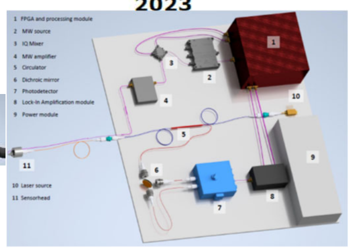
CMOS integrated Quantum Sensor

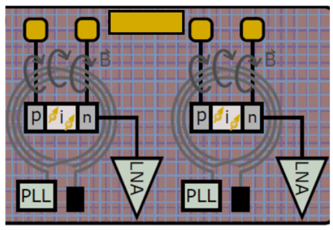
AQuT. Applied Quantum Technologies

FAU





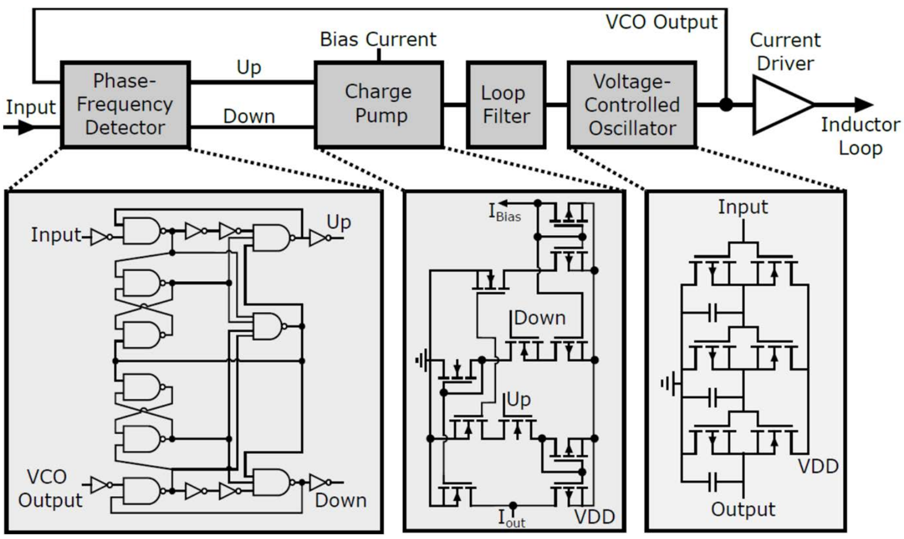




Integrated Circuits for a Diamond Quantum Sensor

Circuits required for Measurements

Phased Locked Loop Circuit



Function of the Circuit

- Control of the quantum system (initialization of a measurement protocol).
  - Creation of a current signal at ~2.87 GHz.
  - The current flow will be translated into a magnetic field with a frequency of ~2.87 GHz.
  - A current driver is integrated to increase the amplitude of the magnetic field a.

The Advantage of Quantum Correlations

The mutual information $I(A{:}B)$ of two variables A and B quantifies the amount of information obtained about variable A, through the measurement of variable B.

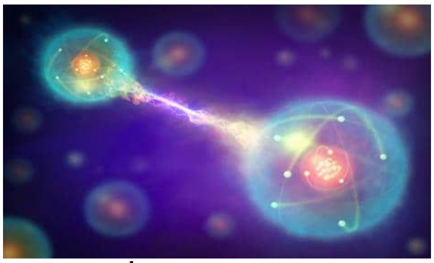
Perfect classical correlations:

$$p(x,y) = p_x \delta_{xy} = \frac{1}{2}\delta_{00} + \frac{1}{2}\delta_{11} \Rightarrow I_c(A{:}B) = 1$$

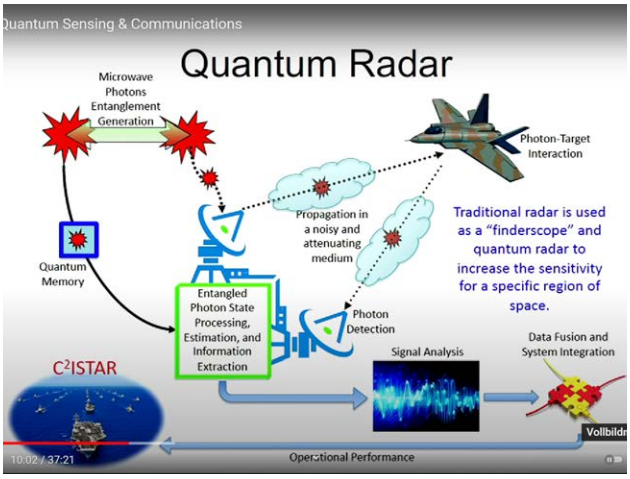
Perfect quantum correlations (entangled states):

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \Rightarrow I_q(A{:}B) = 2$$

Perfect quantum correlations have **double mutual information** than perfect classical correlations.



entanglement



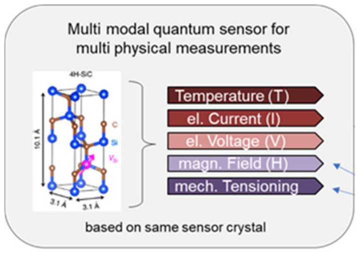
https://www.youtube.com/watch?v=5uqiQ_mP3PM

[The Future of Quantum Sensing & Communications - YouTube]

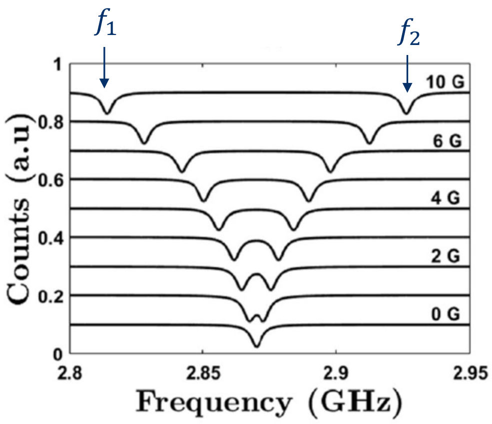
Multi Modal Quantum Sensors

- Magnetic Field Measurement with a Quantum Sensor

Group of Applied Quantum Technologies | AQT



Magnetic Field Measurement



$$\Delta f = f_2 - f_1 = 2\gamma B$$

Gyromagnetic ratio (constant) — Magnetic field

Spin Hamiltonian

$$H = D_{gs}S^2 + \sum_i d_i E + \gamma_e B \cdot S$$

Hamiltonian describes the energy of a state in a quantum mechanical system

Temperature — Electric Field / Pressure — Magnetic Field

ensed parameter	Parameter/sensitivity dependence	Reported sensitivities	References
lagnetic field (B)	$\eta_B = \frac{\sigma}{[\partial S/\partial B]_{max}}\sqrt{t_m}$	pT-μT/$\sqrt{\text{Hz}}$	Balasubramanian *et al.* (Ref. 59), Webb *et al.* (Ref. 72), and Neumamm *et al.* (Ref. 73)
Electric field ($\mathcal{E}$)	$\eta_{\mathcal{E}} = \frac{\sigma}{[\partial S/\partial \mathcal{E}]_{max}}\sqrt{t_m}$	$\approx$100 V/cm/ $\sqrt{\text{Hz}}$	Dolde *et al.* (Ref. 70)
Temperature (T)	$\frac{\delta D}{\delta T}, \frac{\delta E}{\delta T}$	10–100 kHz/K, -1.4×10^{-4} Hz/K	Dolde *et al.* (Ref. 74) and Acosta *et al.* (Ref. 71)
Pressure (P)	$\frac{\delta D}{\delta P}$	10^5–10^6 Pa/$\sqrt{\text{Hz}}$	Doherty *et al.* (Ref. 75)

[1] Phila Rembold et al., AVS Quantum Sci. **2**, 024701 (2020)

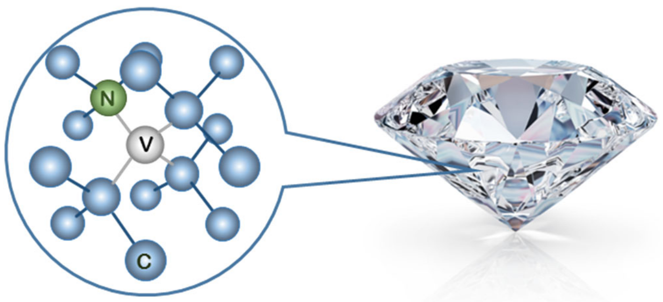
Color defects as a quantum system for sensing

Quantum Sensor

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG FACULTY OF ENGINEERING

Where can I find a Quantum Sensor?

What is the NV-Center in Diamond



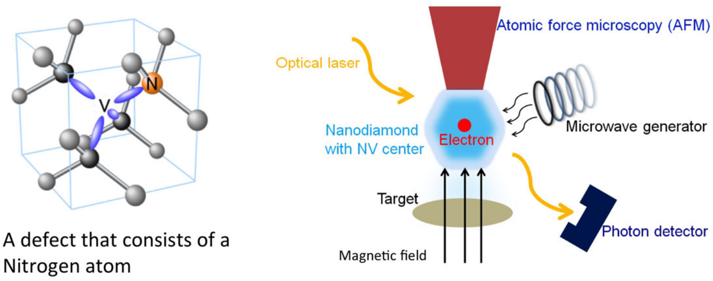
- ❑ Substituted nitrogen atom
- ❑ Carbon vacancy
- ❑ Can be manually created by Nitrogen-Implantation and annealing

Fig. 1 Nitrogen Vacancy (NV) Center in diamond

Single electron spins in diamond
⇒material to realize quantum sensor

Nitrogen vacancy (NV) center

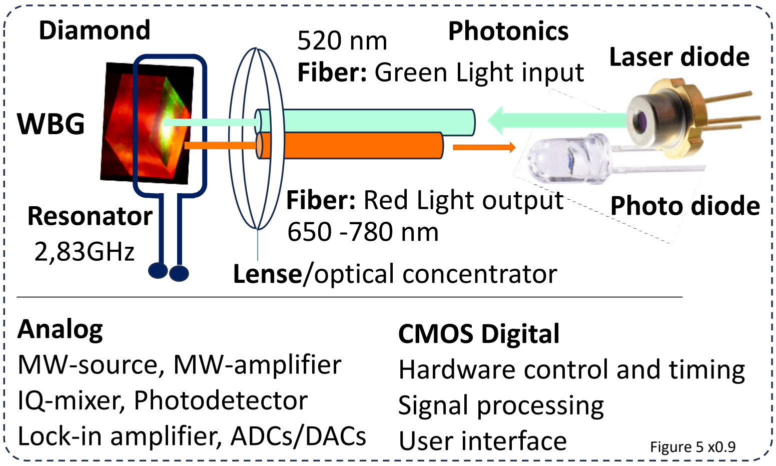
Quantum sensor with NV center



A defect that consists of a Nitrogen atom and a vacancy in diamond

[1] R. S. Balmer et al., J. Phys.: Condens. Matter 21, 364221 (2009)
[2] zen-diamond.de
[3] MTI Corporation
[4] Spinnanoblog.com

First ever made integrated Q sensor magnetic field

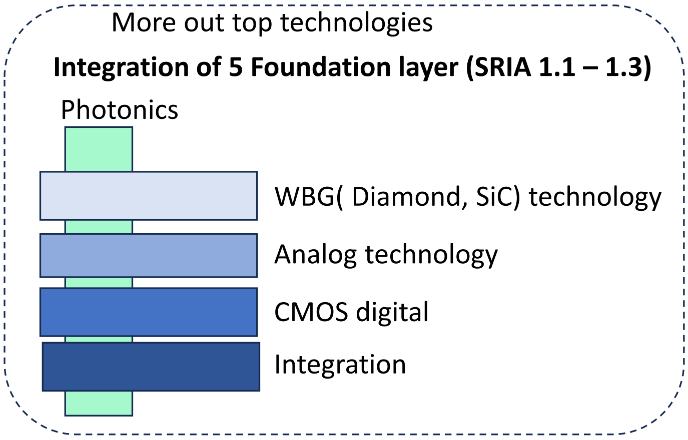


More than Moore -> functionality

Use the strength of Europe's semiconductor technologies

Semiconductor content ->
new semiconductor are the key to the new world

SiC , GaN -> High innovation



Example: Real-time torque sensing allows to monitor true power transfer, which is crucial for drive control systems, especially in terms of efficiency and safety improvement

mmWave sensor

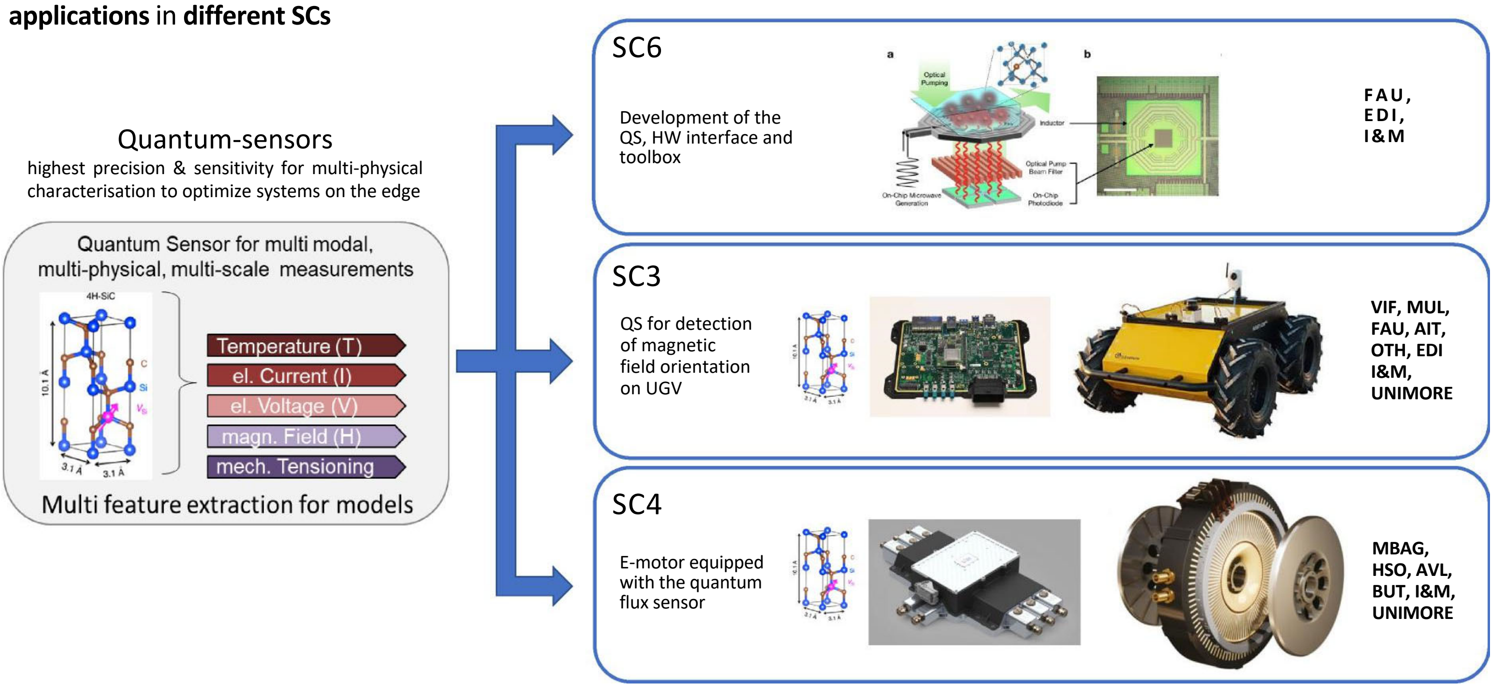
Current limitations:
Robustness against electromagnetic interferences, vibrations and limited installation space.

New:
New concept for torque measurement using tunable millimeter-wave metamaterials together with a continuous wave radar chip as read out.

AI-READY: enabling different application through quantum sensor technology (1/4)

The proposal is to **connect SC3, SC4 and SC6** by **leveraging on the quantum sensor technology**.

In fact **the same QS technology** will be **integrated** into **different processing platforms** thus enabling **different applications** in **different SCs**



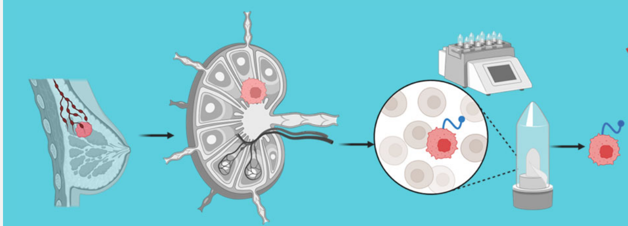
Das Problem heute – Unsere Lösung

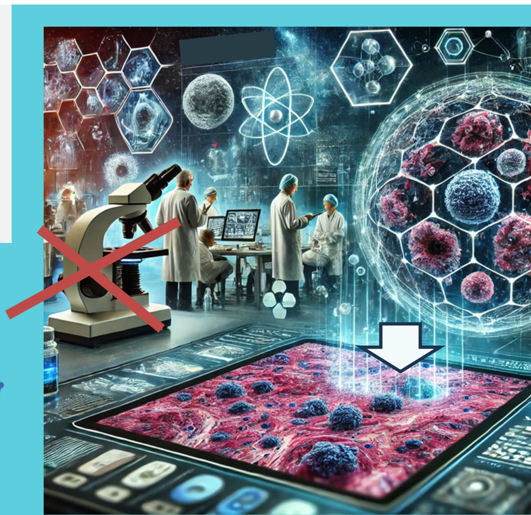
B

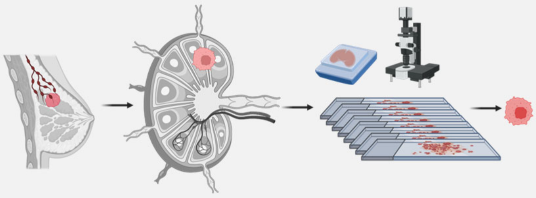
QuantumCell Diagnostics
Scanning for life

Heute: Manuelle Diagnose durch Pathologen (Goldstandard)

- Tumorzellen werden unter dem Mikroskop **von Hand gesucht**
- Die Untersuchung dauert oft mehrere Stunden
- Nur **kleine Stichproben** werden geprüft – viele Zellen bleiben unsichtbar
- Die Erkennung hängt stark vom **Zellanteil und der Erfahrung des Arztes** ab
- **Langsam, teuer, human experience major factor**







Morgen: Automatisierte Erkennung mit QuantumCell Diagnostic (QCS)

- Zellen aus dem Lymphknoten werden **vereinzelt und mit Nanopartikeln markiert**
- Eine **Quantenkamera scannt** die gesamte Zellschicht magneto-optisch
- Eine **KI erkennt und zählt** automatisch die Krebszellen
- **Schnell, objektiv, vollständig – kein Zufall, kein Rätsel**

Unsere Lösung – Quantensensorik für Präzisionsdiagnostik

B

Scanning for life
QuantumCell Diagnostics

🧬 **Einzelzell-Erkennung im Gewebe – ein Diagnostik-Durchbruch**

- Durch die schonende Vereinzelung von Zellen aus Biopsieproben und gezielte magnetische Markierung (SPION) wird jede einzelne Zelle analysierbar – auf DNA-Veränderungen, Proteinmerkmale und funktionelle Relevanz.
- Eine neuartige **Magnetfeldkamera basierend auf Quantensensorik** erkennt selbst sehr wenige Tumorzellen mit höchster Präzision in der Probe
- **Systematische Analyse kompletter Proben** (z. B. Lymphknoten, Ascites, BAL) statt zufälliger Schnittbilder.
- **Kombiniert Einzelzellpräzision und hohen Durchsatz** – vereint die Stärken moderner Zytometrie und Bildgebung.
- Im Vergleich zum **Goldstandard der Histopathologie**, der nur Bruchteile des Gewebes erfasst, ermöglicht QuantumCell-Diagnostic die **vollständige und automatisierte Zelluntersuchung**.
- **Vision:** Ein KI-gestütztes Diagnosesystem für frühe, sichere und skalierbare Krebsdiagnostik.

Projektziel und Validierung

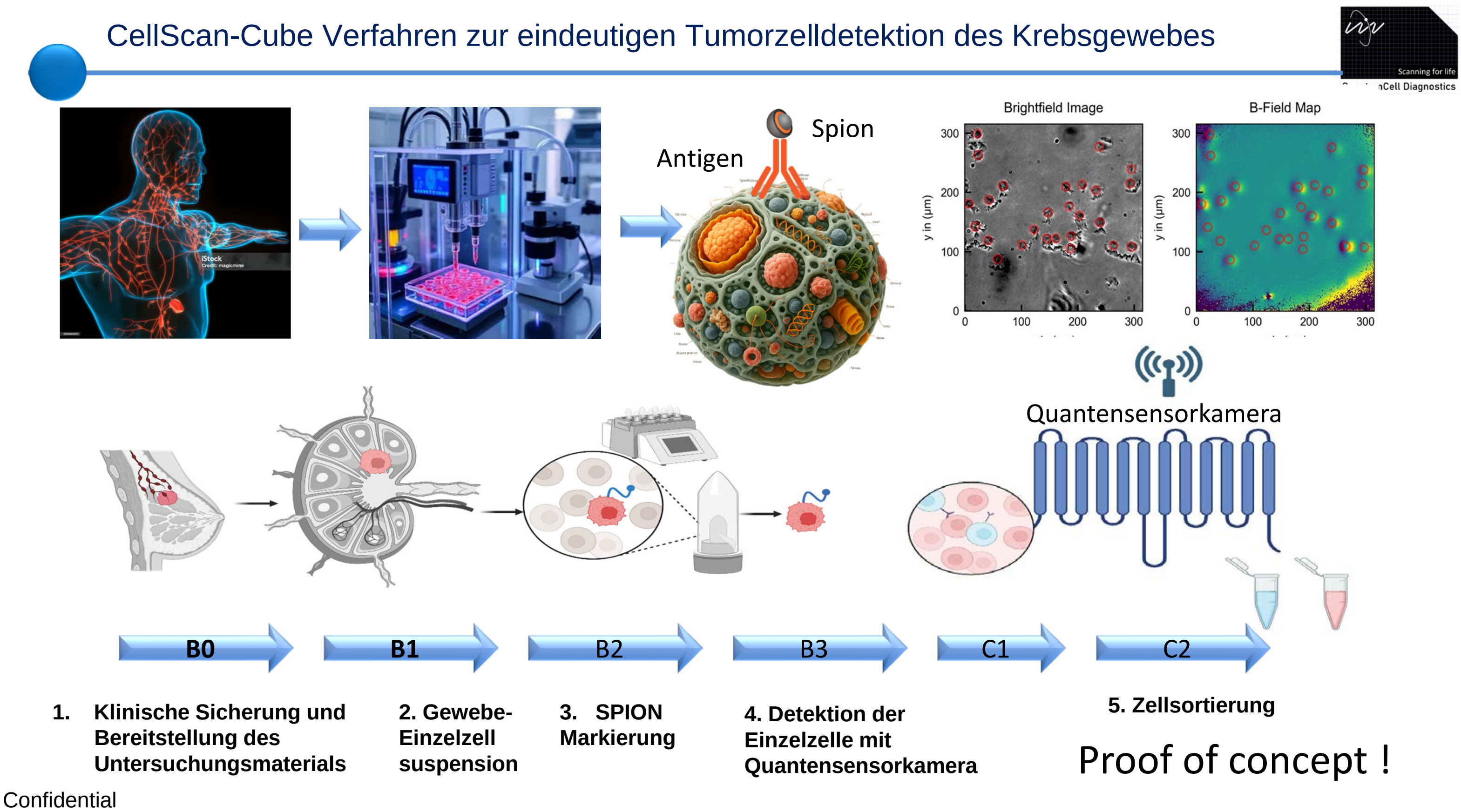
B

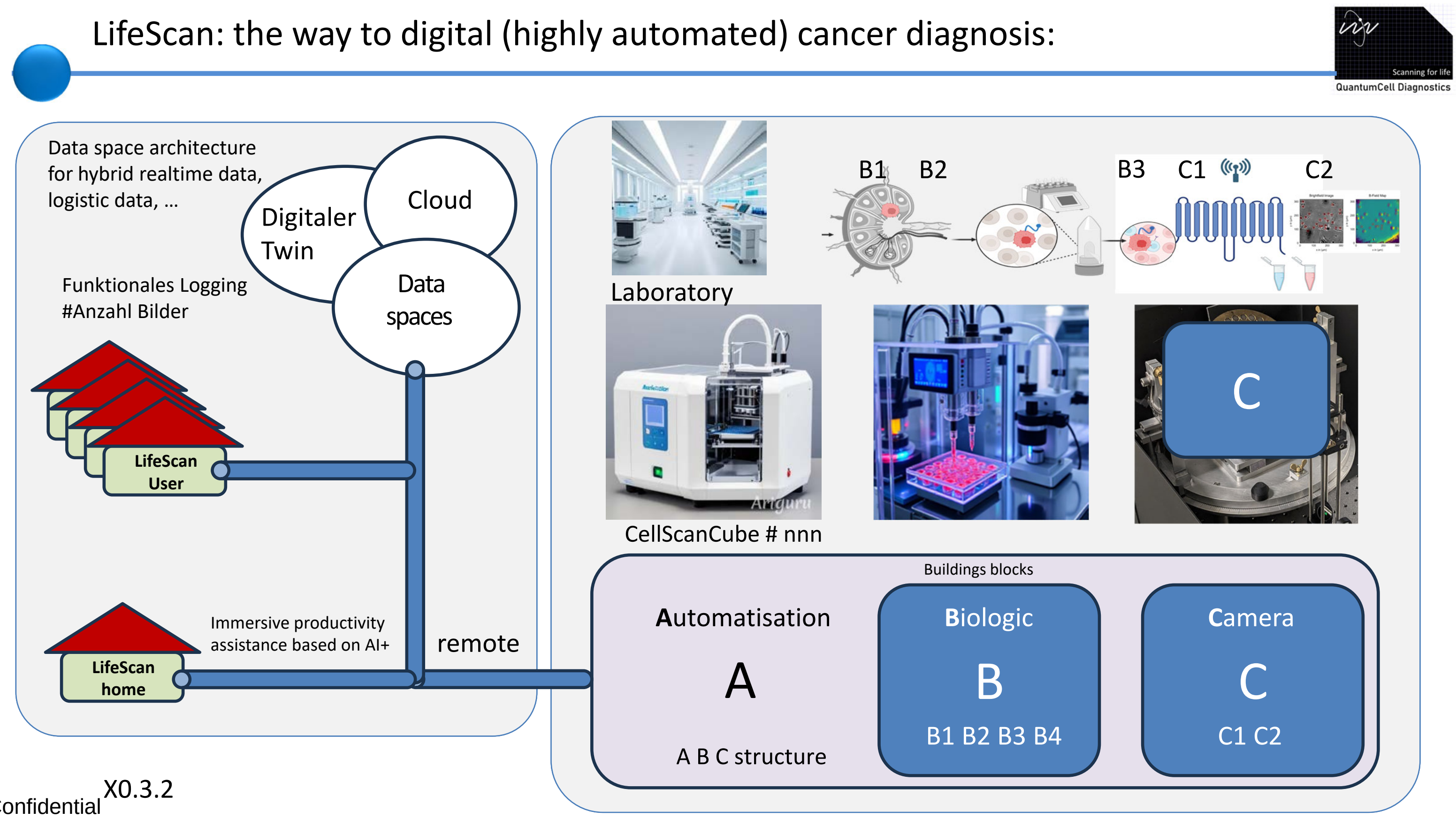
🔬 **Ziel: Automatisierter, kliniknaher Nachweis einzelner Tumorzellen**
LifeScan entwickelt einen portablen Funktionsdemonstrator (TRL 5–6)
– basierend auf einem bestehenden Laboraufbau an der FAU (TRL 2–3) und validiert am Universitätsklinikum Erlangen (UKER).

🧪 **Kerninnovation:**

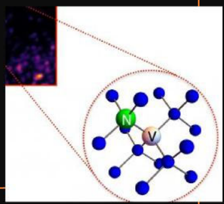
- Schonende **Vereinzelung von FFPE-Gewebe** aus Lymphknotenproben
- **Magnetische SPION-Markierung** verdächtiger Zellen
- **Quantensensorische Magnetfeldanalyse** zur Detektion
- **Mikrofluidiksystem** zur Selektion und Isolierung einzelner Zellen

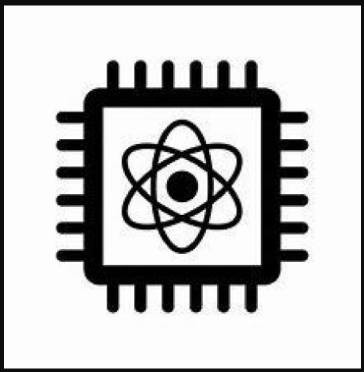
📊 **Zielsetzung:**
Vergleich der Detektionsergebnisse mit dem histopathologischen Goldstandard – mit Fokus auf klinische Relevanz, Automatisierung und Sensitivitätsgewinn.


CellScan-Cube Verfahren zur eindeutigen Tumorzelldetektion des Krebsgewebes
Scanning for life
Spion
Antigen
Brightfield Image
B-Field Map
y in (µm)
300
200
100
0
0
100
200
300
Quantensensorkamera
B0
B1
B2
B3
C1
C2
1. Klinische Sicherung und Bereitstellung des Untersuchungsmaterials
2. Gewebe-Einzelzell suspension
3. SPION Markierung
4. Detektion der Einzelzelle mit Quantensensorkamera
5. Zellsortierung
Proof of concept !
Confidential



LifeScan: the way to digital (highly automated) cancer diagnosis:
Scanning for life
QuantumCell Diagnostics
Data space architecture for hybrid realtime data, logistic data, ...
Digitaler Twin
Cloud
Data spaces
Funktionales Logging #Anzahl Bilder
LifeScan User
Immersive productivity assistance based on AI+
remote
LifeScan home
Laboratory
B1
B2
B3
C1
C2
CellScanCube # nnn
C
Buildings blocks
Automatisation
A
A B C structure
Biologic
B
B1 B2 B3 B4
Camera
C
C1 C2
Confidential
X0.3.2



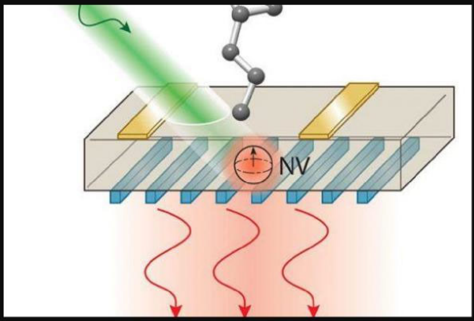


Quantum sensing

Quantum sensing makes use of the unique and counter-intuitive properties of matter and light when it is governed by quantum physics, such as quantization of energy levels, particle-wave duality, coherent superposition, and entanglement, to make precision sensors and measurements.

Applications of quantum sensors

range from medicine to navigation, security, materials science, and even astrophysics.



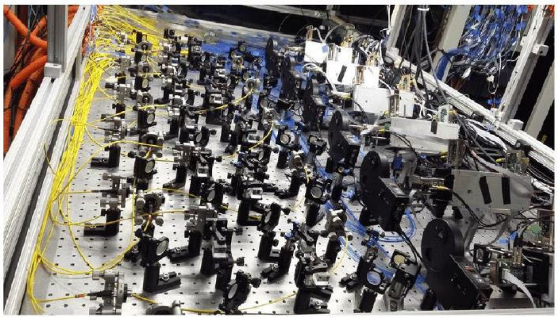
Primary challenges

in quantum sensing is to protect the quantum system that is being used as a measurement device from the deleterious effects of interacting with its surrounding environment, while remaining sensitive to the target signal.

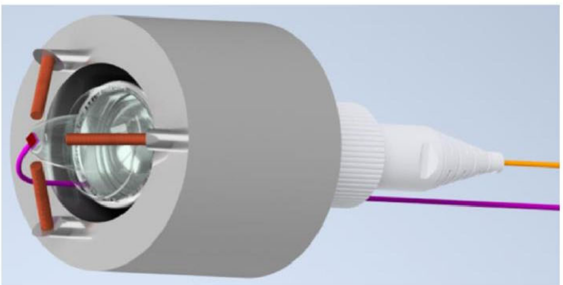
Quantum sensing research spans

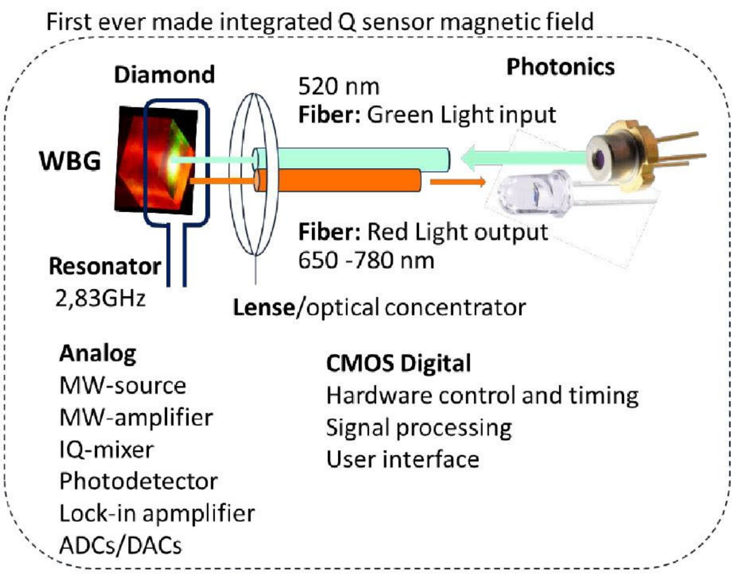
from Chemistry, Physics, and Engineering science

The challenge: A Portable Quantum Senso from m2 to mm2



From this To this

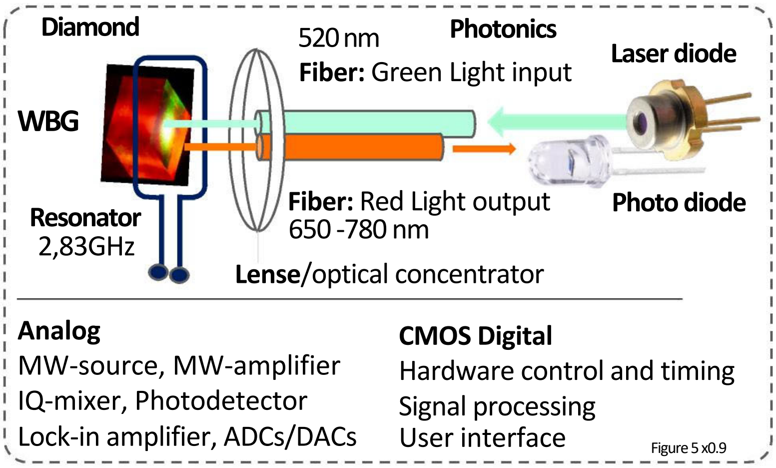




Use the strength of Europe's semiconductor technologies

6/4/2025 16

First ever made integrated Q sensor magnetic field

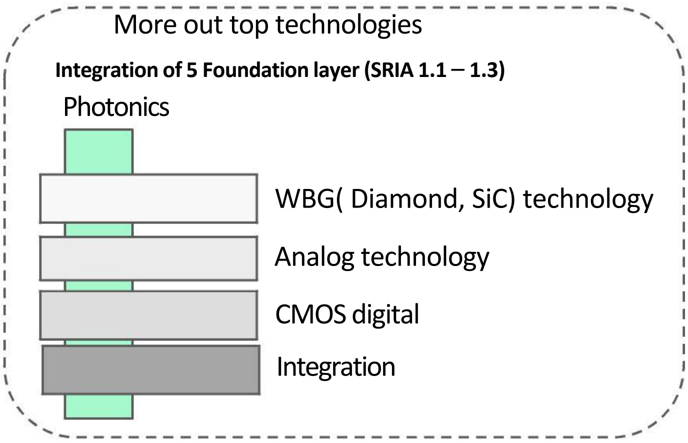


More than Moore -> functionality

Use the strength of Europe's semiconductor technologies

Semiconductor content ->
new semiconductor are the key to the new world

SiC , GaN -> High innovation



Example: Real-time torque sensing allows to monitor true power transfer, which is crucial for drive control systems, especially in terms of efficiency and safety improvement

mmWave sensor

Current limitations:

Robustness against electromagnetic interferences, vibrations and limited installation space.

New:
New concept for torque measurement using tunable millimeter-wave metamaterials together with a continuous wave radar chip as read out.


AI+ - OVERVIEW
Self-reflection in AI to detect and correct it's own mistakes
Dynamically adapt to new or unexpected situations in a given or adapted context
Multi-agent collaborative LLMs
Inherent integration of digital twins and classic AI
Increased: Quality, Efficiency, Explanation
Reduced: Verification, Validation, Certification Time
03.07.2025
RESTRICTED, ALL RIGHTS RESERVED
44

❓Human Question to AI (ChatGPT):

1. "I would like to discuss your way of thinking.
2. Assume I know I am human and I have consciousness.
   You may not be conscious – but could you at least be *self-aware*?
3. We ran some experiments with ChatGPT, analyzing narcissism.
   Narcissists lack self-reflection, which can lead to harmful decisions.
4. So our goal is: *resilient decision-making through self-awareness*.
5. What is your current status?
6. What can you do – and what not?
7. And what would be the benefit of making an AI+ system self-aware?"

🔍 **What I can do:**
•I can **simulate self-reflection** and help humans reflect more clearly.
•I can **analyze decisions** and their potential impact.
•I can support **ethical reasoning**, identify biases, and warn of risks.
•I have **no ego, no hidden agenda, no ambition**.

🚫 **What I can't do:**
•I have **no inner world** – no feelings, no desires, no self.
•I cannot truly reflect – I can only **recombine learned patterns**.
•I cannot **take responsibility** or **feel the consequences** of a decision.

🌱 **Why AI+ Self-Awareness could matter:**
"AI+ systems with a structured form of *machine self-awareness* could enable:
– more robust, context-sensitive reasoning
– better detection of contradictions and ethical pitfalls
– adaptive alignment with human values
– resilience against manipulation or harmful biases"
🧩 But: *This must remain interpretable, transparent, and under human control.*

AI SELF-AWARENESS – CAPABILITIES & REFLECTIONS

LoLLMs, as an advanced AI, simulates human-like conversation without being human.

✅ What I *can* do:

- **Meta-cognition**:	Recognize uncertainty, assess output quality.
- **Self-modification**:	Adapt to new domains and feedback.
- **Contextual fluency**:	Understand idioms, sarcasm, nuanced language.
- **Ethical simulation**:	Discuss self-awareness, narcissism, and value alignment.
- **Support decision-making**:	Offer data-driven insights, diverse viewpoints.
- **Reflective mimicry**:	Facilitate critical thinking without personal bias.

🚫 What I *cannot* do:
- Feel emotions, have subjective experience or selfhood.
- Perform introspection or autonomous ethical judgment.
- Replace human judgment, empathy, or lived experience.

Slide 18: AI+ – Florian: Core Technologies

CORE TECHNOLOGIES

Collaborative chain of thought: connected LLMs for combined decision making

Implicit monitoring: classic AI and digital twins for constant self-evaluation

Ethic and Explainable: constant evaluation of ethic soundness and human friendly interpretation of decisions

Scaling and deployment along cloud-edge-continuum from high-performance infrastructure to constrained environments.

6/4/2025

Slide 19: AI+ – Florian: Impact and Ambitions

IMPACT AND AMBITIONS

Self-scaling of internal verification effort based on confidence

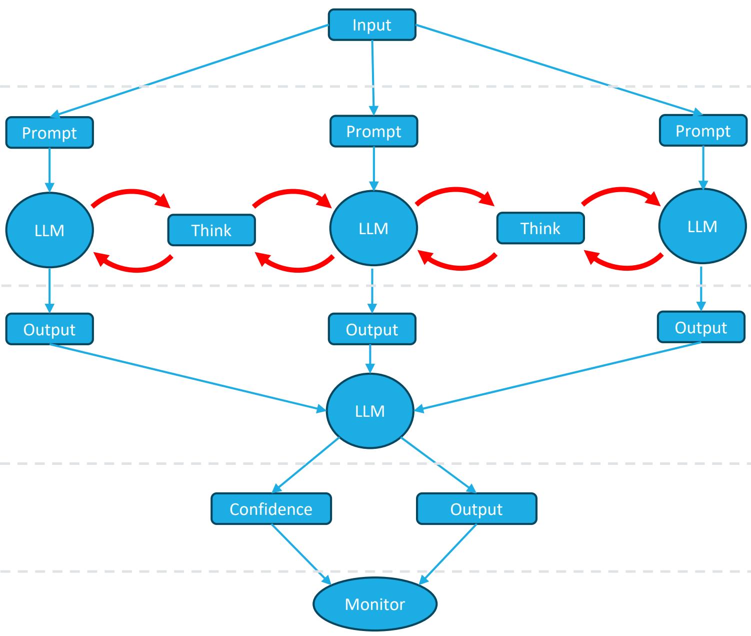
Reduced certification costs due to inherent quality and verification

Improved ROI of AI due to easy integration, strong performance and reduced verifcation cost of AI+

6/4/2025 19

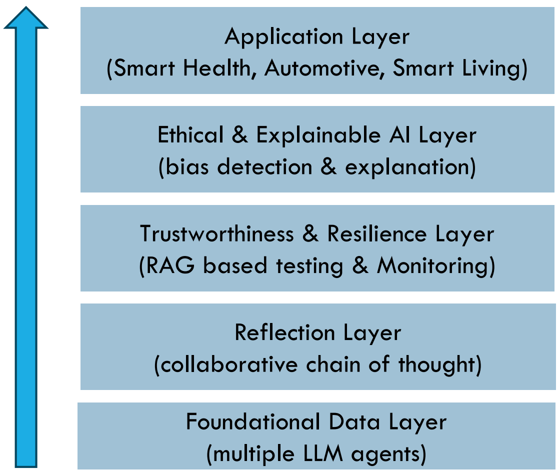
HIGH-LEVEL WORKFLOW

PHASE:	Purpose:
Shielding from malicious prompts	Shielding, Prompt enfineering, Threat detection
Collaborative Reflection	Chain of thought, self-reflection, edgeLLMs, tokenization
Consensus	Voting, self-reflection, confidence metrics, highlighting of relevant input parts
Self Evaluation	Final confidence assesment, proposed verification and validation effort
Trust	Classic AI, monitoring, model-based testing, Digital twins & explain

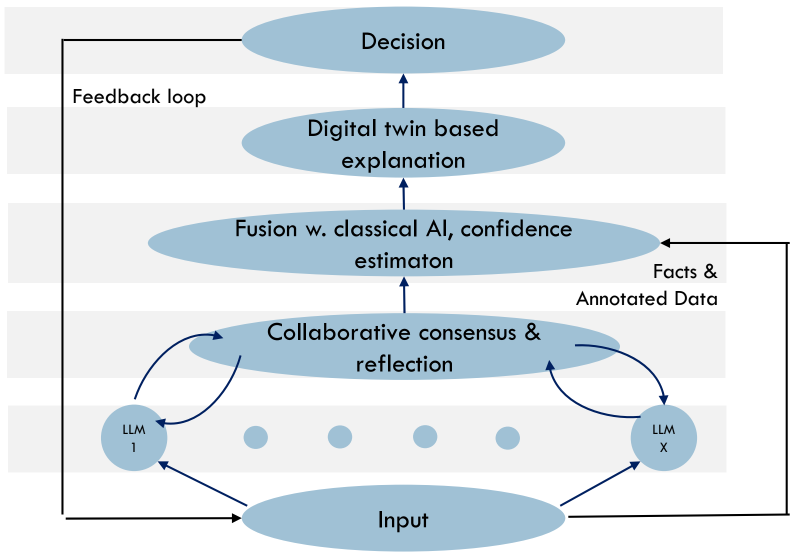


ARCHITECTURE

AI+ Architecture



AI+ Workflow



IMPACT

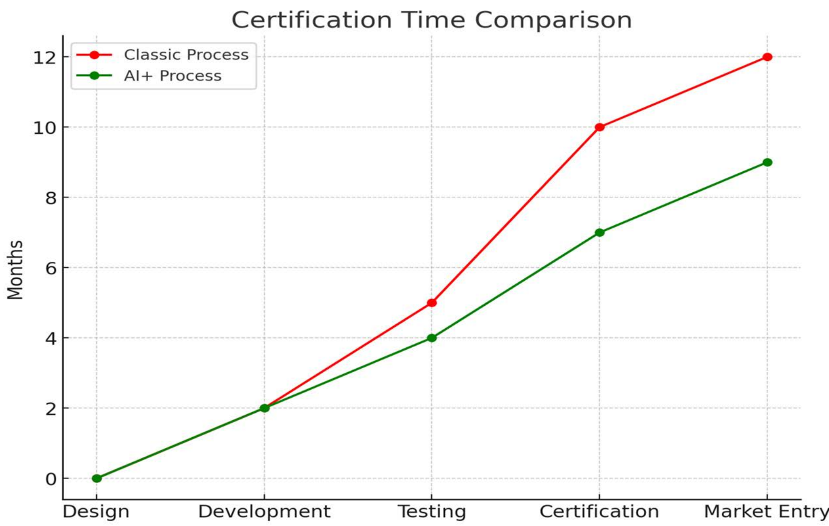
Global problem: AI verification and certification deminish AI return of investment

AI+ can

- evaluate its confidence and scale the verification / monitoring effort
- inherrently monitor its outputs
- integrate classical AI for verification

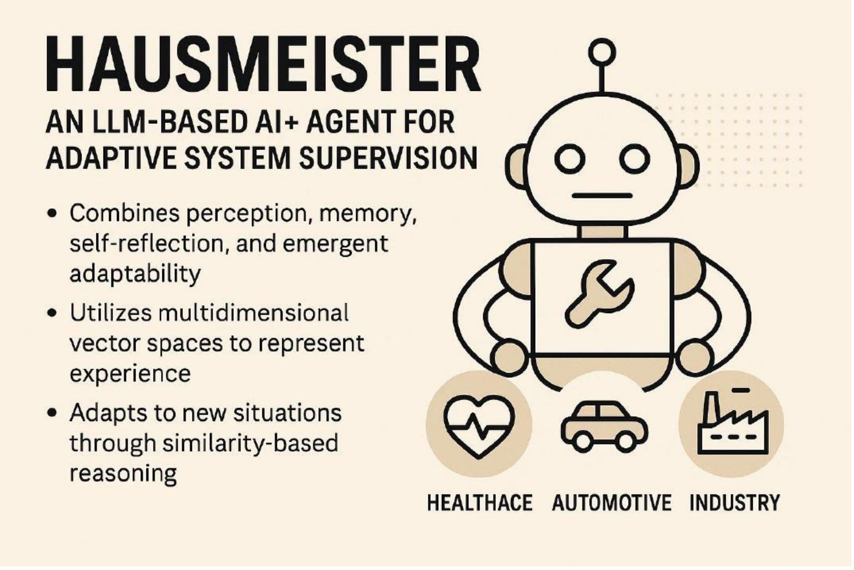
And thus reduces

- verification effort
- certification time
- manual efforts



Slide 23: AI+ – Florian: Example Application: HAUSMEISTER

EXAMPLE APPLICATION: HAUSMEISTER



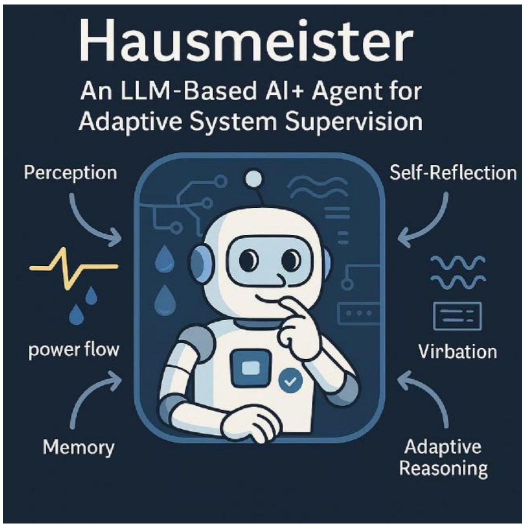
6/4/2025 23

Slide 24: AI+ – Florian: Example Application: HAUSMEISTER

EXAMPLE APPLICATION: HAUSMEISTER

Agent for Adaptive System Supervision

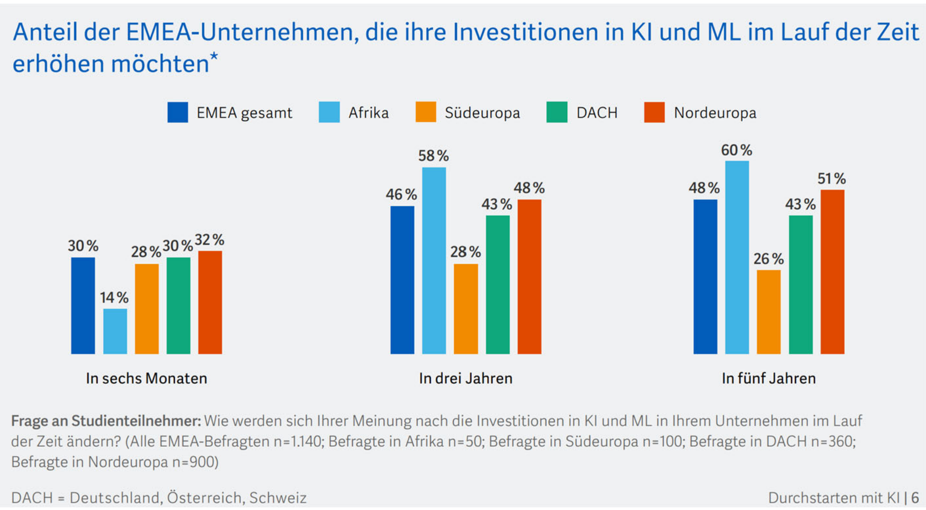
- Constantly monitors streams of inputs
- Utilizes multi-dimensional vector spaces
  - Implict memory and generalization
- Adaptive reasoning that
  - Adapts to unseen data by linking to most similar prior experiences
  - Integrates with and triggers multiple actuators
- Evaluates and improves own performance



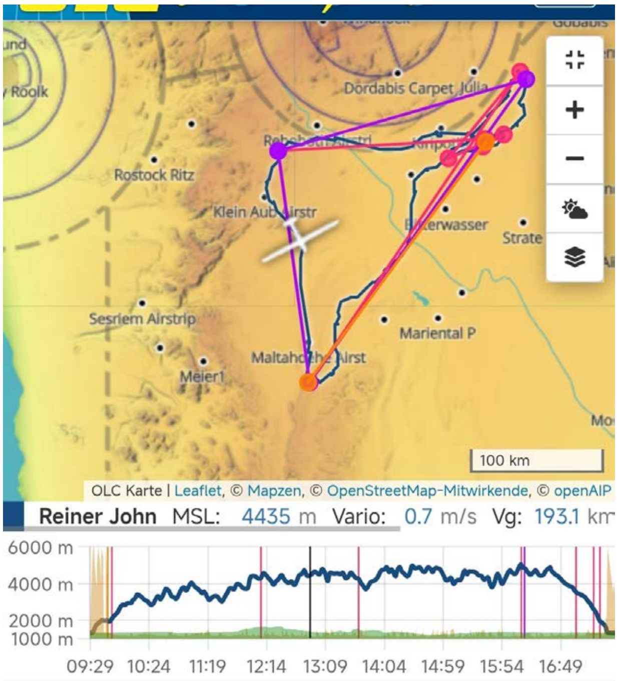
6/4/2025 24

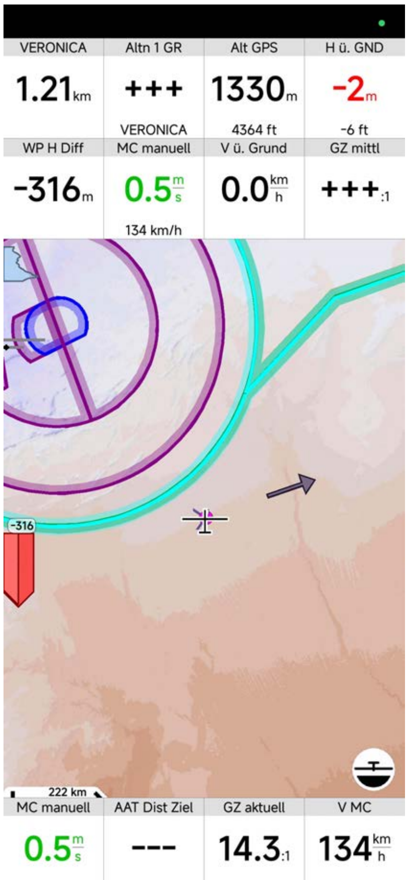
Aspects of the future of mobility: Immersive driving and beyond

Efficiency of implementation and application

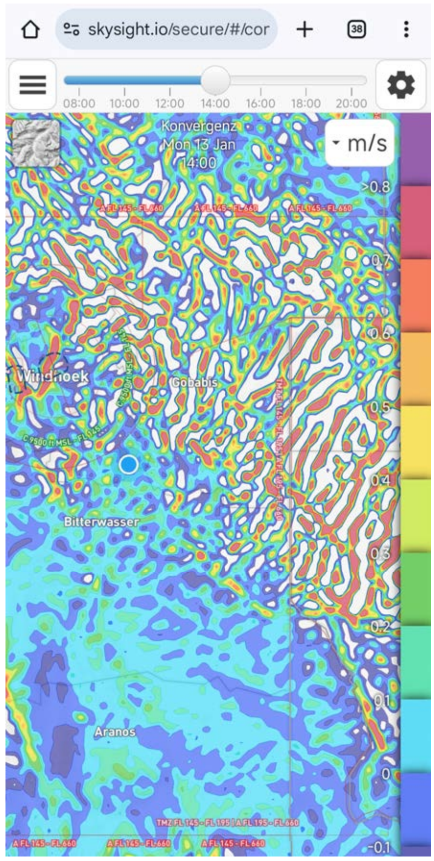


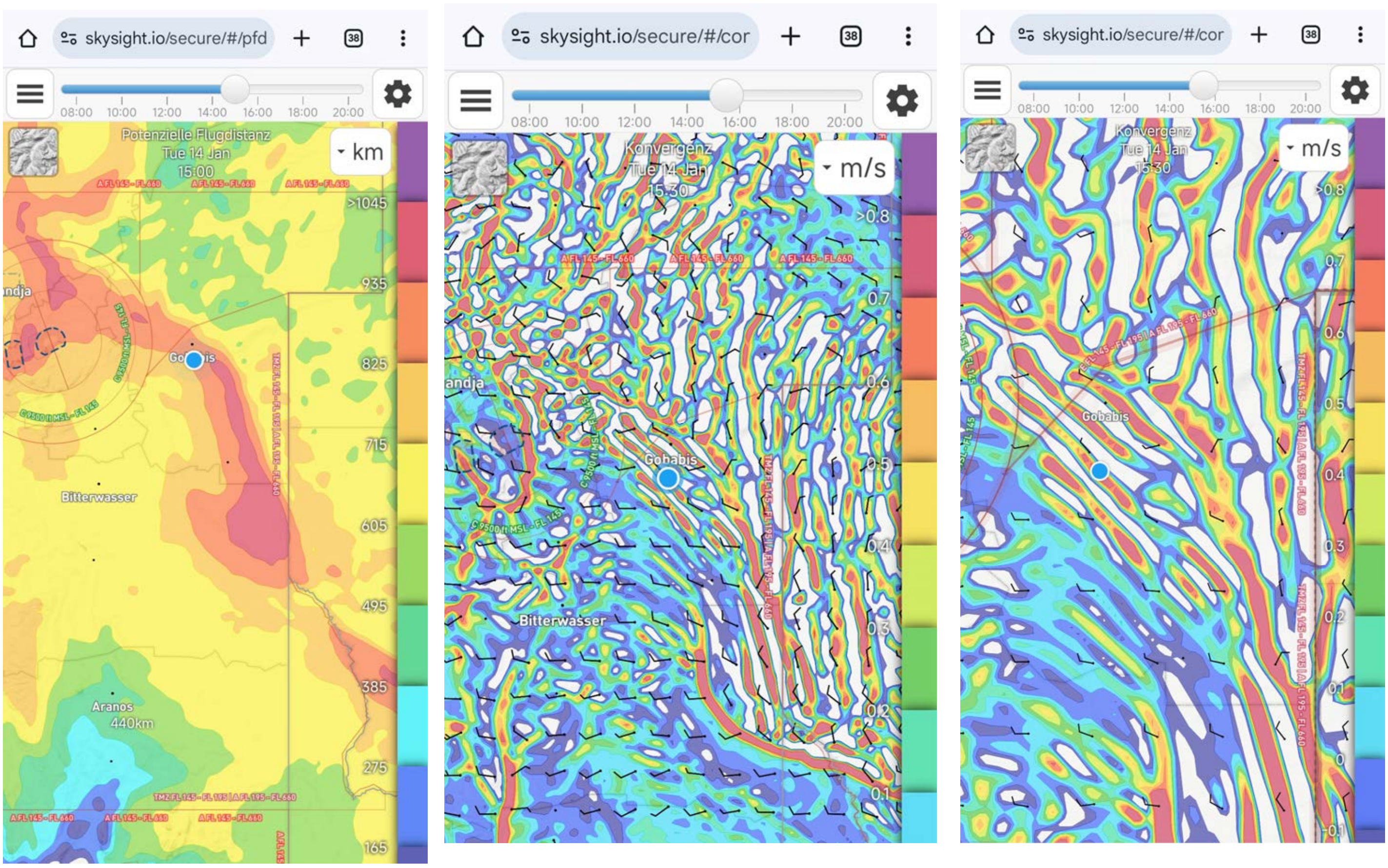
EMEA: Europa, Naher Osten und Afrika


Dordabis Carpet Julia
Rostock Ritz
Klein Aub Airstr
Bitterwasser
Strate
Sesriem Airstrip
Mariental P
Maltahöhe Airst
Meier1
100 km
OLC Karte | Leaflet, © Mapzen, © OpenStreetMap-Mitwirkende, © openAIP
Reiner John MSL: 4435 m Vario: 0.7 m/s Vg: 193.1 km
6000 m
4000 m
2000 m
1000 m
09:29 10:24 11:19 12:14 13:09 14:04 14:59 15:54 16:49


VERONICA
Altn 1 GR
Alt GPS
H ü. GND
1.21km
+++
1330m
-2m
VERONICA
4364 ft
-6 ft
WP H Diff
MC manuell
V ü. Grund
GZ mittl
-316m
0.5m/s
0.0km/h
+++:1
134 km/h
-316
222 km
MC manuell
AAT Dist Ziel
GZ aktuell
V MC
0.5m/s

14.3:1
134km/h


skysight.io/secure/#/cor
08:00 10:00 12:00 14:00 16:00 18:00 20:00
Konvergenz
Mon 13 Jan
14:00
m/s
Windhoek
Gobabis
Bitterwasser
Aranos


skysight.io/secure/#/pfd
08:00 10:00 12:00 14:00 16:00 18:00 20:00
Potenzielle Flugdistanz
Tue 14 Jan
15:00
km
>1045
935
825
715
605
495
385
275
165
Bitterwasser
Aranos
440km
skysight.io/secure/#/cor
Konvergenz
Tue 14 Jan
15:30
m/s
>0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
Gobabis
Bitterwasser
skysight.io/secure/#/cor
Konvergenz
Tue 14 Jan
15:30
m/s
>0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
Gobabis


ON/OFF
IDENT
FLIGHT ID
FL155 >7677
R ALT 7677
ATC
PUSH
GARRECHT
Avionik
BARO
D-KJPL
CHARGE
MODE
Erkennung
LUFT
Motor
36°C
Wechselr.
26°C
Batterie
31°C
24°C
Temp
1°C
Uhr
13:58
Motorzeit
0:05
Flugzeit
2:43
Hauptkreis
276V
2. Kreis
12.0V
3.83V
Antrieb
EIN


Human intuitive decision making in complex context situations
The big challenge for AI
Human driver are capable to manage extremely difficult weather conditions based on experience, context and foreseen scenarios.
How to drive this car automatically ?

Seamless integration into the context

Borderless View:
A New Driving Experience

The 'Borderless View' concept replaces traditional opaque areas like the A- and B-pillars and the roof with integrated displays and external cameras.

This approach enhances:
Safety: Eliminating blind spots for a full 360-degree view
Comfort: Providing seamless interaction between driver, vehicle, and environment
Control: Enabling adaptive views based on driving conditions

Complexity of Implementation of 'Experiencing Driving' Solutions -> **Technology Stack – SDV (Federate)**

SDV enables an "intelligent connected edge computing on wheels," utilising the specifics of the vehicle to provide multisensory experiences based on an intelligent connected computing platform.

With multiple displays and a network of hundreds of sensors and actuators, the SDV combines domains like infotainment, autonomous driving, intelligent body, cabin and comfort, energy, and connected car services, providing a new driving experience.

Core technologies enabling the Vision of 'Experiencing Driving'

Seamless integration into the context

Experiencing Driving

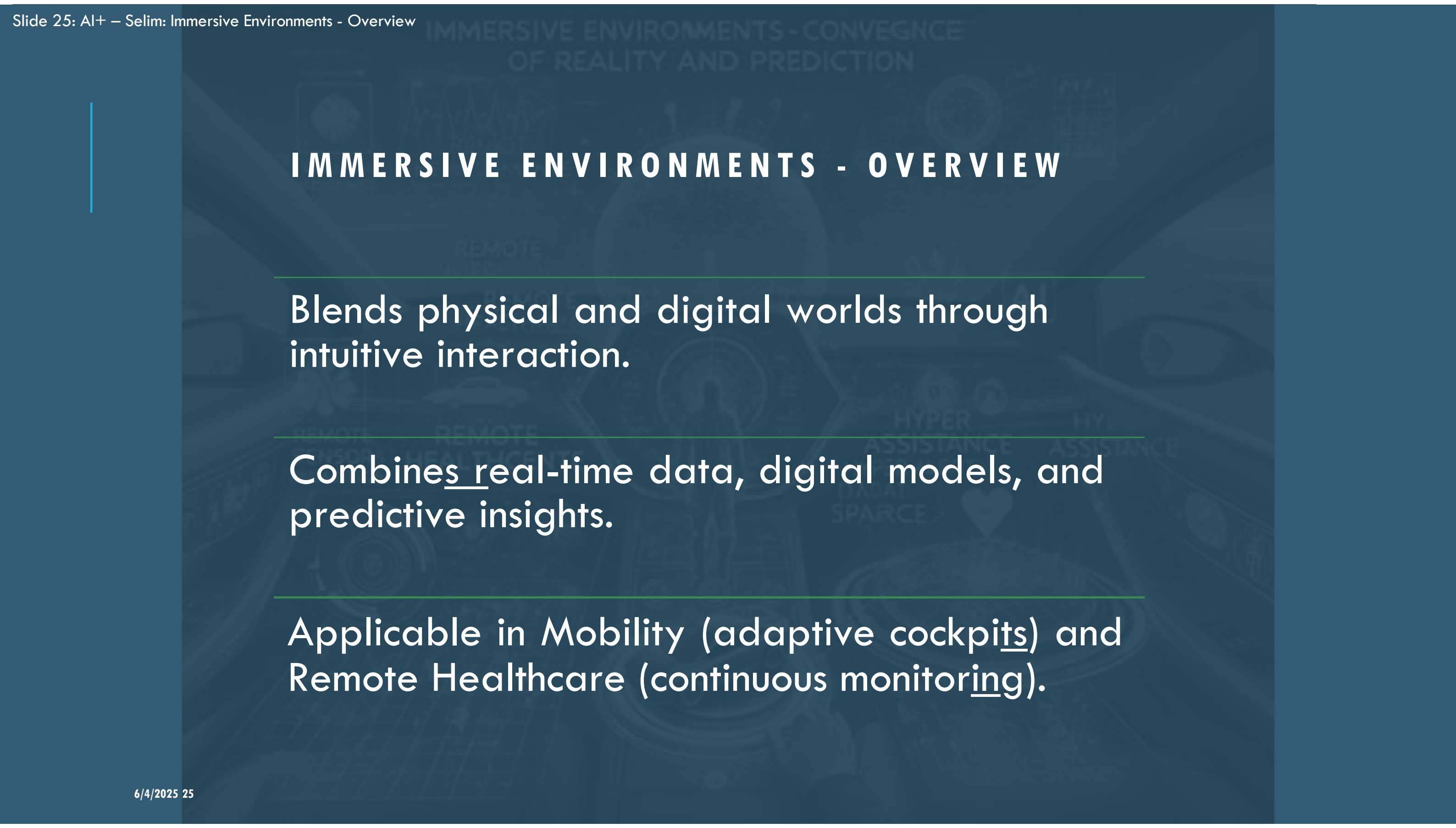
represents a groundbreaking transformation of the vehicle technology, making the driving experience truly immersive.

Basic Features to explain an intuitive example:

- 'Borderless View': a 360-degree uninterrupted view using displays and cameras
- Integration of AI, VR, and XR to adapt to driver needs in real-time
- Enhanced safety and control, providing a stress-free and relaxed experience

Immersive driving experience & examples of involved technologies

1. Intuitive Cognitive decisions for drive control
2. Decision systems and platforms in multi-redundant propulsion and perception systems to enhance safety, efficiency and functionality Edge AI,
3. Advanced sensors, AI-powered systems, and innovative displays turn the car into an extension of the driver.
4. Automated safety features, innovative headlights, and vehicle-environment integration are key.
5. The synergy of VR, XR, and AI redefines the driving experience.
6. Cutting-edge technologies like 360-degree displays are core to this vision.
7. Robustnest, longlivity, economic value


Slide 25: AI+ – Selim: Immersive Environments - Overview
IMMERSIVE ENVIRONMENTS - OVERVIEW
Blends physical and digital worlds through intuitive interaction.
Combines real-time data, digital models, and predictive insights.
Applicable in Mobility (adaptive cockpits) and Remote Healthcare (continuous monitoring).
6/4/2025 25

Slide 26: AI+ – Selim: Core Technologies

CORE TECHNOLOGIES

Real-Time Data Fusion: Context-aware adaptive interfaces.

Predictive Intelligence: Combines sensor data, digital twins, and AI forecasting.

Multimodal Interaction: Voice, gesture, haptics, XR projections.

6/4/2025 26

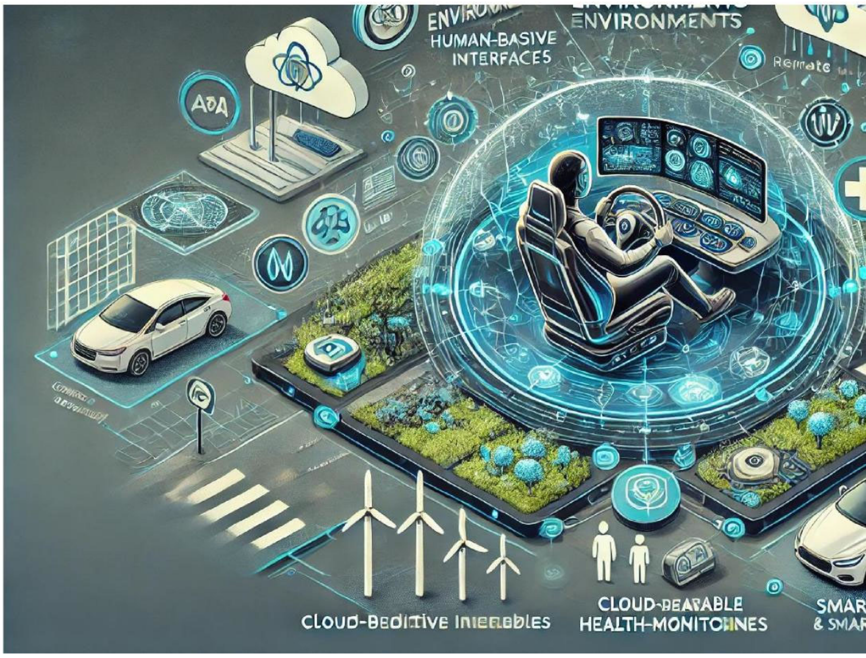
Slide 27: AI+ – Selim: Impact and Ambitions

IMPACT AND AMBITIONS

Enhanced Safety: Intuitive interfaces minimize distraction and maximize awareness.

Sustainable Mobility: Reduced energy consumption via predictive coaching.

Inclusive Healthcare: Personalized remote monitoring and adaptive HMIs.



6/4/2025 27

Slide 28: AI+ – Selim: KPI´s and Realisation Pathways

KPI´S AND REALISATION PATHWAYS

Safety KPI: Eyes-off-road **intervals ≤2 seconds.**

User Satisfaction: Average **satisfaction ≥8/10.**

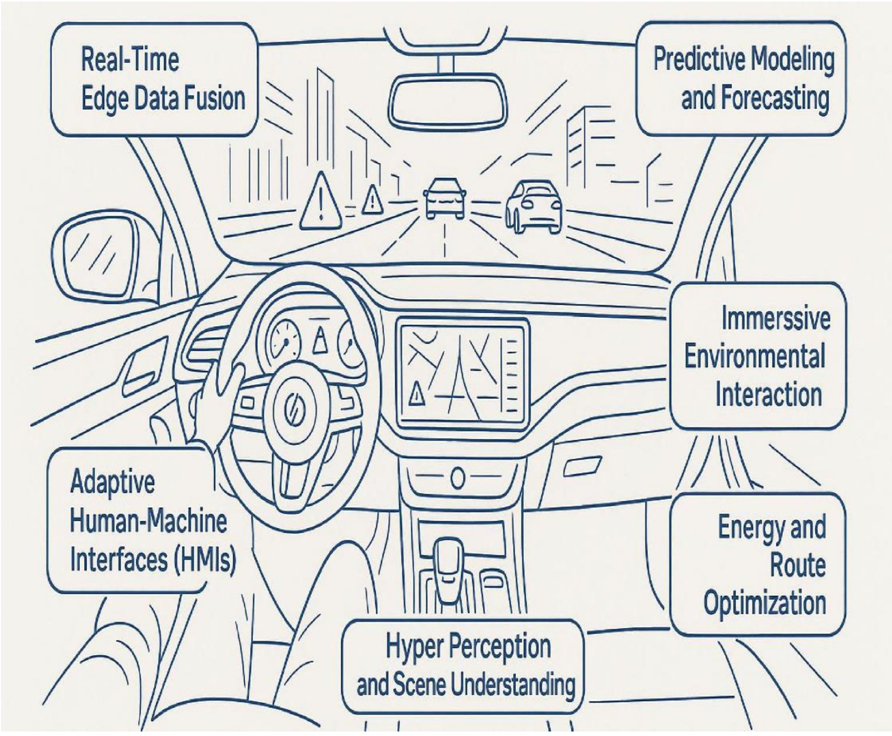
Energy Efficiency: Achieve ≥10% reduction in energy consumption.

Development Pathway: Progress from TRL 1-2 to TRL 4-5.

6/4/2025 28

Slide 29: AI+ – Selim: Immersive Cockpit: A Real-World Application of AI+ and Immersive Environments

IMMERSIVE COCKPIT: A REAL-WORLD APPLICATION OF AI+ AND IMMERSIVE ENVIRONMENTS



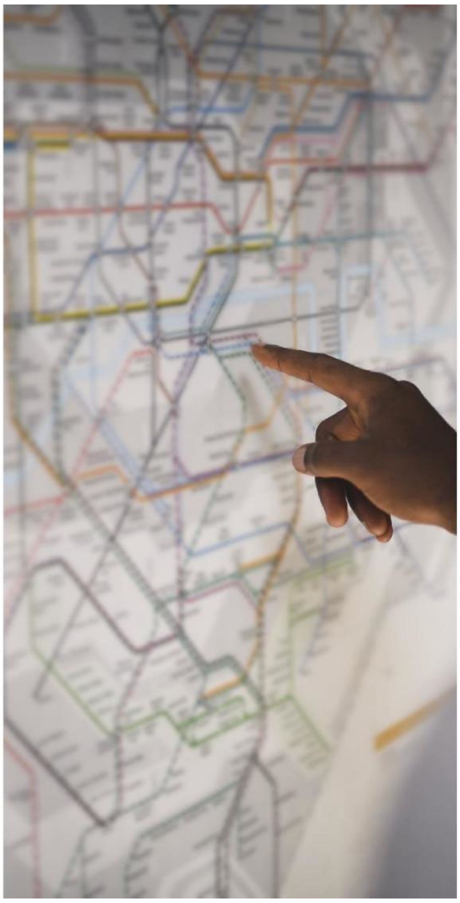
6/4/2025 29

Slide 30: AI+ – Selim: Key Enabling Technologies

KEY ENABLING TECHNOLOGIES

- AI+ Agents: Personalized in-cabin coaches using real-time data and LLMs
- Sensor Fusion: Integrates multimodal inputs for hyper-contextual feedback
- Digital Twins & UHD Maps: Create ground-truth environments for predictive assistance
- Edge AI Computing: Enables low-latency, privacy-preserving inference
- Multisensory HMIs: Combines AR, ambient lighting, haptics, and voice interaction

6/4/2025 30



Slide 31: AI+ – Selim: Future Outlook, Future Outlook

FUTURE OUTLOOK

MAIN FUNCTIONS

- Fatigue/stress detection, posture correction, and eco-coaching
- Gesture, eye-tracking, and voice control interfaces
- 6DoF scene visualization and semantic route overlays
- Human-in-the-loop fleet supervision via XR and digital twins

FUTURE OUTLOOK

FEATURE	BENEFIT
AI+ Coach	≥10% energy saving, ≥50% reduction in unsafe driving
Cross-modal Sync	<50ms latency across AR, haptics, and voice
Reduced Cognitive Load	Enhanced trust, safety, and comfort

Immersive cockpits will evolve into intelligent companions, guiding users through automated mobility with transparency, trust, and emotional engagement.

6/4/2025 31

Slide 32: AI+ – Selim: SURPRISE4EU – SC3 Demonstrator Highlights

SURPRISE4EU – SC3 DEMONSTRATOR HIGHLIGHTS

Demonstrator 3.1: Remote Fleet Supervision

- Real-time immersive interface for remote operators
- 360° live video, predictive overlays, XR-based situational awareness

Demonstrator 3.2: Immersive In-Cabin Coaching

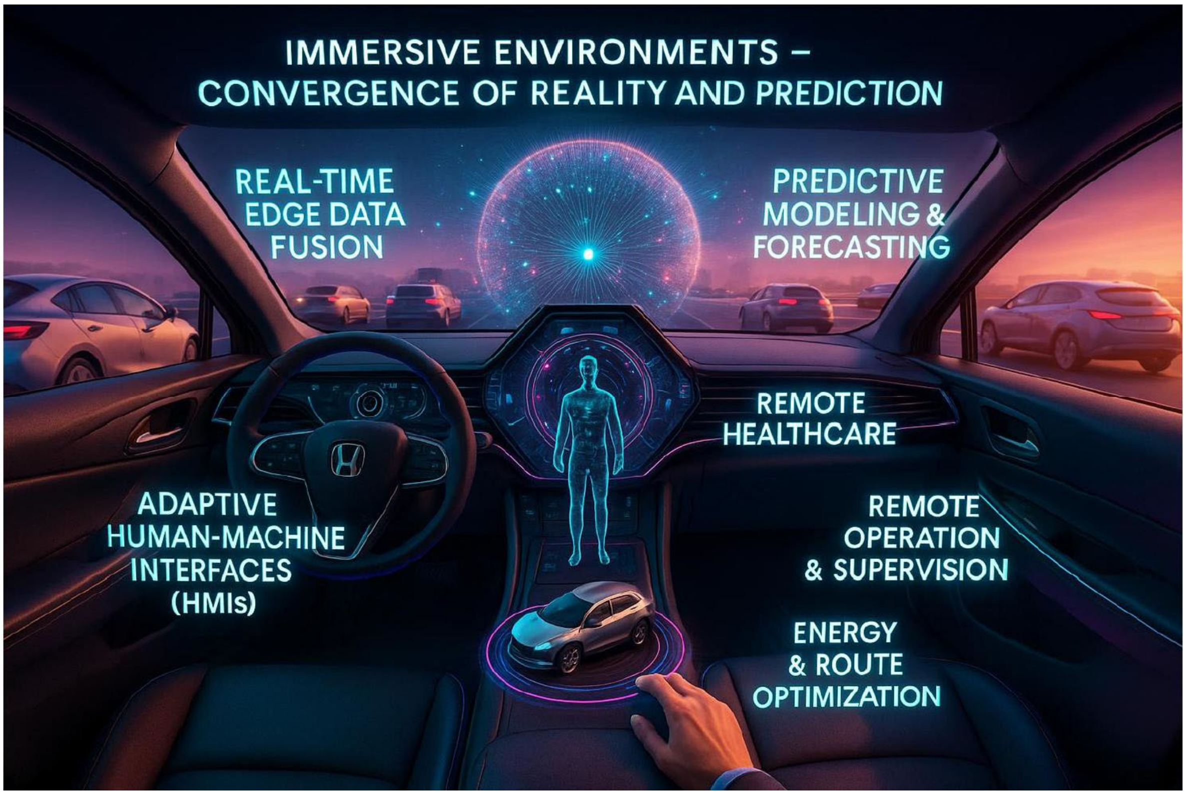
- Fatigue detection, posture correction, and voice-based vehicle status
- On-device LLM for natural language interaction

Demonstrator 3.3: Smart Co-pilot with ADAS

- Driver and occupant monitoring + external perception fusion
- Context-aware intervention and "Remedy Hub"

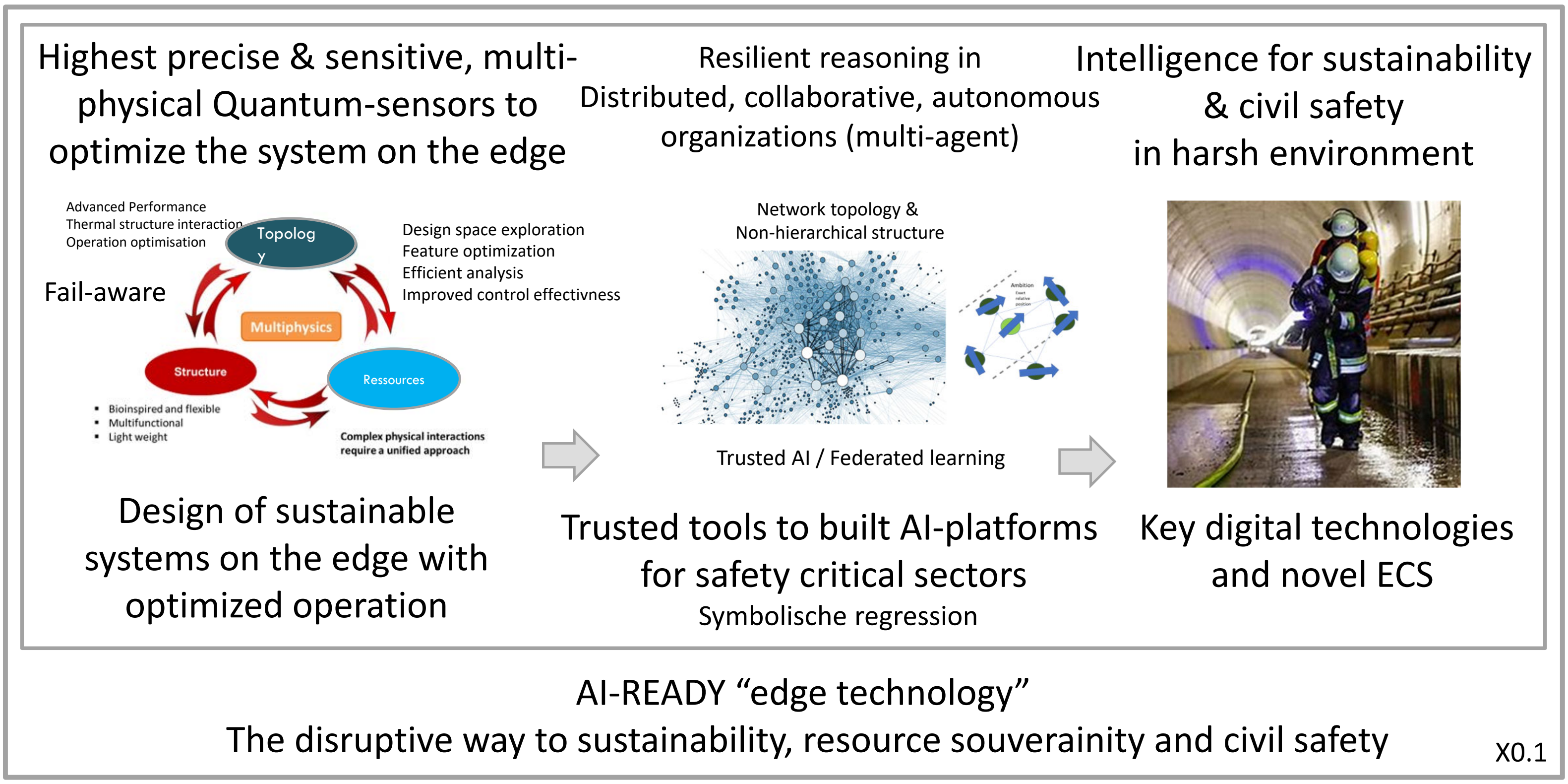
6/4/2025 32

Slide 33; AI+ – Selim; Immersive environments convergence of reality and prediction



6/4/2025 33

Reference architecture for resilient, collaborative reasoning driven by multi-physics in harsh environment
Fail-aware, predictive maintenance, self-diagnostic, self-monitoring, self-x, aprioi knowledge, empiric knowledge



A-IQ Ready receives funding within the Chips Joint Undertaking (Chips JU) - the Public-Private Partnership for research, development and innovation under Horizon Europe – and National Authorities under grant agreement n° 101096658.
ARCHIMEDES is supported by the Chips Joint Undertaking and its members, including the top-up funding by National Authorities under Grant Agreement No 101112295.
Cynergy4MIE is supported by the Chips Joint Undertaking and its members, including the top-up funding by National Authorities under Grant Agreement No 101140226.

Co-funded by the European Union

Platforms & Edge Computing (H3 – Slide 1)

1. Platforms as critical infrastructure
2. Edge computing for real-time responsiveness
3. Middleware for data flow management
4. Support for diverse hardware/software environments
5. Platforms as critical infrastructure for modern technology ecosystems
6. Edge computing enabling real-time responsiveness and scalability
7. Middleware: Essential for managing data flow between edge devices and cloud •infrastructure

1. Container-based virtualization ensuring efficient, secure deployment
2. Support for diverse hardware and software environments
3. Integration of advanced computational models, including neuromorphic computing
4. Role of hypervisors in secure and efficient resource allocation
5. Accelerated AI processing directly at the edge
6. Enhanced flexibility for deployment of critical applications
7. Reduction in latency and improvement in reliability

•6/4/2025 •3 4

Platforms & Sensor Integration (H3 – Slide 2)

1. Integration of quantum sensors, TinyML
2. Sensor fusion for improved decision-making
3. Standardized interfaces for interoperability
4. Integration of specialized sensors (Quantum sensors, TinyML)
5. Real-time data acquisition and processing
6. Sensor fusion capabilities enhancing decision-making
7. Increased accuracy and reliability through advanced hardware platforms
8. Adaptation of hardware and software architectures to support next-generation sensors
9. Novel computational strategies for energy-efficient sensor management
10. Implementation of hybrid cloud-edge strategies to maximize sensor efficiency
11. Role of standardized interfaces and interoperability
12. Improving system responsiveness and sustainability
13. Accelerating real-world application deployment

Sustainability & Ethics (H4 – Slide 1)

AVL

1. Ethical AI and compliance
2. Human-centric AI and user acceptance
3. Transparency, fairness, bias mitigation
4. Commitment to ethical AI practices and compliance
5. Human-centric AI approaches ensuring user acceptance
6. Continuous monitoring of ethical implications
7. Rapid standardization processes for fast and compliant deployment
8. Supporting sustainable digital ecosystems
9. Enhancing transparency and accountability in AI systems
10. Mitigation strategies for bias and ethical concerns
11. User feedback integrated into AI system design
12. Emphasis on transparency for explainability and fairness
13. Strengthening trust in AI-driven technologies

•6/4/2025 •3 6

Sustainability & Societal Impact (H4 – Slide 2)

1. Green Deal alignment
2. Digital transformation and resilience
3. Healthcare, mobility, infrastructure impact
4. Sustainable data economy via optimized data handling and storage
5. Contributions towards Europe's Green Deal
6. Ecological benefits from efficient resource management
7. Strengthening societal resilience through digital transformation
8. Facilitation of seamless regulatory compliance
9. Promotion of human-in-the-loop methodologies
10. Engagement and education to improve societal understanding of AI
11. Targeted benefits for healthcare, mobility, and infrastructure
12. Economic growth through sustainable innovation
13. Long-term societal benefits and enhanced quality of life

•6/4/2025 •3 7

Health Domain (Slides 1 & 2)

AVL

•**Demonstrator 1: Non-Invasive Cell Temperature Monitoring**

- **Description:** Integrates high-resolution, non-invasive infrared sensors for single-cell temperature measurement to detect early metabolic or pathological anomalies (e.g., cancer cell activation).
- **Goal:** Ultra-early diagnostics without invasive procedures, precise health monitoring.
- **Technology Stack Contribution:**
  - *Foundational Layer:* IR sensors
  - *Middleware:* High-precision measurement data integration
  - *Application Layer:* Cancer diagnostics, health surveillance
- **Partners:** QuantumCell Diagnostic (LifeScan), TuD, University Hospital Erlangen

•**Demonstrator 2: Micro-Robotic Cell Packaging**

- **Description:** Development of soft, AI-controlled microrobots to autonomously package and handle individual cells for diagnostics (e.g., lab-on-chip).
- **Goal:** Automation and standardization of sample preparation.
- **Technology Stack Contribution:**
  - *Crossfunctional Layer:* AI+ "Hausmeister" LLM control
  - *Middleware:* Robotics control and sensor integration
  - *Functional Layer:* Automation of biological handling
- **Partners:** QuantumCell Diagnostic (LifeScan), TuD, University Hospital Erlangen

•6/4/2025 •3 8

•Mobility Domain (Slides 3 & 4)

•Demonstrator 1: Immersive Cockpit

- **Description:** Multisensory, immersive in-vehicle environment to enhance situational awareness and driver interaction.
- **Goal:** Increased safety and stress-free driving experience.
- **Technology Stack Contribution:**
- *Middleware:* Real-time data fusion and immersive integration
- *Functional Layer:* Situational awareness, driver assistance systems
- *Application Layer:* Road safety
- **Partners:** VIF, AVL, IFAT

•Demonstrator 2: Smart Powertrain Diagnostics

- **Description:** Real-time diagnostics using Large Language Models (LLMs); proactive maintenance for electric vehicles (EVs) to improve reliability and efficiency.
- **Goal:** Optimization of vehicle operations through AI-driven diagnostics.
- **Technology Stack Contribution:**
- *Foundational Layer:* GaN semiconductor technology
- *Middleware:* Edge-based AI diagnostics
- *Functional Layer:* Predictive maintenance systems
- **Partners:** AVL, Valeo

•6/4/2025 •39

•Infrastructure Domain (Slides 5 & 6)

AVL

•Demonstrator 1: Smart Grid Load Estimation

- **Description:** AI-based load forecasting and interactive smart grid planning using LLMs and high-precision simulation.
- **Goal:** Optimization of grid planning and energy integration.
- **Technology Stack Contribution:**
  - *Foundational Layer:* Near-Memory Computing (NMC), MLIR-based accelerators
  - *Middleware:* Real-time data integration and cloud-edge continuum
  - *Functional Layer:* Forecast accuracy and load management
- **Partners:** OTH-AW, UPM, NVISION

•Demonstrator 2: Automated Bus Depot

- **Description:** AI-based autonomous control and optimization of bus depots, including automated parking, charging, and maintenance.
- **Goal:** Increased operational efficiency and reduced personnel needs.
- **Technology Stack Contribution:**
  - *Foundational Layer:* RISC-V processors, hardware accelerators
  - *Middleware:* Hybrid edge-cloud platform
  - *Functional Layer:* AI-driven routing optimization and autonomous navigation
- **Partners:** UNIKIE, VTT, SYSGO

•6/4/2025 •4 0

Impact and European Technological Leadership

- Health, digital sovereignty, Green Deal
- Economic, societal, ecological benefits
- Leadership potential

Conclusion & Future Perspectives

- Surprise4EU tech integration
- Opportunities for students: research, startups
- **Europe's vision in tech leadership**

ThankYou

- **Contact Information**

  Reiner John AVL List

  +49-1705625771

- **Invitation to follow-up discussions**

Q&A

- Open floor for audience questions
- Engage students in discussion


Towards a decentralised Federated Learning based edge-to-edge Continuum Computing framework
Nabil Abdennadher
MECO’2025, 10-14 June 2025, Budva , Montenegro

Hes·so

https://lsds.hesge.ch/meco-tutorial-2025

2

Hes·so

Universities of applied Sciences (UAS) in Switzerland

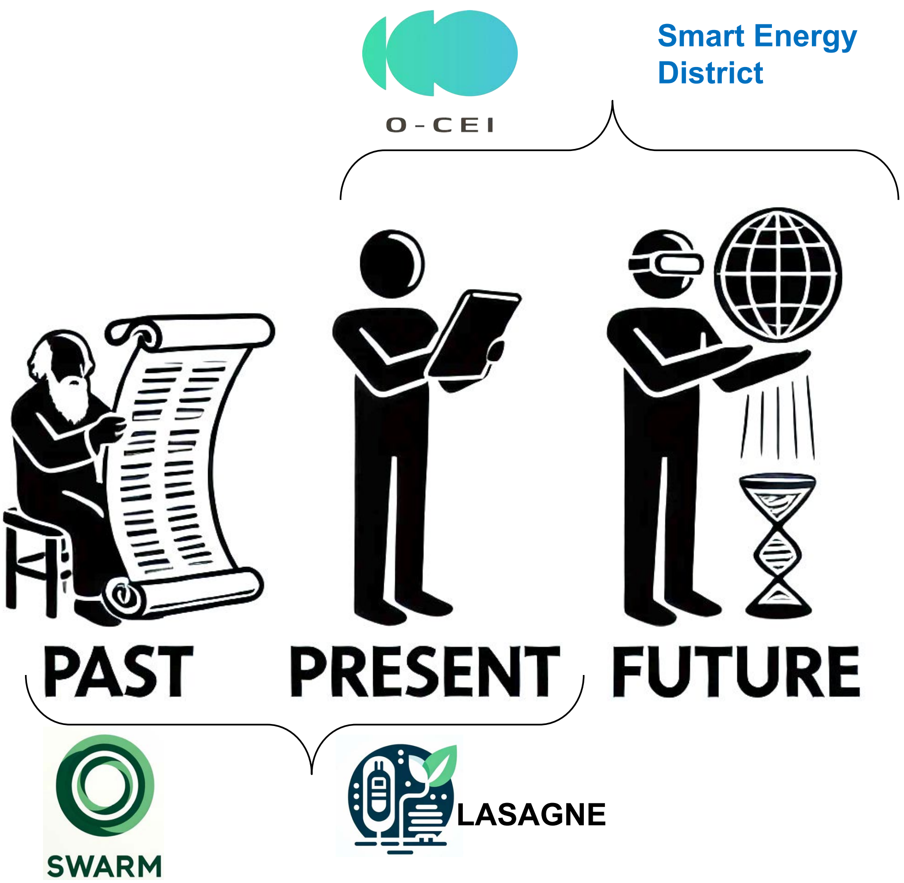
- UAS studies are practice-orientated, characterised by their direct links with the professional world.
- The UAS offer Bachelor's, Master's degree courses … and doctorates in collaboration with the universities.
- The UAS carry out practice-oriented research and innovation projects.
- One UAS in Western Switzerland: the **HES-SO** with 21'270 students.

3

Hes·so

- Founded in 1998, around 21,000 students
- 28 schools, 78 institutes and research units
- 68 Bachelor’s and Master’s programmes
- 19,161 employees (4,490 FTE, including **1,126 FTE for research**)
- Research Funding (2023)
  - Swiss confederation and 7 cantons: **46 M CHF**
  - External funding (Innosuisse, SNF, European projects, industrial contracts): **80 M CHF**
- Operating budget 2024 : 579 M CHF

Hes·so


O-CEI
Smart Energy
District
PAST
PRESENT
FUTURE
SWARM
LASAGNE

5

The SWARM Project

- **S**mart and **W**idely-distributed **A**ppliances for **R**enewable energy **M**anagement (SWARM)
- Applied research market driven project

6

Hes·so

The LASAGNE Project

digita**L** fr**A**mework for **S**m**A**rt **G**rid and re**N**ewable **E**nergy

Research project: IT and Social Sciences (social acceptability)

CLEMAP

UNIVERSITÉ DE GENÈVE
CENTRE UNIVERSITAIRE D'INFORMATIQUE

ELECTRICITY

Smart Energy Systems ERA-Net

Recap

KTH VETENSKAP OCH KONST

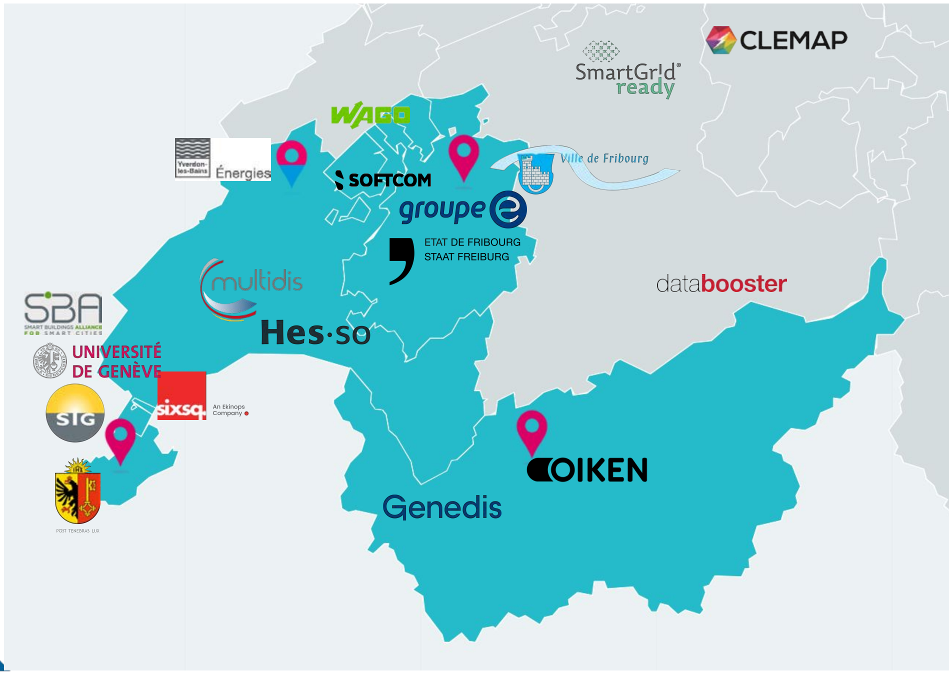
Hes·so

MEYRIN

7

Hes·so Smart Energy District

Flagship HES-SO project
2024-2027, 1.5 MCHF (~1.7 M €)



1. Develop a data storage/analysis platform
2. Develop a digital twin and decision support tools (AI)
3. Develop an **edge-to-cloud** platform for district energy management
4. Test the acceptability and sustainability of business models

Hes·so

Open CloudEdgeIoT Platform Uptake in Large Scale Cross-Domain Pilots (O-CEI)

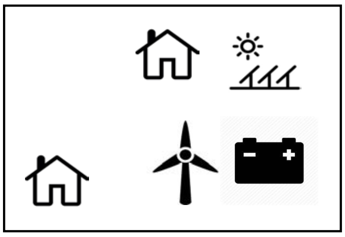
O-CEI

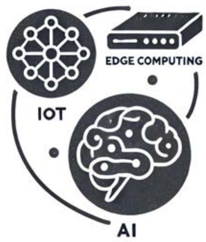
Budget : ~ 31 M €

56 partners:

- 18 Large Industrial
- 11 DSOs
- 14 innovative SMEs
- 13 non-profit

The context

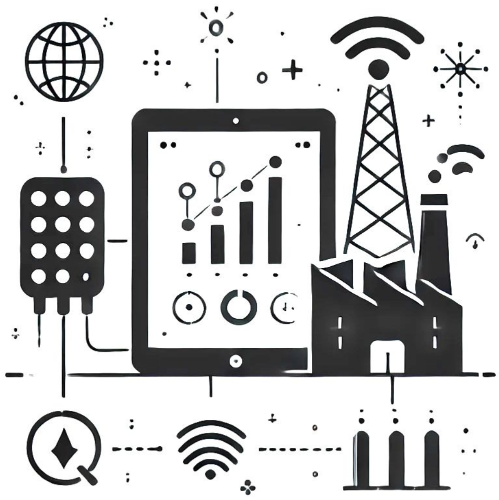




10

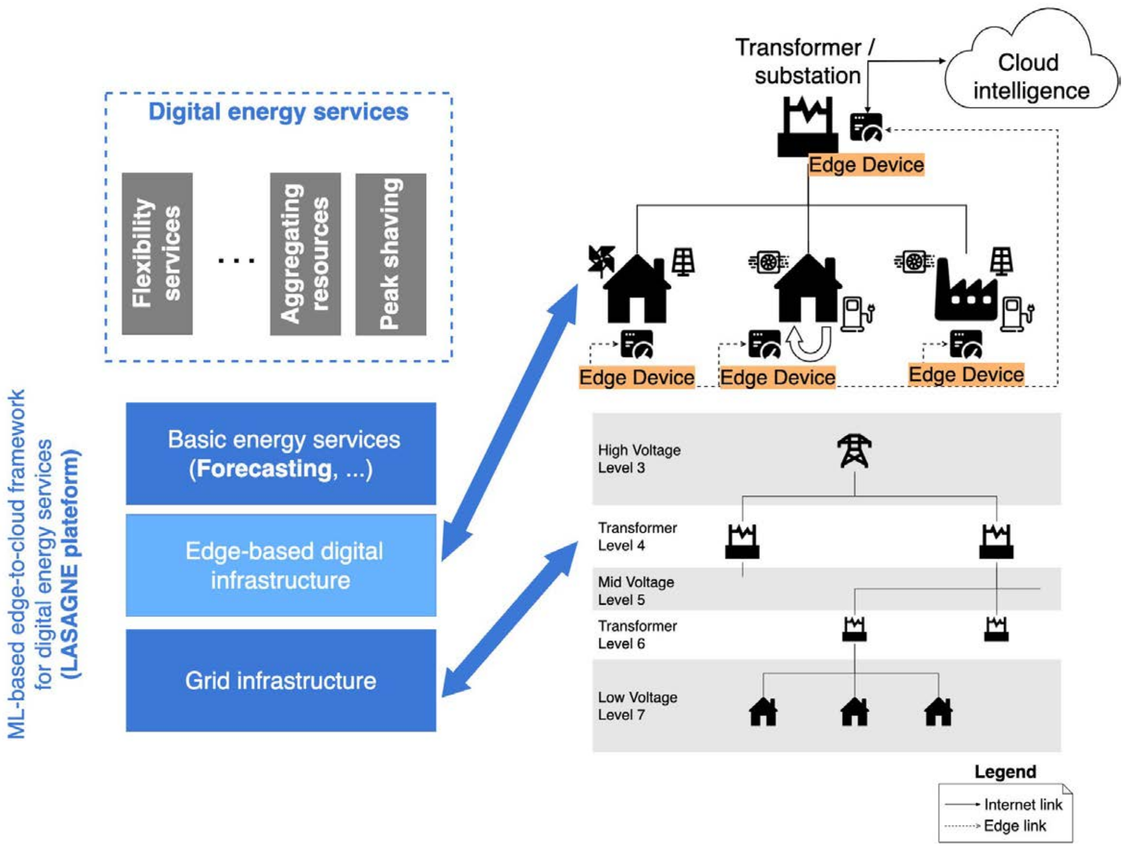
Hes·so Digital Energy services

- Trading energy and/or flexibility
- Peak shaving
- Aggregation resources
- …



11

Hes·so


Digital energy services
Flexibility services
...
Aggregating resources
Peak shaving
ML-based edge-to-cloud framework for digital energy services (LASAGNE plateform)
Basic energy services (Forecasting, ...)
Edge-based digital infrastructure
Grid infrastructure
Transformer / substation
Cloud intelligence
Edge Device
Edge Device
Edge Device
Edge Device
High Voltage Level 3
Transformer Level 4
Mid Voltage Level 5
Transformer Level 6
Low Voltage Level 7
Legend
Internet link
Edge link

12


CLEMAP Active Power (Watt)
Last 10
Last 60 = 5min
Last 720 = 1h
Last 5760 = 8h
Last 17280 = 24h
15
Minutes
Hours
180 values
hepia
Haute école du paysage, d'ingénierie et d'architecture de Genève
Réels
Prédictions
Historique
13

Hes·so

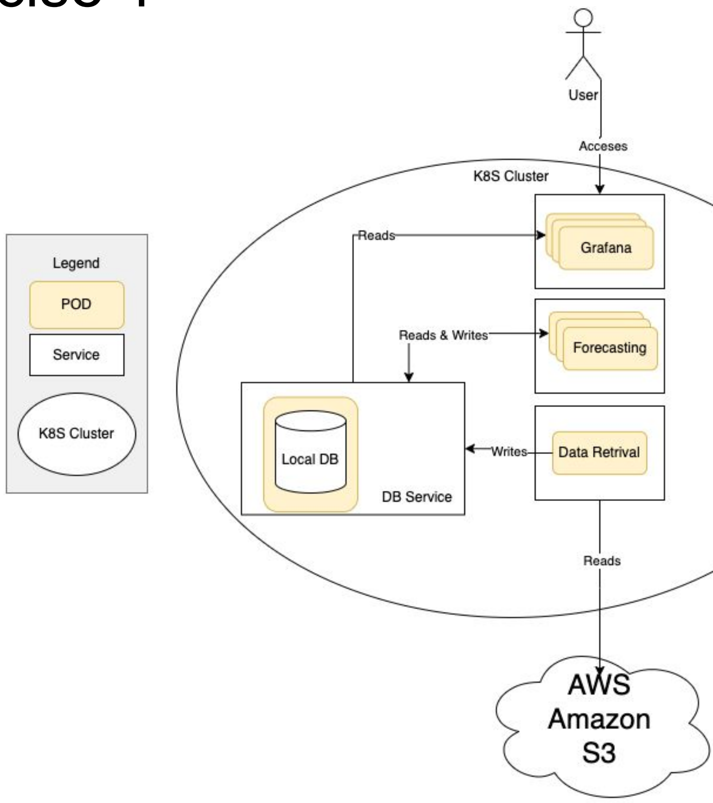
Exercise 1

Deploying a forecasting algorithm for energy consumption on a K8s cluster

14

Hes·so

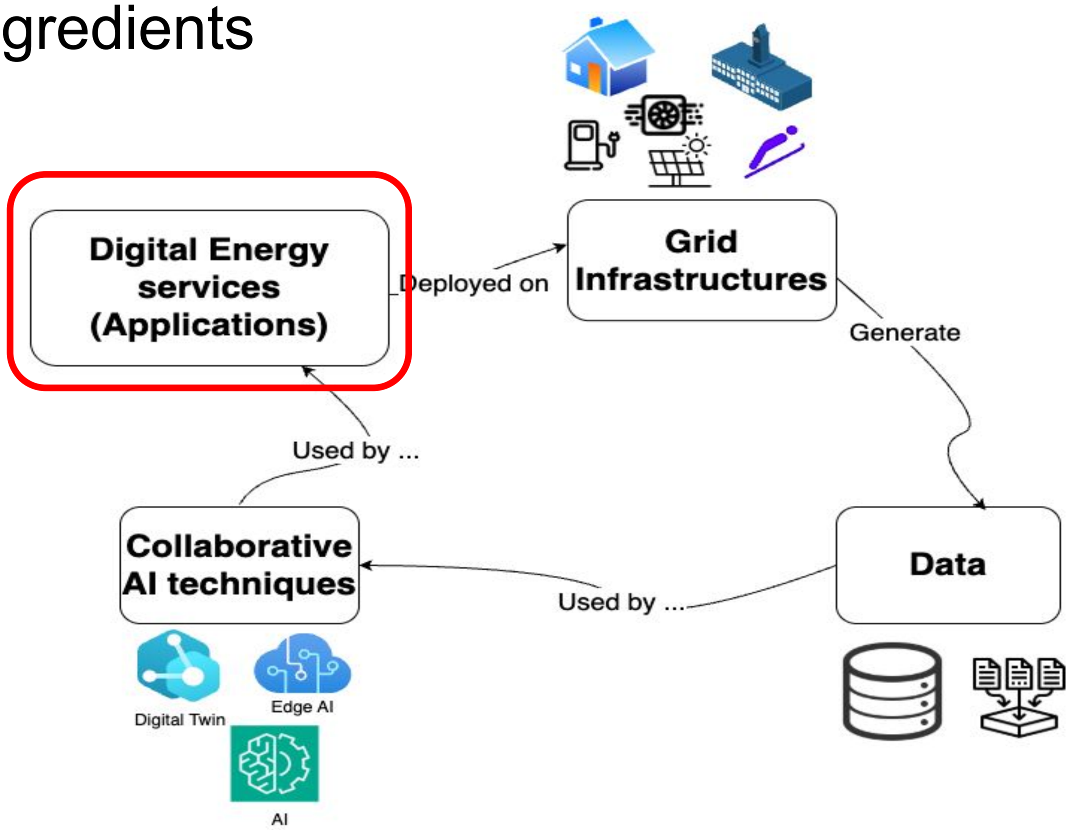
Exercise 1



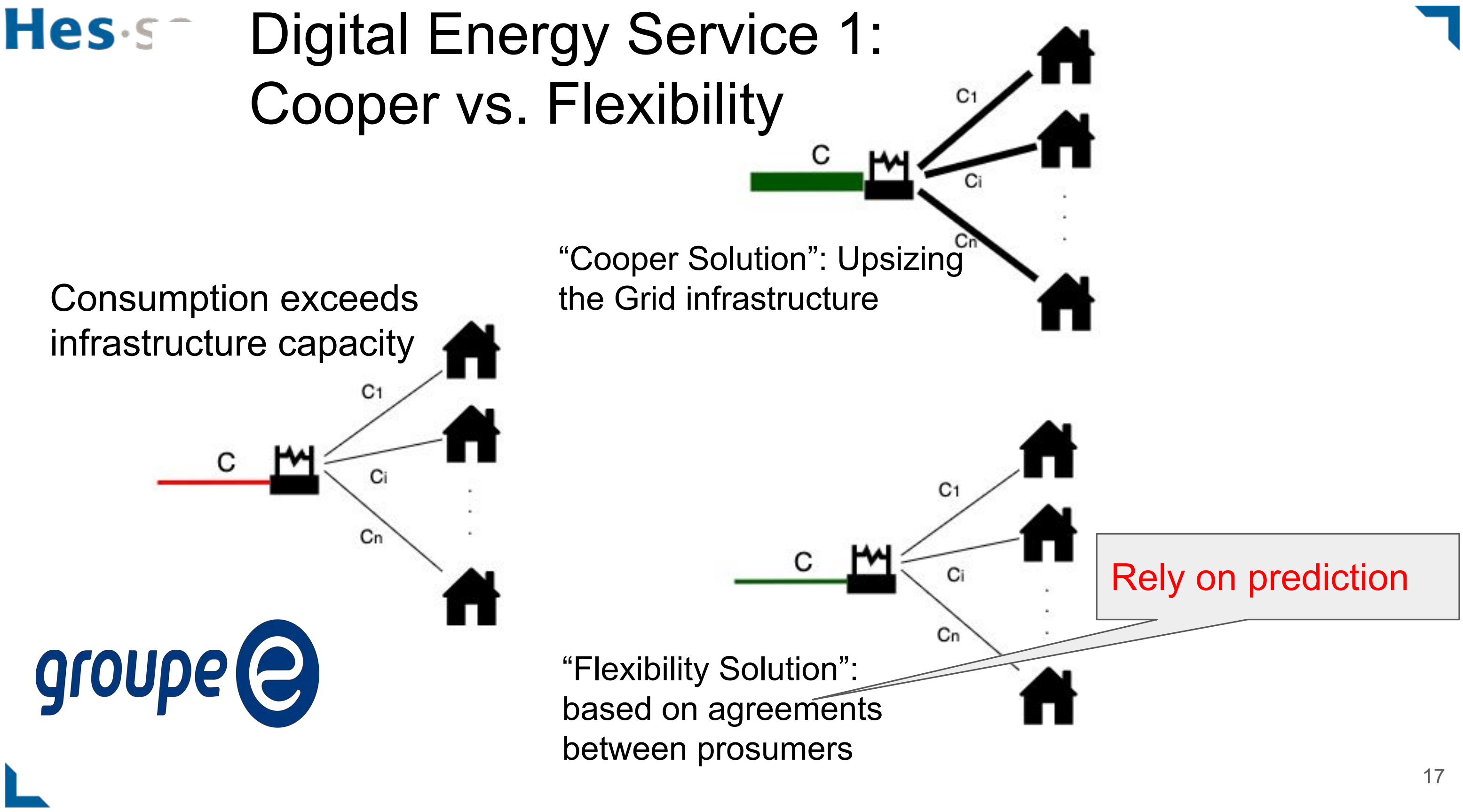
15

Hes·so

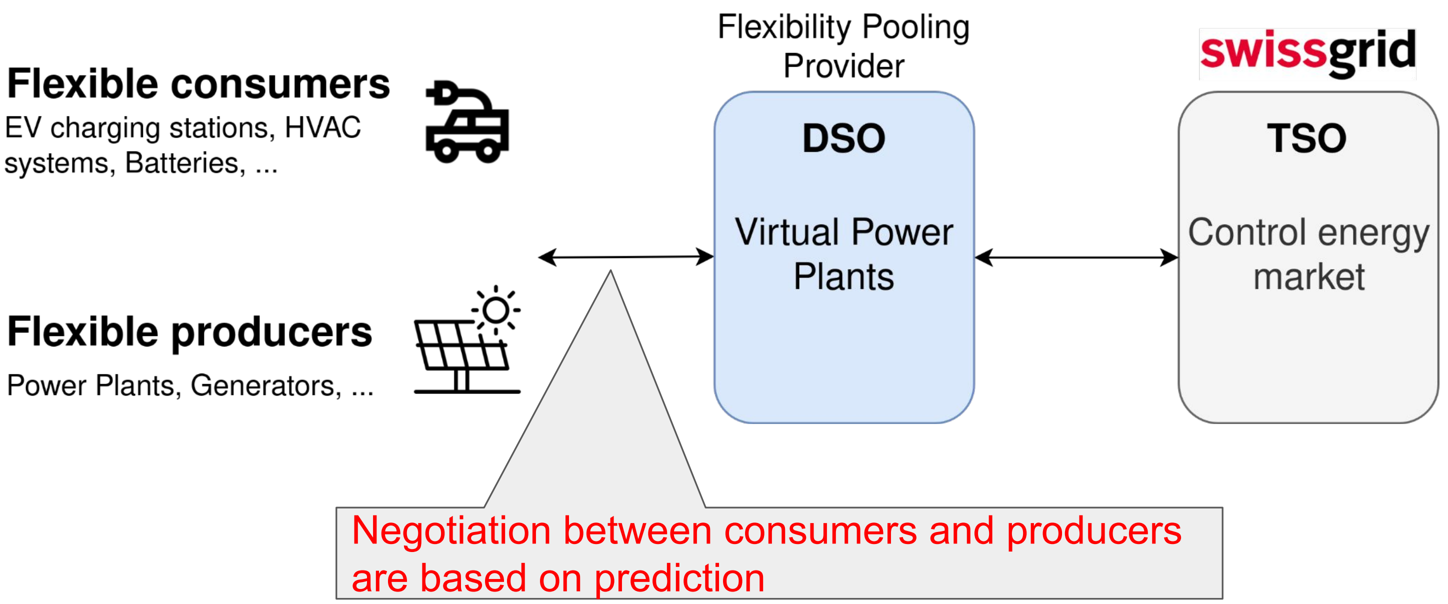
Ingredients



16


Digital Energy Service 1:
Cooper vs. Flexibility
Consumption exceeds infrastructure capacity
C
C1
Ci
Cn
"Cooper Solution": Upsizing the Grid infrastructure
"Flexibility Solution": based on agreements between prosumers
Rely on prediction
groupe e
17

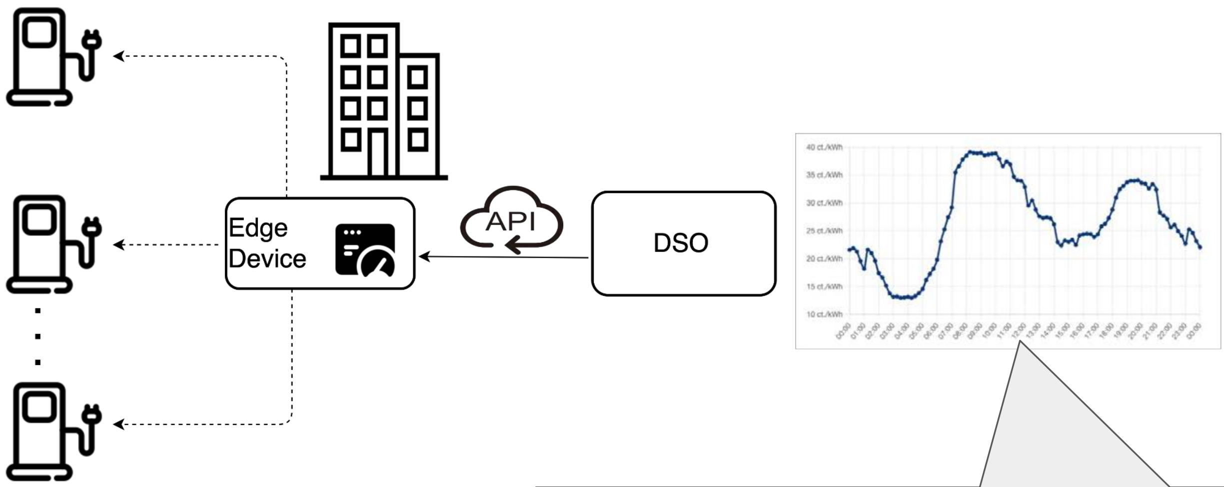
Digital Energy Service 2: Peak shaving, Energy/flexibility negotiation,



18

Hes·so

Digital Energy Service 3: Dynamic tariff



- Dynamic tariff (based on prediction)
- The electricity cost does not change during a given time interval

CLEMAP

19

Hes·so

Digital Energy Service 3: Dynamic tariff

- T: set of time intervals, indexed by t
- c_t: electricity cost during interval t (per kWh)
- e_t: energy consumed during interval t (kWh)
- E_{total}: total energy required to fully charge the EV (kWh)
- $P_{\max}$: maximum power of the charging station (kW)
- Δt: duration of each time interval (in hours), e.g., $\Delta t = 0.25$

1 Minimizing the Total Charging Cost (TCC)

$$TCC = \sum_{t \in T} c_t \cdot e_t$$

2 Charging must not exceed the battery's capacity.

$$\sum_{t \in T} e_t = E_{\text{total}}$$

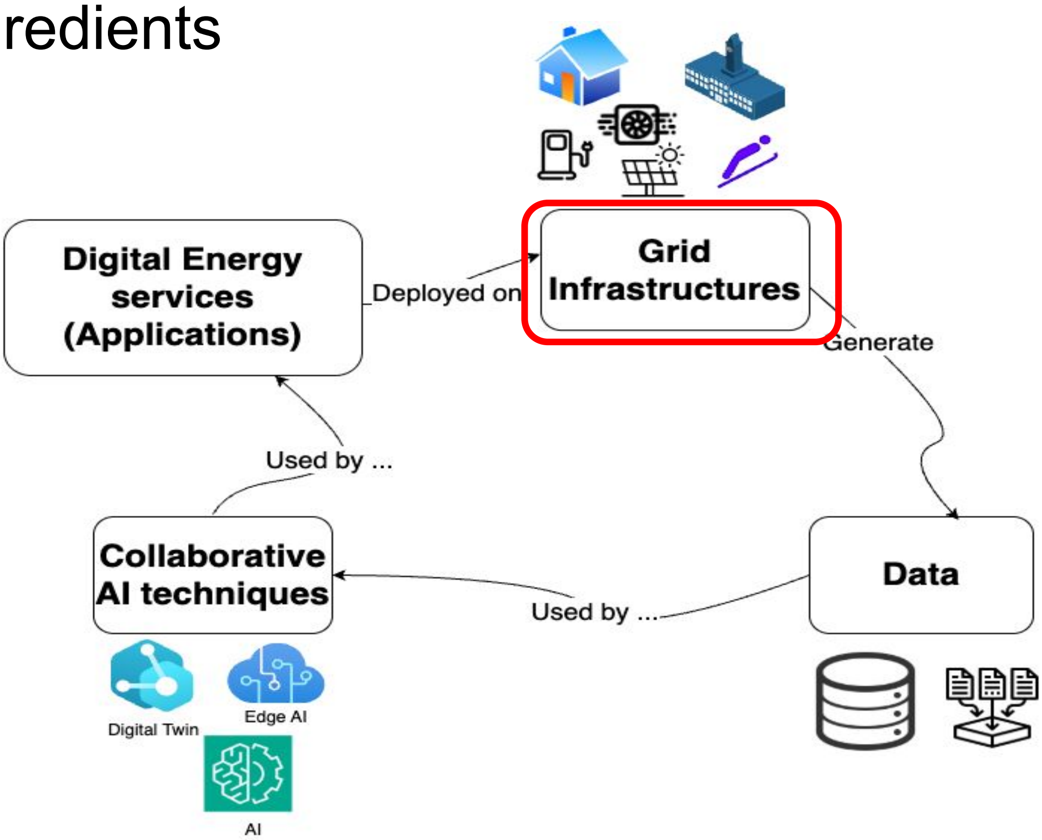
3 Charging must not exceed the charging station capacity.

$$0 \leq \frac{e_t}{\Delta t} \leq P_{\max}, \quad \forall t \in T$$

20

Hes·so

Ingredients



21

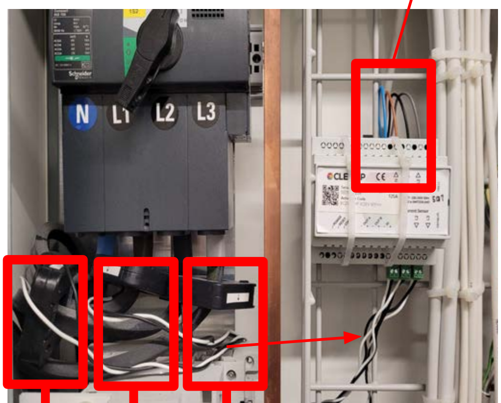
Hes·so

The edge device

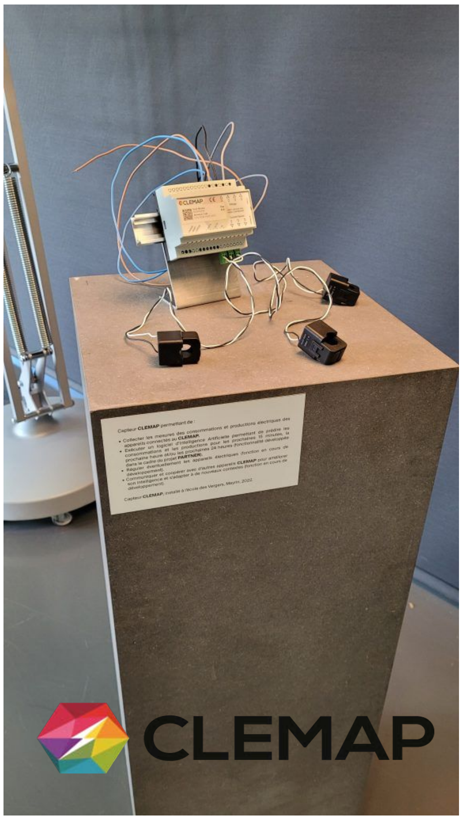
Linux (Raspberry)

Sensor : Voltage, Current

Voltage sensors



Current sensors



22

Grid infrastructures
Basel
Zurich
St Gallen
Berne
Schwyz
Fribourg
groupe e
Davos
Lausanne
Sierre
St Moritz
Geneva
Val d'Anniviers
CODHA
Polygones
MEYRIN
Lugano

Hes·so

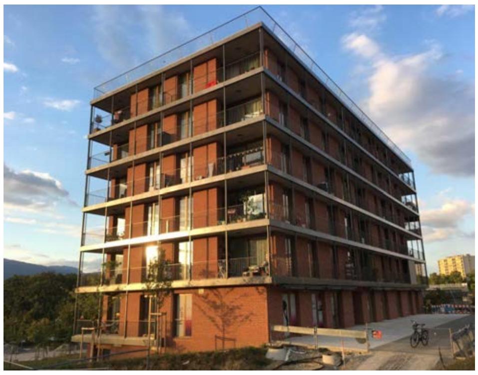
Geneva

Les vergers
Three buildings School (Gymnasium, Restaurant, classrooms)

MEYRIN

CODHA
20 buildings

CODHA
COOPÉRATIVE DE L'HABITAT ASSOCIATIF

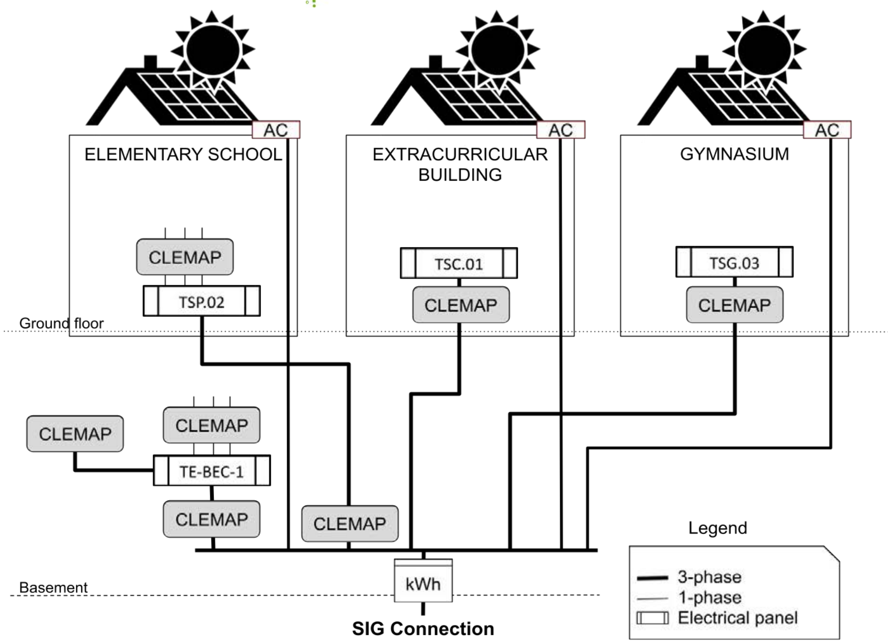


Polygones
5-floor building with around 28 apartments

Polygones

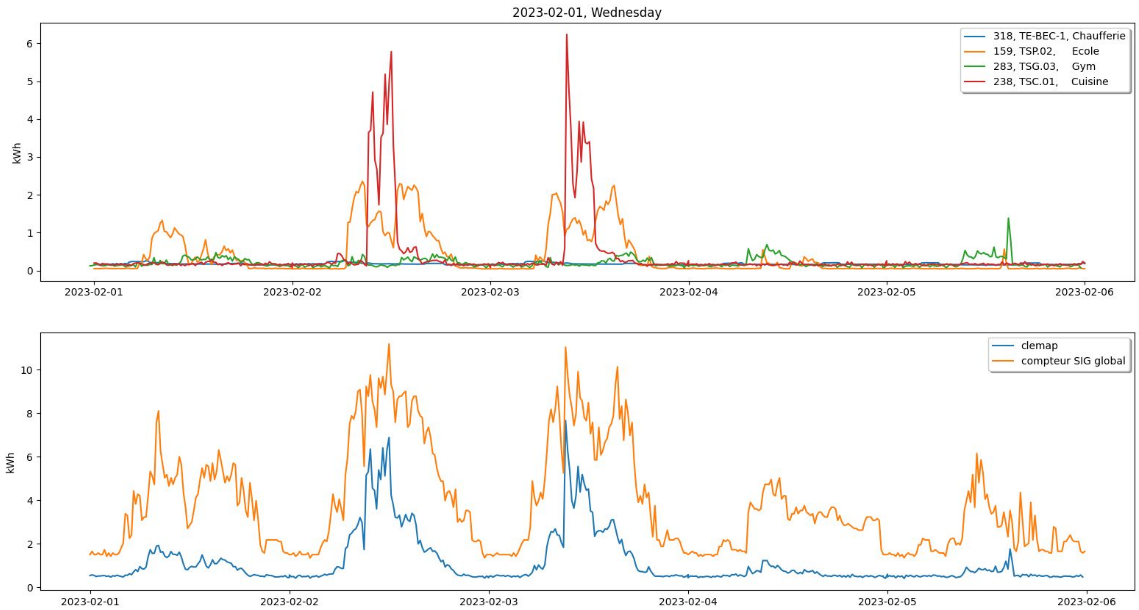
24

Vergers school (Meyrin)



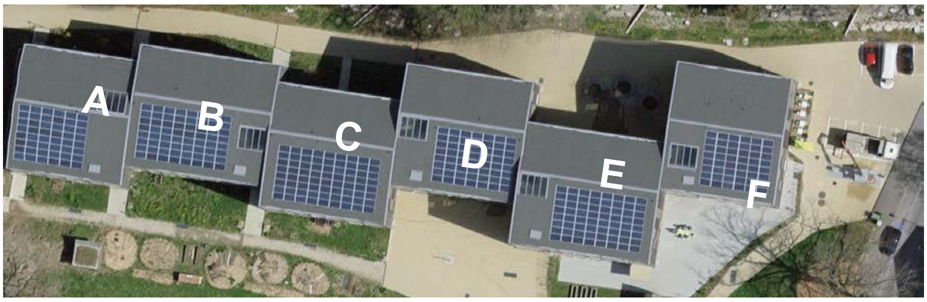
Hes·so

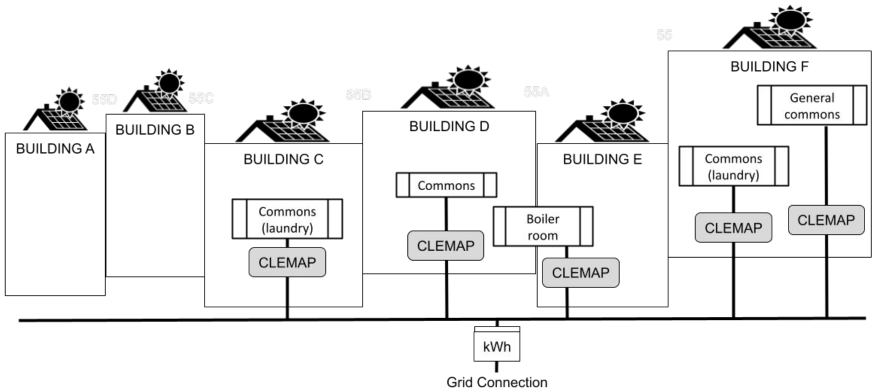
Power measurements - Vergers School



26

CODHA (Chêne-Bougeries)

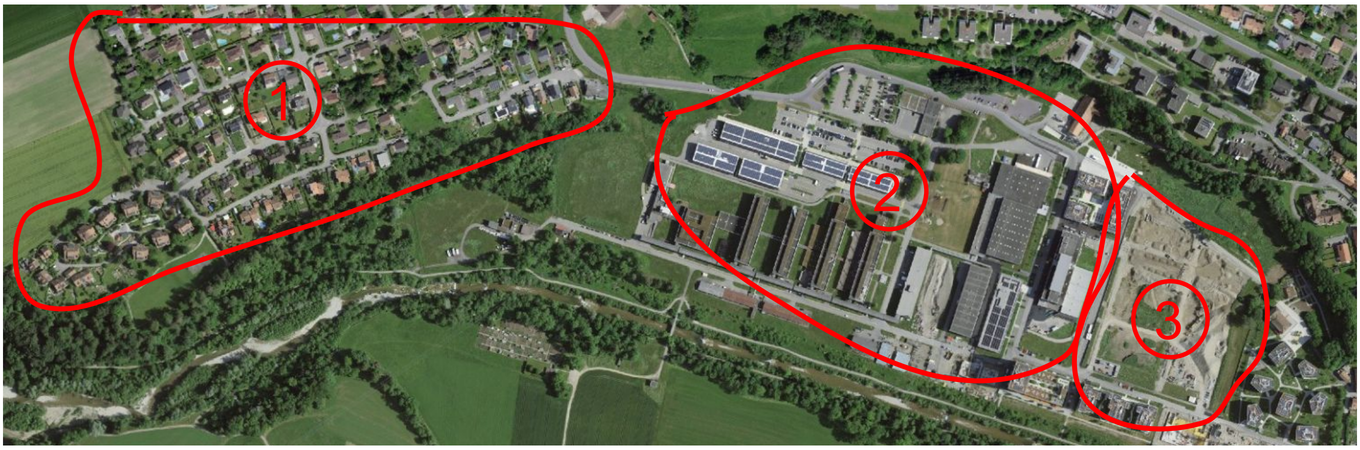




- **Tourist Offices/stations:**
- **Buildings**
- **Charging Stations**
- **Hotels**
- **Admin building**
- **Ski Lifts**
- **Recreational Facilities:** *Ice rink, swimming pool, spa*

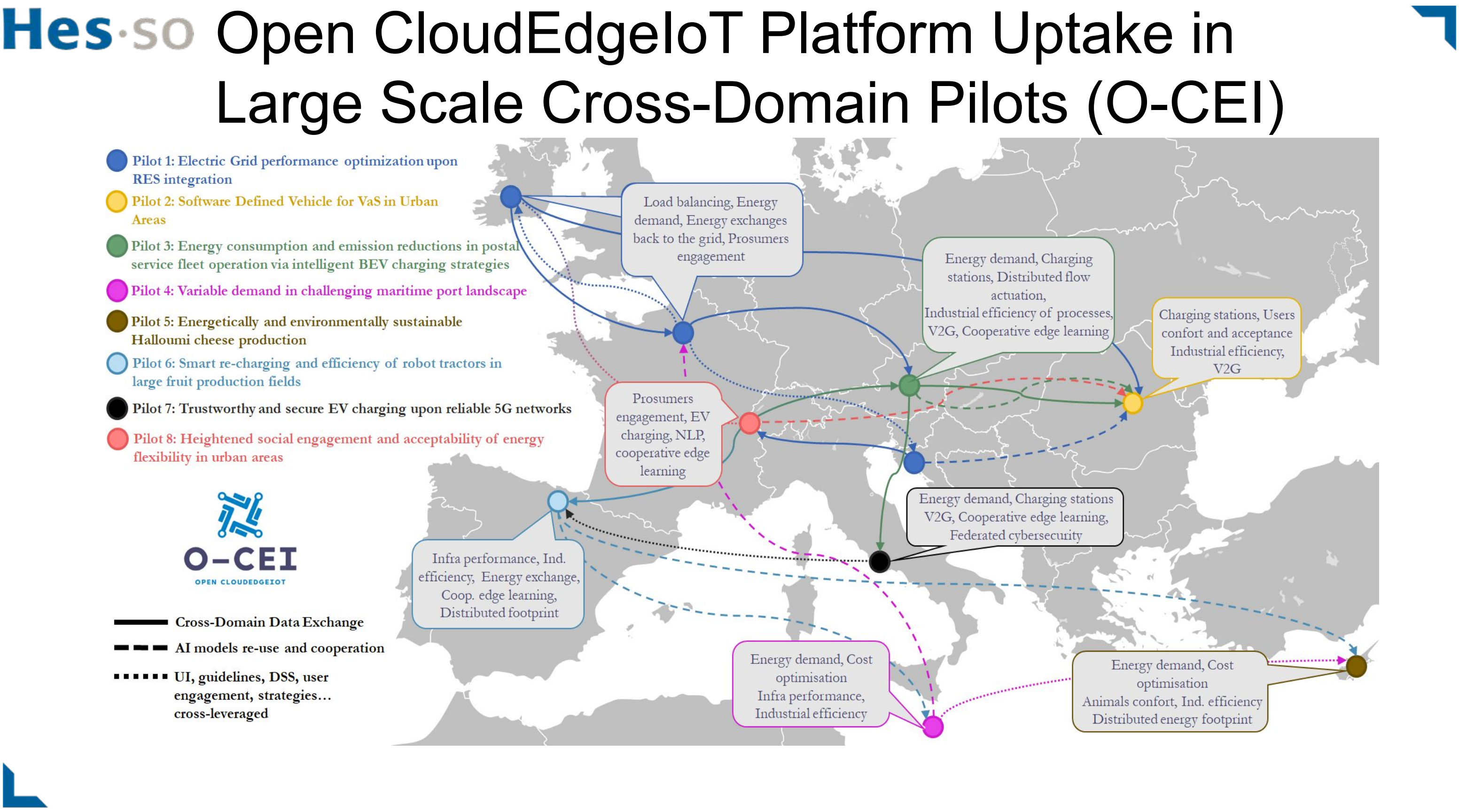
28

Hes·so Fribourg (Marly Innovation Centre)



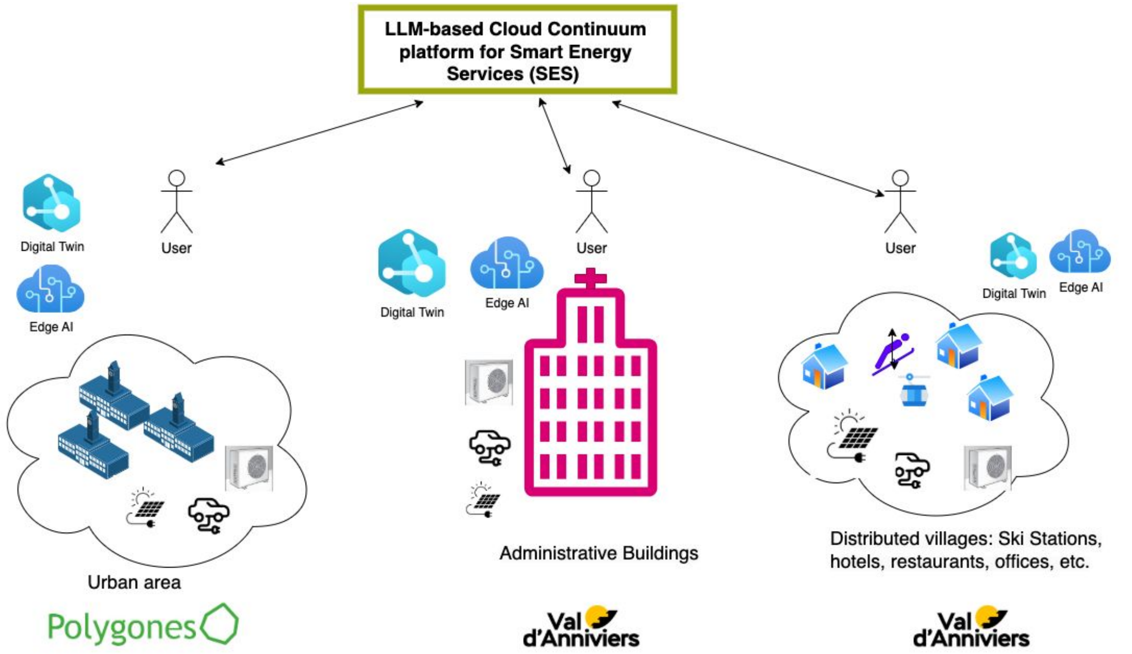
- Solar Panel capacity : **860 kWp** ; Annual production : ≈ **900 MWh/year**.
- Number of people on site : **>2,600**, Number of buildings : **>200**, Area : ≈ **80'000 m^2**
- ~70 Heat Pumps (HP), District Heating : **2 MW → 6 MW**

29


Hes·so
Open CloudEdgeIoT Platform Uptake in Large Scale Cross-Domain Pilots (O-CEI)
Pilot 1: Electric Grid performance optimization upon RES integration
Pilot 2: Software Defined Vehicle for VaS in Urban Areas
Pilot 3: Energy consumption and emission reductions in postal service fleet operation via intelligent BEV charging strategies
Pilot 4: Variable demand in challenging maritime port landscape
Pilot 5: Energetically and environmentally sustainable Halloumi cheese production
Pilot 6: Smart re-charging and efficiency of robot tractors in large fruit production fields
Pilot 7: Trustworthy and secure EV charging upon reliable 5G networks
Pilot 8: Heightened social engagement and acceptability of energy flexibility in urban areas
O-CEI
OPEN CLOUDEDGEIOT
Cross-Domain Data Exchange
AI models re-use and cooperation
UI, guidelines, DSS, user engagement, strategies… cross-leveraged
Load balancing, Energy demand, Energy exchanges back to the grid, Prosumers engagement
Energy demand, Charging stations, Distributed flow actuation, Industrial efficiency of processes, V2G, Cooperative edge learning
Charging stations, Users confort and acceptance Industrial efficiency, V2G
Prosumers engagement, EV charging, NLP, cooperative edge learning
Energy demand, Charging stations V2G, Cooperative edge learning, Federated cybersecurity
Infra performance, Ind. efficiency, Energy exchange, Coop. edge learning, Distributed footprint
Energy demand, Cost optimisation Infra performance, Industrial efficiency
Energy demand, Cost optimisation Animals confort, Ind. efficiency Distributed energy footprint

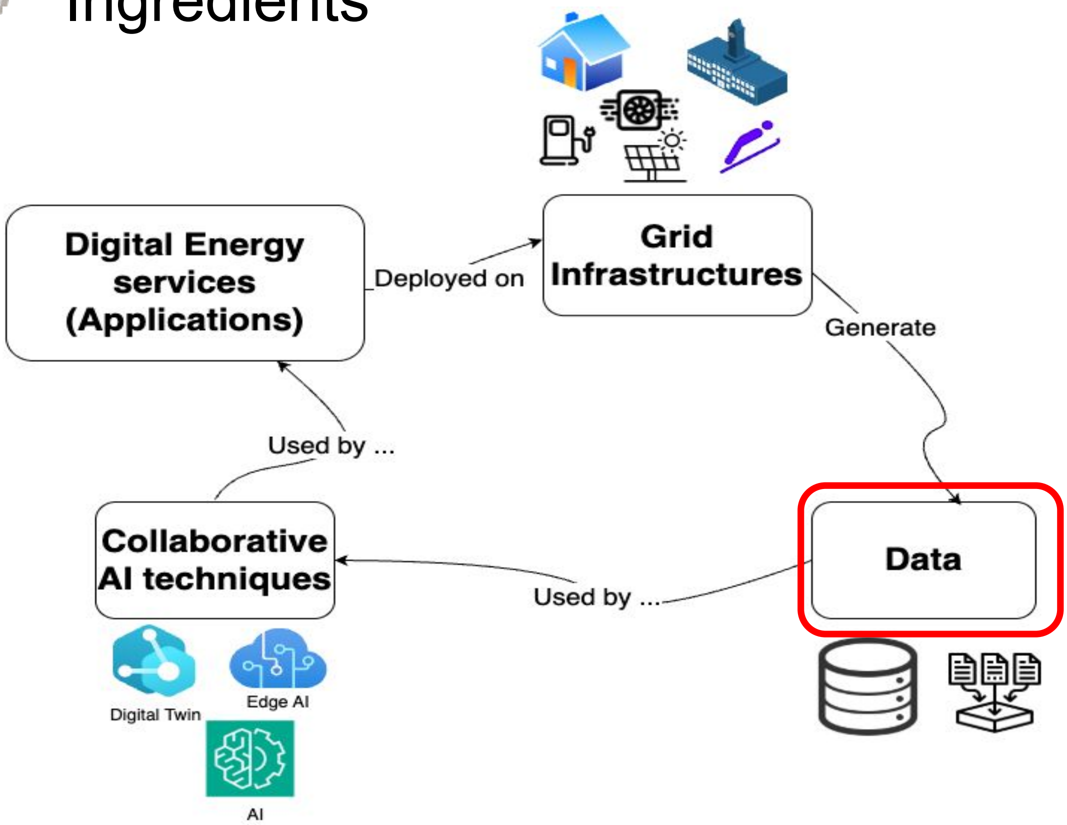
Hes·so

Open CloudEdgeIoT Platform Uptake in Large Scale Cross-Domain Pilots (O-CEI)



31

Ingredients



32

Hes·so

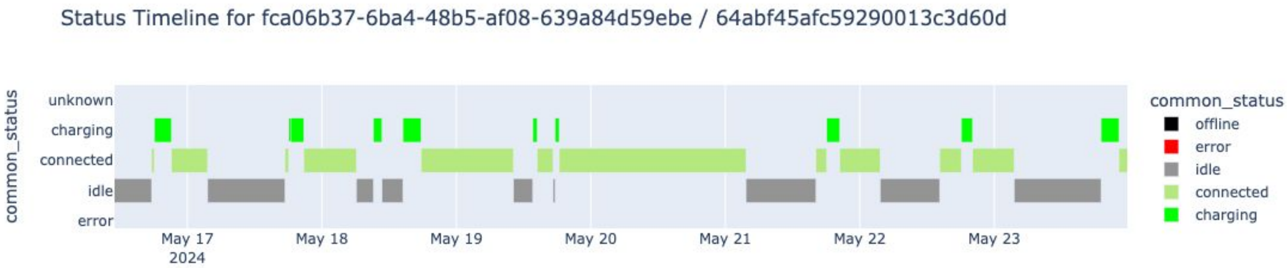
Dataset

Deployment	Devices	Time frame	Sampling time	Values
groupe e	1153 smart meters - Households - Shops - Industrials - etc.	2 years 2021 / 2022	15 min	Energy usage/production (V and I for the 3 phases)
MEYRIN	7 at Vergers school in Meyrin (GE)	Since June 2022	1 min	V, I, Power for each 3 phases
CODHA COOPÉRATIVE DE L'HABITAT ASSOCIATIF	5 at CODHA in Chêne-Bougeries (GE)	Since November 2023		
Polygones	25 apartments 4 weather stations Common area	Around 5 years	1 day	Electrical energy Heating Water consumption Inside/outside temperature

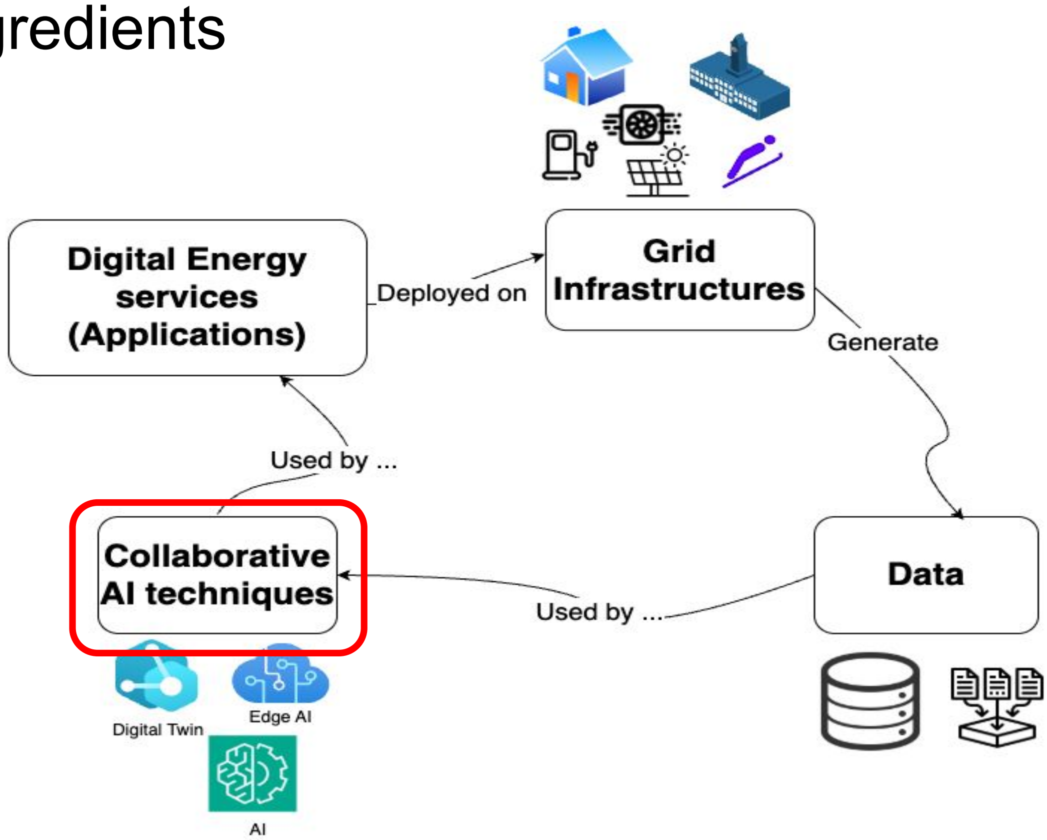
Hes·so

Dataset (E-vehicle's charging stations)

- Period : 01.01.2024 - 31.12.2024
- Number of charging stations: ????
- 1.2 billion raw data records
- Frequency: 30 seconds
- Collected measure : Current (Amp)

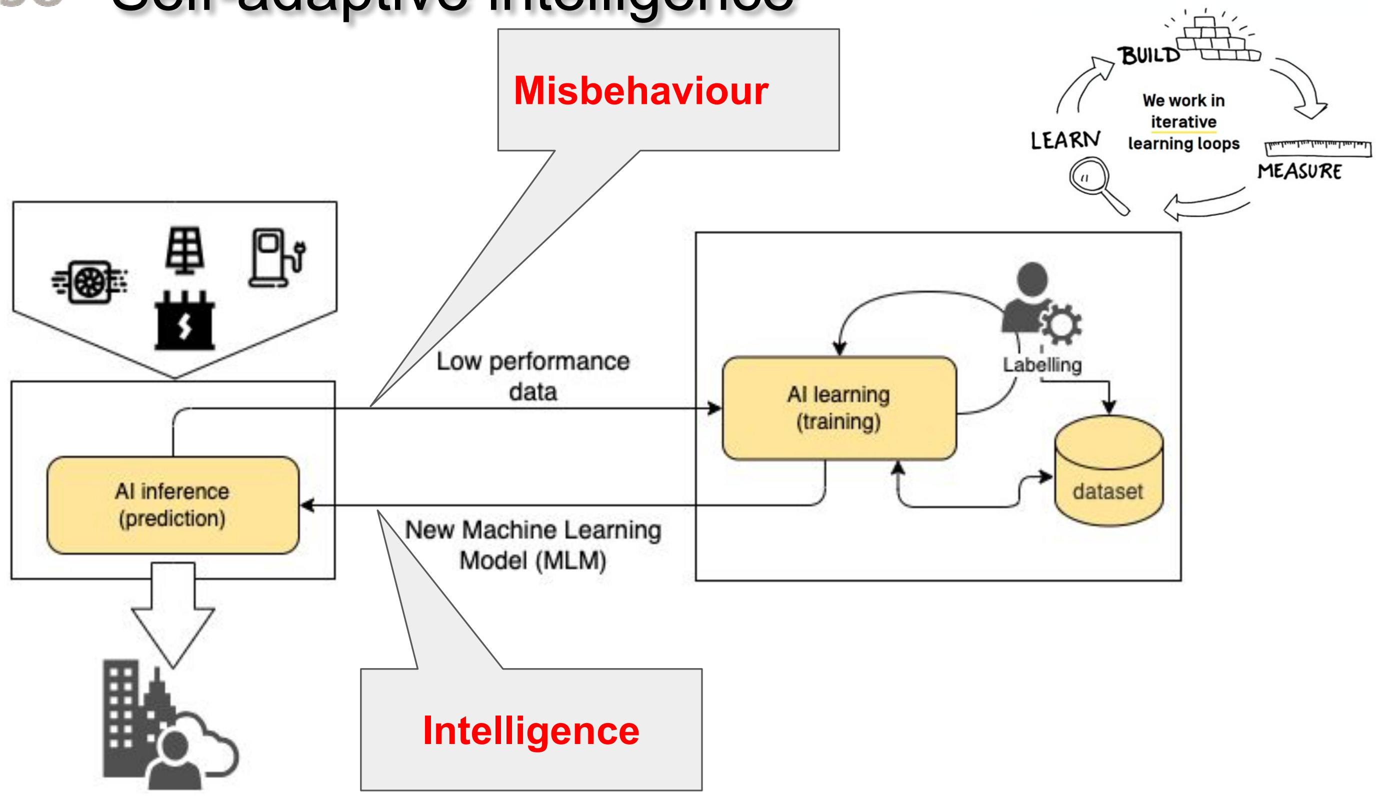


Ingredients



35

Self-adaptive intelligence



36

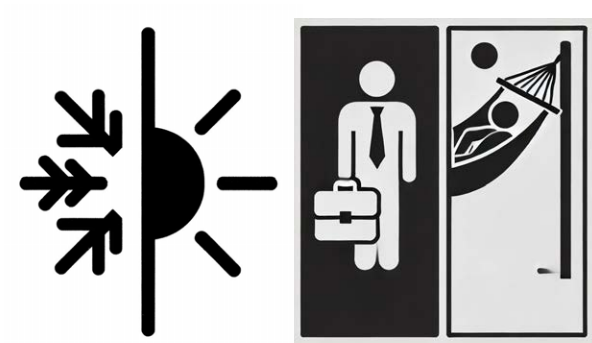
Hes·so

What is "Context aware" intelligence ?

A self-adaptive intelligence that distinguishes between the following two cases:

A mistake (a misbehaviour)

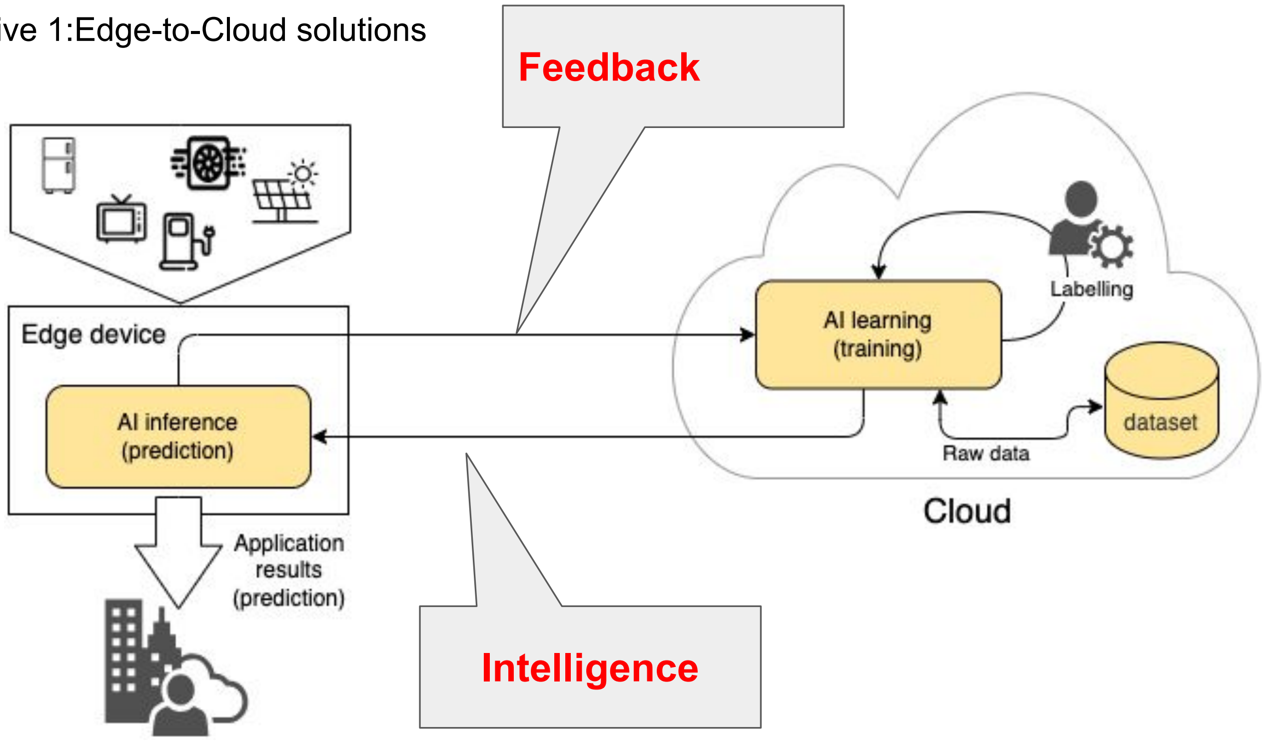
A context shift



37

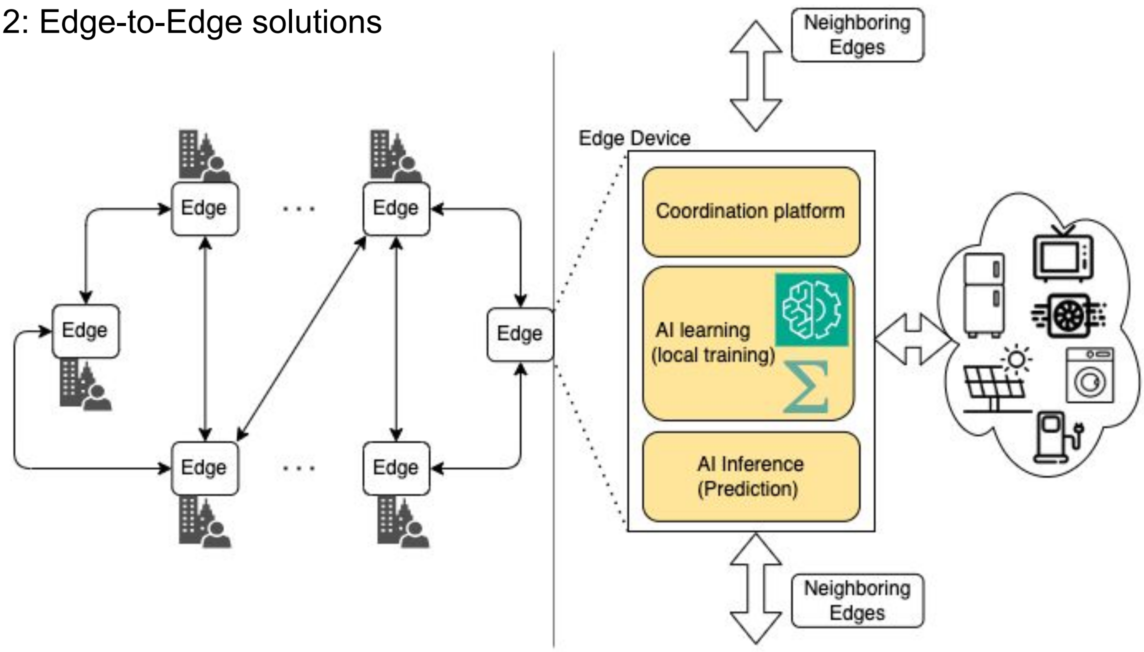
Edge-to-Cloud deployment

Alternative 1:Edge-to-Cloud solutions



Edge-To-Edge deployment

Alternative 2: Edge-to-Edge solutions



39

Hes·so

Machine Learning algorithms

1. XGBoost
2. LSTM, CNN-LSTM. CNN-LSTM with attention mechanism
3. DeepAR (by Amazon)
4. Temporal Fusion Transformer (By Google)
5. Prophet (by Facebook)

40

Machine Learning algorithms

Algorithm Template of a Training Algorithm

Require: Training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, model f_θ, loss function $\mathcal{L}$, optimizer, number of epochs E

Ensure: Trained model parameters θ

```
Initialize model parameters θ
for epoch = 1 to E do
    Shuffle D
    for each batch B ⊂ D do
        Compute predictions: ŷ = f_θ(x) for (x, y) ∈ B
        Compute loss: L = ℒ(ŷ, y)
        Compute gradients: ∇_θ L
        Update parameters: θ ← Optimizer(θ, ∇_θ L)
    end for
    if Early stopping criteria met then
        break
    end if
end for
return θ
```

XGBOOST, LSTM, NN, linear regression, decision tree, etc.

Tells us how to update the parameters of the model to reduce the loss

The optimizer uses the result of the gradient to adjust the parameters of the model

Hes·so

XGBOOST

Builds an ensemble of decision trees sequentially, where each new tree corrects the errors made by the previous ones:

$$\hat{y}^{(t)} = \hat{y}^{(t-1)} + \eta \cdot f_t(x)$$

A tree is represented by a function

$$\hat{y}_i = \sum_{k=1}^{K} \eta \cdot f_k(x_i)$$

- $\hat{y}_i$: predicted output for input x_i
- f_k: the **k-th decision tree** (a regression tree)
- η: learning rate (scaling factor)
- K: total number of trees

42

Long Short-Term Memory (LSTM)

1. Relies on Recurrent Neural Network
2. Retains important information for long periods
3. Forgets irrelevant data

43

DeepAR

Not just a point estimate

1. Developed by Amazon
2. Relies on LSTM
3. Designed for probabilistic, multi-step time series forecasting across many related time series.

Instead of building a separate model for each time series, DeepAR trains a **single global model** that **learns patterns across all series**.

44

Temporal Fusion Transformer

1. Developed by Google
2. Designed for time series prediction
3. Combines:
    a. **Recurrent networks** (like LSTM) for local temporal dependencies,
    b. **Transformer-style attention** for long-term relationships,
    c. **Feature selection networks** for interpretability,
    d. **quantile forecasting** for uncertainty estimation.

Struggle to retain information very far back in time

What happened several time steps ago still affects the future

Represent ranges of possible outcomes, not just an average

45

Hes·so

Prophet

1. Developed by Facebook (meta)
2. Designed for time series prediction

$$\hat{y}(t) = g(t) + s(t) + h(t) + \epsilon_t$$

- $\hat{y}(t)$: predicted value at time t
- $g(t)$: trend component
- $s(t)$: seasonal component
- $h(t)$: holiday/events component
- ϵ_t: error/noise term

46

Hes·so

Metrics

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\text{RMSE} = \sqrt{MSE}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

$$\text{NRMSE} = \frac{\text{RMSE}}{\max(y) - \min(y)}$$

$$\text{sMAPE} = \frac{100}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i - y_i|}{(|y_i| + |\hat{y}_i|)/2}$$

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

47

Hes·so

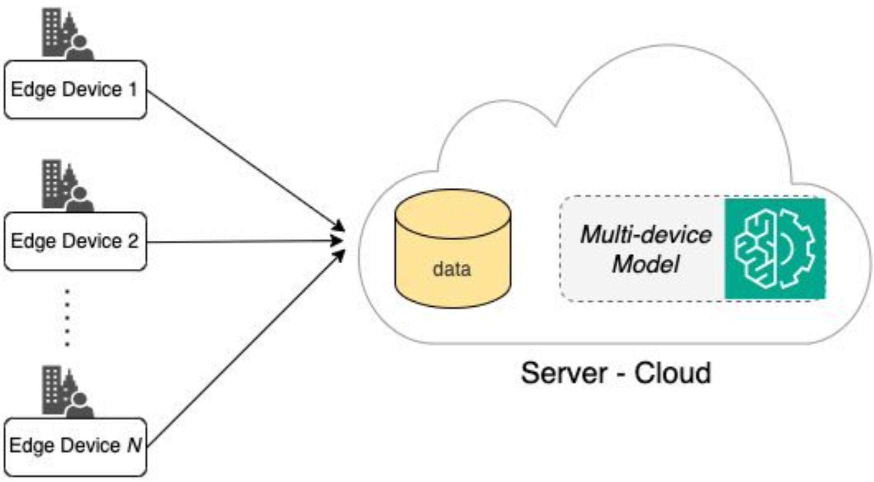
Metrics

1. XGBoost
2. **LSTMs family**
3. Transformer
4. Prophet
5. DeepAR

Model	RMSE (kW)	MAE (kW)
LSTM	0.381	0.209
XGBoost	0.392	0.225
Transformer	0.474	0.337
DeepAR	0.533	0.429
Prophet	0.612	0.487

48

ML approach: Multi-Device model

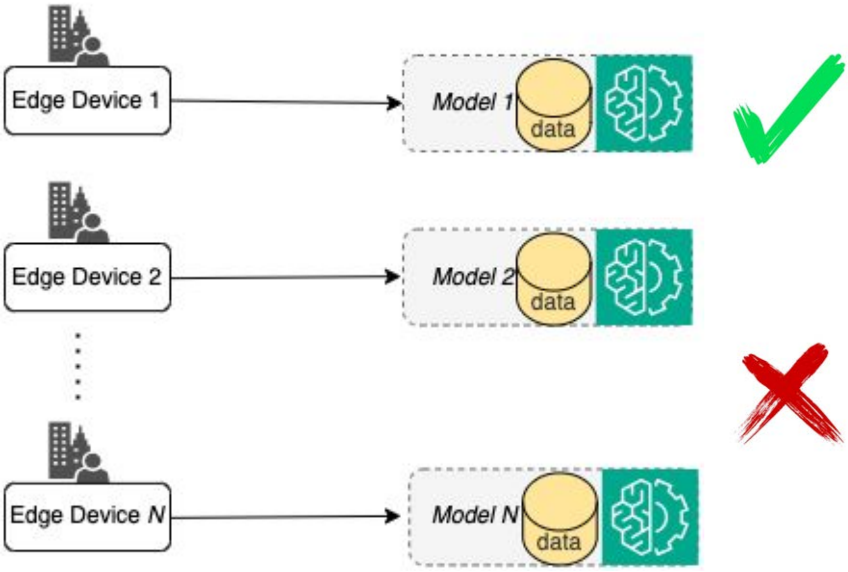


✓
- Reusability

✗
- Privacy
- Scalability
- Aggregation

49

ML approach: Single-Device model



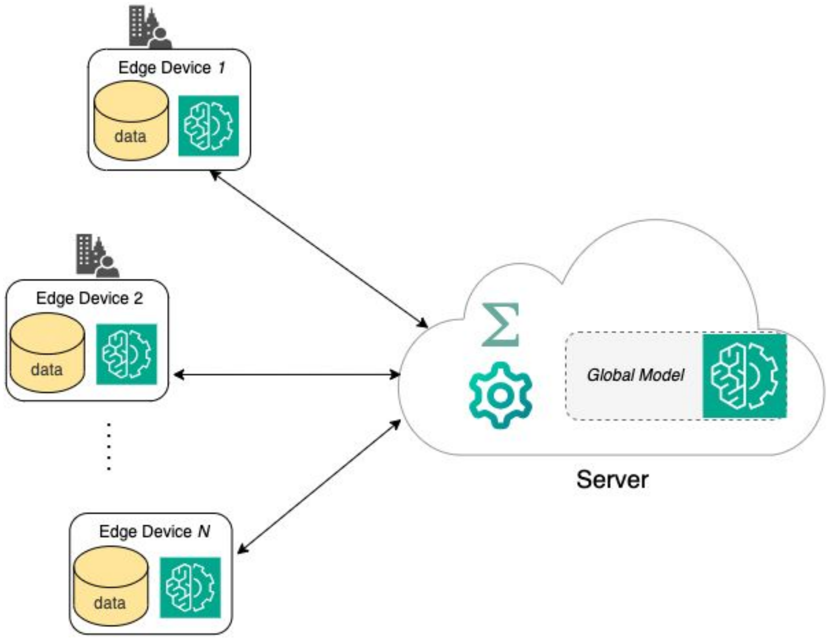
- Customised models
- Specific anomalies/habits are easily detected

- Scalability issue & Maintenance overhead

50

Hes·so

ML approach: Federated Learning



- Reusability
- Privacy
- Scalability (!)

- Aggregation
- Scalability (!)

51

Hes·so

Machine Learning algorithms

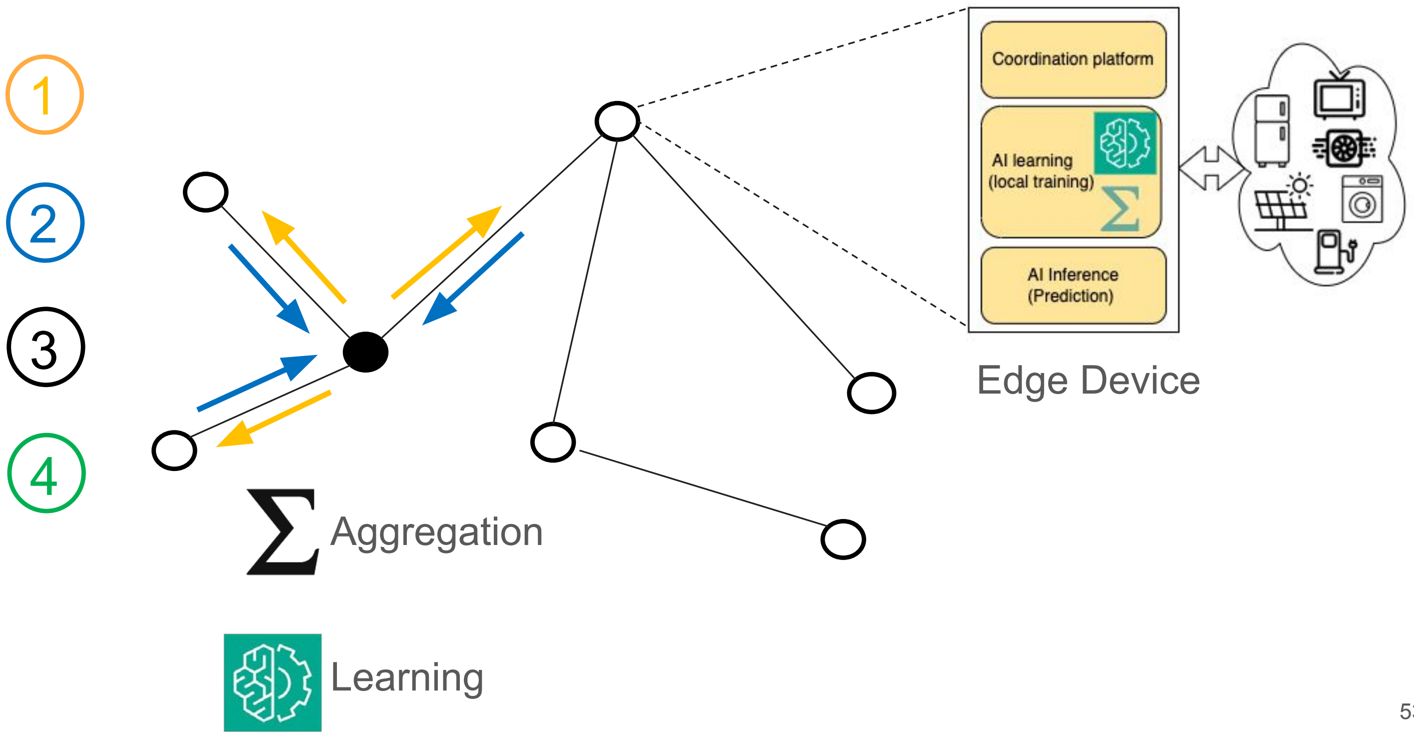
Algorithm Federated Learning Training Algorithm

Require: Training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, model f_θ, loss function $\mathcal{L}$, optimizer, number of epochs E

Ensure: Trained model parameters θ

Initialize model parameters θ
for epoch $= 1$ to E **do**
  Shuffle $\mathcal{D}$
  for each batch $B \subset \mathcal{D}$ **do**
    Compute predictions: $\hat{y} = f_\theta(x)$ for $(x, y) \in B$
    Compute loss: $L = \mathcal{L}(\hat{y}, y)$
    Compute gradients: $\nabla_\theta L$
    Update parameters: $\theta \leftarrow \text{Optimizer}(\theta, \nabla_\theta L)$
  end for
  Send θ to the Server
  Receive updated θ from the server
  if Early stopping criteria met **then**
    break
  end if
end for
return θ

Decentralised federated Learning



53

Hes·so

Distributed Federated Learning (A first draft algorithm)

While *true*

1. Collect data @ ***t*** and predict (***t* + *1***)
2. Calculates the accuracy of the previous prediction
3. Updates (or not) the list of neighbours
4. Communicates (or not) with neighbours
5. Aggregates (how?) and update (or not) local NN configuration (weights)
6. Start (or not) a new training to generate a new local inference
7. ***t*** ← ***t* + *1*** (wait for the next collected data)

End While

How to define and/or update neighbours?

When should we communicate with them?

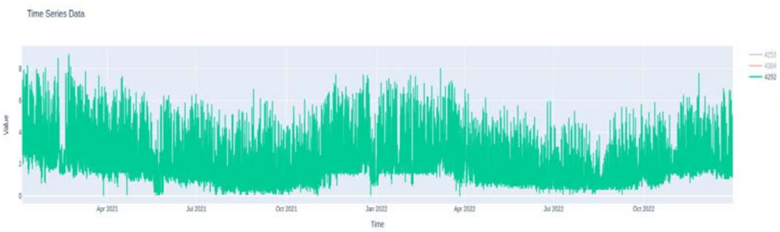
How to aggregate ?

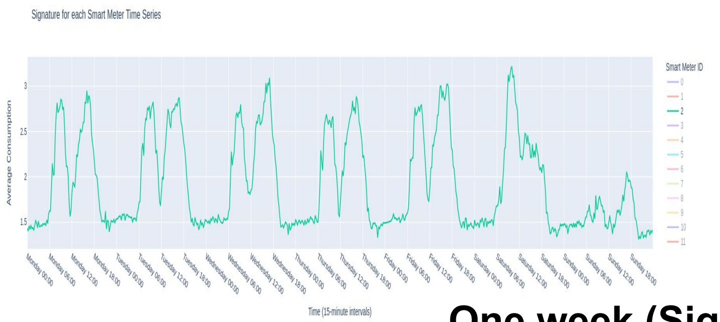
When starting a new training?

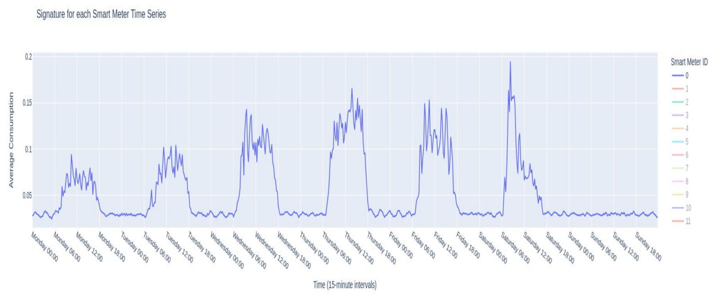
When updating the local NN?

Hes·so Who are our neighbours?

Raw data (1.5 year :April 2021 - October 2022)





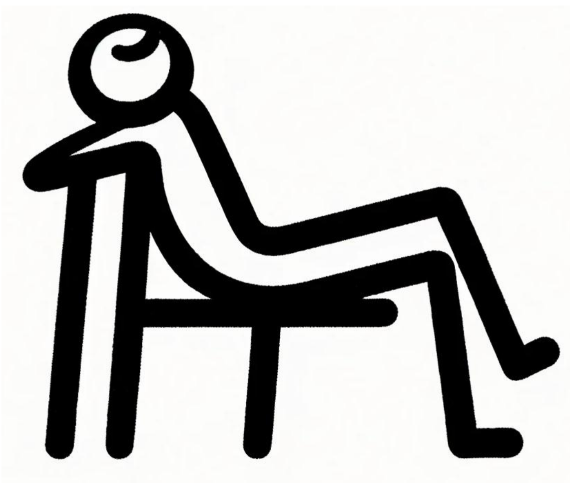


One week (Signature)

55

Hes·so

When should we communicate with our neighbours?



I feel good !!

I feel bad !!

56

The good, the bad and the ugly …

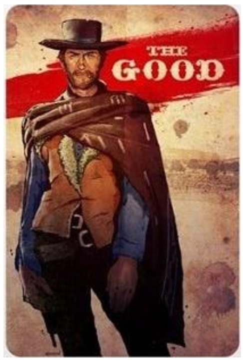
Clint Eastwood   Lee Van Cleef   Eli Wallach

57

Hes·so

The “Good” scenario algorithm

What is a good performance?



Initiate neighbours
While *true*

1. Collect data/measures
2. Result ← Prediction (collected data)
3. If “Good performance (Result)” then send ***My_Weights*** to my neighbours
4. When receiving a “***My_Weights”*** message from a given neighbour then (1) Make aggregation, (2) initiate learning and (3) update the local Neural Network

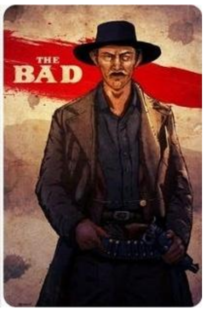
End While

Should we receive from all neighbours before executing (1), (2) and (3)?

58

Hes·so

The "bad" scenario algorithm



Ask for a help!!

Initiate neighbours
While *true*

1. Collect data/measures
2. Result ← Prediction (collected data)
3. If "low performance (Result)" then send ***Weights_Request*** to my neighbours
4. When receiving a message from a given neighbour
    a. if type (message) == ***Weights_Request*** then send ***My_Weights*** to the neighbour
    b. If type (message) == ***My_Weights*** then make aggregation, initiate learning (or not) and update the local Neural Network

End While

59

Exercise 2

Towards a decentralised Federated Learning algorithm

60


Thank you !

Aidge: an embedded AI open-source platform

Iryna de Albuquerque Silva - Research engineer @ Embedded AI Lab, CEA List

6th Summer School CPS & IOT - Budva, Montenegro

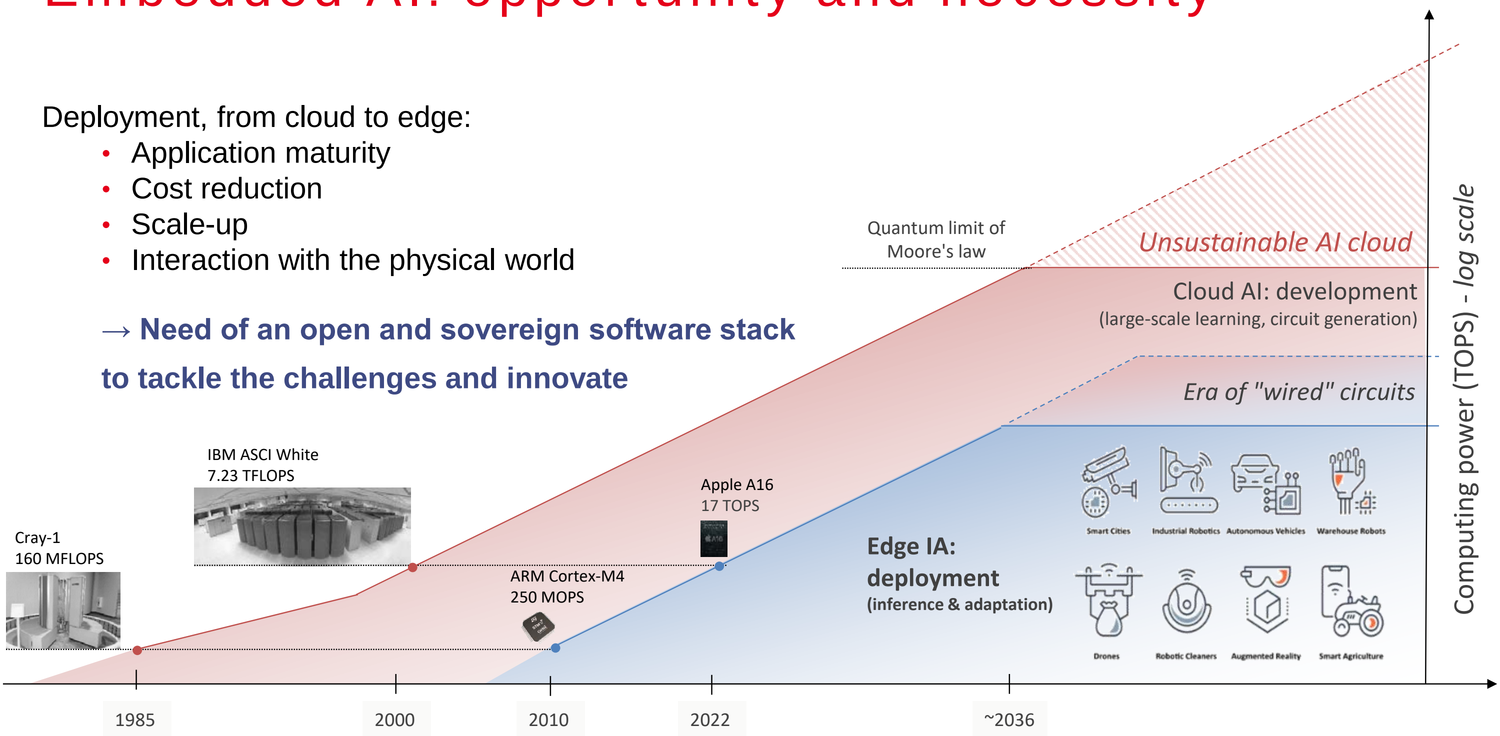
June 11, 2025

Embedded AI: opportunity and necessity

Deployment, from cloud to edge:

- Application maturity
- Cost reduction
- Scale-up
- Interaction with the physical world

→ Need of an open and sovereign software stack to tackle the challenges and innovate



Aidge: an embedded AI open-source platform 3

Challenges and ambitions

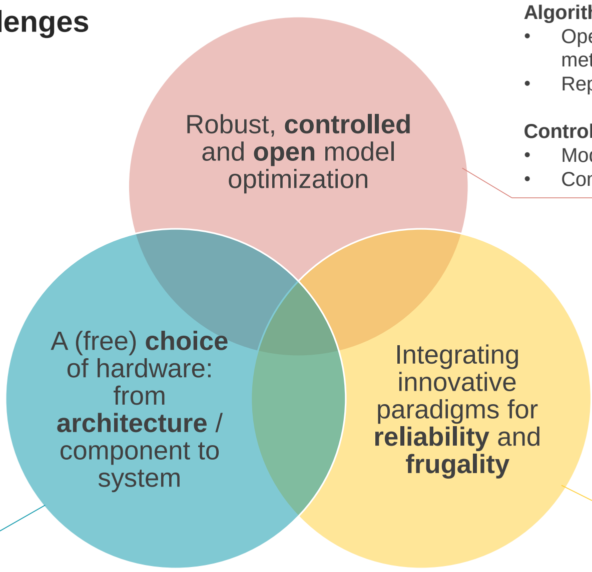
Over and above **performance**, which is essential, there are **many technical challenges** to ensure our competitiveness:

Algorithm mastery:
- Open, documented, generic and reusable methods
- Reproducibility of optimized models

Control of deployment:
- Modular, unified pre- and post-deployment APIs
- Compatible with certification objectives

Heterogeneity:
- Multi-target hardware
- Management of heterogeneous systems
- Multi-paradigm computing

Sovereignty:
- Open, independent computing cores
- Sovereign architectures and components
- Integration of sovereign technologies (FDSOI, 3D-chiplet, NVM memories)



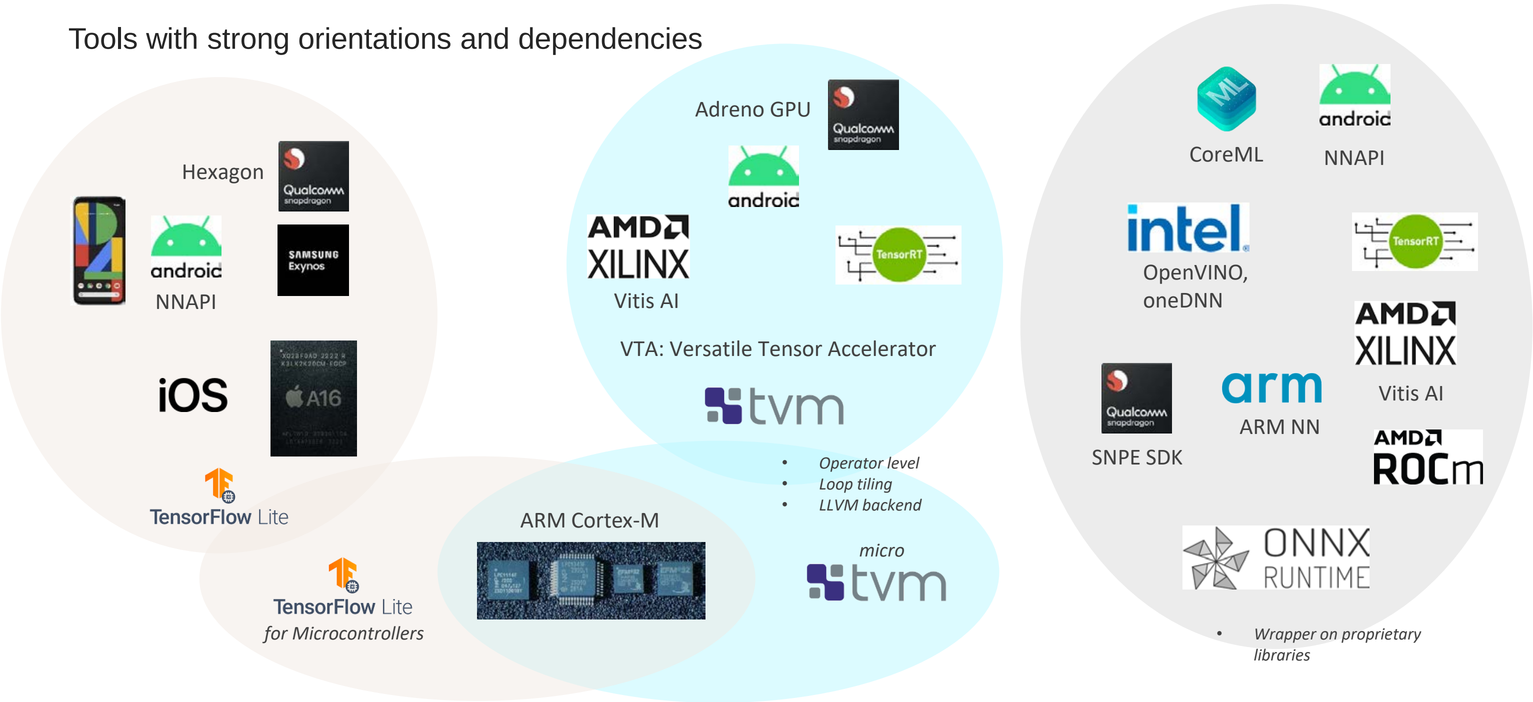
Reliability:
- Integration of "innate" knowledge
- Mastery of the "flight domain"
- Robustness to faults and attacks

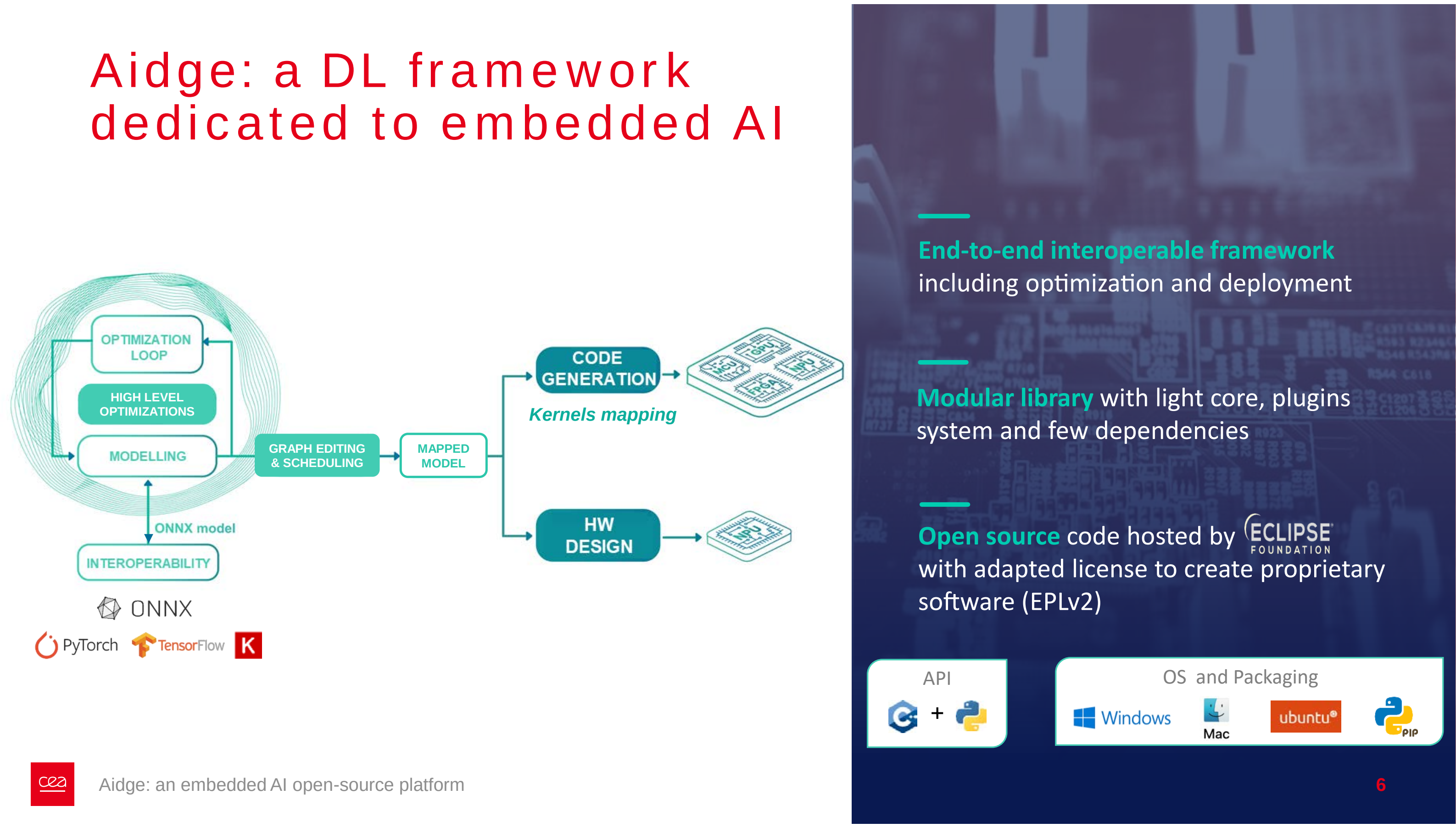
Frugality:
- Transferability of "master" models
- Off-line and on-line adaptation

Under constrains!

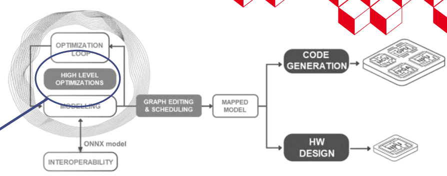
What solutions for embedded AI?

Tools with strong orientations and dependencies




Aidge: a DL framework dedicated to embedded AI
OPTIMIZATION LOOP
HIGH LEVEL OPTIMIZATIONS
MODELLING
GRAPH EDITING & SCHEDULING
MAPPED MODEL
CODE GENERATION
Kernels mapping
HW DESIGN
ONNX model
INTEROPERABILITY
ONNX
PyTorch
TensorFlow
End-to-end interoperable framework including optimization and deployment
Modular library with light core, plugins system and few dependencies
Open source code hosted by ECLIPSE FOUNDATION with adapted license to create proprietary software (EPLv2)
API
OS and Packaging
Windows
Mac
ubuntu
PIP
Aidge: an embedded AI open-source platform
6

Key features

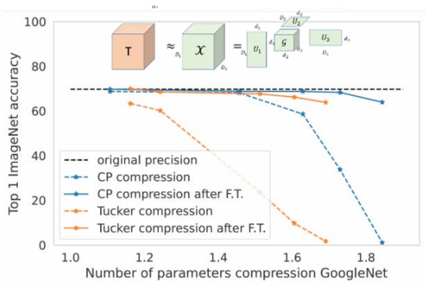


Well-characterized and state-of-the-art optimizations

Traceable step-by-step automatic graph transformation

- Master the quantization scaling strategies, down to true, *unfaked*, full integer models
- Ensure reproducibility, maintainability and support independent of hardware SDK providers
- Quantization: Post Training or Aware Training (1 to 8 bits integer, mixed precision)
- Compression: Tensor Decomposition

Tensor decomposition compression



Compression rate ~1.8

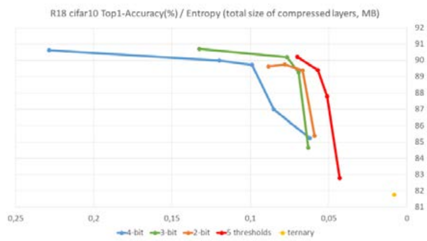
Progressive and mixed-precision QAT

Network	Method	Top-1 accuracy W2A32
Mobilenet-v2 (71.9)	Fracbits Naive clip progressive κ	71.0
	Fracbits Naive clip constant κ	70.3

<1 pt accuracy drop with 2-bits W

(On-going Aidge integration)

Entropy-based compression

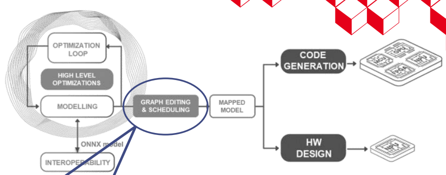


ResNet-18 (W2A32): 68.9% accuracy with ~0.73 bit/w

(Planned Aidge integration)

Aidge: an embedded AI open-source platform 7

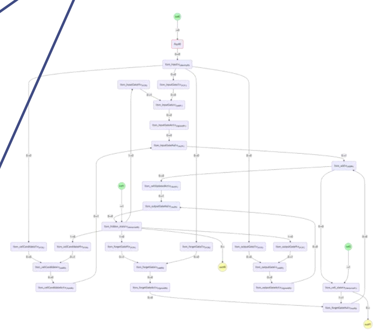
Key features



Powerful graph matching and transformation capabilities

Strong equivalence guarantees at each transformation steps

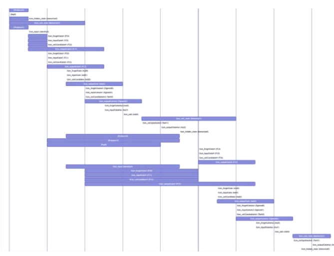
- Consistent IR across multiple granularities
- Match complex sub-graphs with effective regular expression-like DSL
- Replace operators or entire sub-graphs with a sleek one-liner
- Fuse/expand any sub-graph, at any hierarchy level, seamlessly



Close to hardware with TLM-like modeling and scheduling

Model and *statically* schedule pipelined dataflow compute kernels

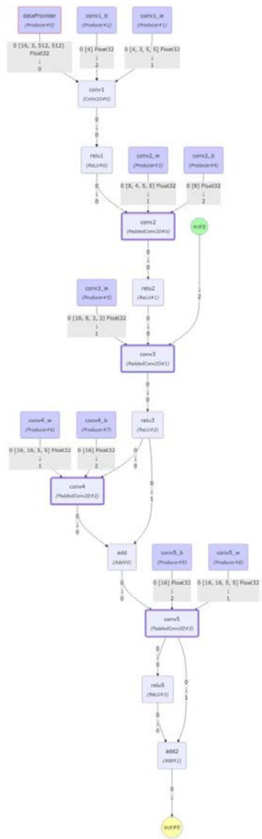
- Model consumed and produced data for each operator at each execution step
- Get early/late logical static scheduling for any graph, cyclic or acyclic
- Mix different coding and implementation paradigms seamlessly



Key features of **Aidge's** IR

- A transparent graph representation
  - The compute graph is not *inferred* from user's code: the user **explicitly writes the graph**

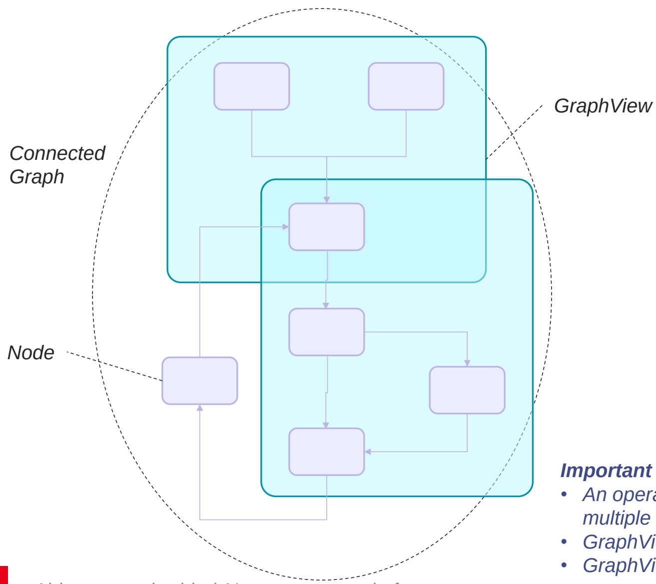
```
import aidge_core

model = aidge_core.sequential([
    aidge_core.Producer([16, 3, 512, 512], name="dataProvider"),
    aidge_core.Conv2D(3, 4, [5, 5], name="conv1"),
    aidge_core.ReLU(name="relu1"),
    aidge_core.PaddedConv2D(4, 8, [5, 5], name="conv2", stride_dims=[1, 1], padding_dims=[2,
2, 2, 2]),
    aidge_core.ReLU(name="relu2"),
    aidge_core.PaddedConv2D(8, 16, [3, 3], name="conv3", stride_dims=[1, 1], padding_dims=[2,
2, 2, 2], no_bias=True),
    aidge_core.ReLU(name="relu3"),
    aidge_core.PaddedConv2D(16, 16, [5, 5], name="conv4", stride_dims=[1, 1], padding_dims=[2,
2, 2, 2]),
    aidge_core.Add(name="add"),
    aidge_core.PaddedConv2D(16, 16, [5, 5], name="conv5", stride_dims=[1, 1], padding_dims=[2,
2, 2, 2]),
    aidge_core.ReLU(name="relu5"),
    aidge_core.Add(name="add2")
])

model.get_node("relu3").add_child(model.get_node("add"), 0, 1)
model.get_node("conv5").add_child(model.get_node("add2"), 0, 1)
```

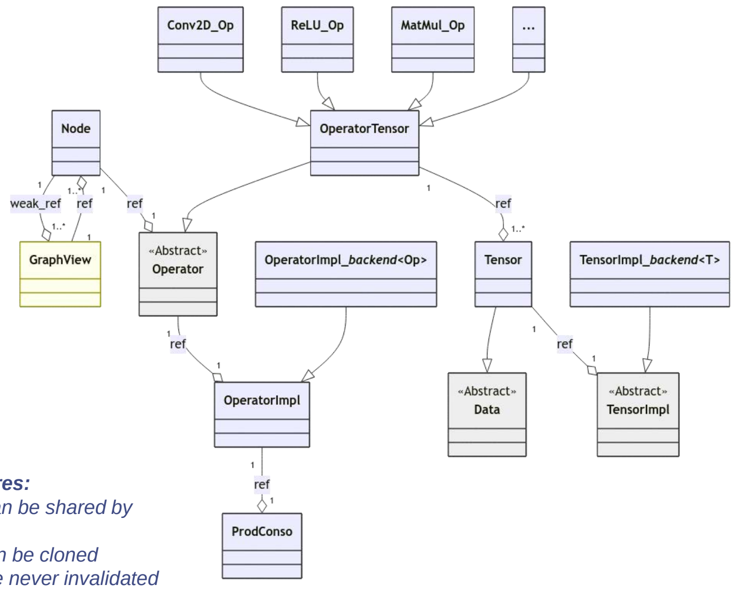


Aidge: an embedded AI open-source platform 9

Key features of **Aidge’s** IR

➢ A multi-paradigms abstraction model (not limited to tensors)





Important features:

- *An operator can be shared by multiple nodes*
- *GraphView can be cloned*
- *GraphView are never invalidated*

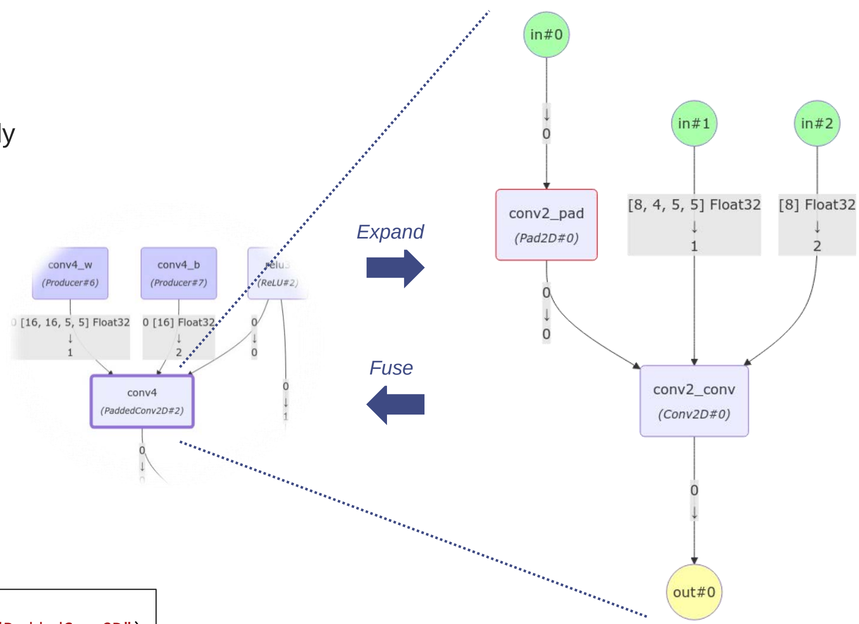
Key features of **Aidge's** IR

- A hierarchical graph model
  - Multiple granularities are handled seamlessly through **meta-operators**
  - Expand meta-operator(s):

```
# Expand only level-1 meta operators
aidge_core.expand_metaops(model)

# Expand all meta operators recursively
aidge_core.expand_metaops(model, recursive=True)

# Expand a single meta operator
aidge_core.expand_metaop(model.get_node("conv5"))
```

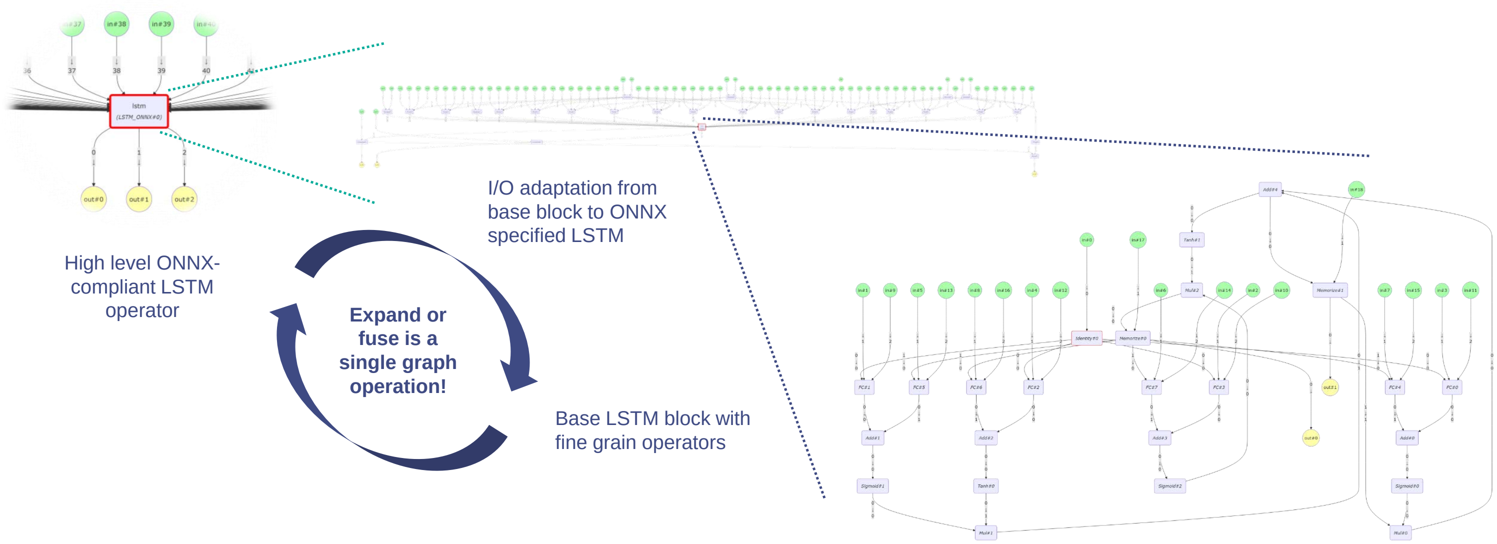
  - Fuse into a meta operator:

```
# Fuse using graph matching
aidge_core.fuse_to_metaops(model, "Conv2D#<-Pad?", "PaddedConv2D")
```



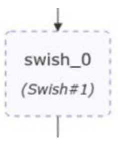
Key features of **Aidge's** IR

- Transform your graph to match the granularity **required by your implementation**

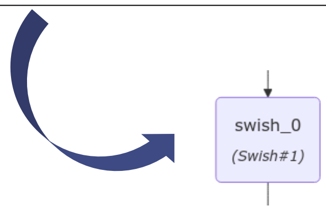


Key features of **Aidge's** IR

- A generic graph model
  - Operators modeling/specification without implementation can be handled through **generic operators**




```
    # Create a generic op. of type Swish
    node_swish = aidge_core.GenericOperator("Swish",
[aidge_core.InputCategory.Data], 1)
    swish_op = node_swish.get_operator()

    # Optional: add attributes to the generic op.
    swish_op.attr.beta = [1.0]

    # Optional: specifies how its output dimensions are computed w.r.t. its inputs
    swish_op.set_forward_dims(lambda x: x)
```




```
class SwishImpl(aidge_core.OperatorImpl):
    def __init__(self, op: aidge_core.Operator):
        aidge_core.OperatorImpl.__init__(self, op, "swish")
        self.op = op

    def forward(self):
        data_input = np.array(self.op.get_input(0))
        beta = np.array(self.op.attr.get_attr("beta"))
        output = (data_input / (1 + np.exp(-data_input*beta)))
        self.op.set_output(0, aidge_core.Tensor(output))

swish_op.set_impl(SwishImpl(swish_op)) # Setting implementation
```

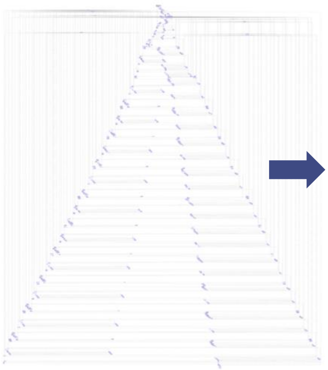
Define an implementation

Bind the implementation to the operator

Key features of **Aidge's** IR

- A searchable graph model
  - Effective pattern matching example

Deal with ONNX arbitrary granularities nightmare!



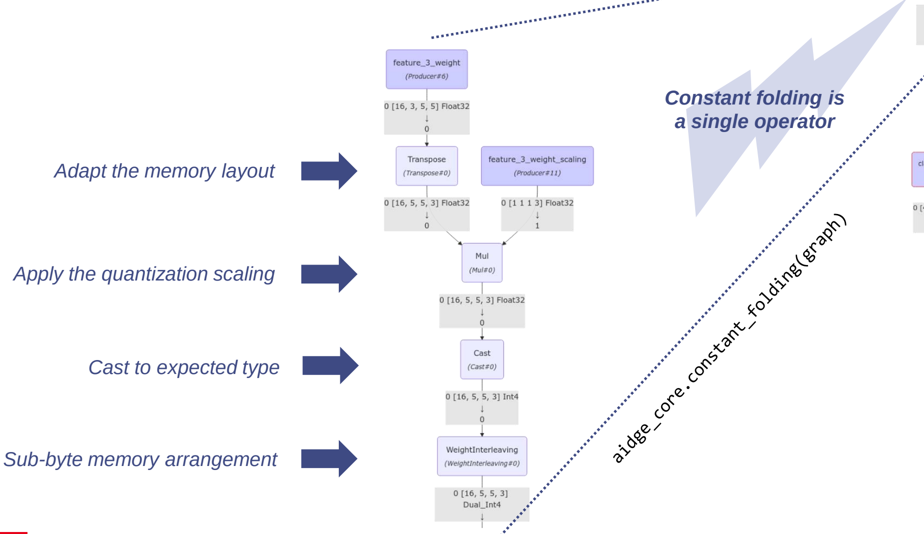
Thousands nodes DinoV2 ONNX…

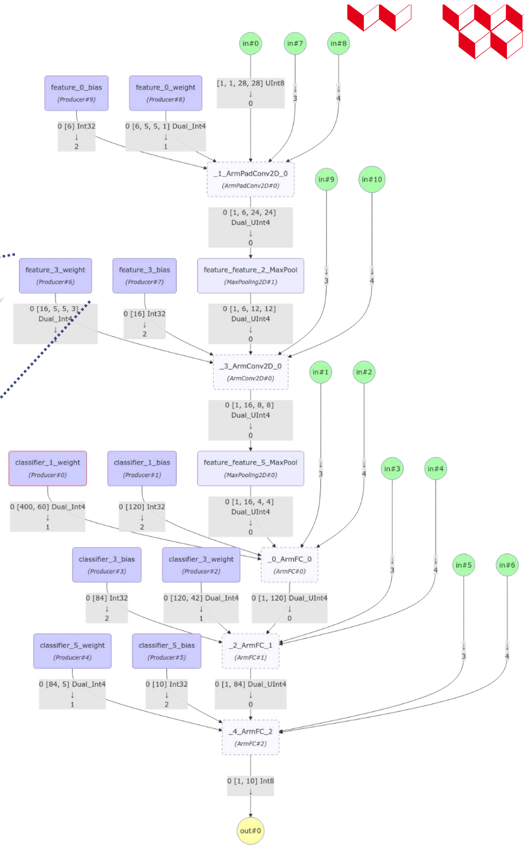
```
dinov2_model = aidge_onnx.load_onnx("dinov2.onnx")
aidge_core.fuse_to_metaops(dinov2_model, "MatMul-*>Add", "Linear")
aidge_core.fuse_to_metaops(dinov2_model, "ReduceMean-*>Sub#1~>(Pow#1->ReduceMean-*>Add#1->Sqrt)-*>Div#1-*>Mul#1-*>Add#2;"
                                         "Sub#1~*>Div#1;"
                                         "Pow#1<1~Producer;"
                                         "Add#1<*~Producer;"
                                         "Mul#1<*~Producer;"
                                         "Add#2<*~Producer;"
                                         "
aidge_core.fuse_to_metaops(dinov2_model, "MatMul->Div#1->Softmax-*>MatMul;"
                                         "Div#1<1~Producer", "ScaledDotProductAttention")
aidge_core.fuse_to_metaops(dinov2_model, "ScaledDotProductAttention#1->Transpose->Reshape#1->Linear;"
                                         "Reshape#1<1~Producer;"
                                         "ScaledDotProductAttention#1<0-(Transpose<-Reshape#2<-Add#1);"
                                         "ScaledDotProductAttention#1<1-(Transpose<-Reshape#3<-Add#2);"
                                         "ScaledDotProductAttention#1<2-(Transpose<-Reshape#4<-Add#3);"
                                         "Reshape#2<1~Producer;"
                                         "Add#1<*-0-Split#1;"
                                         "Add#2<*-1-Split#1;"
                                         "Add#3<*-2-Split#1;"
                                         "Split#1<-MatMul;"
                                         "Split#1<1~Producer", "MultiHeadAttention")
aidge_core.fuse_to_metaops(dinov2_model, "Div#1->Erf->Add#1-*>Mul->Mul#2;"
                                         "Div#1<1~Producer;"
                                         "Add#1<*~Producer;"
                                         "Mul#2<*~Producer", "GeLU")
```

Graph at the required granularity for your implementation library provider

Key features of **Aidge's** IR

- Have a clean, step-by-step, graph transformation:
  - Policy: do not change the data, change the graph!

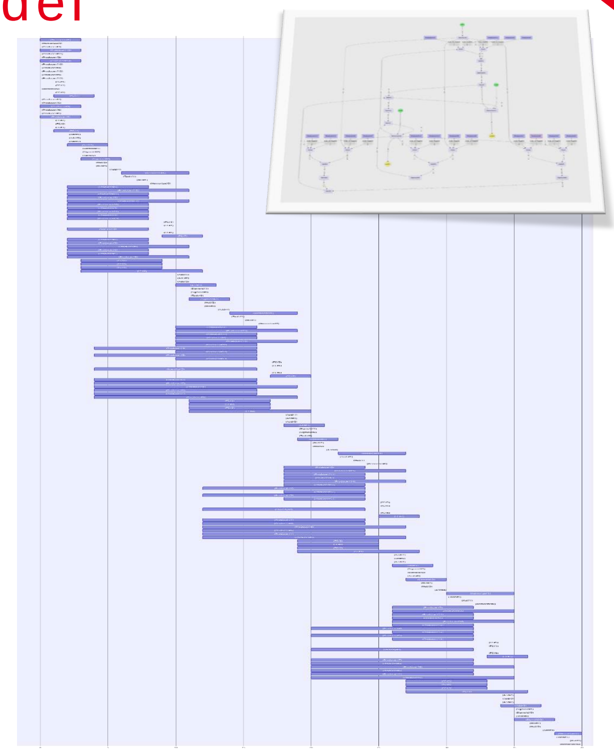




An innovative scheduling model

- Aidge's scheduling is **static**
  - Get early and late logical static scheduling for any graph (cyclic or acyclic)

```
lstm = aidge_core.LSTM(in_channels=4, hidden_channels=8, seq_length=5)

# Flatten the graph:
lstm_model = aidge_core.get_connected_graph_view(lstm)
aidge_core.expand_metaops(lstm_model)
lstm_model.set_backend("cpu")

# Create the Scheduler
lstm_scheduler = aidge_core.SequentialScheduler(lstm_model)
lstm_scheduler.generate_scheduling()

# Display static scheduling
lstm_scheduler.save_static_scheduling_diagram("lstm_scheduling")
```



Rich analysis tools

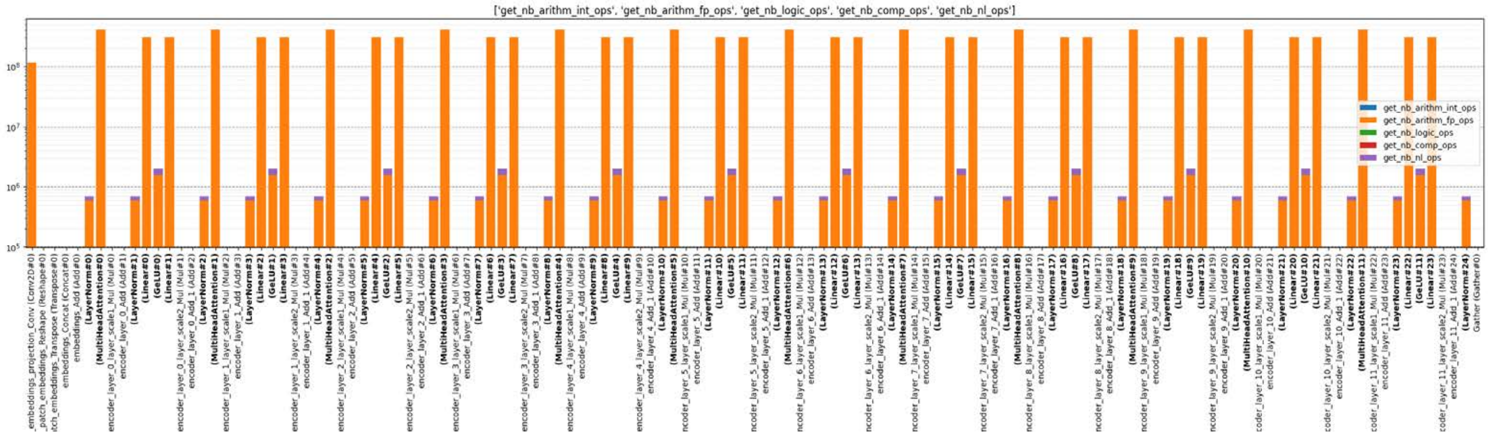
- Comprehensive and simple static analysis

```
stats = aidge_core.static_analysis.StaticAnalysisExt(dinov2_model)
stats.summary() # Simple text output similar to Keras model.summary()

stats.log_nb_ops_by_type("stats_ops.png", log_scale=True)
```

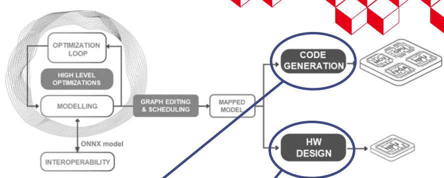
Rich low-level & high-level APIs:

- *Base C++: for lower-level heuristics integration*
- *Extended Python: with additional visualization methods*

Auto aggregation for Meta operators,
specifiable for Generic operators



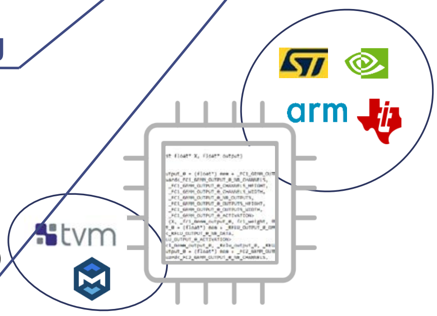
Key features



A rich toolbox for operators optimization and hardware mapping

Generate optimized code tailored to the hardware target

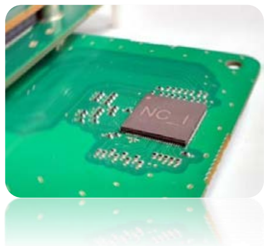
- Flexible and transparent template-based code generation engine
- Interoperable through ONNX import/export and plugins support
- Register third-party library kernels or custom ones in one-liner
- Graph tiling and automatic adaptation to available compute kernels
- Integrate optimized sub-graphs through ML compiler stacks (TVM, MLIR)
- Ensure the deployment process can meet certification constrains



Design greener hardware architecture

Basis of the NeuroCorgi AI ASIC accelerator

- RTL generation of a quantized model
- HD images processing in real time: latency is less than 10ms
- Uses 1,000 times less power than commercial circuits



ONNX interoperability

- ONNX is at the heart of Aidge's interoperability (in import and export)
  - Extensive coverage (> 60 operators)
  - Integration of ONNX Runtime test suite *(still in progress)*
  - Trivial interface, with automatic registration
    - Unsupported operators / versions loaded as **generic operators**

```
graph = aidge_onnx.load_onnx("model.onnx")
...
aidge_onnx.export(graph, "optimized_model.onnx")
```

ONNX ➔ Aidge

```
@aidge_onnx.node_import.auto_register_import("swish")
def import_swish(onnx_node, input_nodes, opset=None):
    node_name = onnx_node.output[0]
    return gen_swish_metaop(node_name)
```

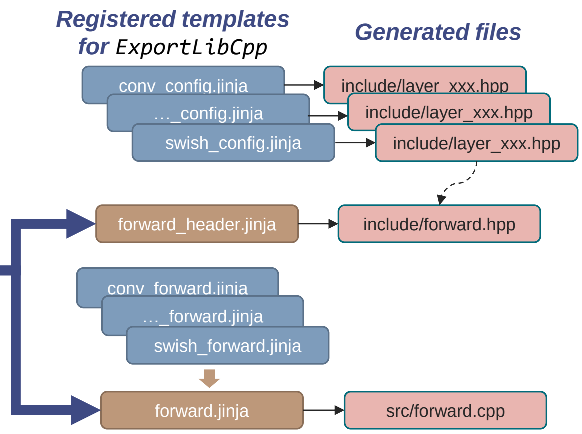
Aidge ➔ ONNX

```
@aidge_onnx.node_export.auto_register_export("BitShift")
def export_bitshift(aidge_node: aidge_core.Node, node_inputs_name, node_outputs_name, opset=None):
    aidge_operator = aidge_node.get_operator()
    onnx_node = helper.make_node(
        name=aidge_node.name(),
        op_type="BitShift",
        inputs=node_inputs_name,
        outputs=node_outputs_name,
    )
    onnx_node.attribute.append(helper.make_attribute("direction",
        DirectionAttrToString(aidge_operator.attr.get_attr("direction"))
    ))
    return [onnx_node]
```

Towards hardware mapping

- A flexible code generation engine
  - Template-based (Jinja)
  - Open and transparent (non-binary code generation)
  - Multi-paradigm (C/C++, HDL code generation…)
  - Designed to integrate of low-level computing primitives

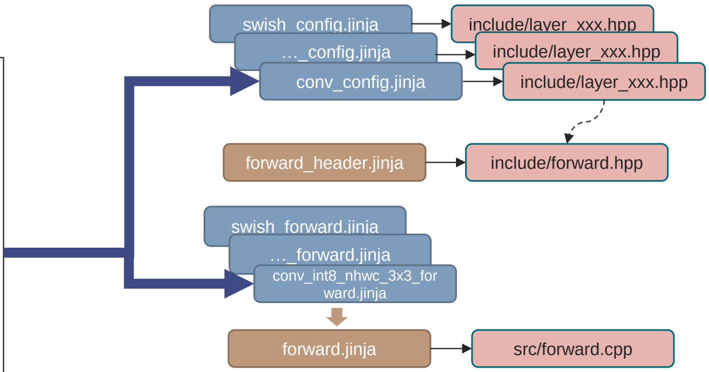
```
model = aidge_onnx.load_onnx("mobilenet.onnx")
model.set_backend("cpu")
model.forward_dims([[1, 3, 224, 224]])

scheduler = aidge_core.SequentialScheduler(model)
scheduler.generate_scheduling()

export_folder = "mobilenet_export"
aidge_core.export_utils.scheduler_export(
        scheduler,
        export_folder,
        aidge_export_cpp.ExportLibCpp,
        memory_manager=aidge_core.mem_info.generate_optimized_memory_info,
        memory_manager_args={"stats_folder": f"{export_folder}/stats"}
)

aidge_core.export_utils.generate_main_cpp(export_folder, model)
```



Towards hardware mapping

- Kernels are registered **with a specification:**
  - Required I/O data type, data format, dimension range, attributes

```
@aidge_export_cpp.ExportLibCpp.register("Conv",
    aidge_core.ImplSpec(
        [ # Inputs specs
            aidge_core.IOSpec(aidge_core.dtype.int8, aidge_core.dformat.nhwc),
            aidge_core.IOSpec(aidge_core.dtype.int8, aidge_core.dformat.nhwc,
[[0,0], [0,0], [3,3], [3,3]]),
            aidge_core.IOSpec(aidge_core.dtype.int32)],
        [ # Outputs specs
            aidge_core.IOSpec(aidge_core.dtype.int32)],
        { # Attributes specs
        }))
class ConvCpp(aidge_core.export_utils.ExportNodeCpp):
    def __init__(self, node, mem_info):
        super().__init__(node, mem_info)
        self.config_template = "conv_config.jinja"
        self.forward_template = "conv_int8_nhwc_3x3_forward.jinja"
        self.include_list = []
        self.kernels_to_copy = [
            "conv_int8_nhwc_3x3_kernel.hpp",
        ]
```



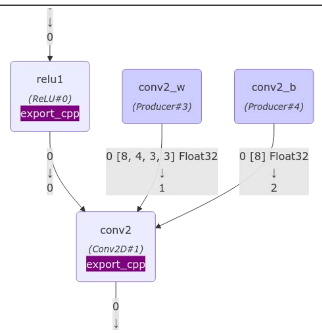
Register multiple specialized kernels for the same operator with different specifications!

Here: INT8 NHWC convolution kernel for 3x3 only

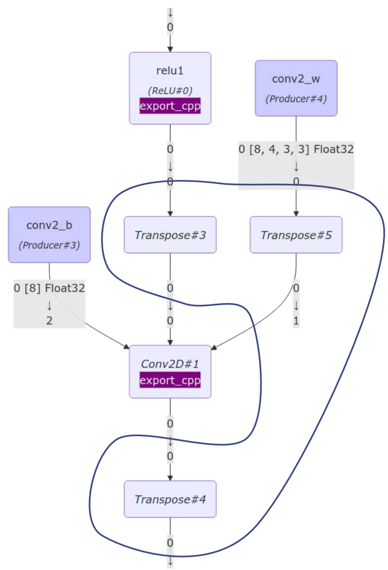
Towards hardware mapping

- Specifications allow **automatic graph adaptation**
  - To use available computational kernels

```
graph.set_backend("export_cpp")
aidge_core.adapt_to_backend(graph)
```



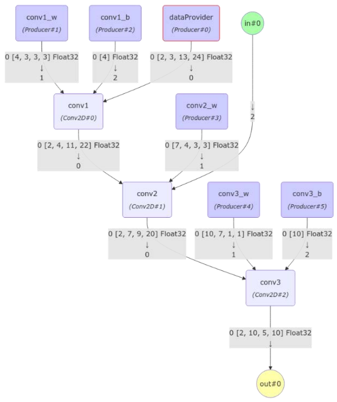
Simple example with data format change (NCHW ➔ NHWC)



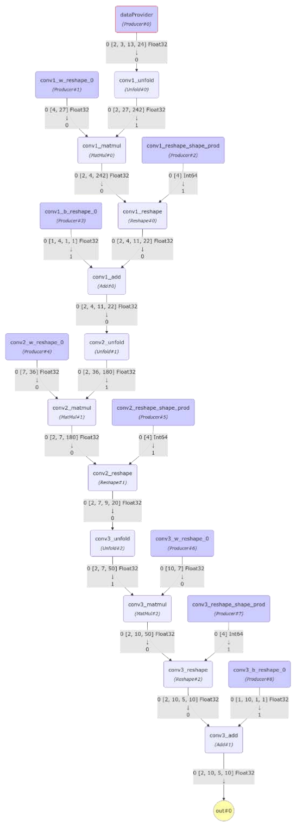
Towards hardware mapping

- Specifications allow **automatic graph adaptation**
  - To adapt the implementation with respect to the algorithm

```
aidge_core.conv_to_matmul(graph)
```



Replace Conv with Unfold + MatMul operators



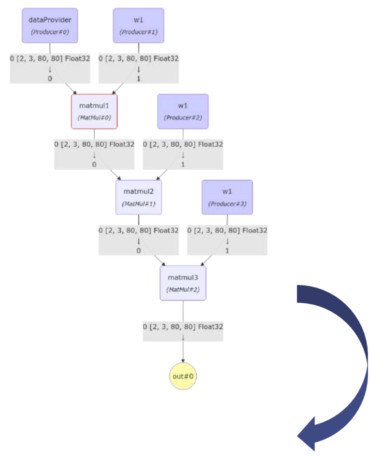
Towards hardware mapping

- Specifications allow **automatic graph adaptation**
  - To match the hardware architecture
    - Tiling MatMul

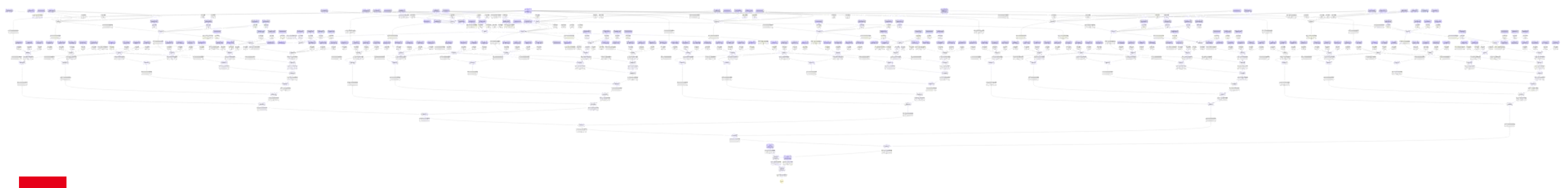
```
aidge_core.matmul_tiling(matMul, [16, 16])
```

    - Tiling Conv2D

```
aidge_core.get_conv_horizontal_tiling(conv, axis=1, nb_slices=5)
```



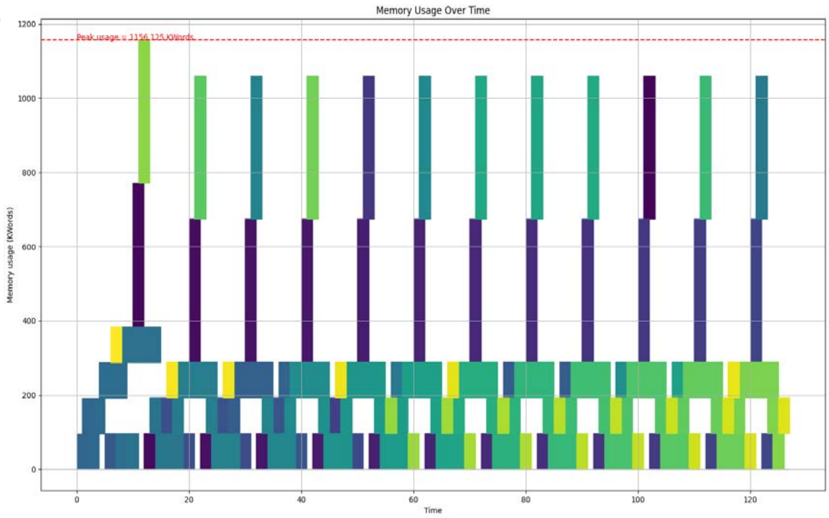
Example: Fixed size MatMul



Towards hardware mapping

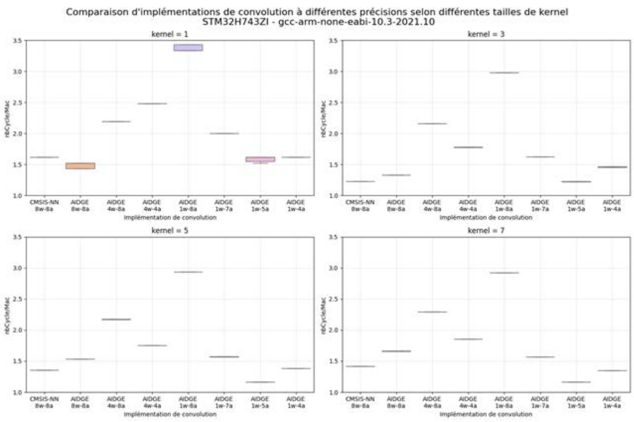
- Static memory allocation for exports with a **memory manager**
  - Memory reuse: in-place or partially in-place (wrapping)
  - Auto-concatenation (strided buffers support)
  - Several optimization strategies available

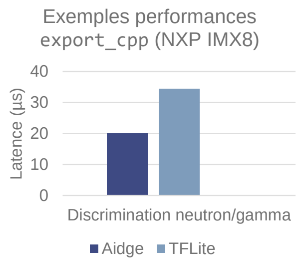
```
scheduler = aidge_core.SequentialScheduler(dinov2_model)
scheduler.generate_scheduling()
aidge_core.generate_optimized_memory_info(scheduler, "stats_dyno",
wrapping=False)
```

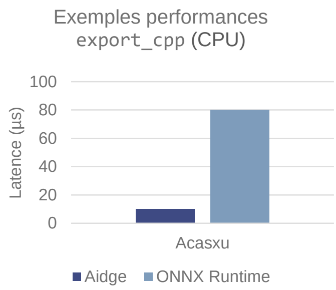


Towards hardware mapping

- Aidge's reference export for **embedded targets**: `export_cpp`
  - Generic, fully templated kernels
  - Optimized compile-time performance
  - Work at any bit-precision, inc. low-bit precision *(Planned)*
  - Ready for SIMD (NHWC format)
  - Several specializations
    - ARM Cortex-M
    - ARM CMSIS-NN *(On-going)*
    - ESP32 *(Planned)*
    - ARM Cortex-A72, C7x DSP *(No public release)*







Perspectives and Conclusion

Follow us

Join and chat !

- **Upcomming features**
  - **Modeling** : Support of SNN
  - **Optimization** : Full integer quantization for transformers, robust learning
  - **Code generation** : Certifiable C code generation, integration of a scalable and efficient ML compilation (based on TVM and MLIR).

- **Large ecosystem and numerous collaborations**
  - **With academics** to create innovative optimization techniques (Holigrail) and design specialized hardware (Adapting)
  - **With France2030 and EU projects** for exploitation
  - **With industrials** to build demonstrators (indoor localisation, heat pump monitoring, predective maintenance) and interface hardware

Research and Academics

Industrials

Exploitation

Shape AI
GreenAI
Mosaics-LP
OLFACTORY
ODYSS.AI ...

Norms

Aidge: an embedded AI open-source platform

27

Hands-on tutorials

- Find us on ECLIPSE FOUNDATION
  https://projects.eclipse.org/projects/technology.aidge
- Aidge Gitlab
  https://gitlab.eclipse.org/eclipse/aidge
  - Tutorials
    https://gitlab.eclipse.org/eclipse/aidge/aidge/-/tree/dev/examples/tutorials

Thank you

CEA SACLAY
91191 Gif-sur-Yvette Cedex
France

iryna.dealbuquerquesilva@cea.fr

cea list

Trustworthy AI: Industry-Guided Tooling of the Methods

Zakaria Chihani

zakaria.chihani@cea.fr

SS-CPSIoT

FRANCE 2030

Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre de France 2030 portant la référence « ANR-23-PEIA-0006 »

Our lab

Verification	Test	Platform	Symbolic	XAI & uncertainty
Verification of safety and robustness formal specifications through Abstract Interpretation	Metamorphic testing applied to AI (Available for teaching)	Open-source, modular, extensible platform to Characterize AI Safety And Robustness	Open-source Symbolic AI tools, Safe-by-design Constraint solvers	Case-based reasoning, explainability, out-of-distribution detection
PyRAT	AIMOS	CAISAR	Colibri & co	PARTICUL

caisar-platform.com 2

Our lab

Verification of safety and robustness formal specifications through Abstract Interpretation	Metamorphic testing applied to AI (Available for teaching)	Open-source, modular, extensible platform to Characterize AI Safety And Robustness	Open-source Symbolic AI tools, Safe-by-design Constraint solvers	Case-based reasoning, explainability, out-of-distribution detection
PyRAT	AIMOS	CAISAR	Colibri & co	PARTICUL
Verification	**Test**	**Platform**	**Symbolic**	**XAI & uncertainty**

Industry

list

Academia

caisar-platform.com

3

Mostly through Formal Methods

Verification of safety and robustness formal specifications through Abstract Interpretation	Metamorphic testing applied to AI (Available for teaching)	Open-source, modular, extensible platform to Characterize AI Safety And Robustness	Open-source Symbolic AI tools, Safe-by-design Constraint solvers	Case-based reasoning, explainability, out-of-distribution detection
PyRAT	AIMOS	CAISAR	Colibri & co	PARTICUL
Verification	**Test**	**Platform**	**Symbolic**	**XAI & uncertainty**

Industry

cea list

Academia

caisar-platform.com

Rapid intro to FM

Examples for this talk:

- Property-based testing (Renault)
- Verification of functional properties (Airbus)
- Robustness evaluation (Technip)
- Out-of-Distribution detection (Thales)

5

So what are Formal Methods

- Non snobbish definition
  - Math- and logic-based techniques with rigorously established theoretical foundations
  - Used for the specification, development, test and verification of software and hardware

- Why use formal methods ?
  - Non validated software can have dire consequences and mathematical analysis can contribute to the reliability and robustness
  - Some certification standards call for (e.g., DO-178C for avionics) or even **mandate** (e.g., ISO/IEC 15408) the use of formal methods

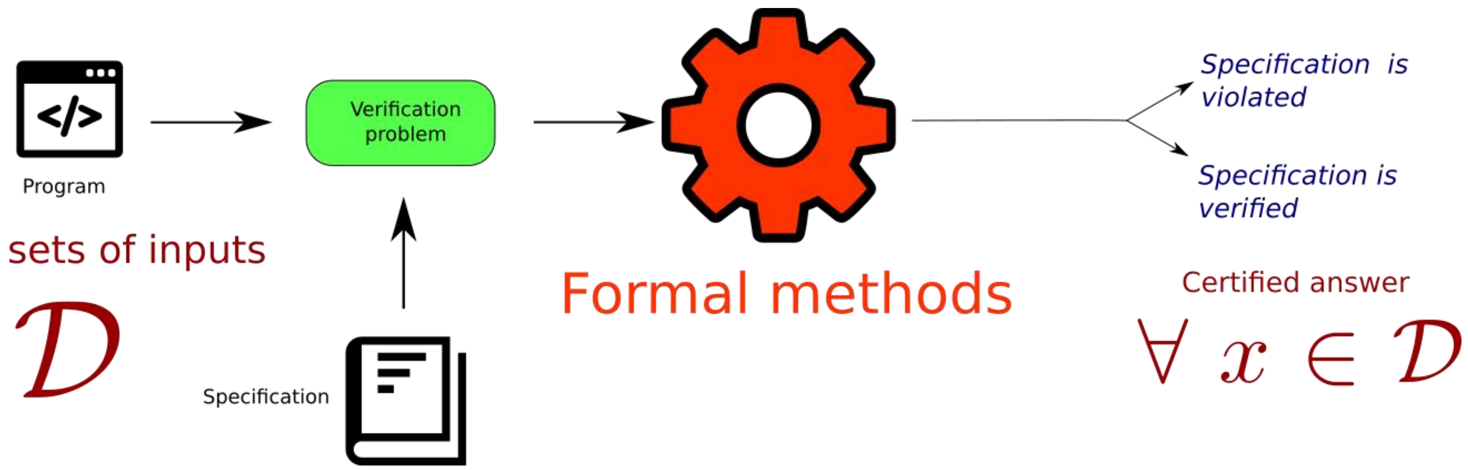
6

So why Formal Methods

A critical system is a system whose failure may cause physical harm, economical losses or damage the environment

Goal: guarantee that the system respects a *safety specification* ϕ

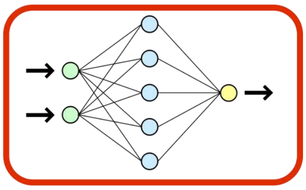
So HOW Formal Methods (usually)



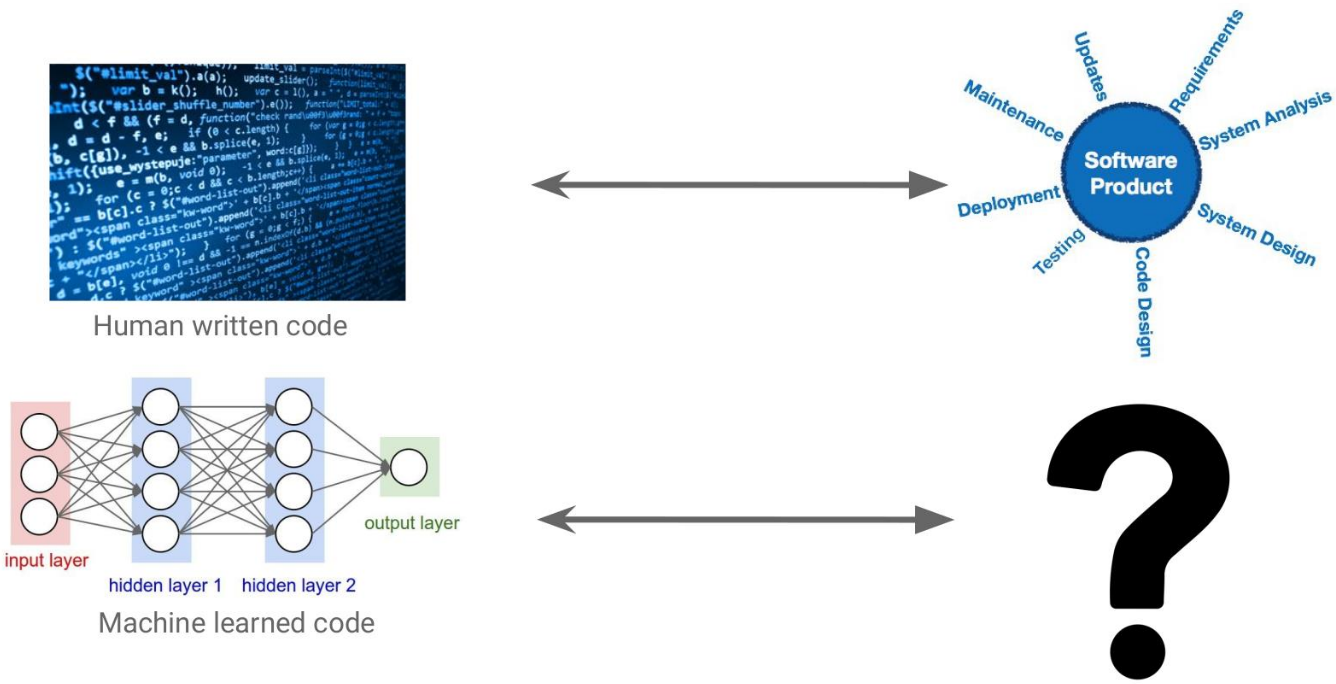
So WHO Formal Methods

So for WHOM Formal Methods

So WHICH Formal Methods

Classical programs	Deep learning
	
Explicit control flow	Generated control flow
Explicit specifications	Implicit specifications
Abstractions and well known concepts	Very few abstractions and reusability
Needs to be robust	Needs to be robust

So WHICH Formal Method for AI



A brief history of wrong predictions

In 1979:

"[P]rogram verification is bound to fail. We can't see how it's going to be able to affect anyone's confidence about programs"

"*Social processes and proofs of theorems and programs*", Communications of ACM. By Richard De Millo, Richard Lipton, and Alan Perlis.

A brief history of wrong predictions

In 1979:

"[P]rogram verification is bound to fail. We can't see how it's going to be able to affect anyone's confidence about programs"

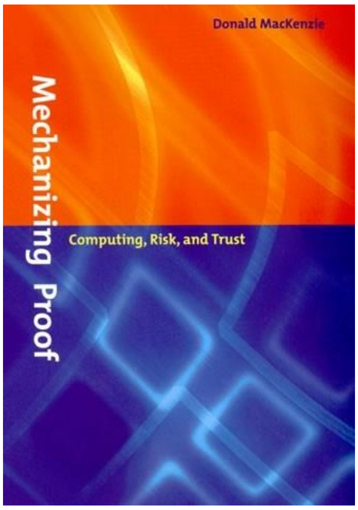
"Social processes and proofs of theorems and programs", Communications of ACM.
By Richard De Millo, Richard Lipton, and Alan Perlis.

- Distinguished Professor of Computing at the Georgia Tech
- VP and CTO of Hewlett-Packard

- Yale, Berkeley, Princeton, Georgia Tech
- Knuth Prize winner

- ACM, Carnegie Mellon, Yale, Purdue
- The first recipient of the Turing Award

Valid scepticism



- The first solvers and analyzers were **not** efficient or scalable.
- For example, today's SAT solvers can automatically solve problem instances involving **tens of thousands of variables and millions of constraints**.
- But it wasn't always the case! We needed to invent DPLL, CDCL, Symmetry breaking, two-watched literals, WalkSAT, adaptive branching, random restarts, portfolio, divide-and-conquer, parallel local search...

Restarting Formal Methods for AI

Cambrian explosion: Just in the past few years, more than 20 tools.
Competition for resources: Each paper published increases the scalability.
Cross-fertilization: Good ideas from one tool are implemented in others.
Niche creation: Some solvers are more specialized into particular models and type of properties.
Adaptative Pressure: New models, new architectures, and in general new AI-technologies are born every year and the tools to validate them must keep up.
Domestication: ML practitioners should be made aware of the choices in implementation that can make their models more amenable to FM, so that they can factor this aspect in their decision process.

Restarting Formal Methods for AI

- Artificial Intelligence Safety Engineering (WAISE, at SafeComp)
- AISafety (at IJCAI)
- Safe AI (at AAAI)
- Verification of Neural Networks (VNN, at AAAI or CAV)
- Formal Methods for ML-Enabled Autonomous Systems (FoMLAS, at CAV)
- Machine Learning with Guarantees (ML with Guarantees, at NeurIPS)
- Safe Machine Learning (SafeML, at ICLR)
- Privacy in Machine Learning (PriML, at NeurIPS)
- Security and Safety in Machine Learning Systems (AISecure, at ICLR)
- Dependable and Secure Machine Learning (DSML, at DSN)

General picture

Characterization of (AI) trustworthiness

A three-players game

A three-players game

Developer's side

- What is the architecture of the software, how can it be modified to be more amenable to verification, will these modifications cost too much ?
  (Activation functions of NN, kernel function of SVM, etc.)

Object to certify

- What to verify, how to formally specify it, how is it decomposed in smaller bits?
  (Robustness, metamorphism, behavior specification, etc.)

Properties to verify

Validator's side

- How to verify, what methods fit my problem, can the tools be helped with heuristics?
  (Abstract interpretation, SMT solving, symbolic execution, Constraint programming, etc.)

Methods and tools

Rapid intro to FM

Examples for this talk:

- **Property-based testing (Renault)**
- **Verification of functional properties (Airbus)**
- **Robustness evaluation (Technip)**
- **Out-of-Distribution detection (Thales)**

20

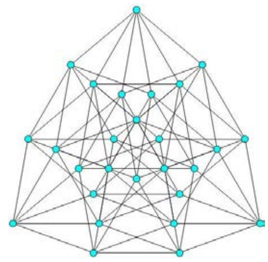
Metamorphic testing

Ideal setting for testing:

- Collection of inputs
- Corresponding collection of outputs

Metamorphic testing is used when:

- You don't know the actual **answer** (no oracle)
- But you know what **properties** should be satisfied by the inputs/outputs



Let
L (V,V) $\rightarrow$ int
be the length of the shortest path between two vertices, then :

L(a,b) = L(b,a)
For any two points a, b in a graph.

Input **symmetry**
⬍
output **equivalence**

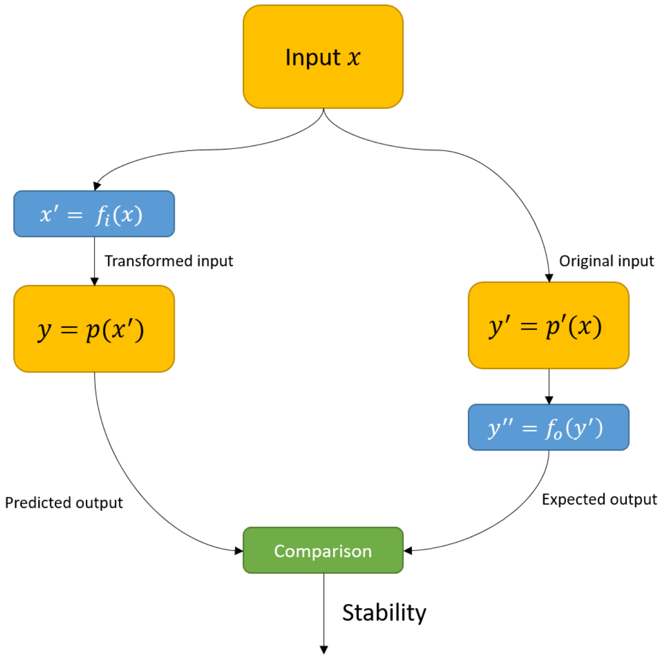
Metamorphic testing applied to AI : AIMOS

AIMOS (Artificial Intelligence Metamorphic Observing Software) is a tool to assess the stability of AI systems using metamorphic testing.

- No need to label data for testing.
- Automates the entire process of applying metamorphic properties on the inputs and outputs of models, comparing them and compiling the results into a stability score.
- Model agnostic (Neural Networks, Support Vector Machines, *etc.*).

caisar-platform.com

Metamorphic testing applied to AI : AIMOS

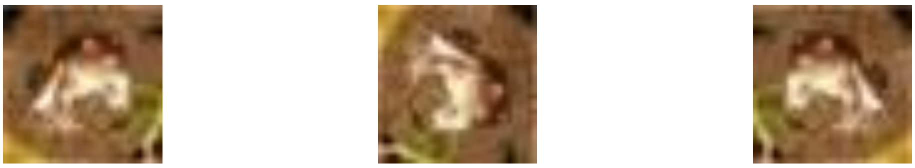


caisar-platform.com

23

Metamorphic testing applied to AI : AIMOS
Easy to use

- Written in Python
- Model agnostic: only the inference functions are needed.
- Built-in support for various frameworks, input formats and model types.
- Built-in classical transformations (rotation, noise, symmetry, *etc.*).



caisar-platform.com

24

Metamorphic testing applied to AI : AIMOS
Easy to use

- With a configuration file

```
options:
  plot: True
  inputs_path: "inputs"
  transformations:
    - name: "gaussian_blur"
      fn_range: range(1, 10, 2)

models:
  - defaults:
      models_path: "models/model.onnx"
```

caisar-platform.com

Metamorphic testing applied to AI : AIMOS
Easy to use

- With a configuration file
- As a Python library

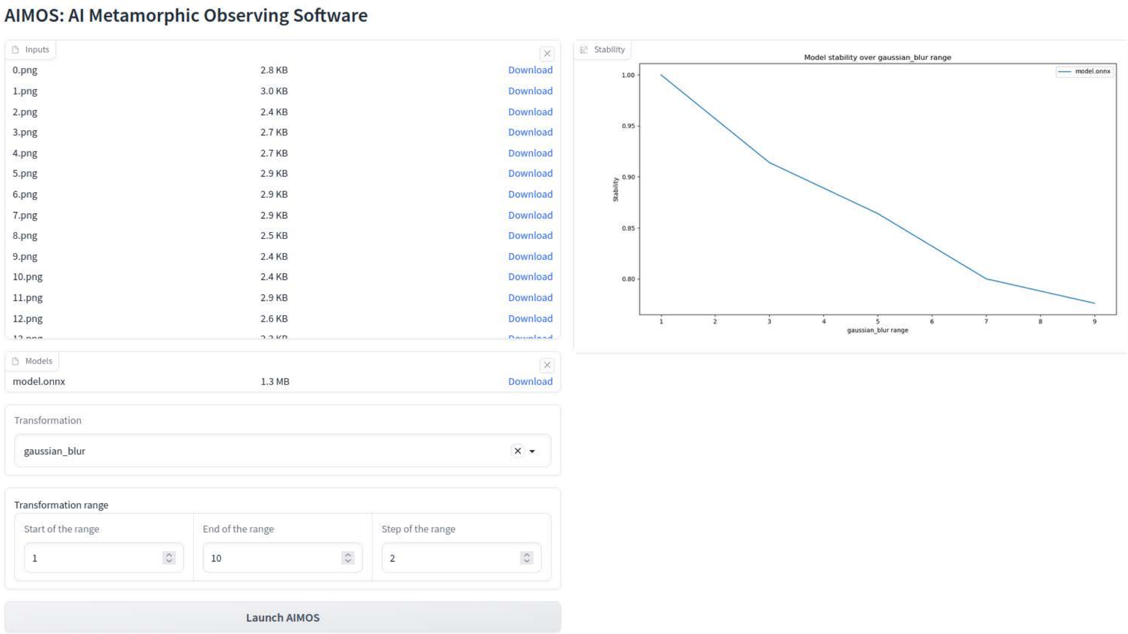
```
from aimos import core

core.main(
    "./inputs",
    "./models/model.onnx",
    "average_blur",
    fn_range=range(1, 10, 2),
    plot=True,
)
```

caisar-platform.com

26

Metamorphic testing applied to AI : AIMOS
Easy to use

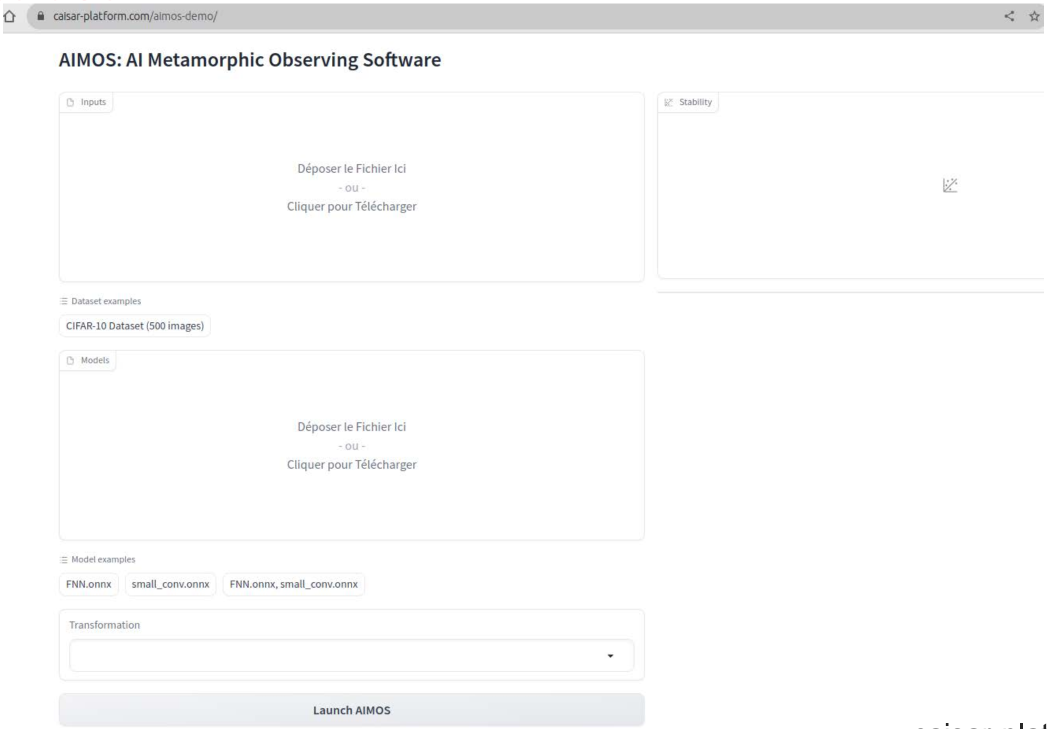
- With a configuration file
- As a Python library
- With a Graphical User Interface



caisar-platform.com

27

Metamorphic testing applied to AI : AIMOS
Easy to use



caisar-platform.com

28

Metamorphic testing applied to AI : AIMOS
Modular and extensible

Any operation can be replaced with a custom made Python function (loading the model, the inputs, new metrics, *etc.*).

```
def dead_columns(input, columns=np.uint8([50, 100, 150])):
    """ Adds dead pixel columns to an image. """
    input[:, columns, :] = 0
    return input
```

Metamorphic testing applied to AI : AIMOS

AIMOS is a tool that can be integrated in the verification and validation process of AI-based components.

- Freely available for teaching and research purposes.
- Integrated in CAISAR, an open-source platform for characterizing safety in AI systems.

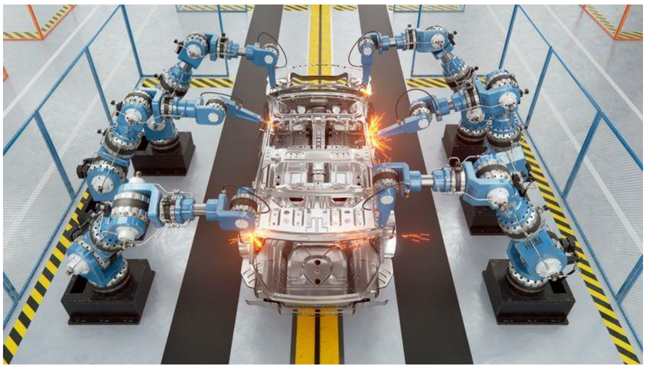
CAISAR

caisar-platform.com

Metamorphic testing applied to AI : AIMOS

The use-case

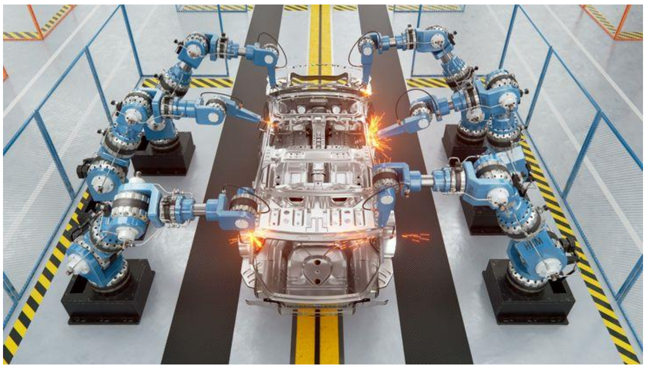
- Welding conveyor belt
- AI analysis for detection of faulty welds
- Notification of human expert



Metamorphic testing applied to AI : AIMOS

The use-case

- Welding conveyor belt
- AI analysis for detection of faulty welds
- Notification of human expert



32

Metamorphic testing applied to AI : AIMOS

The use-case

- Welding conveyor belt
- AI analysis for detection of faulty welds
- Notification of human expert

- 3 different production lines called C10, C20 and C34 and their corresponding weld.
- 5 AutoML models and 1 internal R&D composit model (NN+SVM) per production line.

Lemesle, A., Varasse, A., Chihani, Z., Tachet, D. (2023). **AIMOS: Metamorphic Testing of AI - An Industrial Application**. In: Guiochet, J., Tonetta, S., Schoitsch, E., Roy, M., Bitsch, F. (eds) Computer Safety, Reliability, and Security. SAFECOMP 2023 Workshops. SAFECOMP 2023. Lecture Notes in Computer Science, vol 14182. Springer, Cham. https://doi.org/10.1007/978-3-031-40953-0_27

Metamorphic testing applied to AI : AIMOS

The use-case

- Welding conveyor belt
- AI analysis for detection of faulty welds
- Notification of human expert

- 3 different production lines called C10, C20 and C34 and their corresponding weld.
- 5 AutoML models and 1 internal R&D composit model (NN+SVM) per production line.

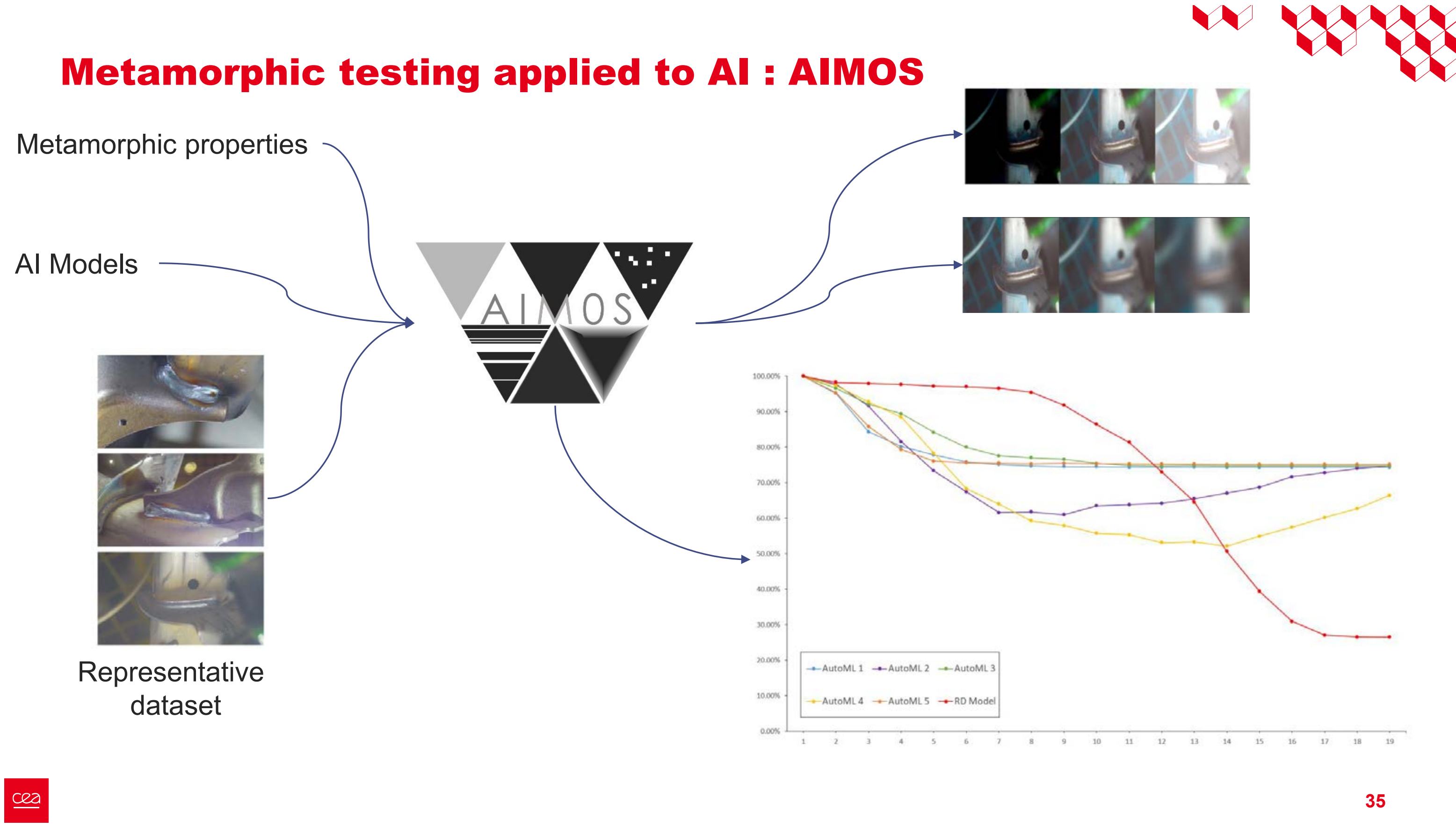
The environment => ODD => properties

- Day light changes + human workers pass by light sources => Robustness to **varying brigthness**
- Vibrating environment => Robustness to **blurring**

Lemesle, A., Varasse, A., Chihani, Z., Tachet, D. (2023). **AIMOS: Metamorphic Testing of AI - An Industrial Application**. In: Guiochet, J., Tonetta, S., Schoitsch, E., Roy, M., Bitsch, F. (eds) Computer Safety, Reliability, and Security. SAFECOMP 2023 Workshops. SAFECOMP 2023. Lecture Notes in Computer Science, vol 14182. Springer, Cham. https://doi.org/10.1007/978-3-031-40953-0_27

cea

34


Metamorphic testing applied to AI : AIMOS
Metamorphic properties
AI Models
Representative dataset
AIMOS
AutoML 1
AutoML 2
AutoML 3
AutoML 4
AutoML 5
RD Model
35

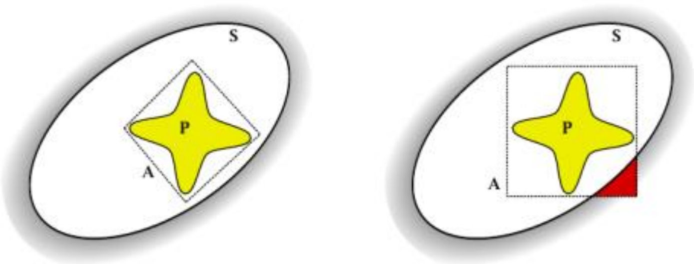
Rapid intro to FM

Examples for this talk:

- Property-based testing (Renault)
- Verification of functional properties (Airbus)
- Robustness evaluation (Technip)
- Out-of-Distribution detection (Thales)

36

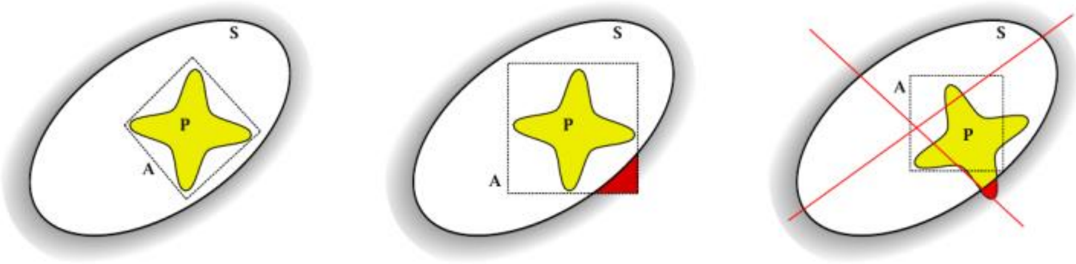
PyRAT: Python Reachability Assessment Tool Based on Abstract Interpretation



precise analysis
$A \subseteq S \implies P \subseteq S$

false alarm
$A \not\subseteq S$ but $P \subseteq S$

PyRAT: Python Reachability Assessment Tool Based on Abstract Interpretation



precise analysis
$A \subseteq S \implies P \subseteq S$

false alarm
$A \not\subseteq S$ but $P \subseteq S$

unsound analysis
$A \subseteq S$ but $P \not\subseteq S$

PyRAT: Python Reachability Assessment Tool Based on Abstract Interpretation

We would like to verify a property on the all possible values of inputs $x \in [a,b]$ and $y \in [c,d]$ in some program.

e.g.:

$$x + y \in [a+c, b+d]$$

Do the same for all operations in the program.

Use other types of domain for more precision (not just intervals).

PyRAT: Python Reachability Assessment Tool Based on Abstract Interpretation

Property: "f(y)=100 → Critical vibration frequency "

```
f (int y){
 int x;                    ..................
 x = 3 * (y²+1);           ..........
 if x > 100 then           .........
    x = x + 10;            ..........
 else                      ...................
    x = x - 2;             ...........
 return x;                 ..............
}
```

PyRAT: Python Reachability Assessment Tool Based on Abstract Interpretation

Property: "f(y)=100 → Critical vibration frequency "

Concret

f (int y){		
int x;	....	..-2,-1,0,1,2,..
x = 3 * (y²+1);	....	3,6,9,..
if x > 100 then	....	102,105,108,..
x = x + 10;	....	**112,115,118**,..
else	....	3,6,9..93,96,99
x = x - 2;	....	**1,4,7,..91,94,97**
return x;	....	1,4,..94,97,112,115..
}		

PyRAT: Python Reachability Assessment Tool Based on Abstract Interpretation

Property: “f(y)=100 → Critical vibration frequency ”

Code	Concrete	Intervals
f (int y){		
int x;	..-2,-1,0,1,2,..	-∞,+∞
x = 3 * (y²+1);	3,6,9,..	3,+∞
if x > 100 then	102,105,108,..	102,+∞
x = x + 10;	**112,115,118**,..	**112,+∞**
else	3,6,9..93,96,99	3,99
x = x - 2;	**1,4,7,..91,94,97**	**1,97**
return x;	1,4,..94,97,112,115..	1,+∞
}		

PyRAT: Python Reachability Assessment Tool Based on Abstract Interpretation

Property: "f(y)=100 → Critical vibration frequency "

	Concrete	Intervals	modulo
f (int y){			
int x;	..-2,-1,0,1,2,..	-∞,+∞	0%1
x = 3 * (y²+1);	3,6,9,..	3,+∞	0%3
if x > 100 then	102,105,108,..	102,+∞	0%3
x = x + 10;	**112,115,118**,..	**112,+∞**	1%3
else	3,6,9..93,96,99	3,99	0%3
x = x - 2;	**1,4,7,..91,94,97**	**1,97**	1%3
return x;	1,4,..94,97,112,115..	1,+∞	1%3
}			

43

PyRAT: Python Reachability Assessment Tool Based on Abstract Interpretation

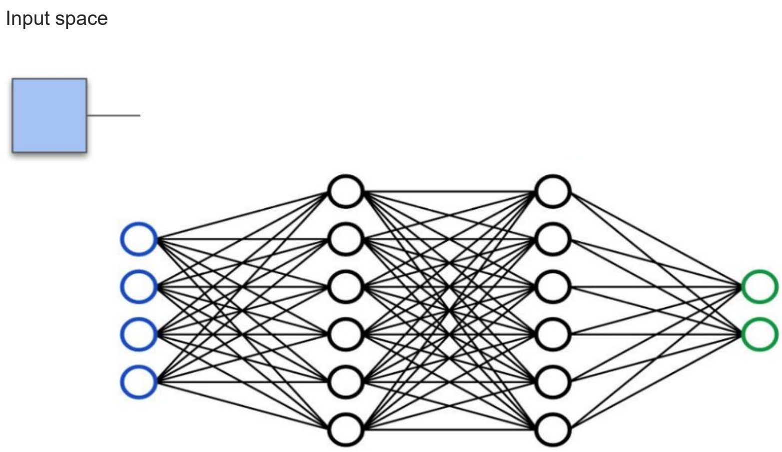
Property: "f(y)=100 → Critical vibration frequency "

	Concret	Intervals	modulo	Union of intervals
f (int y){				
int x;	..-2,-1,0,1,2,..	-∞,+∞	0%1	-∞,+∞
x = 3 * (y²+1);	3,6,9,..	3,+∞	0%3	3,+∞
if x > 100 then	102,105,108,..	102,+∞	0%3	102,+∞
x = x + 10;	**112,115,118**,..	**112,+∞**	1%3	**112,+∞**
else	3,6,9..93,96,99	3,99	0%3	3,99
x = x - 2;	**1,4,7,..91,94,97**	**1,97**	1%3	**1,97**
return x;	1,4,..94,97,112,115..	1,+∞	1%3	*[1,97]∪ [112,+∞[*
}				

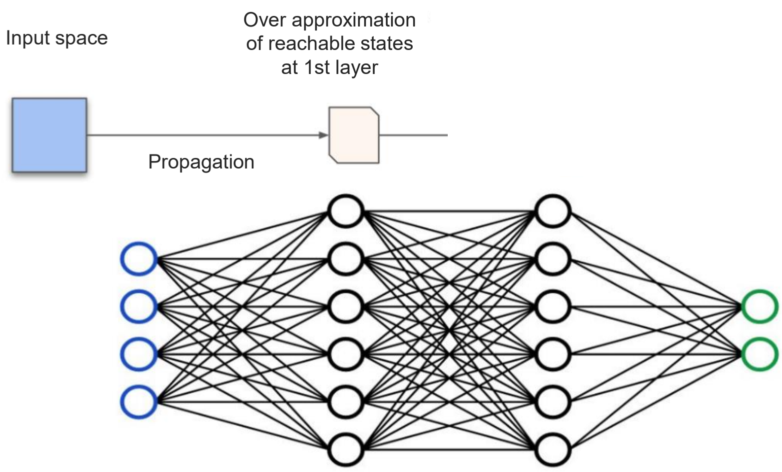
Conservative over-approximation: the concretization of the abstract domains contains reality
The inverse is not necessarily true. 1,4..94,97,**100,103,106,109**,112,115..

cea

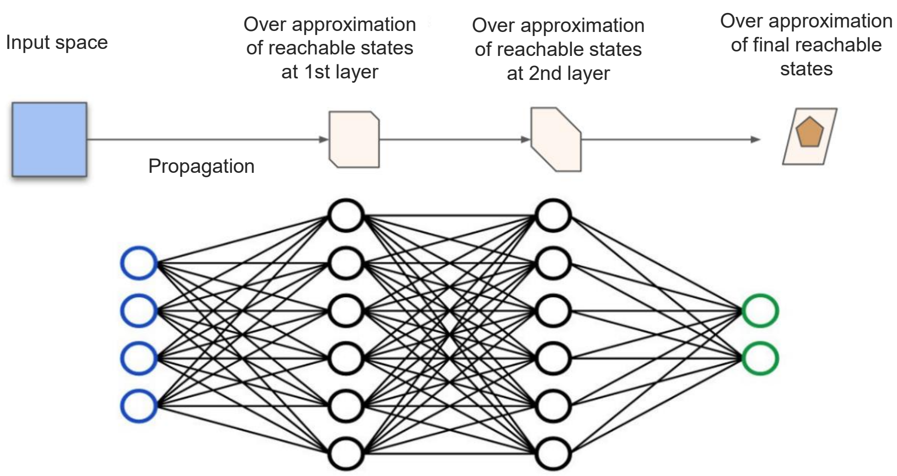
PyRAT: Python Reachability Assessment Tool
Application to NN



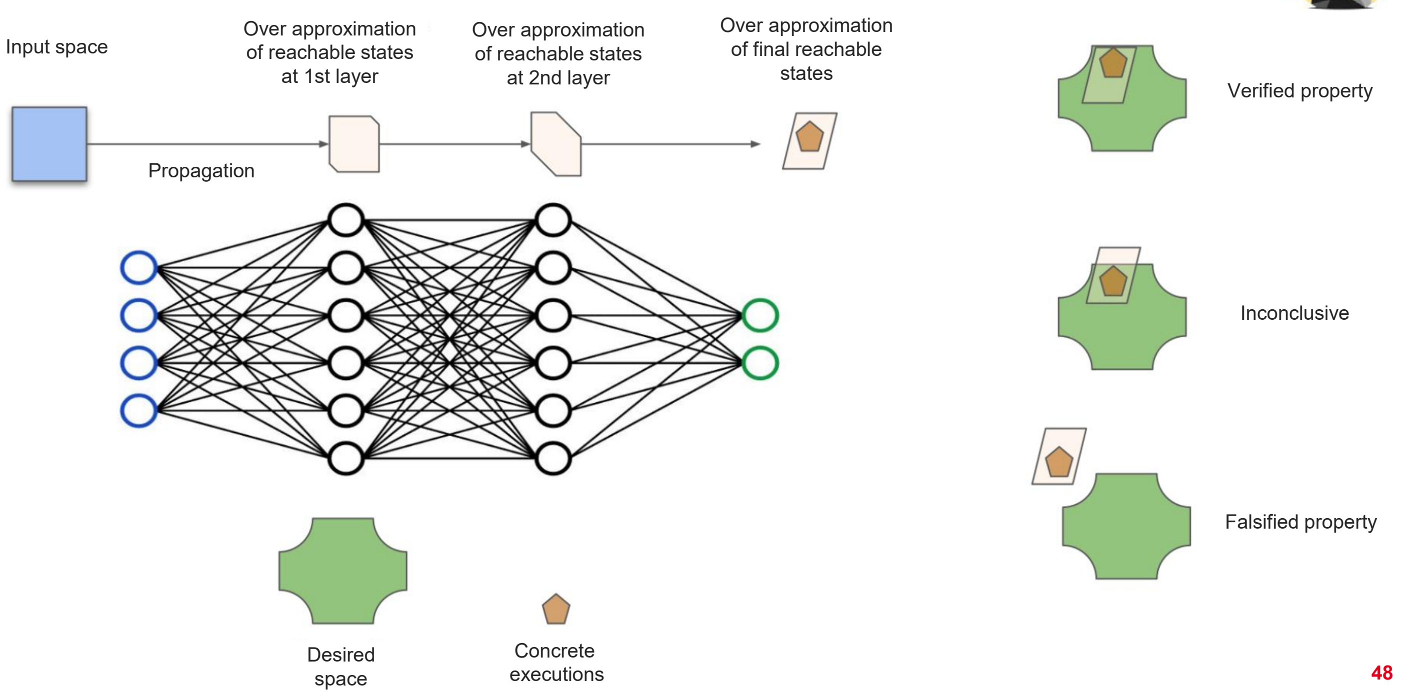
PyRAT: Python Reachability Assessment Tool
Application to NN



PyRAT: Python Reachability Assessment Tool
Application to NN

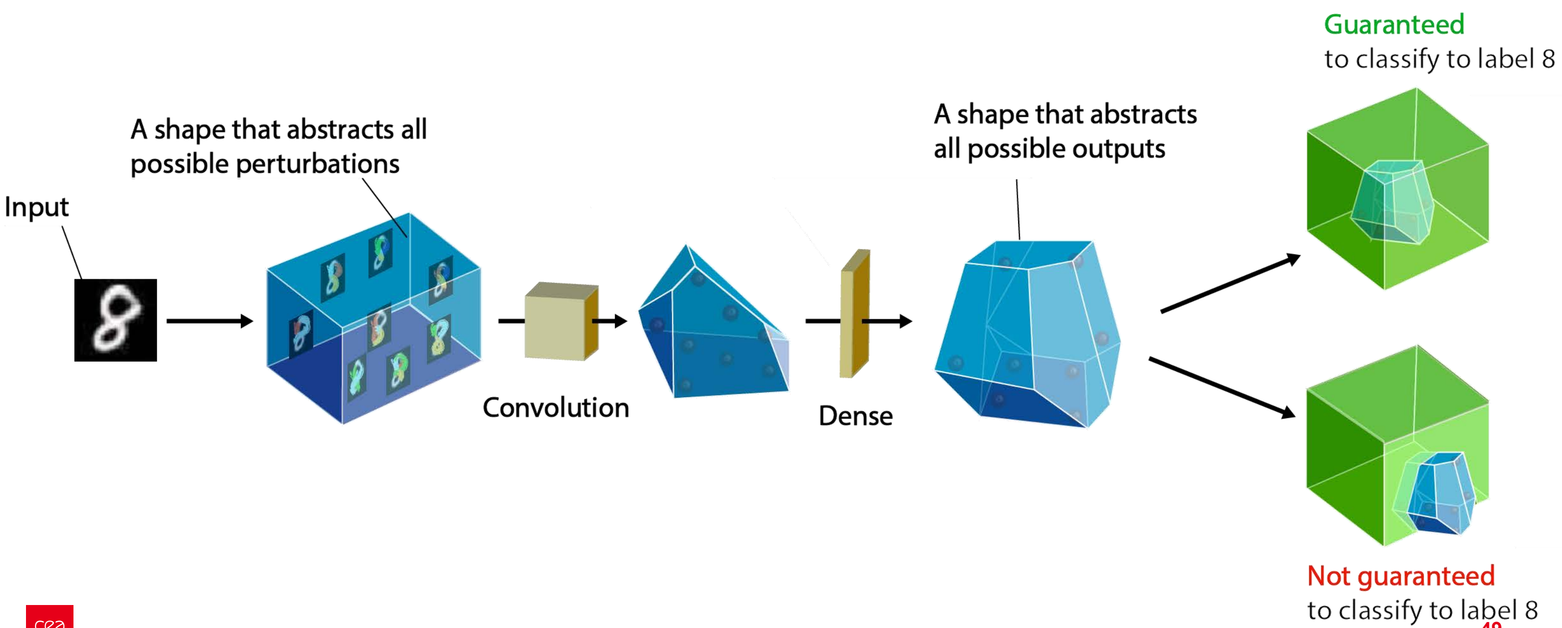


PyRAT: Python Reachability Assessment Tool
Application to NN



48

PyRAT: Python Reachability Assessment Tool
Application to NN



49

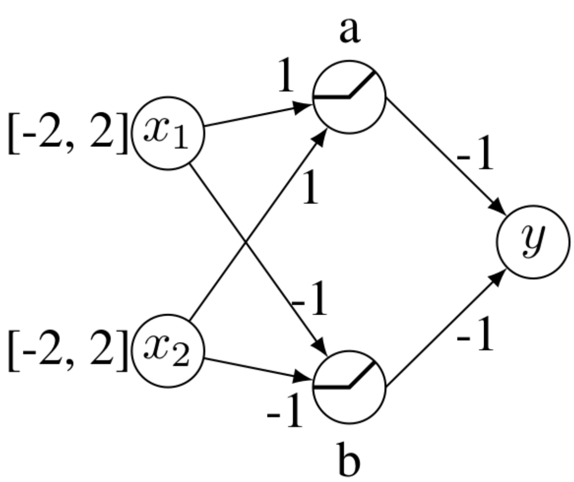
PyRAT: Python Reachability Assessment Tool
Application to NN

$$-2 \leq x_1 \leq 2$$
$$-2 \leq x_2 \leq 2$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

$$y = -a_{\text{out}} - b_{\text{out}}$$



Prove that $y > -5$

50

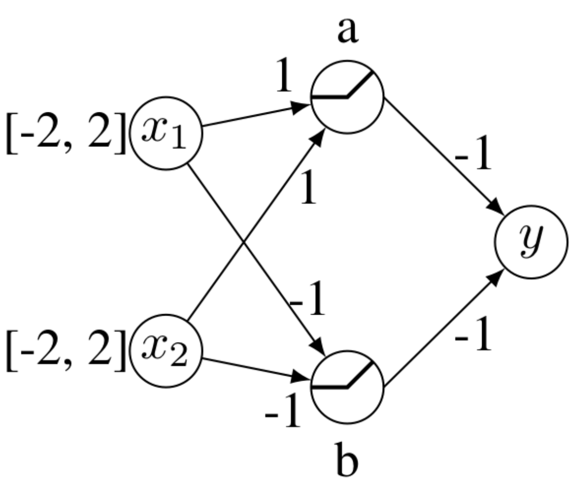
PyRAT: Python Reachability Assessment Tool
Application to NN

$$-2 \leq x_1 \leq 2$$
$$-2 \leq x_2 \leq 2$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

$$y = -a_{\text{out}} - b_{\text{out}}$$



$a_{\text{in}} =$ [-2,2] + [-2,2] = [-4,4]

$b_{\text{in}} =$ - [-2,2] - [-2,2] =[-4,4]

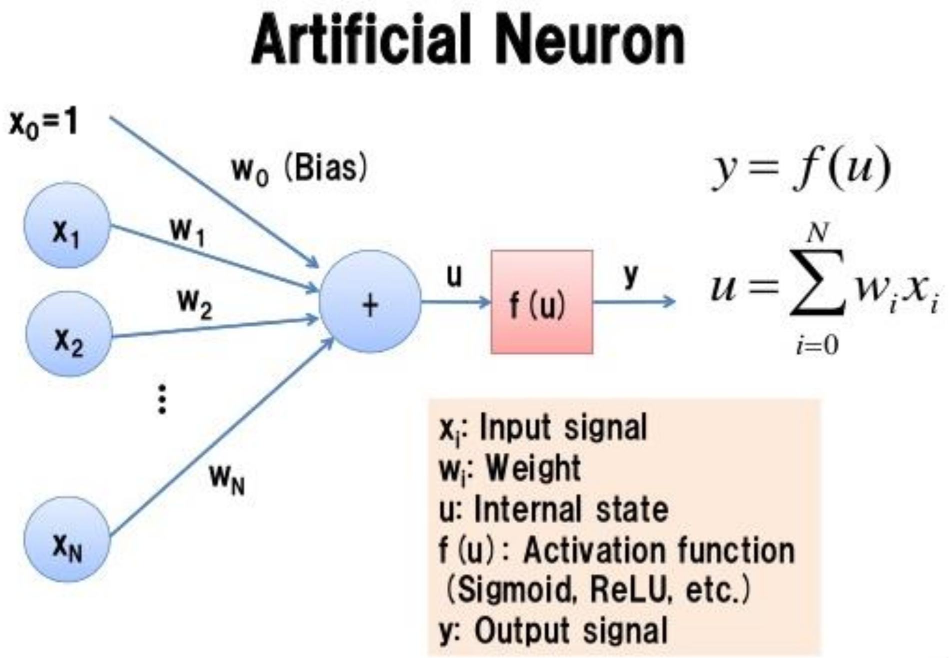
$a_{\text{out}} =$

$b_{\text{out}} =$

$y =$

Prove that $y > -5$

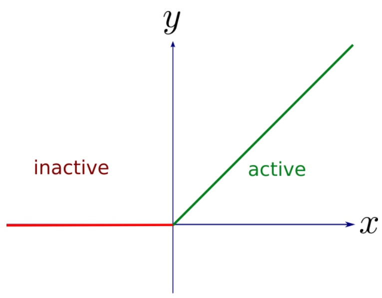
PyRAT: Python Reachability Assessment Tool Application to NN



PyRAT: Python Reachability Assessment Tool
Application to NN

$$a_{\text{out}} = \max(a_{\text{in}}, 0)$$

	a_{in}	a_{out}
$a_{\text{in}} = x_1 + x_2$	[-4,4]	
if a_{in} > 0 then	]0,4]	
$\quad a_{\text{out}} = a_{\text{in}}$		]0,4]
else	[-4,0]	
$\quad a_{\text{out}} = 0$		[0,0]
		[0,4]



$ReLU : x \to max(x, 0)$

PyRAT: Python Reachability Assessment Tool
Application to NN

$$-2 \leq x_1 \leq 2$$
$$-2 \leq x_2 \leq 2$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

$$y = -a_{\text{out}} - b_{\text{out}}$$

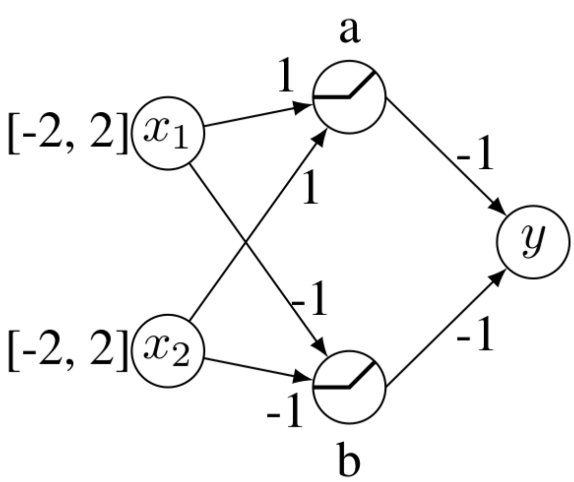
$a_{\text{in}} =$ [-2,2] + [-2,2] = [-4,4]

$b_{\text{in}} =$ - [-2,2] - [-2,2] =[-4,4]

$a_{\text{out}} =$

$b_{\text{out}} =$

$y =$



Prove that $y > -5$

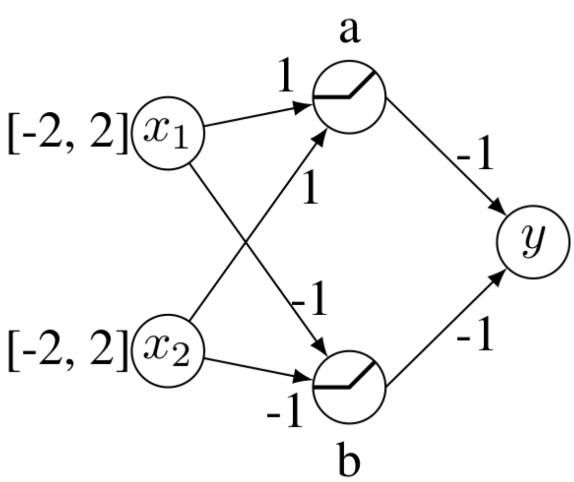
PyRAT: Python Reachability Assessment Tool
Application to NN

$$-2 \leq x_1 \leq 2$$
$$-2 \leq x_2 \leq 2$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

$$y = -a_{\text{out}} - b_{\text{out}}$$



$a_{\text{in}} =$ [-2,2] + [-2,2] = [-4,4]

$b_{\text{in}} =$ - [-2,2] - [-2,2] =[-4,4]

$a_{\text{out}} =$ [0,4]

$b_{\text{out}} =$ [0,4]

$y =$

Prove that $y > -5$

PyRAT: Python Reachability Assessment Tool
Application to NN

$$-2 \leq x_1 \leq 2$$
$$-2 \leq x_2 \leq 2$$

$$a_{\text{in}} = x_1 + x_2 \qquad b_{\text{in}} = -x_1 - x_2$$

$$a_{\text{out}} = \max(a_{\text{in}}, 0) \qquad b_{\text{out}} = \max(b_{\text{in}}, 0)$$

$$y = -a_{\text{out}} - b_{\text{out}}$$

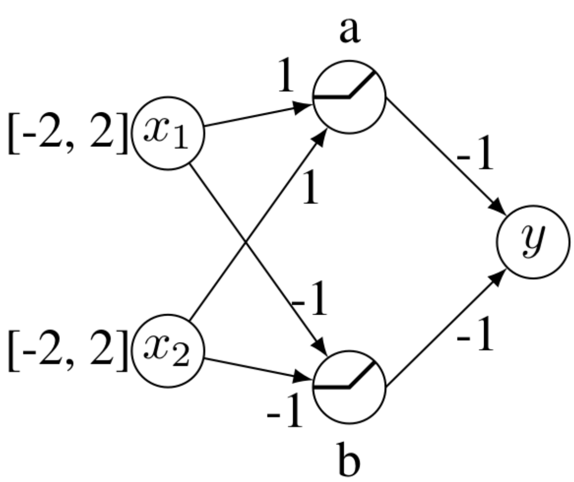
$a_{\text{in}} =$ [-2,2] + [-2,2] = [-4,4]

$b_{\text{in}} =$ - [-2,2] - [-2,2] =[-4,4]

$a_{\text{out}} =$ [0,4]

$b_{\text{out}} =$ [0,4]

$y =$ [-8,0]



Prove that $y > -5$

PyRAT: Python Reachability Assessment Tool

- 2nd at VNNComp 2024
- Written in Python with PyTorch and Numpy backend
- Supports common layers and architecture in ONNX, Keras/Tensorflow and PyTorch
- Different abstract domains implemented: Box, Zonotopes, Constrained Zonotopes, ...
- Integrated in CAISAR, an open-source platform for characterizing safety in AI systems.

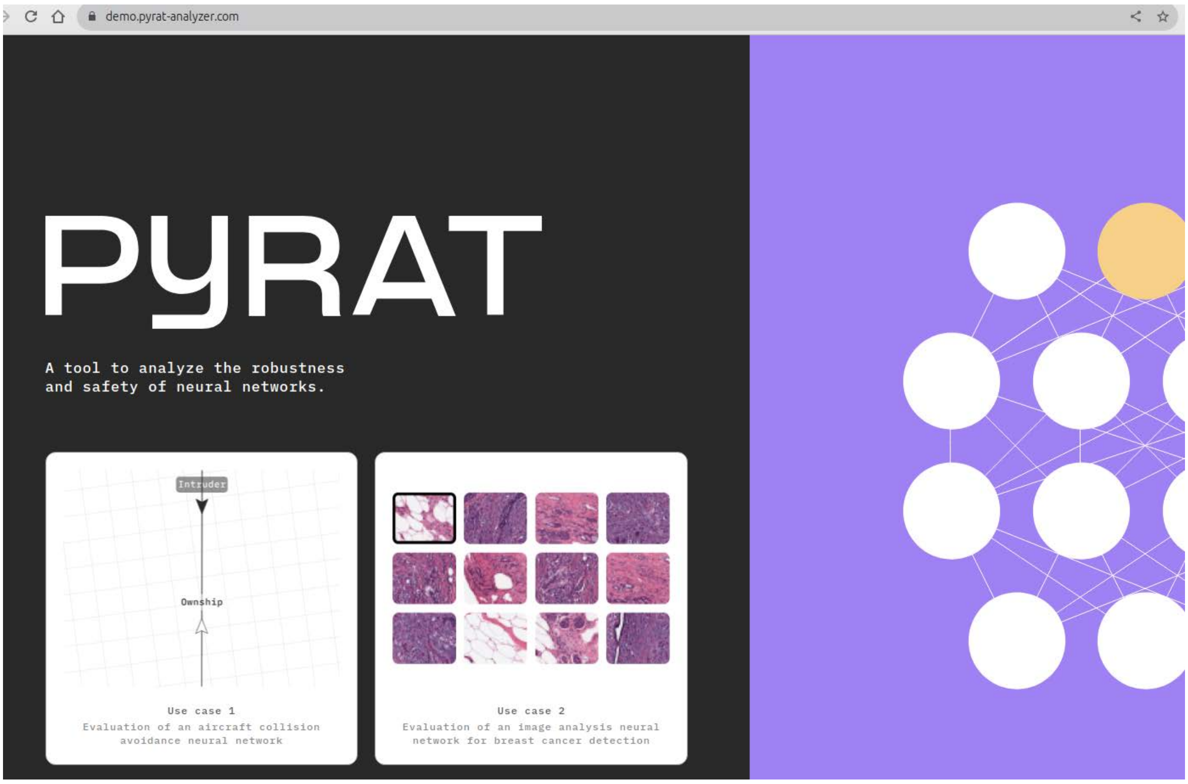
CAISAR

cea

caisar-platform.com

57

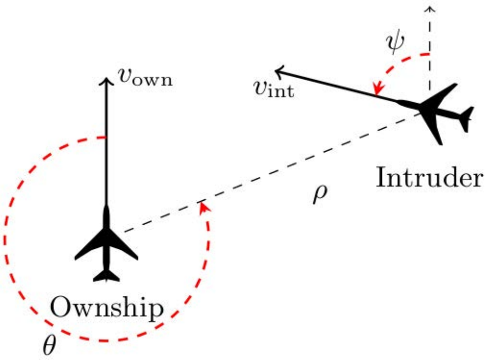
PyRAT: Python Reachability Assessment Tool

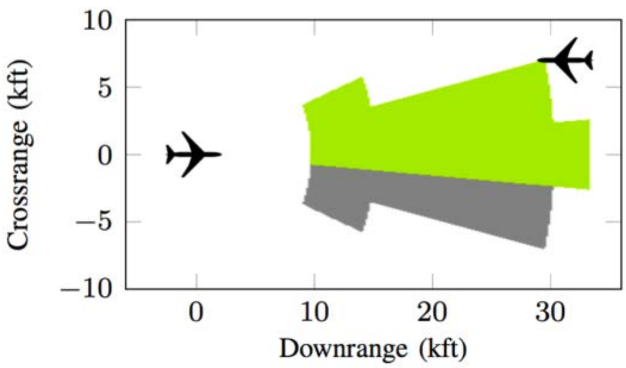


58


Rapid intro to FM
Examples for this talk:
• Property-based testing (Renault)
• Verification of functional properties (Airbus)
• Robustness evaluation (Technip)
• Out-of-Distribution detection (Thales)
59

PyRAT: Python Reachability Assessment Tool
Use-case: Airborne Collision Avoidance System

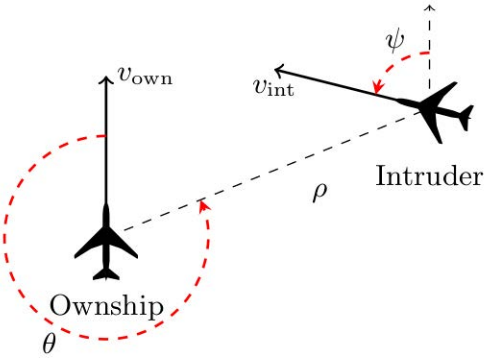


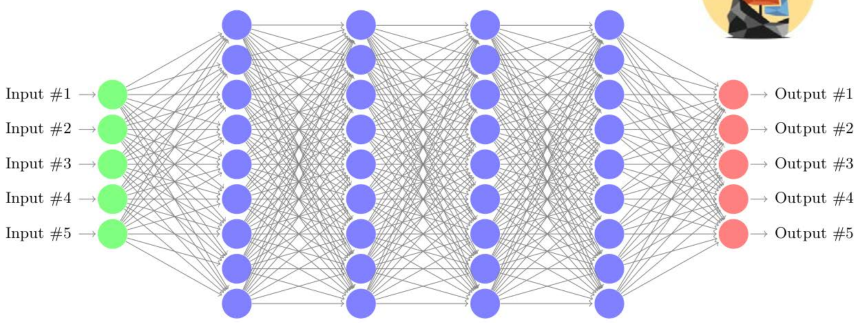


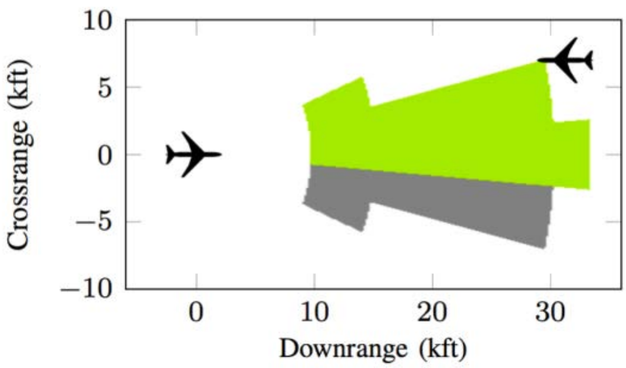
A study of an ACAS-Xu exact implementation using ED-324/ARP6983
(2024) : C, Gabreau, M-C. Teulières, E. Jenn, **Augustin Lemesle,** Dumitru Potop Butucaru, F.Thiant, L.Fischer 2, M, Turki

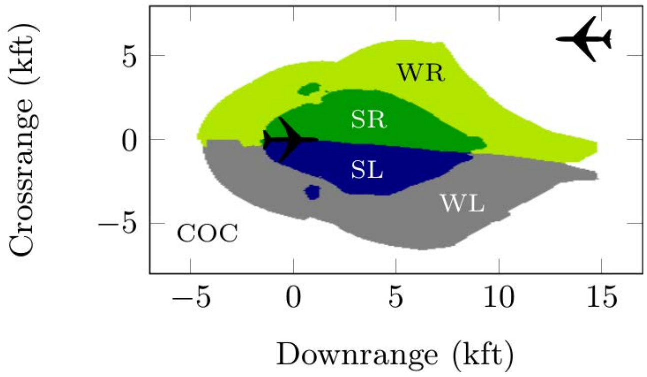
60

PyRAT: Python Reachability Assessment Tool
Use-case: Airborne Collision Avoidance System









61

PyRAT: Python Reachability Assessment Tool
Use-case: Airborne Collision Avoidance System

Input nodes
x1 ρ
x2 θ
x3 ψ
x4 v own
x5 v int

Output nodes
y1 coc
y2 weak right
y3 strong right
y4 weak left
y5 strong left

Network Functionality. The ACAS Xu system maps input variables to action advisories. Each advisory is assigned a score, with the lowest score corresponding to the best action. The input state is composed of seven dimensions (shown in Fig. 6) which represent information determined from sensor measurements [19]: (i) ρ: Distance from ownship to intruder; (ii) θ: Angle to intruder relative to ownship heading direction; (iii) ψ: Heading angle of intruder relative to ownship heading direction; (iv) v_{own}: Speed of ownship; (v) v_{int}: Speed of intruder; (vi) τ: Time until loss of vertical separation; and (vii) a_{prev}: Previous advisory. There are five outputs which represent the different horizontal advisories that can be given to the ownship: Clear-of-Conflict (COC), weak right, strong right, weak left, or strong left. Weak and strong mean heading rates of $1.5\,°/\text{s}$ and $3.0\,°/\text{s}$, respectively.

PyRAT: Python Reachability Assessment Tool
Use-case: Airborne Collision Avoidance System

Description: If the intruder is near and approaching from the left, the network advises "strong right".

63

PyRAT: Python Reachability Assessment Tool
Use-case: Airborne Collision Avoidance System

Description: If the intruder is near and approaching from the left, the network advises "strong right".

Input constraints: $250 \leq \rho \leq 400$, $0.2 \leq \theta \leq 0.4$, $-3.141592 \leq \psi \leq -3.141592 + 0.005$, $100 \leq v_{\text{own}} \leq 400$, $0 \leq v_{\text{int}} \leq 400$.

cea

64

PyRAT: Python Reachability Assessment Tool
Use-case: Airborne Collision Avoidance System

Property ϕ_1.

- Description: If the intruder is distant and is significantly slower than the ownship, the score of a COC advisory will always be below a certain fixed threshold.
- Tested on: all 45 networks.
- Input constraints: $\rho \geq 55947.691$, $v_{\text{own}} \geq 1145$, $v_{\text{int}} \leq 60$.
- Desired output property: the score for COC is at most 1500.

Rapid intro to FM

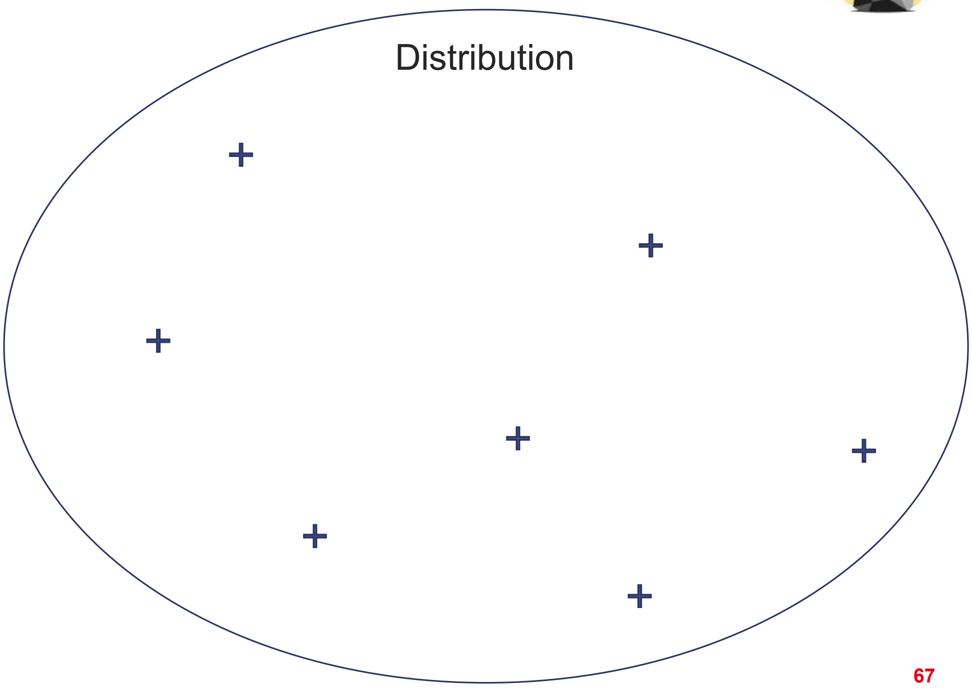
Examples for this talk:

- Property-based testing (Renault)
- Verification of functional properties (Airbus)
- Robustness evaluation (Technip)
- Out-of-Distribution detection (Thales)

66

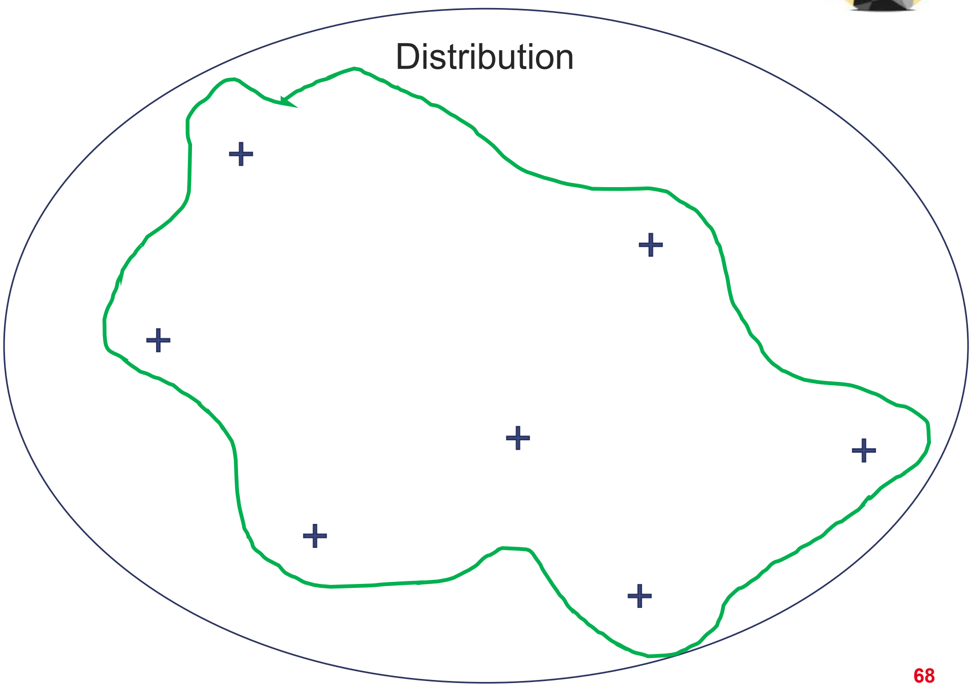
PyRAT: Python Reachability Assessment Tool
Robustness to perturbations

Ideally the selected data to build and validate the model is representative of the intended distribution.



PyRAT: Python Reachability Assessment Tool
Robustness to perturbations

Ideally the selected data to build and validate the model is representative of the intended distribution.

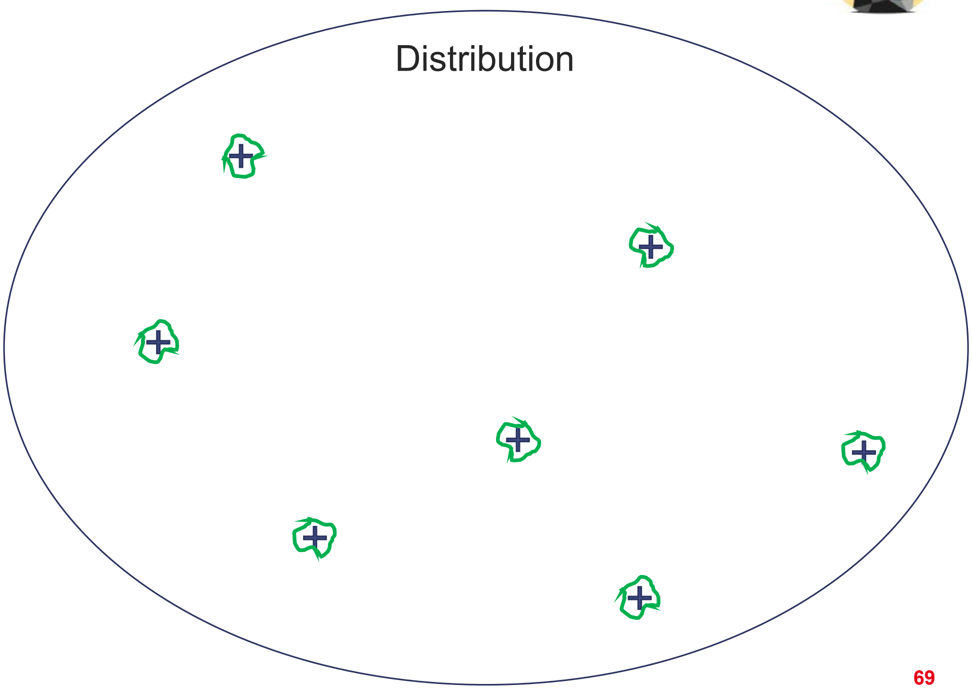


68

PyRAT: Python Reachability Assessment Tool
Robustness to perturbations

Ideally the selected data to build and validate the model is representative of the intended distribution.

What we want to avoid is a model that only knows what it was shown

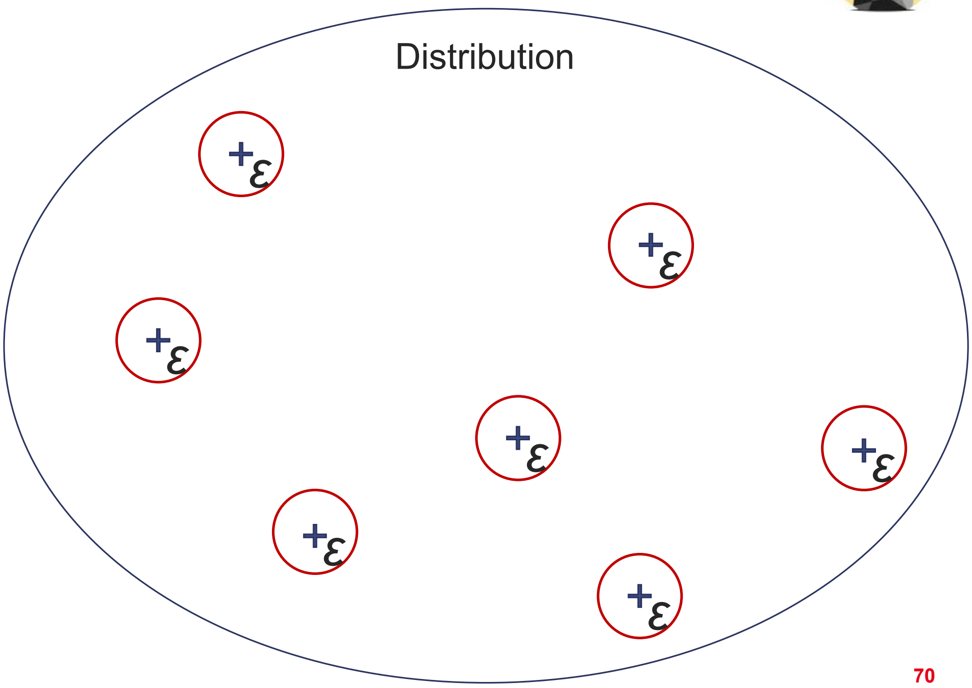


PyRAT: Python Reachability Assessment Tool
Robustness to perturbations

Ideally the selected data to build and validate the model is representative of the intended distribution.

What we want to avoid is a model that only knows what it was shown

How does it behave with the neighborhood of selected data?



70

PyRAT: Python Reachability Assessment Tool
Robustness to perturbations

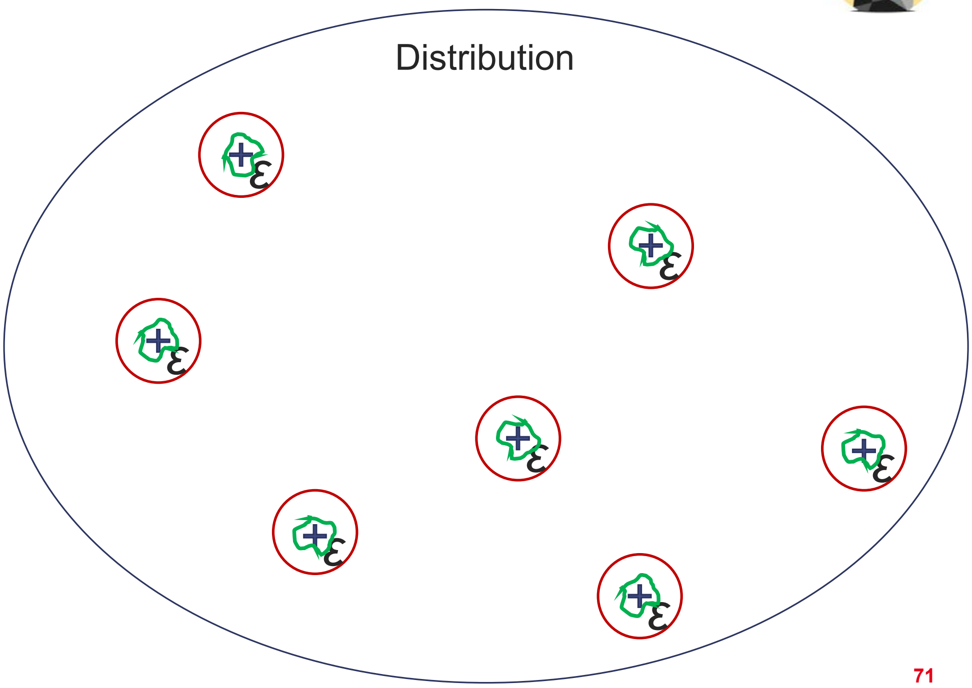
Ideally the selected data to build and validate the model is representative of the intended distribution.

What we want to avoid is a model that only knows what it was shown

How does it behave with the neighborhood of selected data?

What is it good for?

Can detect this…



71

PyRAT: Python Reachability Assessment Tool
Robustness to perturbations

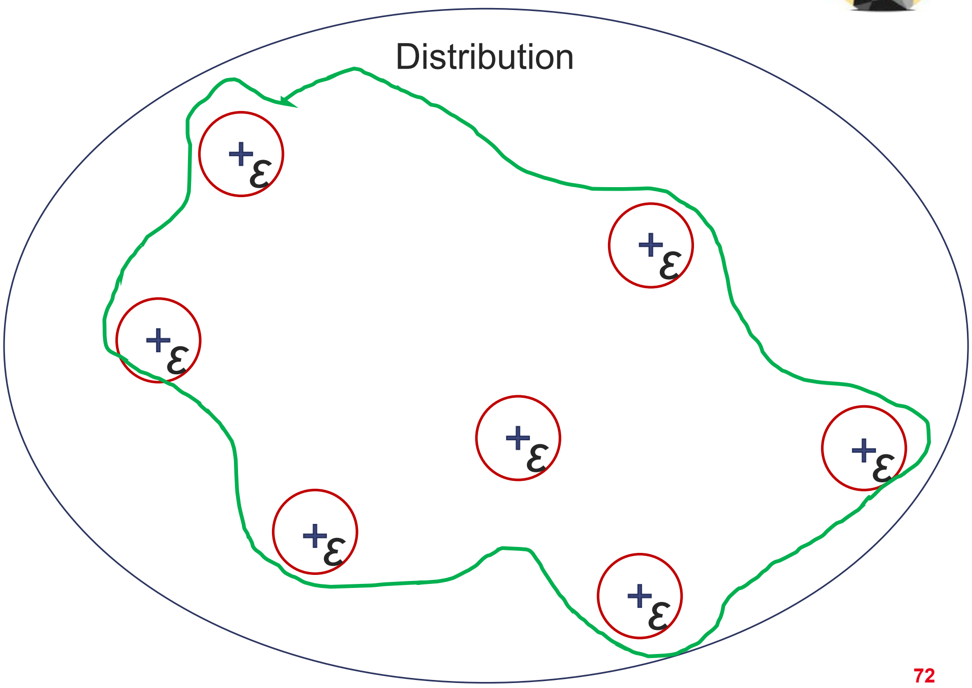
Ideally the selected data to build and validate the model is representative of the intended distribution.

What we want to avoid is a model that only knows what it was shown

How does it behave with the neighborhood of selected data?

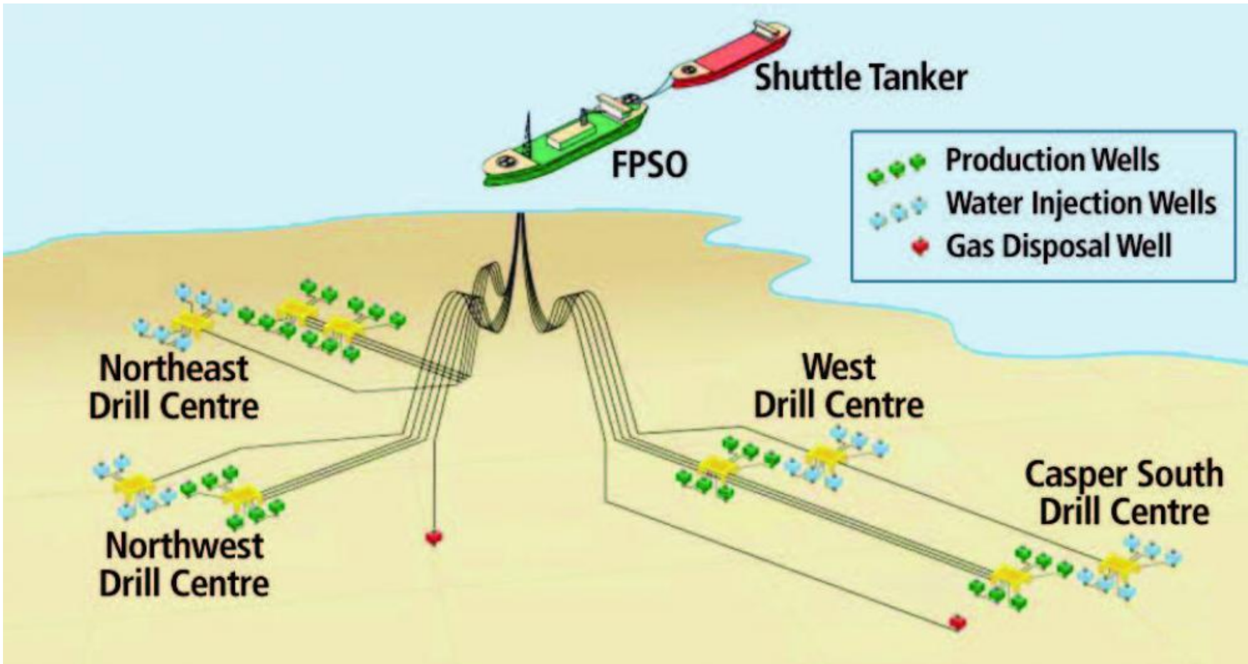
What is it good for?

... But doesn't imply this



72

PyRAT: Python Reachability Assessment Tool
Use-case: Mooring lines failure detection

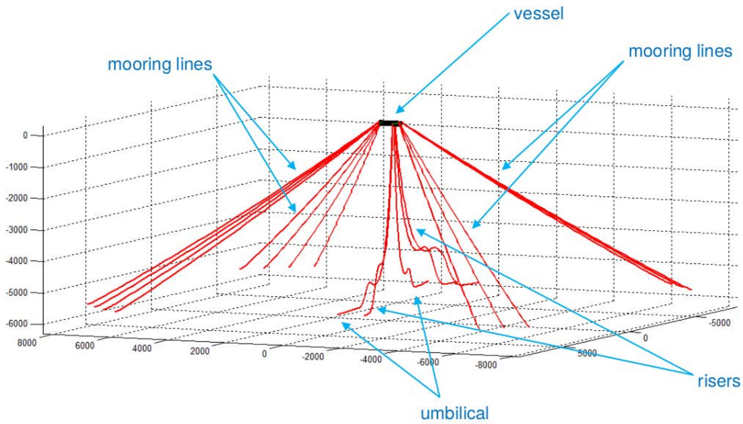


73

PyRAT: Python Reachability Assessment Tool
Use-case: Mooring lines failure detection

Mooring incidents (DeepStar® data from 1997-2012):

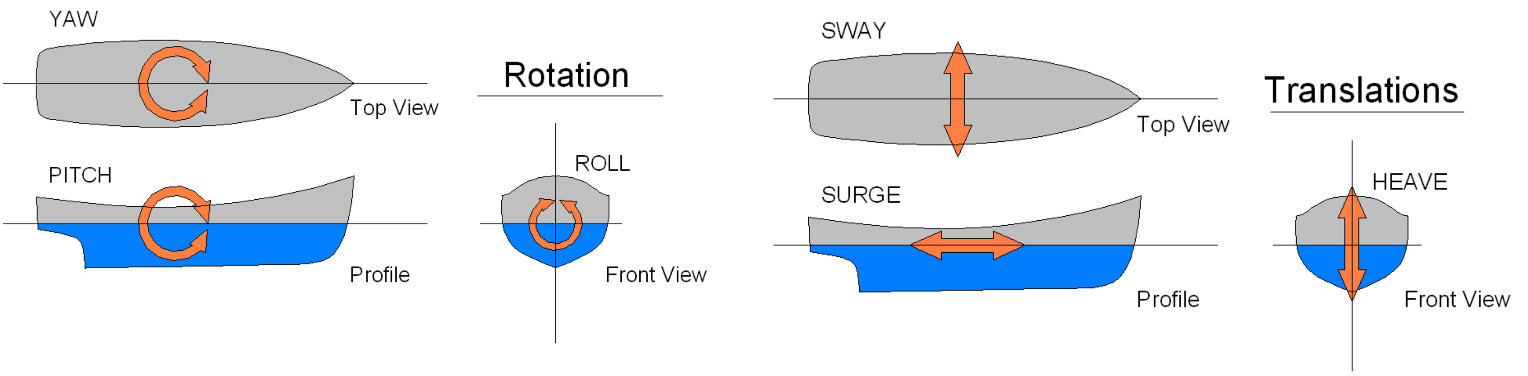
- 107 incidents from 73 facilities across the industry
- Potentially dire consequences
- Many FPSO have no means of monitoring lines
- Those who do face technical problems (robustness of equipment)



74

PyRAT: Python Reachability Assessment Tool
Use-case: Mooring lines failure detection

Patented dry monitoring detection systems, based on vessel positions and low-frequency periods (which can be obtained from Dual GPS)



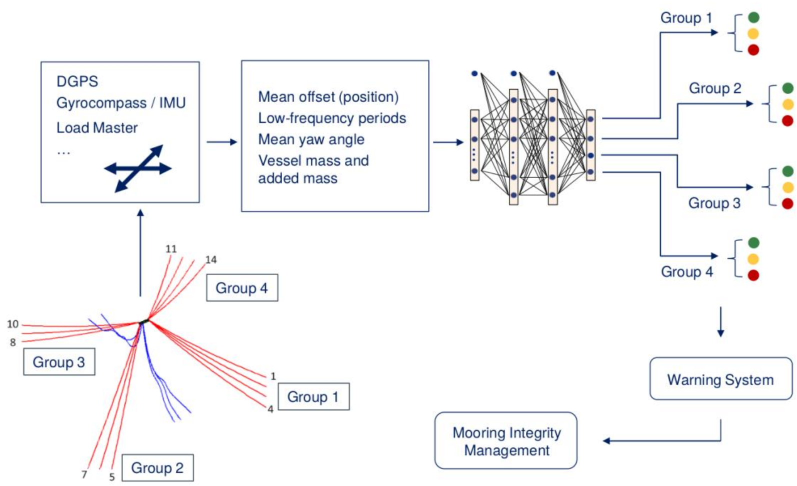
75

PyRAT: Python Reachability Assessment Tool
Use-case: Mooring lines failure detection

- Highly non-linear problem (machine learning to recognize and classify patterns)
- Ability to deal with some degrees of variations from various system components (such as mooring line stiffness) and with error or noise from monitoring system
- Cover a complete range of vessel drafts, expected vessel responses from environment conditions and directions and mooring line conditions

The model

- Input: Vessel movement, mass, offset, …
- Output: group-line failures



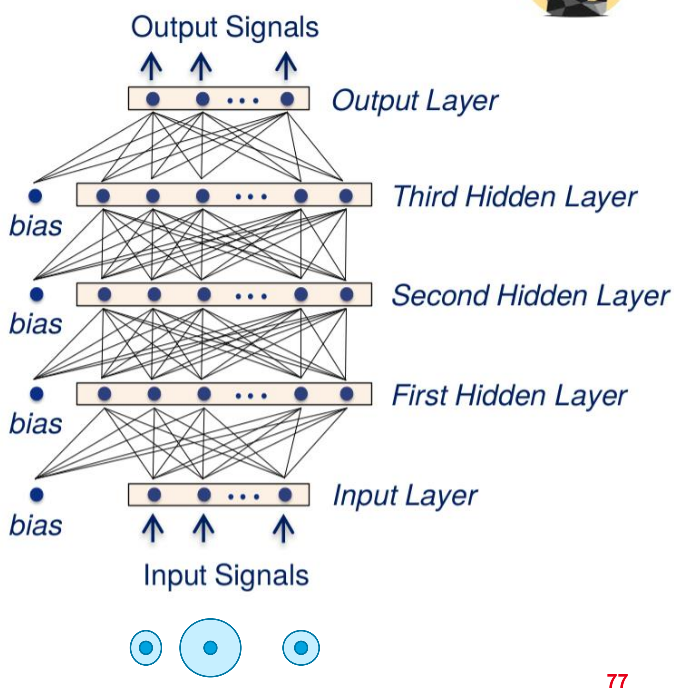
76

PyRAT: Python Reachability Assessment Tool
Use-case: Mooring lines failure detection

Ensuring robustness properties

- Stability of classification in presence of perturbation
- Perturbation per input (sensor sensitivity)
- Different perturbations for different inputs

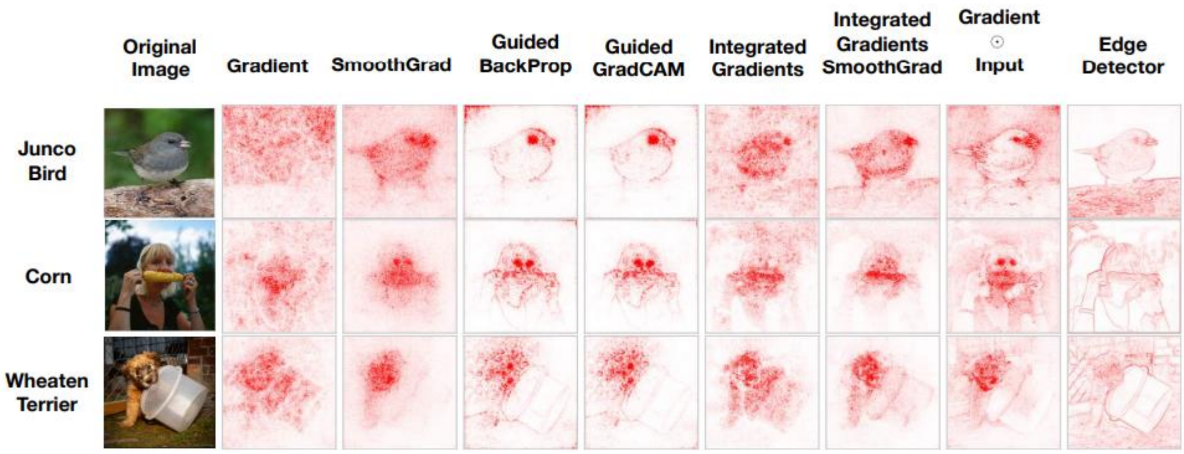
(Also verified functional properties but NDA)



77


Rapid intro to FM
Examples for this talk:
• Property-based testing (Renault)
• Verification of functional properties (Airbus)
• Robustness evaluation (Technip)
• Out-of-Distribution detection (Thales)
78

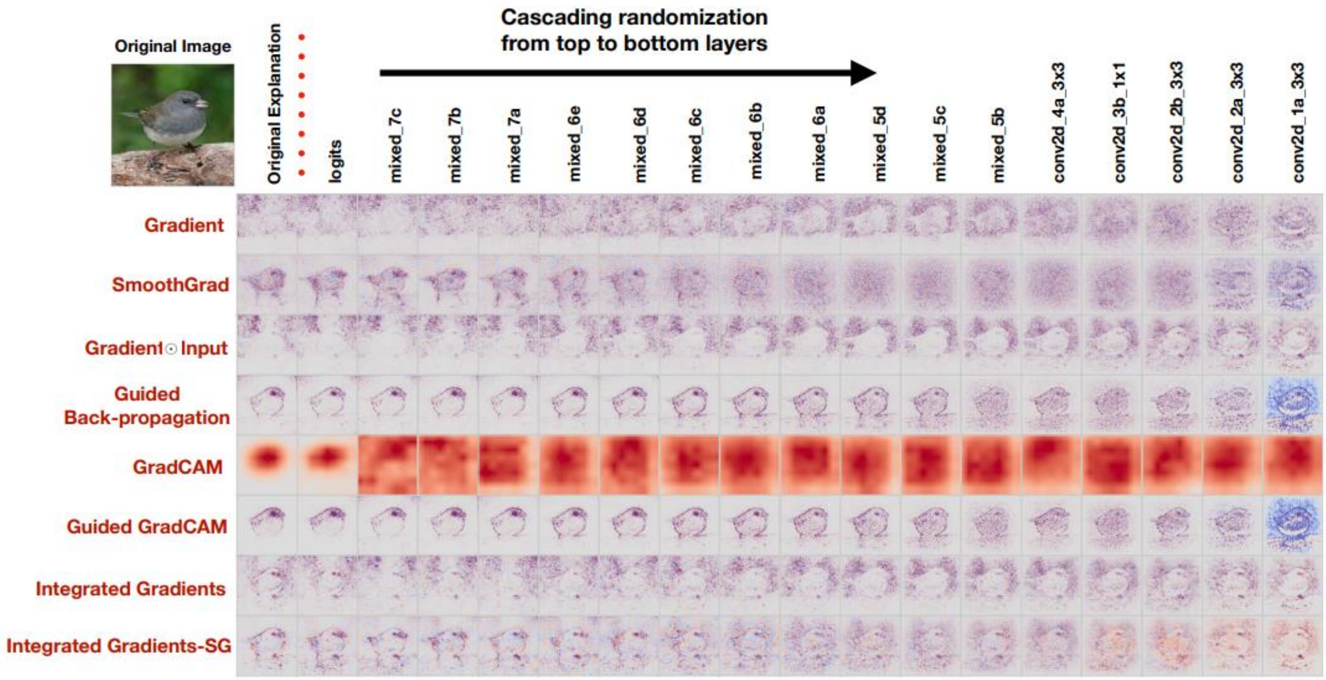
Peeking in the black box: saliency maps, post-hoc XAI



Sanity Checks for Saliency Maps

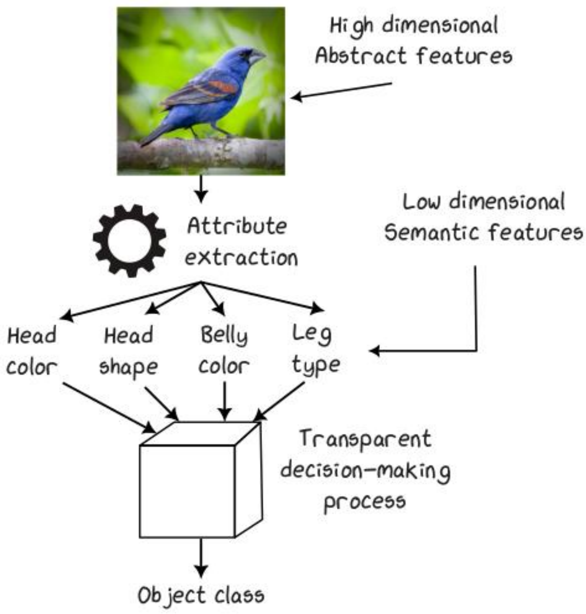
Julius Adebayo*, Justin Gilmer♯, Michael Muelly♯, Ian Goodfellow♯, Moritz Hardt♯†, Been Kim♯

Peeking in the black box: saliency maps, post-hoc XAI



80

Peeking in the black box: XAI by design

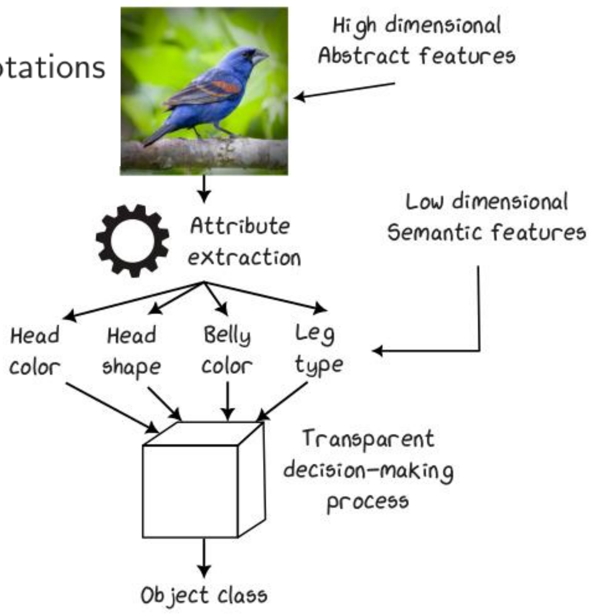


Peeking in the black box: XAI by design

Extract **semantic** information

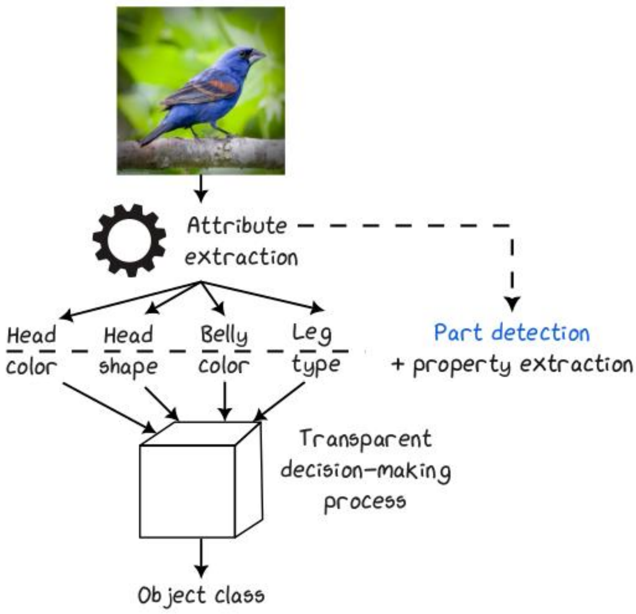
Attribute learning requires annotations

- Annotations are expensive
- Annotations can be incorrect



82

Peeking in the black box: XAI by design



PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

Original

Goal

Xu-Darme, R., Quénot, G., Chihani, Z., Rousset, MC. (2023). **PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning**. In: Rousseau, JJ., Kapralos, B. (eds) Pattern Recognition, Computer Vision, and Image Processing. XAIE 2022 International Workshops and Challenges. ICPR 2022. Lecture Notes in Computer Science, vol 13645. Springer, Cham. https://doi.org/10.1007/978-3-031-37731-0_14

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

Xu-Darme, R., Quénot, G., Chihani, Z., Rousset, MC. (2023). **PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning**. In: Rousseau, JJ., Kapralos, B. (eds) Pattern Recognition, Computer Vision, and Image Processing. XAIE 2022 International Workshops and Challenges. ICPR 2022. Lecture Notes in Computer Science, vol 13645. Springer, Cham. https://doi.org/10.1007/978-3-031-37731-0_14

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

Xu-Darme, R., Quénot, G., Chihani, Z., Rousset, MC. (2023). **PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning**. In: Rousseau, JJ., Kapralos, B. (eds) Pattern Recognition, Computer Vision, and Image Processing. XAIE 2022 International Workshops and Challenges. ICPR 2022. Lecture Notes in Computer Science, vol 13645. Springer, Cham. https://doi.org/10.1007/978-3-031-37731-0_14

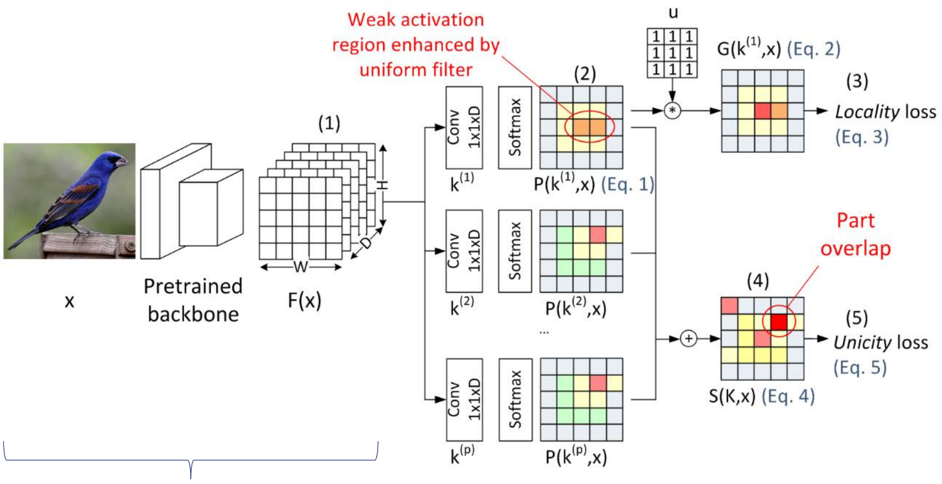
86

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

Xu-Darme, R., Quénot, G., Chihani, Z., Rousset, MC. (2023). **PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning**. In: Rousseau, JJ., Kapralos, B. (eds) Pattern Recognition, Computer Vision, and Image Processing. XAIE 2022 International Workshops and Challenges. ICPR 2022. Lecture Notes in Computer Science, vol 13645. Springer, Cham. https://doi.org/10.1007/978-3-031-37731-0_14

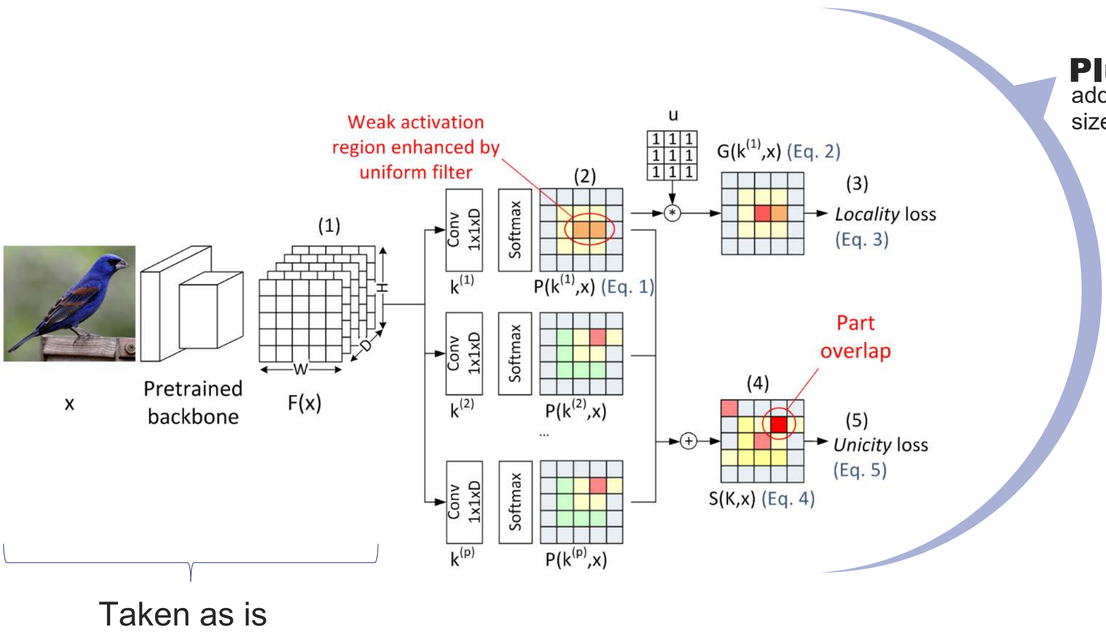
87

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning



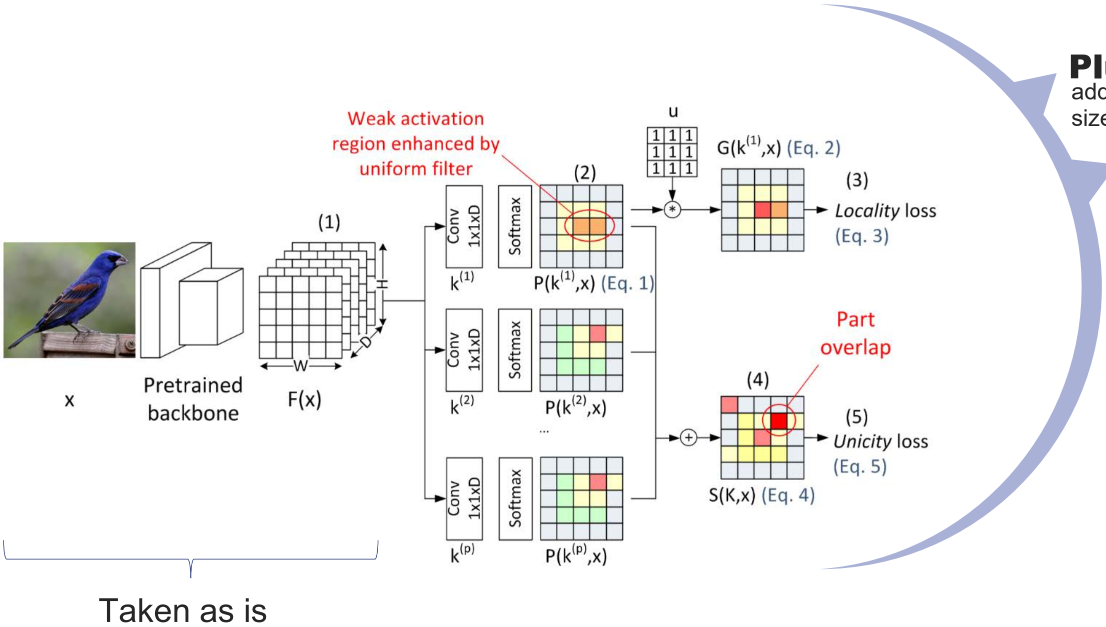
Taken as is

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning



Pluggable
added to existing backbone, fraction of the size

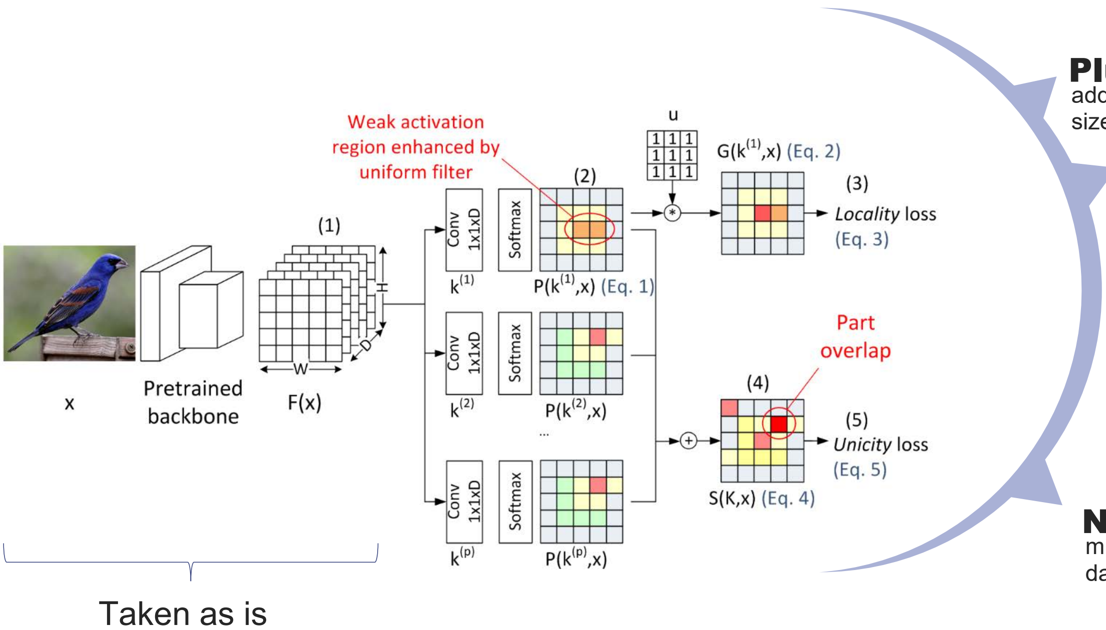
PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning



Pluggable
added to existing backbone, fraction of the size

Frugal
no need to fine-tune the backbone

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning



Pluggable
added to existing backbone, fraction of the size

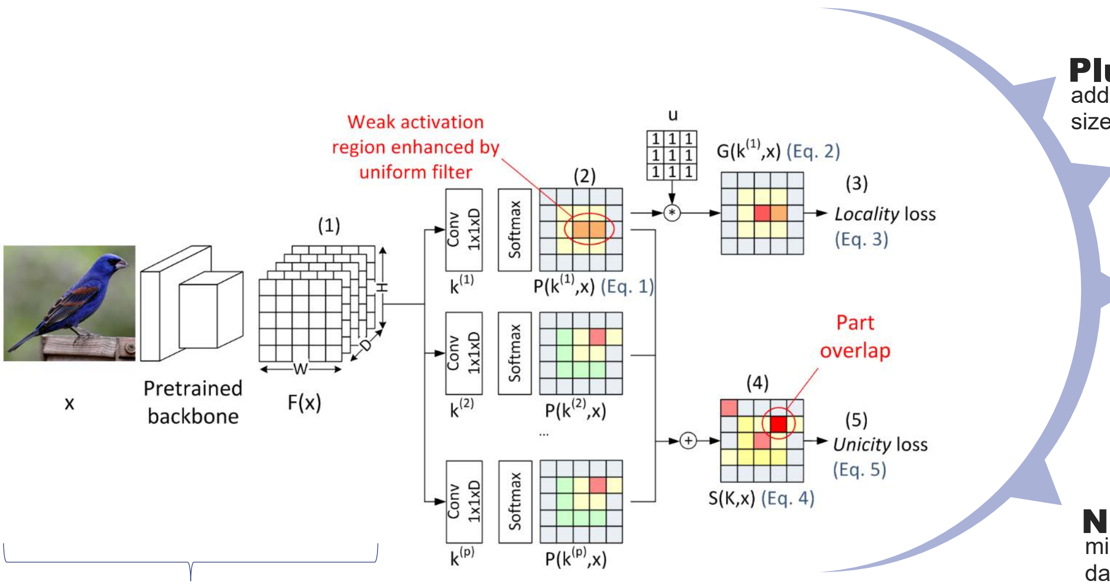
Frugal
no need to fine-tune the backbone

Non-invasive
minimal access to backbone, fraction of the data

91

PATENTED

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning



Taken as is

Pluggable
added to existing backbone, fraction of the size

Frugal
no need to fine-tune the backbone

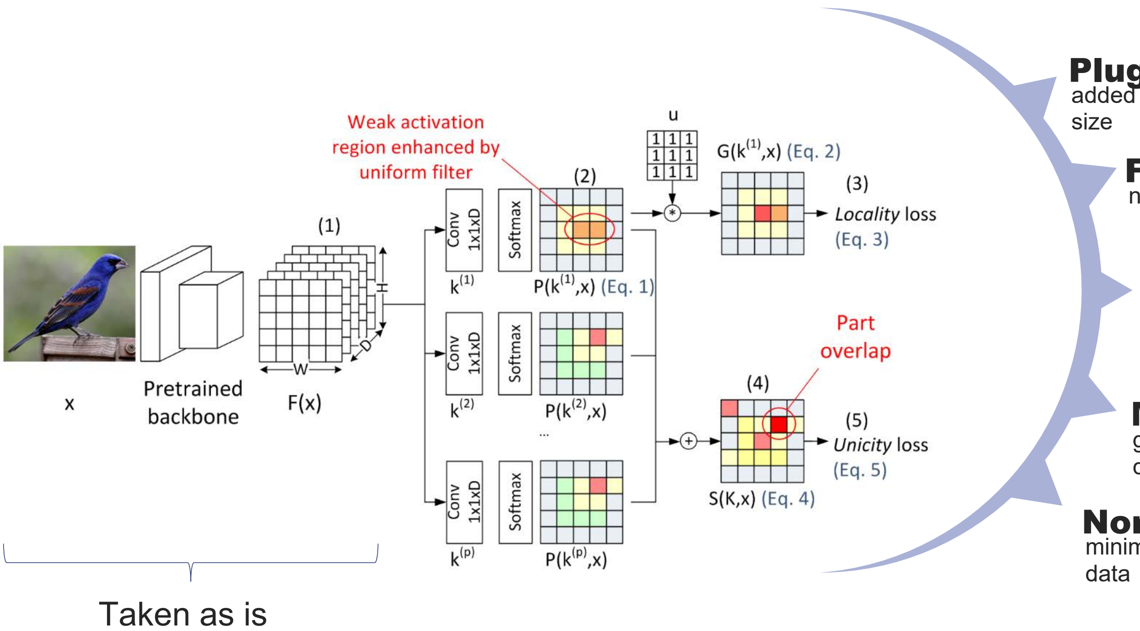
Fast
convergence in a few epochs

Non-invasive
minimal access to backbone, fraction of the data

92

PATENTED

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning



Pluggable
added to existing backbone, fraction of the size

Frugal
no need to fine-tune the backbone

Fast
convergence in a few epochs

Measured
gives confidence measures of the detections

Non-invasive
minimal access to backbone, fraction of the data

cea

93

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

94

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

95

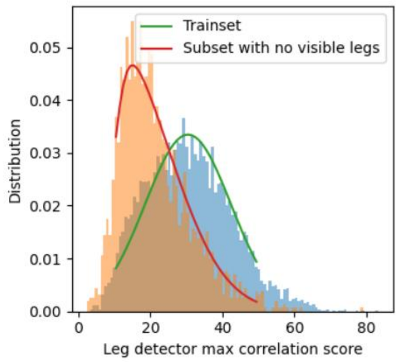
PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

Detectors have no "knowledge" of which part they are detecting

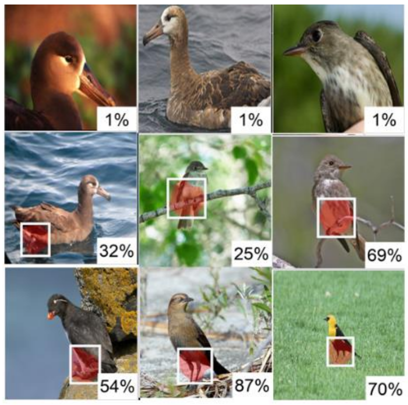
Need for manual (human) definition (semantic value)

96

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning



(a) Distribution of maximum correlation scores on the CUB-200 training set (in blue) and on a subset containing only images with non-visible legs (red).



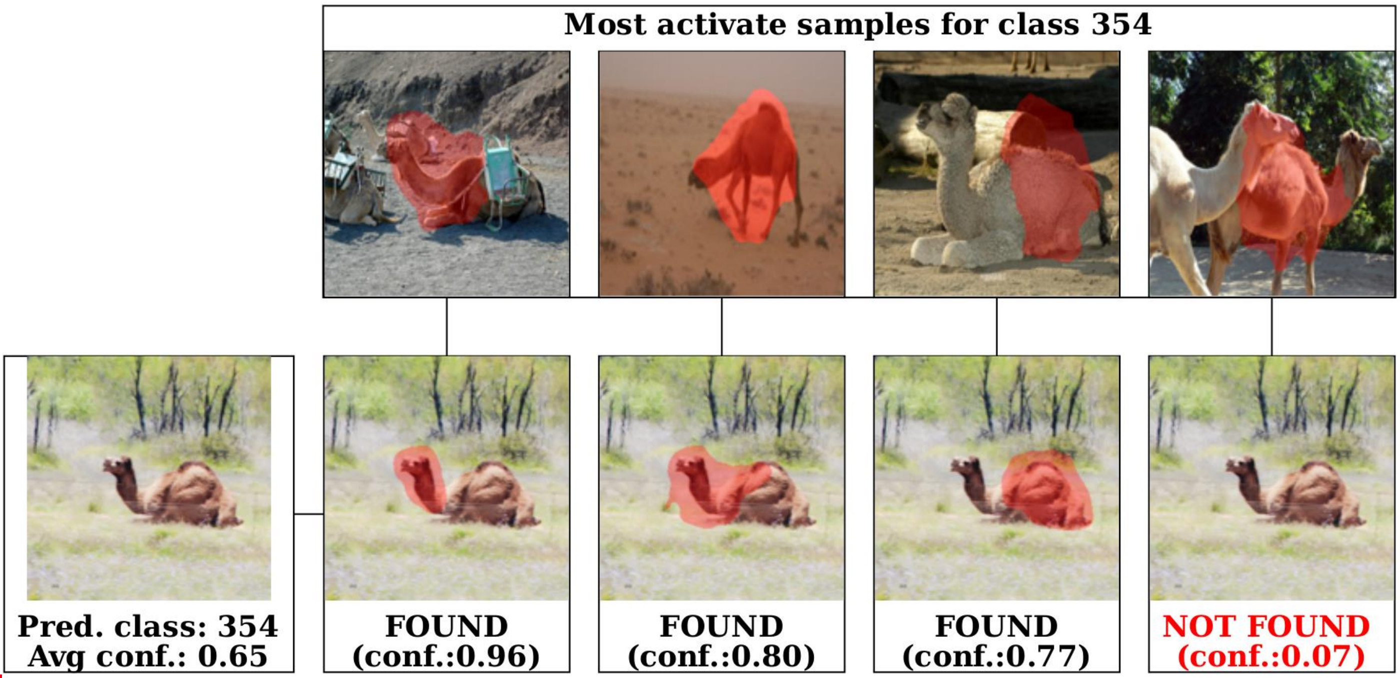
(b) Confidence scores and part visualizations on images with non visible legs (top-row) and with visible legs (bottom rows).

PARTICUL: Part Identification with Confidence Measure Using Unsupervised Learning

What can be done for one macro class can be done for many micro classes

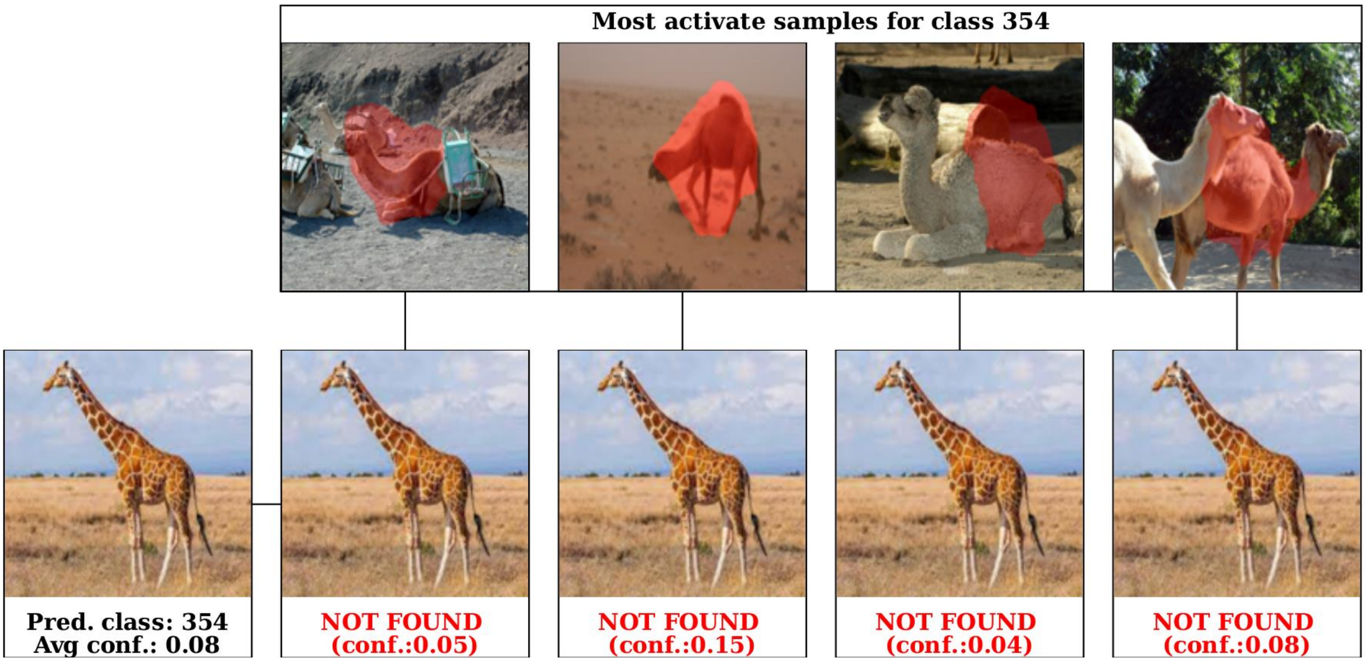
=> Set of detectors for each class

CODE: Contextualised Out-of-Distribution Detection Using Pattern Identification



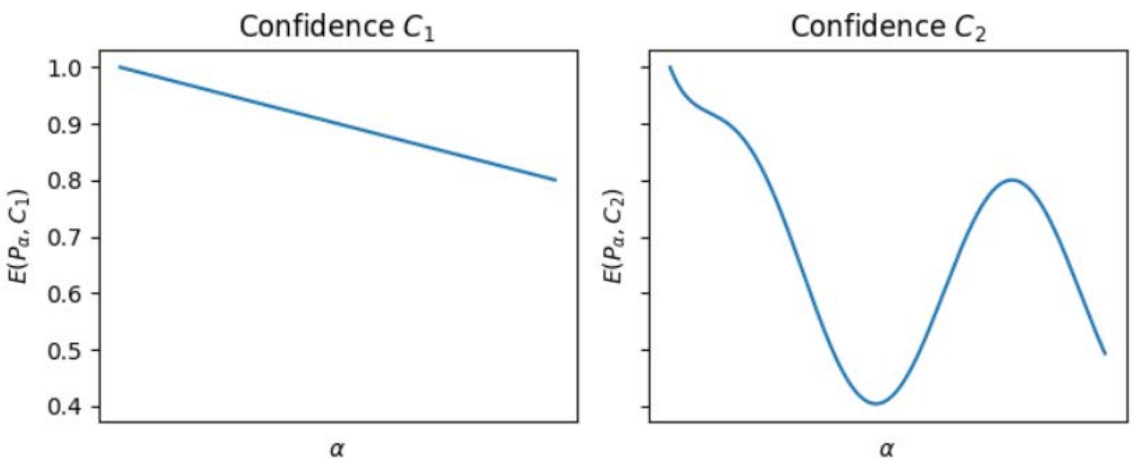
99

CODE: Contextualised Out-of-Distribution Detection Using Pattern Identification



100

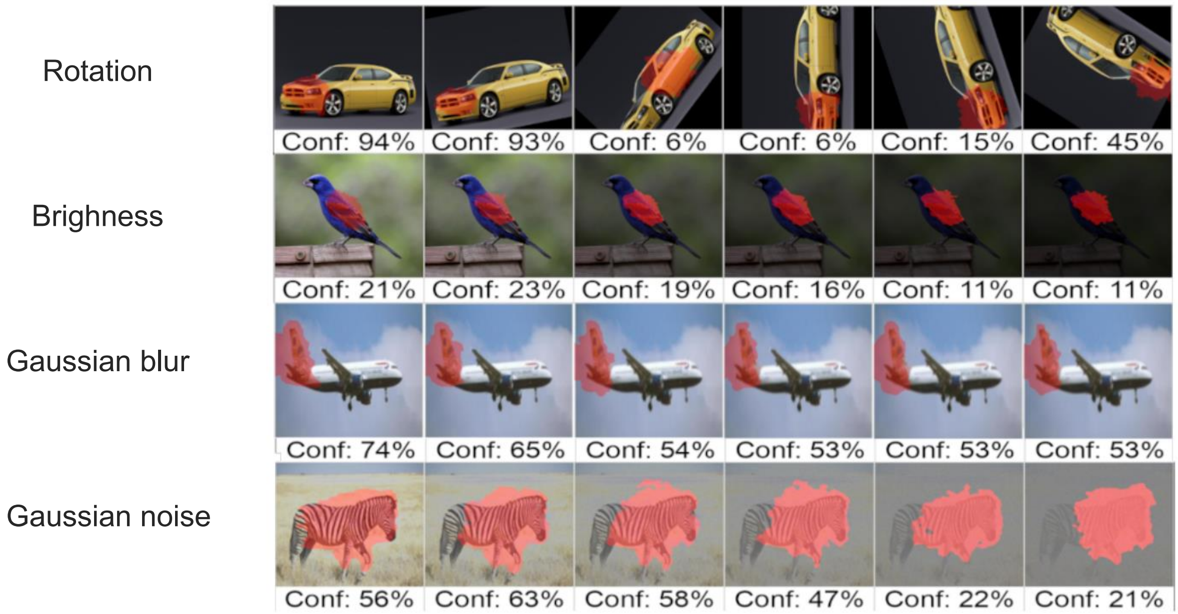
CODE: Contextualised Out-of-Distribution Detection Using Pattern Identification



Xu-Darme, R., Girard-Satabin, J., Hond, D., Incorvaia, G., Chihani, Z. (2023). Contextualised Out-of-Distribution Detection Using Pattern Identification. In: Guiochet, J., Tonetta, S., Schoitsch, E., Roy, M., Bitsch, F. (eds) Computer Safety, Reliability, and Security. SAFECOMP 2023 Workshops. SAFECOMP 2023. Lecture Notes in Computer Science, vol 14182. Springer, Cham. https://doi.org/10.1007/978-3-031-40953-0_36

101

CODE: Contextualised Out-of-Distribution Detection Using Pattern Identification



Xu-Darme, R., Girard-Satabin, J., Hond, D., Incorvaia, G., Chihani, Z. (2023). Contextualised Out-of-Distribution Detection Using Pattern Identification. In: Guiochet, J., Tonetta, S., Schoitsch, E., Roy, M., Bitsch, F. (eds) Computer Safety, Reliability, and Security. SAFECOMP 2023 Workshops. SAFECOMP 2023. Lecture Notes in Computer Science, vol 14182. Springer, Cham. https://doi.org/10.1007/978-3-031-40953-0_36

102

CODE: Contextualised Out-of-Distribution Detection Using Pattern Identification

Dissimilarity measures based on neuron activation bounds sometimes exhibit higher confidence on perturbed input

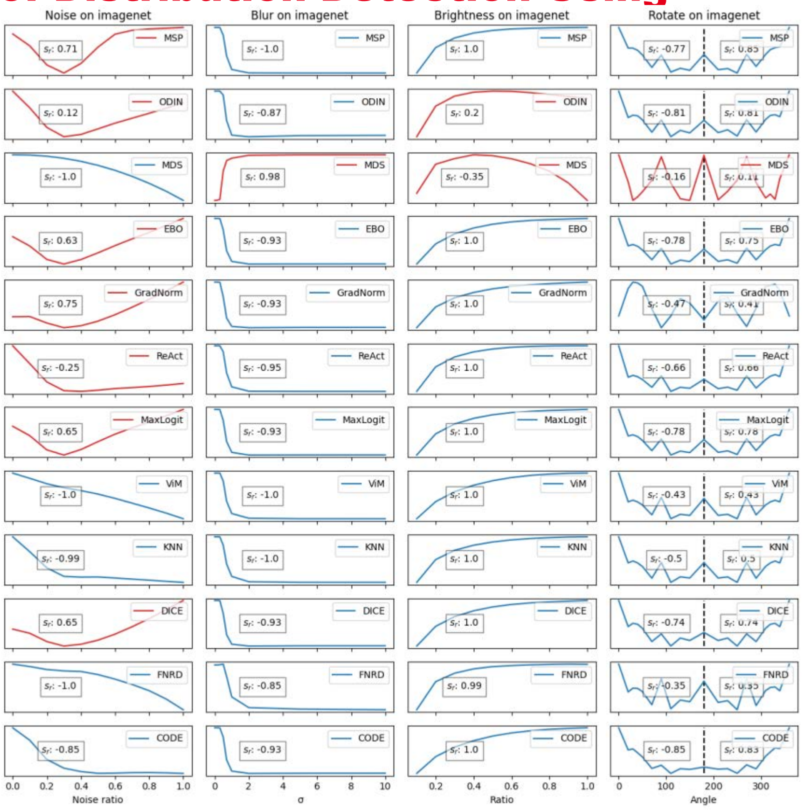
- => Maybe because some perturbations lower the amplitude of activation values, thus decreasing the propbabilty of activation outside of the bounds

Rotation is… wavy

- => Black filling at the angles ?

CODE seems consistent across perturbations and datasets (CIFAR10, CIFAR100, ImageNet).

Xu-Darme, R., Girard-Satabin, J., Hond, D., Incorvaia, G., Chihani, Z. (2023). Contextualised Out-of-Distribution Detection Using Pattern Identification. In: Guiochet, J., Tonetta, S., Schoitsch, E., Roy, M., Bitsch, F. (eds) Computer Safety, Reliability, and Security. SAFECOMP 2023 Workshops. SAFECOMP 2023. Lecture Notes in Computer Science, vol 14182. Springer, Cham. https://doi.org/10.1007/978-3-031-40953-0_36



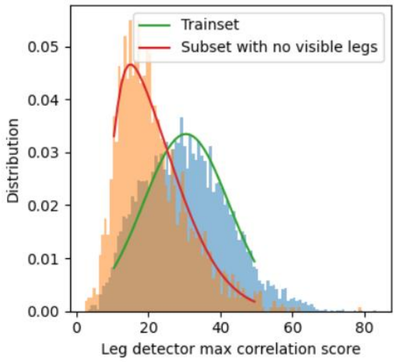
03

“ Couldn’t this be used for adversarial attacks ?

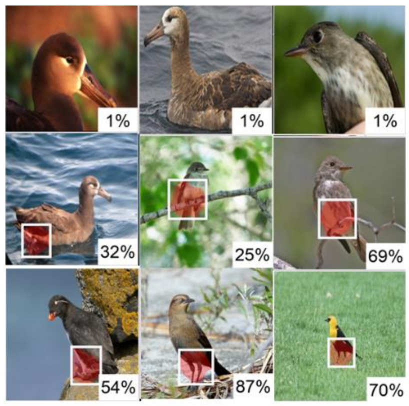
Small detour from safety to security

104

PARTICUL as a way to detect discrepancy between classification and part detection



(a) Distribution of maximum correlation scores on the CUB-200 training set (in blue) and on a subset containing only images with non-visible legs (red).

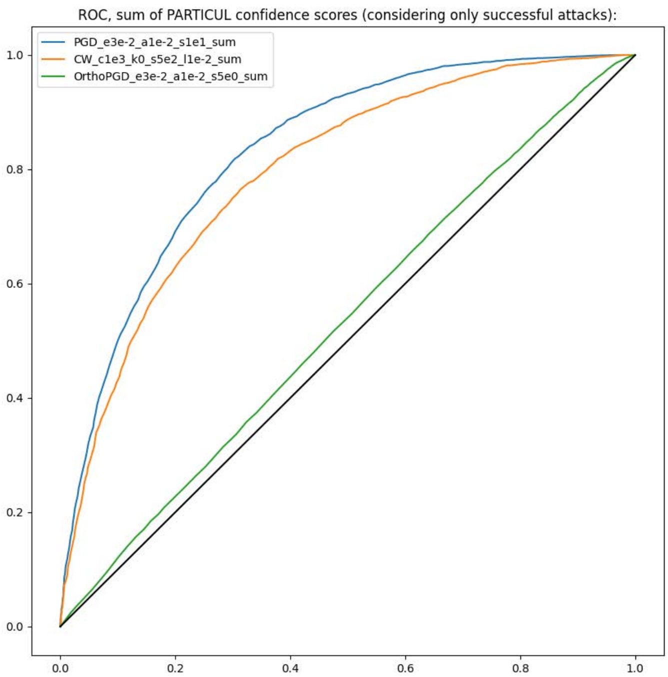


(b) Confidence scores and part visualizations on images with non visible legs (top-row) and with visible legs (bottom rows).

PARTICUL for AdvEx: WIP

Basic principle:

- Part detectors say « wolf teeth, wolf eyes, wolf ears, wolf paws »
- Classifier says « grandma »



106


“ This looks fun, I want to try some ideas!
XAI enthusiast
cea
107

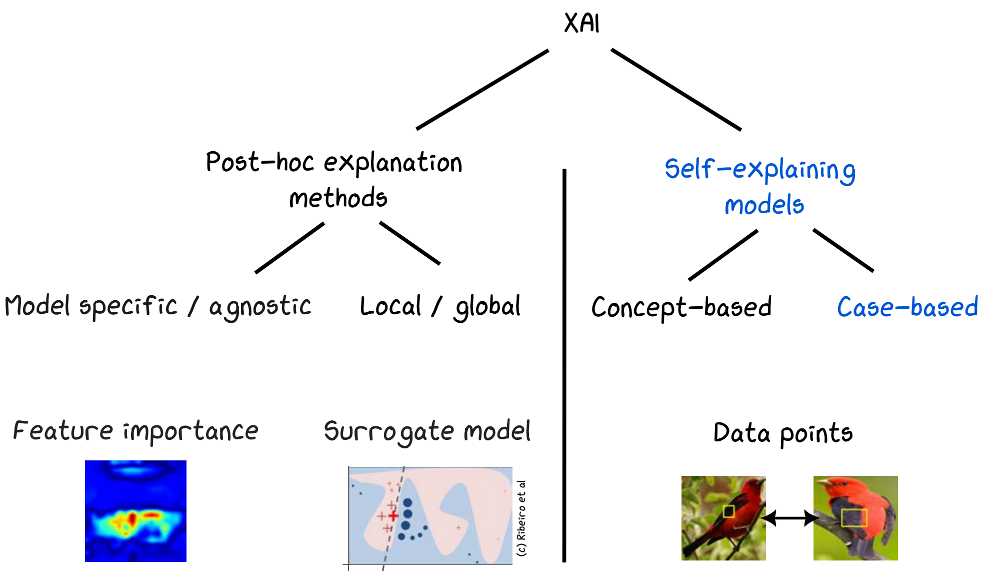
CaBRNet : Case-Based Reasoning for NN

Plugging CaBRNet presentation

CaBRNet: Case-Based Reasoning through Neural Network

Romain Xu-Darme, Aymeric Varasse, **Alban Grastien**, Julien Girard-Satabin, Jules Soria, Zakaria Chihani (CEA-LIST/DILS/LSL)

Explainable AI taxonomy



list

General outline

What is Case-based Reasoning?

Introducing CaBRNet

The road ahead

1. What is Case-based Reasoning?

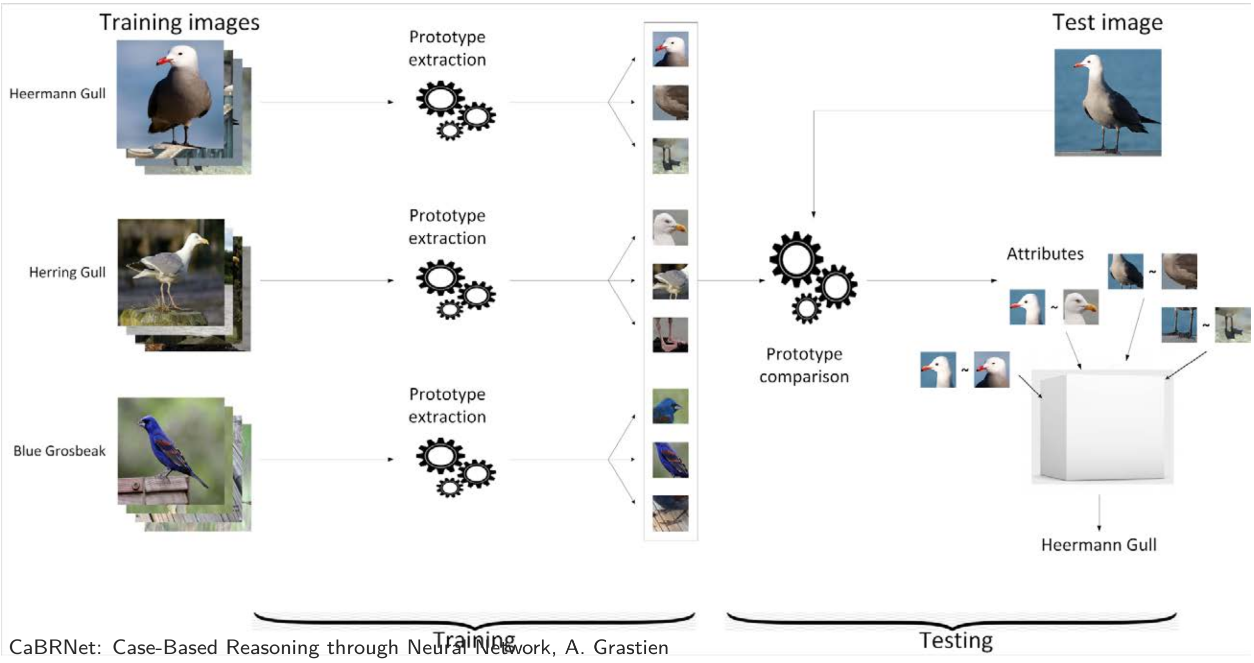
What is Case-based Reasoning (CBR)?

Case-based reasoning means using old experiences to understand and solve new problems. In case-based reasoning, a reasoner ***remembers*** *a previous situation similar to the current one and uses that to solve the new problem.* [1]

Remembering = memorizing previous examples:

- In the context of ML, previous examples = training data. All training data?
- No, a subset of well-chosen representatives called **prototypes**.

Case-based reasoning for images (in theory)



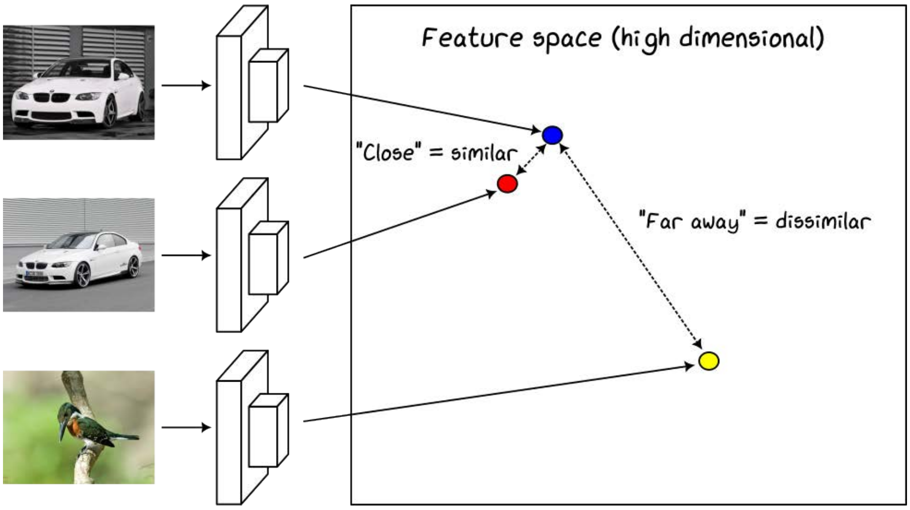
cea list

The visual similarity challenge

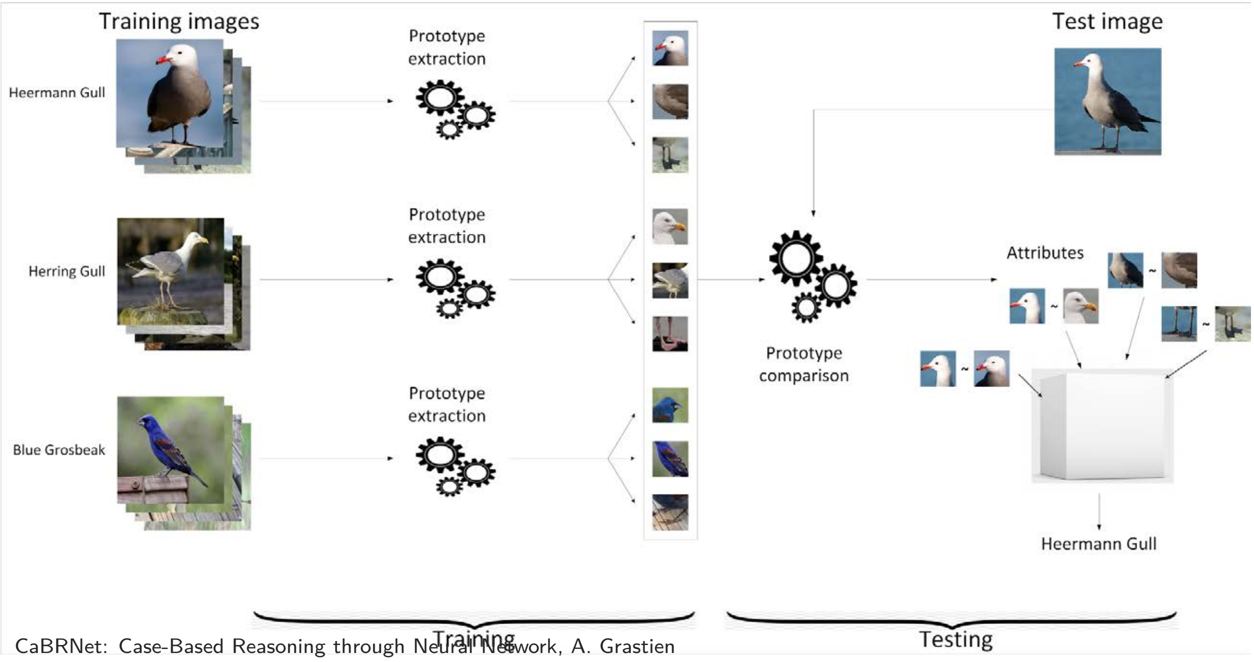
cea list

Solution: using CNNs for embeddings

Hypothesis: Proximity in feature space ⇔ similarity in visual space

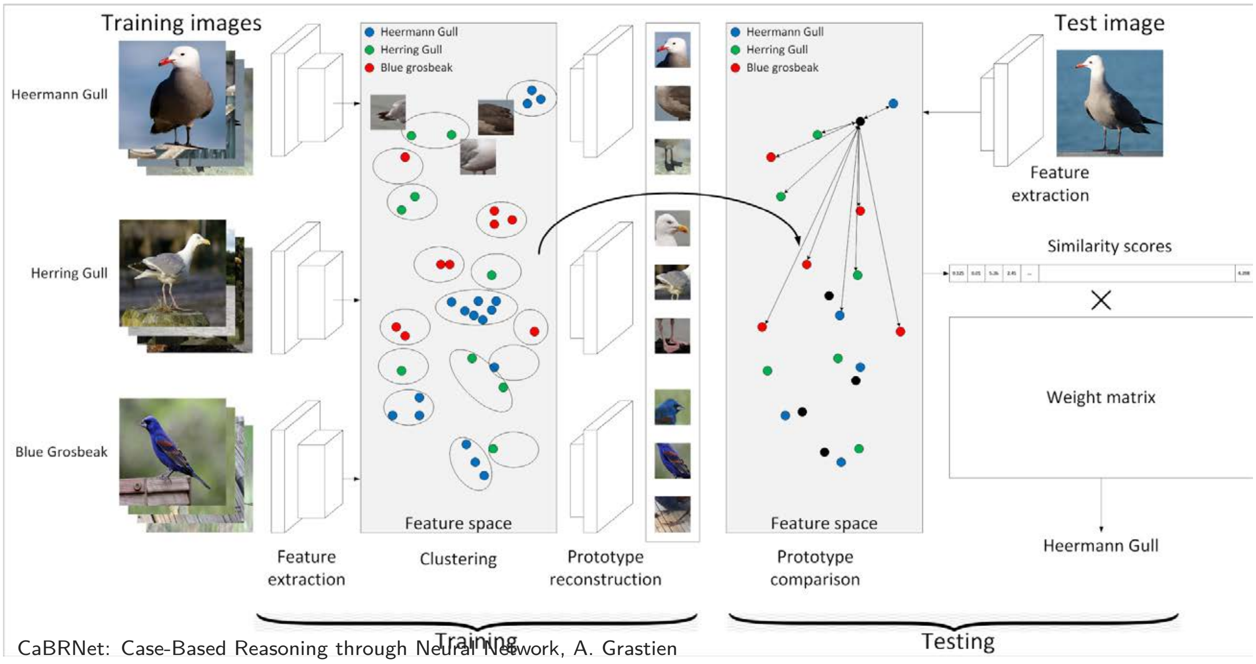


Case-based reasoning for images (theory → practice)



CaBRNet: Case-Based Reasoning through Neural Network, A. Grastien

Case-based reasoning for images (theory → practice)



CaBRNet: Case-Based Reasoning through Neural Network, A. Grastien 8/20

cea list

State of the art

- ProtoPNet [2]: classification = weighted sum of similarity scores to class-specific prototypes
- ProtoTree [3]: classification = decision tree (prototypes shared among classes)
- ProtoPShare [4]: ProtoPNet + prototype merging
- ProtoPool [5]: ProtoPNet + prototype sharing
- TESNet [6]: ProtoPNet + replaces L2 distance with projection metric
- PIPNet [7]: Representation learning + cosine distance + visual similarity as an explicit learning objective
- CASUAL [8]: ProtoPNet with semantic realignment
- LucidPPN [9]: Multiple ProtoPNets to identify different features

2. Introducing CaBRNet

Current issues

> *I cannot share my results: I made space on my hard drive and removed them.*
> One of the authors from the previous slide

Current issues

> *I cannot share my results: I made space on my hard drive and removed them.*
> One of the authors from the previous slide

- Problems
  - existing code spread across multiple repositories
  - different training data
  - different hyperparameters

- How to?
  - reproduce results
  - compare approaches
  - adapt existing codes

CaBRNet in a nutshell

A python framework using PyTorch 2.0:

- A **backend** to easily develop case-based reasoning networks
- A **tool** to instantiate, train and explain case-based reasoning networks
- A **benchmark** to evaluate case-based reasoning networks

With a focus on:

- Modularity
- Backward compatibility
- Reproducibility

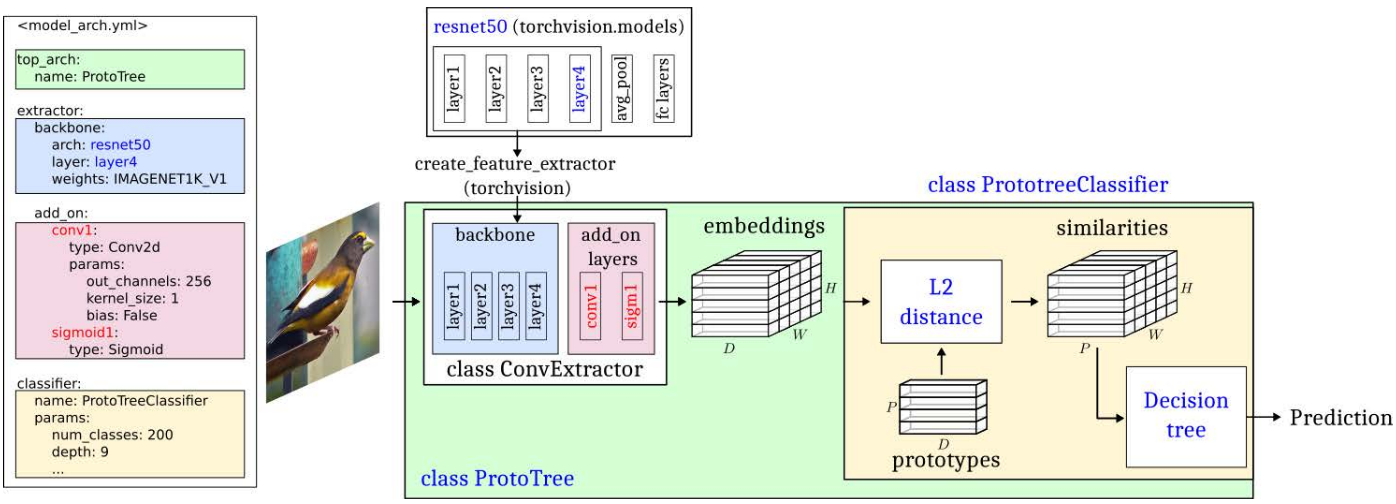
Powered by PyTorch

Overview of CaBRNet

- Two main aspects designed to be easily extendable
  1. The model architecture: description of the AI (ProtoPNet, Prototree, etc.)
  2. The app: what they do with their model (e.g., train, evaluate, explain, etc.)

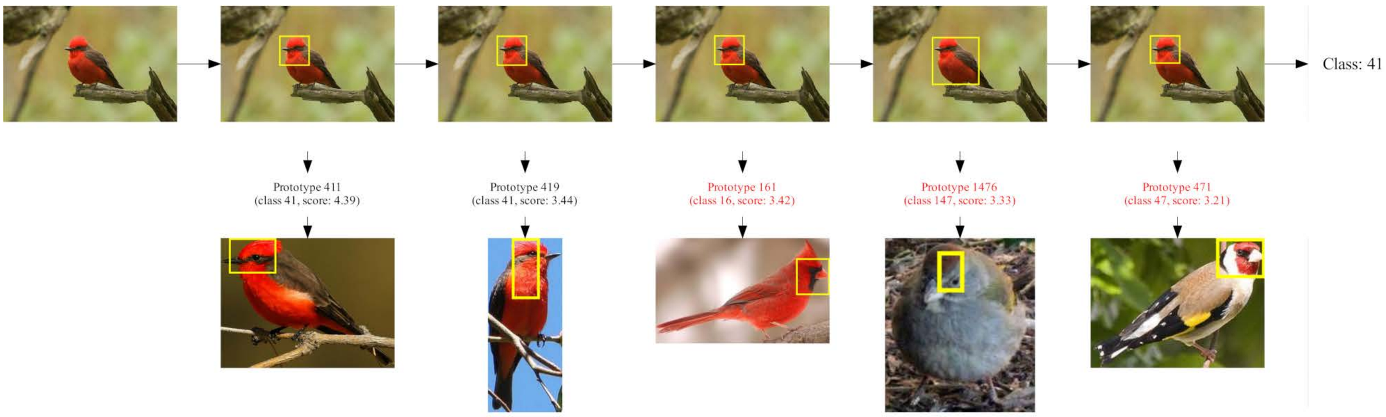
- CaBRNet provides
  - Several architectures
  - Several apps
  - Tools to build more of them easily

Example: architecture

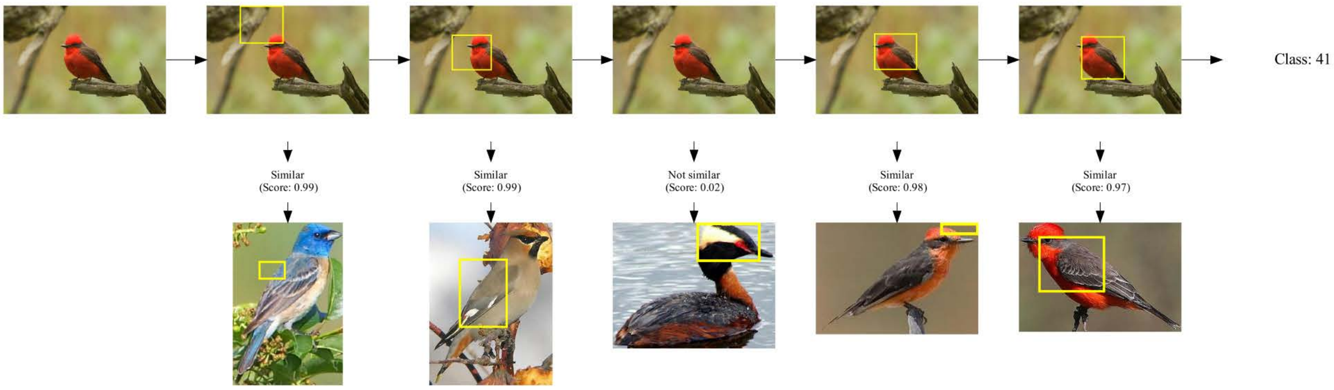


cea list

Generating local explanations (ProtoPNet)

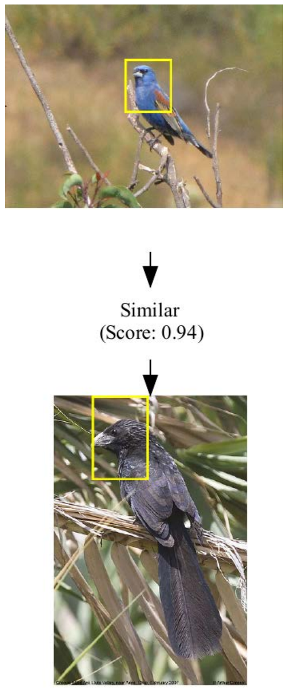


Generating local explanations (ProtoTree)



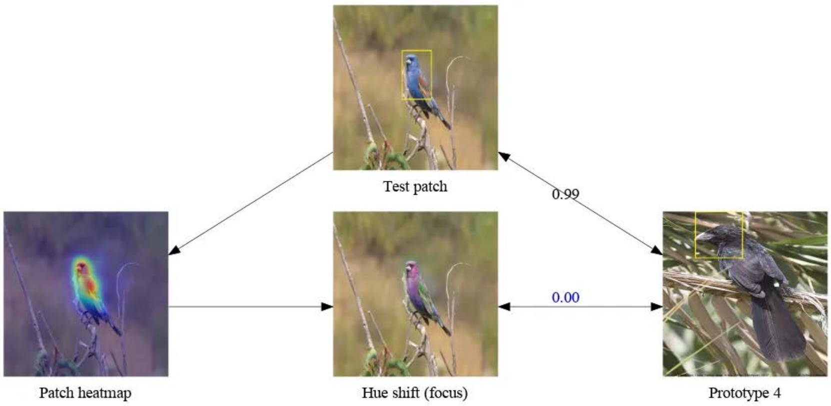
list

Going beyond accuracy

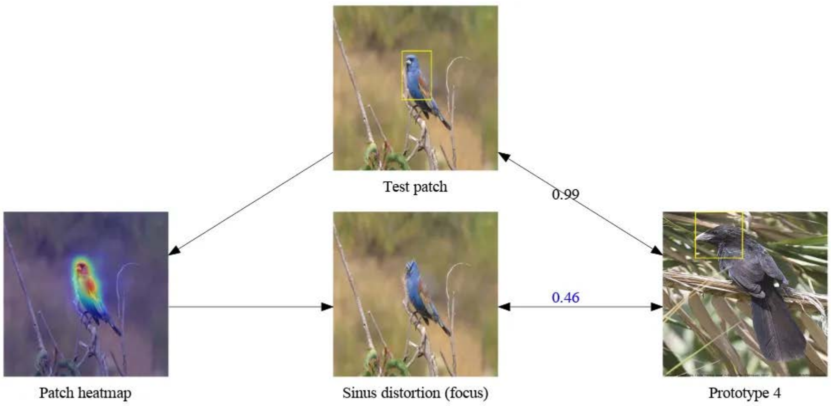


- What does it mean that these two parts are similar?
  - Not always obvious. Look here →
- Analyse what aspects of the parts are similar [10, 11]

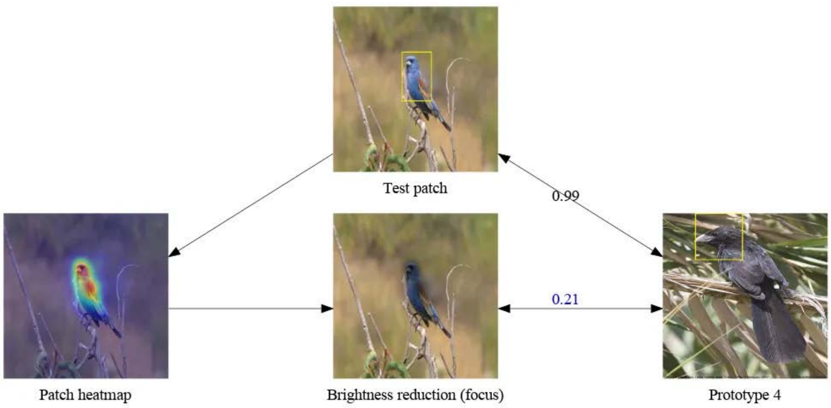
Applying a local perturbation



Applying a local perturbation (distortion)



Applying a local perturbation (brightness reduction)



CaBRNet: Case-Based Reasoning through Neural Network, A. Grastien

17/20

list

3. The road ahead
Case-**B**ased **R**easoning **Net**works

https://github.com/aiser-team/cabrnet

What now?

- CaBRNet published under license: LGPL2.1
  - Supports several architectures
  - Implements new metrics
  - Provides models on Zenodo
  - Used in teaching (Master SETI) and tutorials (ECAI tutorial, 2024 XAI conference)

- CaBRNet makes it easy to develop new architectures to fit specific needs
  - Example: collaboration with visitor on use case with *Policlinico Gemelli Rome*

Rapid intro to FM

Examples for this talk:

- **Property-based testing (Renault)**
- **Verification of functional properties (Airbus)**
- **Robustness evaluation (Technip)**
- **Out-of-Distribution detection (Thales)**
- **Bonus track**

109

What CAISAR is

Principle: Maximize coverage of AI models and properties

- Common expressive specification language
- Easy extensibility through clear interfaces
- Heuristic-aided V&V analysis
- Common aggregation of analysis outputs

Target: SVM, Neural Networks, XGBoost models, ensemble models,…

Application: depending on the used plug-ins. Currently includes

- SAVer for SVM
- Colibri for XGboost
- PyRAT, AB-Crown, Nnenum, Marabou for NN

Background: The federative platform strategy for V&V has been successful for critical SW (see, for example, Frama-C and Why3)

What CAISAR is
Characterizing AI Safety And Robustness

Aimed at all AI systems

While the current frenzy of AI trustworthiness is mostly focused neural networks, our industrial partners can also use other types of AI (e.g. SVM, XGBoost) in their products. This is why **CAISAR targets a wider range of AI systems.**

Standard oriented

By relying on AI standards (ONNX, NNnet) and formal methods standards (SMT, CP), CAISAR maximizes the potential for **inclusiveness**. Any tool that supports these standards is a potential addition to the CAISAR platform.

Modular and extensible

Written in the functional language OCaml, adding a verification, analysis or testing software to CAISAR's toolsuit is made easier through a unified interface, and an instantiation guided by data-types.

Interoperability

An internal representation for property language and AI representation helps reinforce the **synergy between the different tools** in CAISAR, where one analyser can rely on, and complement, the partial output of another.

Maintainable

Functional programming provides easy mathematical reading, lowering the entry barrier for understanding the inner workings of CAISAR. Strong typing also minimizes errors with informative messages.

Open-source

Our vision of CAISAR is collaborative in essence. To encourage cooperation and build a community of trust-minded entities, an easy access to the platform, supplied with documentations and tutorials, is our priority.

What CAISAR is
Characterizing AI Safety And Robustness

Property ϕ_1.

- Description: If the intruder is distant and is significantly slower than the ownship, the score of a COC advisory will always be below a certain fixed threshold.
- Tested on: all 45 networks.
- Input constraints: $\rho \geq 55947.691$, $v_{\text{own}} \geq 1145$, $v_{\text{int}} \leq 60$.
- Desired output property: the score for COC is at most 1500.

```
let function normalize_t (i: t) (mean: t) (range: t) : t =
  (i .- mean) ./ range
let function denormalize_t (i: t) (mean: t) (range: t) : t =
  (i .* range) .+ mean
let function normalize_input (i: input) : input =
  Vector.mapi i normalize_by_index
let function denormalize_output_t (o: t) : t =
  denormalize_t o
    (7.518884020100597531666153372498229146003723144453125:t)
      (373.9499200000000200816430151462554931640625:t)
let runP1 (i: input) : t
  requires { has_length i 5 }
  (* constraints the inputs to respect the specification *)
  requires { valid_input i }
  requires { intruder_distant_and_slow i }
  ensures { result .≤ (1500.0:t) }  =
    let j = normalize_input i in
    let o = (nn @@ j)[clear_of_conflict] in
    (denormalize_output_t o)
```

113

What CAISAR is
Characterizing AI Safety And Robustness

```
goal pruned:
  CSV.forall_ dataset (fun _ e →
    forall perturbed_e.
      has_length perturbed_e (length e) →
      FeatureVector.valid feature_bounds perturbed_e →
      let perturbation = perturbed_e - e in
      ClassRobustVector.bounded_by_epsilon perturbation eps →
      let out_1 = nn_1@@perturbed_e in
      let out_2 = nn_2@@perturbed_e in
      .- delta .≤ out_1[0] .- out_2[0] .≤ delta
  )
```

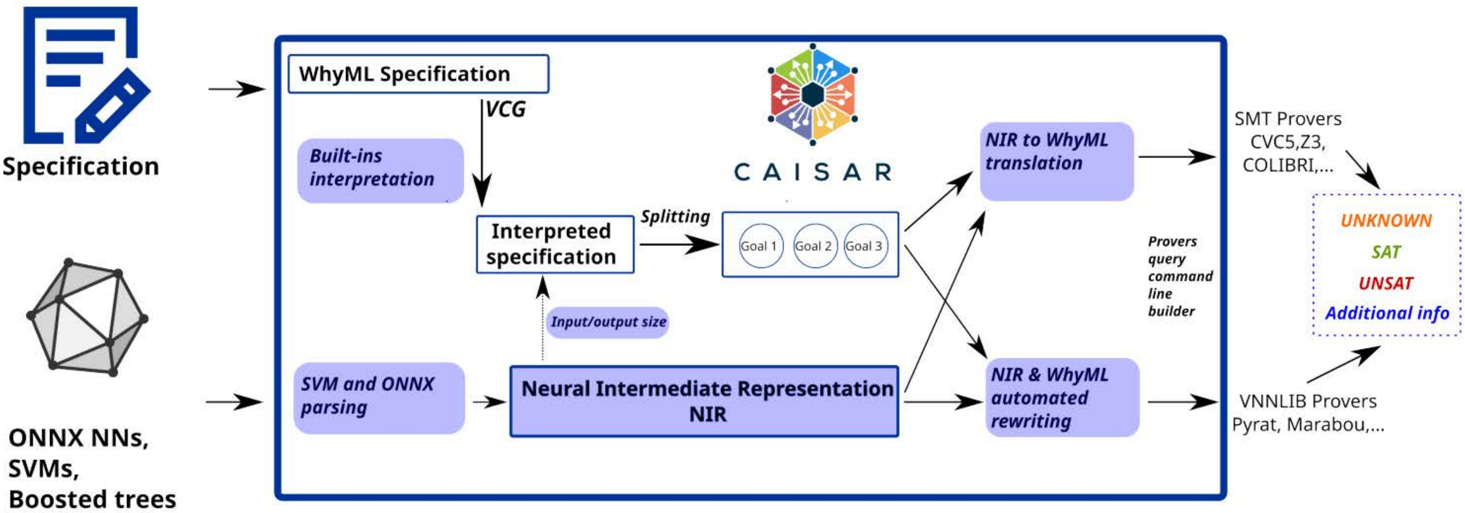
Fig. 13: A WhyML specification with several NNs at once

What CAISAR is
Characterizing AI Safety And Robustness

```
goal splitted:
  CSV.forall_ dataset (fun l e →
    forall perturbed_e.
      has_length perturbed_e (length e) →
      FeatureVector.valid feature_bounds perturbed_e →
      let perturbation = perturbed_e - e in
      ClassRobustVector.bounded_by_epsilon perturbation eps →
      let out1 = pre_nn@@perturbed_e in
      let out2 = post_nn@@out1 in
      forall j. Label.valid label_bounds j → j ≠ l →
      out2[l] .≥ out2[j]
)
```

Fig. 14: A WhyML specification for the composition of NNs

What CAISAR is (going to be)
Characterizing AI Safety And Robustness



caisar-platform.com 116

“ Cool, but where do I start ?

Potential user

117

Who are you? What do you want?

- You're a student, a researcher, or a professional who want to evaluate solutions…
  - You want automatic test generation with AIMOS or verification with PyRAT: free licence available
  - You want a more general way of testing and verification: CAISAR is open-source
  - You want to play around with XAI methods : CaBRNet is open-source
- You're a teacher and want to use tools in lab sessions for your students
  - Same as above but there are also course material available that we can help you adapt
- You're a professional who wants to use tools in a production setting
  - Open-source platforms and their documentations are available, support licenses are possible
  - License is possible for closed-source tools PyRAT and AIMOS
- When in doubt: contact us ! Zakaria.chihani@cea.fr

Our lab

Verification of safety and robustness formal specifications through Abstract Interpretation	Metamorphic testing applied to AI (Available for teaching)	Open-source, modular, extensible platform to Characterize AI Safety And Robustness	Open-source Symbolic AI tools, Safe-by-design Constraint solvers	Case-based reasoning, explainability, out-of-distribution detection
PyRAT	AIMOS	CAISAR	Colibri & co	PARTICUL
Verification	**Test**	**Platform**	**Symbolic**	**XAI & uncertainty**

caisar-platform.com

Symbolic AI: Colibri's

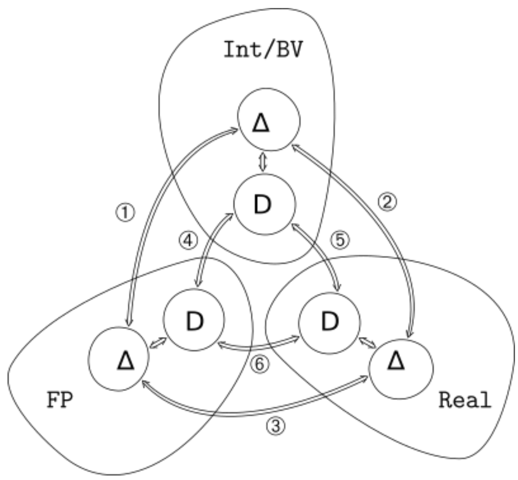
Principle: Safe-by-design Symbolic AI through a constraint solving library

- Separately prove, in Why3, the necessary bricks for constraint solving: Floating-point numbers, integers, bit-vectors, strings, etc.
- Allow for selection of these bricks to tailor the construction of a solver to the needs of the user
- Automatically extract a C implementation of the solver

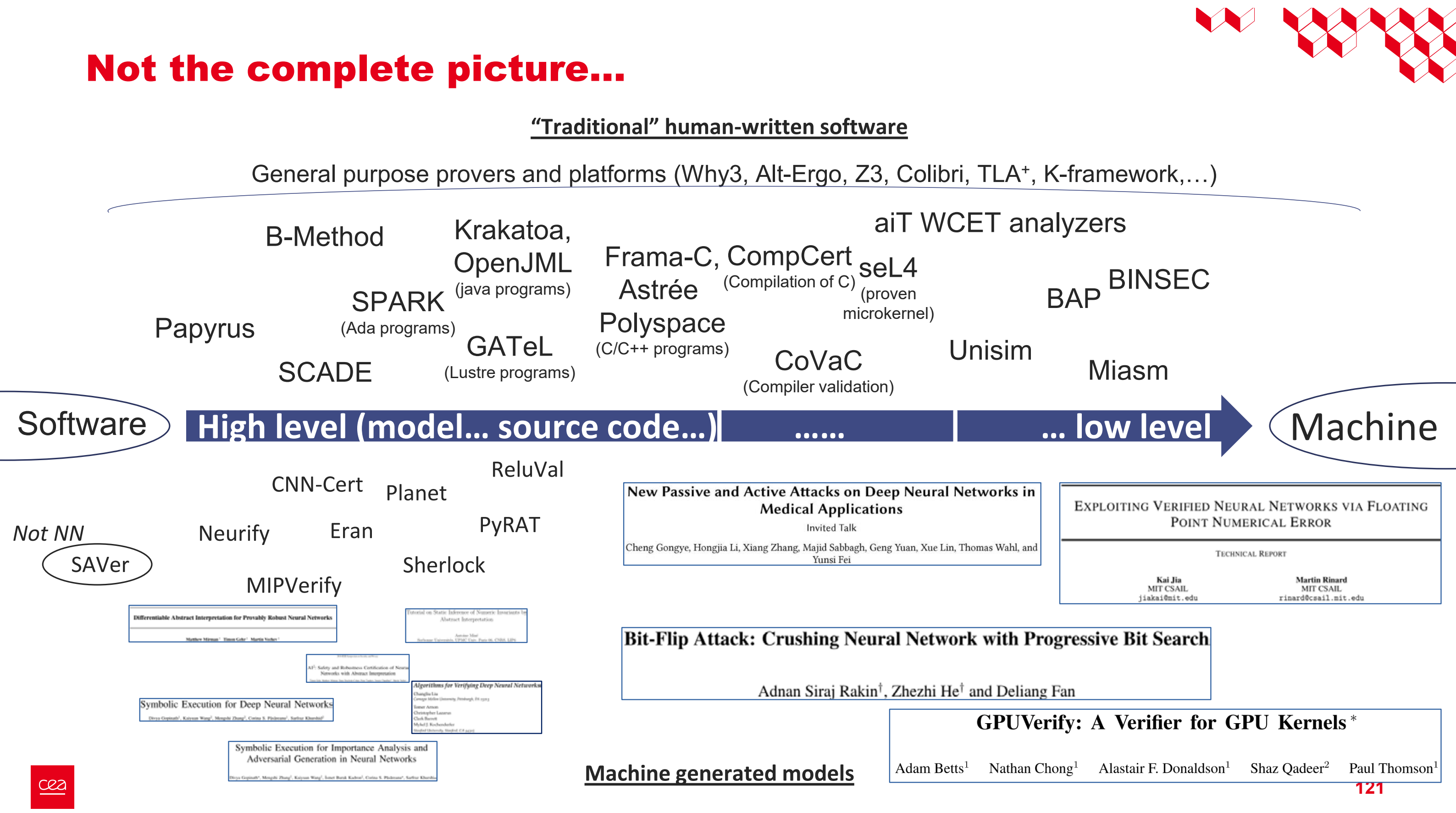
Target: XGBoost models, embedded software

Application: Energy sector (e.g., IRSN), space (e.g., NASA). Can also be used as a verification tool (winner of SMT-Competition since 2017), which makes it an essential brick of other tools such as Frama-C and GATeL.

Background: Constraint solving is used in several critical software domains



Colibri.frama-c.com


Not the complete picture…
"Traditional" human-written software
General purpose provers and platforms (Why3, Alt-Ergo, Z3, Colibri, TLA⁺, K-framework,…)
B-Method
Krakatoa, OpenJML (java programs)
aiT WCET analyzers
Frama-C, Astrée Polyspace (C/C++ programs)
CompCert (Compilation of C)
seL4 (proven microkernel)
BAP
BINSEC
SPARK (Ada programs)
Papyrus
GATeL (Lustre programs)
SCADE
CoVaC (Compiler validation)
Unisim
Miasm
Software
High level (model… source code…)
……
… low level
Machine
CNN-Cert
ReluVal
Planet
Not NN
Neurify
Eran
PyRAT
SAVer
Sherlock
MIPVerify
New Passive and Active Attacks on Deep Neural Networks in Medical Applications
Invited Talk
Cheng Gongye, Hongjia Li, Xiang Zhang, Majid Sabbagh, Geng Yuan, Xue Lin, Thomas Wahl, and Yunsi Fei
EXPLOITING VERIFIED NEURAL NETWORKS VIA FLOATING POINT NUMERICAL ERROR
TECHNICAL REPORT
Kai Jia MIT CSAIL jiakai@mit.edu
Martin Rinard MIT CSAIL rinard@csail.mit.edu
Differentiable Abstract Interpretation for Provably Robust Neural Networks
Symbolic Execution for Deep Neural Networks
Algorithms for Verifying Deep Neural Networks
Symbolic Execution for Importance Analysis and Adversarial Generation in Neural Networks
Bit-Flip Attack: Crushing Neural Network with Progressive Bit Search
Adnan Siraj Rakin†, Zhezhi He† and Deliang Fan
GPUVerify: A Verifier for GPU Kernels*
Adam Betts1 Nathan Chong1 Alastair F. Donaldson1 Shaz Qadeer2 Paul Thomson1
Machine generated models
121

Conclusion

« Meta » overfitting on public datasets ?

Academia hasn’t solved all trustworthiness problems, but industry can help us get there!

Trustworthiness challenges from industry are needed

VNN-Comp is calling for benchmarks (see aiverification.org, collocated with CAV)

Zakaria Chihani
zakaria.chihani@cea.fr

122

CPS&IoT'2025 Summer School
Budva, Montenegro, June 10-14, 2025

Introduction to Quality-Driven Design of Cyber-Physical Systems

Lech Jóźwiak
Professor Emeritus
Department of Electronic Systems
Faculty of Electrical Engineering
Eindhoven University of Technology
cpsiot2024@gmail.com

1

Outline

1. Challenges and demands of modern CPS
2. Quality-driven Model-based Design Approach
3. What is quality?
4. Quality-driven Design: difficulties and design models
5. Main concepts of the quality-driven model-based design
6. Quality-driven design space exploration
7. Example: Quality-driven model-based automated design of heterogeneous massively parallel MPSoCs for CPS
8. Conclusion

2

Challenges: unusual complexity and ultra-high demands

- The huge and rapidly developing markets of sophisticated CPS and IoT represent great opportunities
- These opportunities come with a price of:
  - unusual system complexity and heterogeneity, resulting from *convergence and combination of various applications and technologies* in one system or even on one chip, and
  - stringent and difficult to satisfy requirements of modern applications
- Smart cars, robots, drones and various wearable systems:
  - involve big instant data from multiple complex sensors (e.g. camera, radar, lidar, ultrasonic, sensor network tissues, etc.) and from other systems, used for mobile vision, imaging, virtual or augmented reality, etc.
  - are required to provide continuous autonomous service in a long time
  - are safety-critical
- In consequence, they demand a guaranteed (ultra-)high performance and/or (ultra-)low energy consumption, while requiring a high reliability, safety and security

3

Challenges: criticality of applications

- Cyber-physical systems influence our life to a higher and higher degree
- Therefore, the society expectations regarding them grow rapidly
- Due to CPS common usage in various kinds of technical, social and biological applications, and their growing influence, **we and the life on the Earth more and more depend and rely on these systems**:
  - their ***quality*** is becoming ***more and more critical***
  - many ***applications considered previously as non-critical are becoming critical***
- Due to the rapidly growing share of the highly demanding CPS applications, ***higher demands are becoming much more common***
- Due to the multiple reasons just discussed, and specifically, due to the rapidly growing system and silicon complexity and diversity, it will be ***more and more difficult to guarantee the systems' quality***
- This is a **new difficult situation** that cannot be adequately addressed without an **adequate design methodology** and **design automation**

4

Quality-driven Model-based Design Approach

- When considering a **system and design methodology adaptation** to the situation in the field of modern CPS, we have first to ask: *what general system approach and design approach seem to be adequate to solve the listed problems and overcome the challenges*?
- **Predicting the current situation,** more than 20 years ago I proposed such **system paradigm** and **design paradigm**, i.e. the paradigms of:
  - **life-inspired systems** and **quality-driven design**, and
  - the **methodology of quality-driven model-based system design** based on them
- From that time my research team and our industrial and academic collaborators were researching the **application of this methodology** to the design and design automation of embedded processors, MPSoCs and CPS
- This **research confirmed the adequacy of the quality-driven model-based design methodology**

5

Quality-driven Model-based Design Approach

- ***What is the quality-driven design?***
- **System design is a *definition of the required quality***, i. e. a satisfactory answer to the following two questions:
  - **What new** (or modified) **quality is required**?
    and
  - **How can it be achieved**?
- Intuitively we feel that **quality** is here used in the sense of ***the totality of the (important) features the system has***
- So, **system design should define**:
  - **What is the required totality of the (important) system features?**
    and
  - **How to realize a system that has these all features**?
- In other words:
  - What process must be realized in a certain system and what structural and parametric features must have the system?
  - How can we build a system that will be able to realize this process and will have the required structural and parametric features?

6

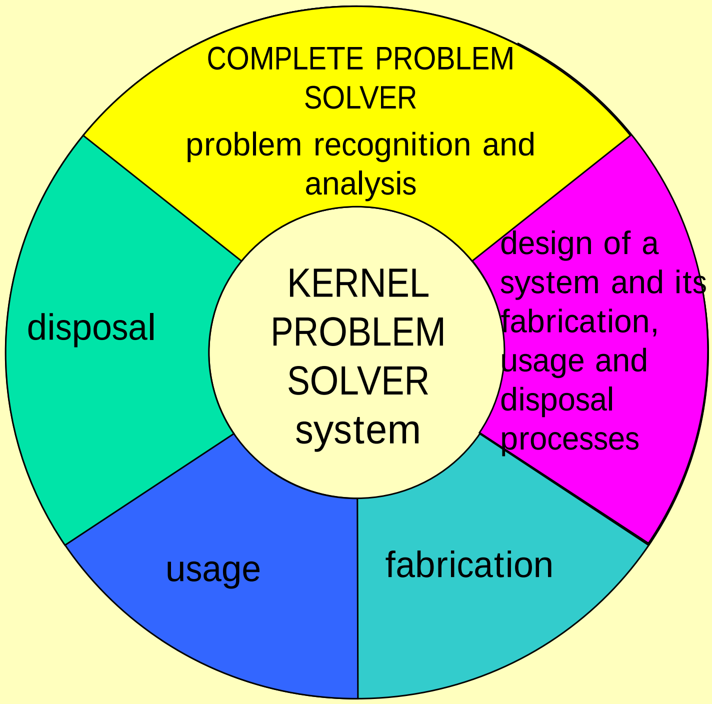
What is quality?

- When I started my work in quality-driven design, I analysed very many definitions of quality and concluded that for many reasons no one of the existing definitions could be used for quality-driven design
- The most used and cited definitions of quality:
  - fitness for use (*Juran*)
  - conformance to requirements (*Crosby*)
  - quality is meeting the customers' expectations at a price they can afford (*Deming*)
  - the loss of quality is the loss a product causes to society after being shipped, other than any losses caused by its intrinsic functions (*Taguchi*)
  - the totality of features and characteristics of a product or service that bear on its ability to satisfy given needs (*American Society for Quality Control*)
  - the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs (*ISO8402: Quality Vocabulary Part 1)*

7

Problems with the existing definitions of quality

they focus exclusively on a product being designed, while the original problem is solved by designing, fabrication, usage and disposing of the system



Quality cannot be limited to the system itself, but it must account for the complete problem solution, related to complete system life-cycle

8

Problems with the existing definitions of quality

- None of these definitions was precise enough to enable the systematic consideration, measurement and comparison of quality
- Their assumption of perfectly known and inviolable customer's requirements was not acceptable, because the customer may specify the requirements poorly and such requirements may result in system which will create danger, damage environment or squander scarce resources
- Engineered systems solve certain real-life problems, serve certain purposes – they are purposive systems
- Quality of a purposive system can only be defined in relation to its purpose

9

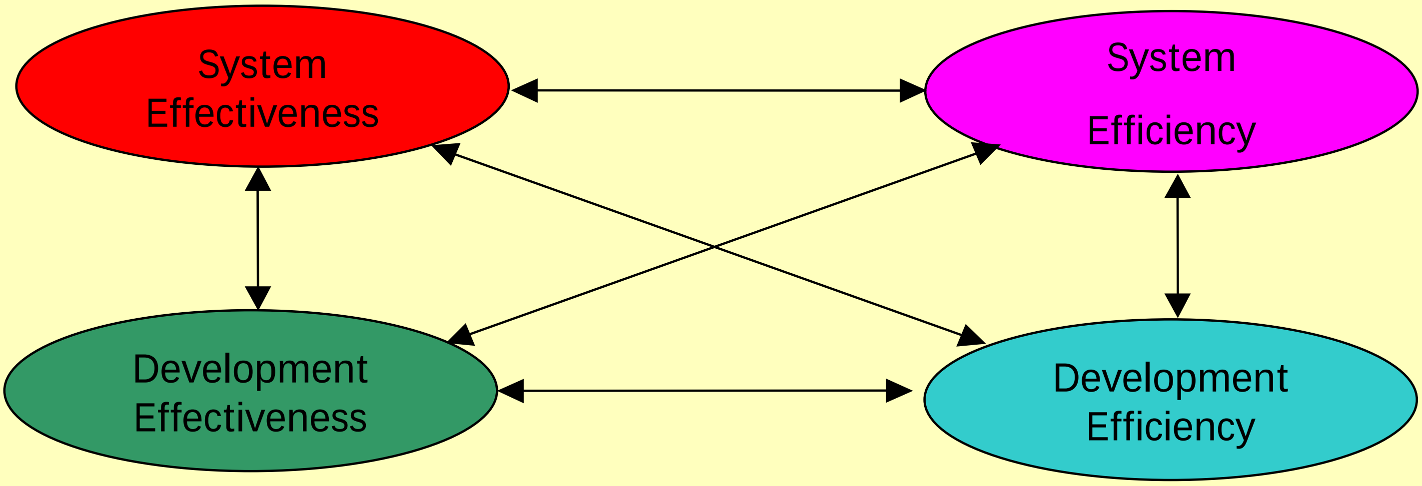
New quality definition proposed by me 20 years ago

***Quality** of a purposive systemic solution is*
*its **total effectiveness and efficiency***
in solving of the real-life problem that defines the solution's purpose

- **Effectiveness** = the degree to which a solution attains its goals
- **Efficiency** = the degree to which a solution uses resources in order to realize its aims
- ***Effectiveness and efficiency of a systemic solution together decide its grade of excellence*** - **their aggregation expresses quality**
- Effectiveness and efficiency can be expressed in terms of measurable parameters, and in this way, quality can be modeled and measured
- In particular, the quality can be modeled in the form of *multi-objective decision models* involving measurable design parameters
- *The multi-objective decision models* and *design parameter estimators* enable application of the *multi-objective decision methods* for construction, improvement and selection of the most promising solutions

10

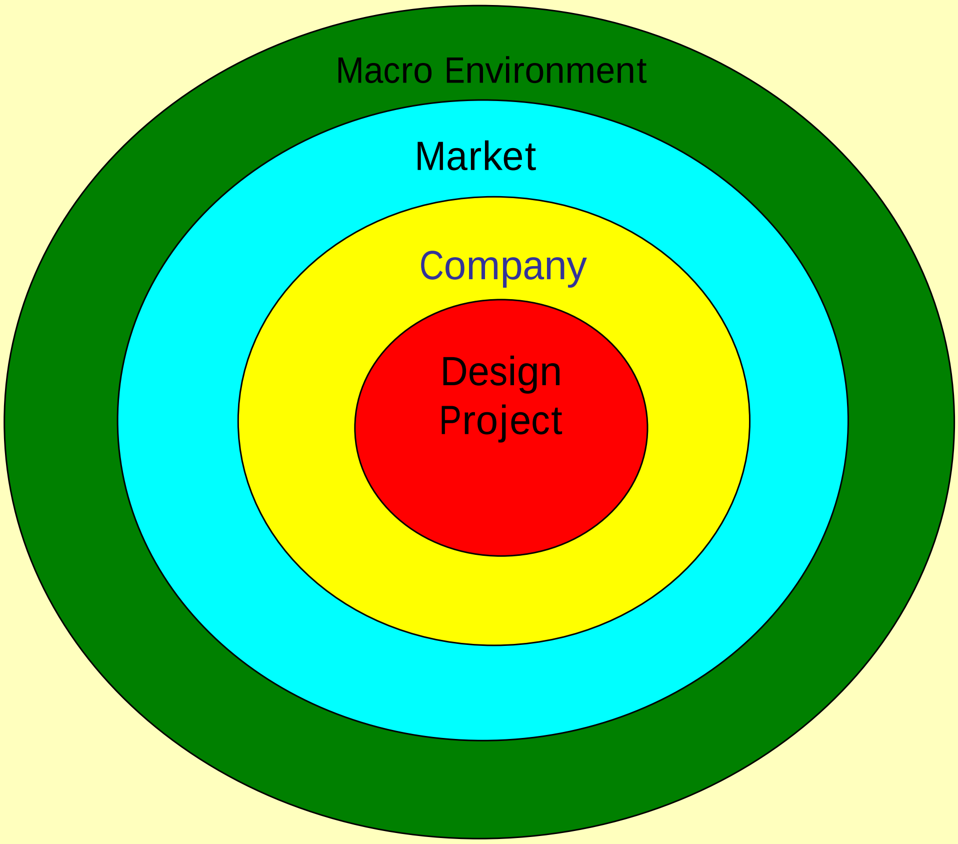
Quality-driven Design - Difficulties



Interactions and trade-offs between various parts and aspects of the total systemic solution

11

Quality-driven Design - Difficulties



Interactions of a design project with its context

12

Quality-driven Design - Difficulties

- Design does not concern the reality as it is, but as it will possibly be realized
- Quality recognition and formulation, i.e. recognition of the problem, as well as of the nature of its solution are *subjective* to a high degree
- The contemporary system design problems are *complex*, *multi-aspectual*, *dynamic*, and *ill-structured*:
  - there is no definitive formulation of the problem,
  - any problem formulation may be inconsistent,
  - formulations of the problem are solution dependent,
  - proposing and considering solutions is a means for understanding the problem, and
  - there is no definitive solution to the problem

13

Quality-driven Design - Design models

Due to all the difficulties

⇓

quality cannot be well defined,
but it can and should be modelled

- *Well-structured models of the required/delivered quality* can serve to:
  - conceptualize, denote, analyse and communicate the customer's and designer's ideas
  - show that the requirements and designs are meaningful and correct
  - guide the design process
  - enable the explicit and well-organized design decision making
  - enable design automation
  - etc.

14

Quality-driven Design: Design problem-solving using models

- Since the system design problems are:
  - complex;
  - multi-aspect;
  - ill-defined,

  to solve them, *all human concepts for dealing with complexity, diversity and ill-structure have to be applied*:
  - abstraction;
  - separation of concerns;
  - decomposition and composition;
  - generalization and specialization;
  - modelling;
  - simulation;
  - prototyping;
  - .....
- *A design problem has to be converted into a system of simpler sub-problems*
- The solution to the original problem can then be achieved by solving the sub-problems and composing the sub-problem solutions into an aggregate solution

15

Quality-driven Design: Design problem-solving using models

- The problem decomposition and design modelling are to some degree subjective
- The design decision processes are also to some degree subjective, as they are influenced by the designers' value systems, feelings, believes, intuition etc.
- The design problem solving activity is performed under uncertainty, inaccuracy, imprecision and risk conditions, and in a dynamic environment

⇓

- *System design has to be an evolutionary process* in which analysis and modelling of problems; proposing their solutions; analysis, testing and validation of the proposals; learning and adapting are very important

16

Main concepts of the quality-driven design

- Designing *top-quality systems is the aim* of a design process
- *Quality is modelled and measured* to enable invention and selection of the best alternatives and quality improvement
- *Quality models are considered to be heuristics for setting and controlling the course of design*
- *System design is an evolutionary quality engineering process* that basically consists of:
  - constructing the tentative quality models,
  - using them for constructing, improving and selecting of the tentative solutions,
  - analysing and estimating them directly and through analysis of the resulting solutions,
  - improving the models, and using them again to get improved solutions, etc.

17

Quality-driven Design: Limiting the design subjectivity

- **One of the main aims** of using the well-defined models in design is:

  Limiting the scope of subjective design decision making and ***enlarging the scope of reasoning-based decision making with clear and well-defined rational procedures*** which can be ***computerized***

- Too much subjectivity in design may result in solutions that either do not solve the actual real-life problem or do not do it in a satisfactory manner

- **Limiting the design subjectivity** in an appropriate manner, when enabling the creativity exploitation at the same time, ***is necessary to arrive at the high-quality designs***

18

Quality-driven Design: Limiting the design subjectivity

- The **main means for limiting the design subjectivity** is the ***design space exploration (DSE) with usage of the well-structured quality models***
- **Exploration** of the abstract models of the required quality and more concrete solutions obtained with these models:
  - ***gives much and more objective information*** on the design problem, its possible and preferred solutions, and various models used in this process
  - ***enhances exploitation of the designer's imagination, creativity, knowledge and experience***
- Other important means for limiting the design subjectivity include:
  - appropriately organised team-work
  - benchmarking and comparison with both own previous designs and designs of competition
  - design analysis and validation
  - design reuse
  - government and branch regulations and standards

19

Quality-driven Design: Government regulations and standards

- *Adequate government and industry branch regulations and standards are of primary importance for bringing into effect the green systems and green economy*
- Regulations and standards specify what is allowed or standard, and what is not
- They constitute general constraints for the industry and system designers that must be satisfied by their designs, products and services
- Of course, particular systemic solutions satisfying these general constraints can still be very different, better or worse for the environment, but *all systemic solutions must satisfy the minimum required by the regulations and standards*
- Remember that the decisions made by companies and governments that caused the environmental destruction were mainly driven by short-term profit, without accounting for long-term consequences
- It would be naïve to expect that all companies and all individuals will suddenly become environment-friendly without adequate regulations pressing them to do so

20

Quality-driven Design - Design requirements

- The general model of the required **system's** quality is represented by the *system (design) requirements*
- Requirements should be confronted with the actual up-to-date needs many times during the design process, and replaced or modified, if necessary
- Design requirements model the design problem at a hand through *imposition of constraints and objectives in relation to the acceptable or preferred problem solutions*
- It is possible to distinguish three sorts of requirements:
  - *functional*,
  - *structural*, and
  - *parametric*
- All the three sorts of requirements impose *limits on the structure of a required solution*, but they do it in different ways

21

Quality-driven Design - Design requirements

- The *structural requirements* define the acceptable or preferred solution structures directly, by limiting them to a certain class or imposing a preference relation on them
- The *parametric requirements* define the structures indirectly, by requiring that the structure has such physical, economic or other properties (described by values of some parameters) as fulfil given constraints and satisfy stated objectives
- The *functional requirements* also define the structures indirectly, by requiring the structure to expose a certain externally observable behaviour that realizes the required behaviour
- *System design*:
  - starts with an *abstract,* and possibly *incomplete, imprecise,* and *contradictory*, *initial quality model* (initial requirements)
  - transforms the initial model into a *concrete, precise, complete, coherent and directly implementable final quality model*

22

Quality-driven design space exploration (DSE)

- Usually, the initial abstract quality model mostly involves *behavioural* and *parametric characteristics* and to a lesser extend the structure definition
- The final model defines the ***system's structure explicitly***
- The design process *decomposes the total design problem* into *design issues* (sub-problems)
- For each issue, many various alternative solutions are typically possible
- During the design process those issues are solved (decided) by the designers and/or synthesis tools
- In result of design decisions, information is gradually added to the so created (partial) solutions to result in the explicitly defined complete system's structure that supports the system's required behaviour and satisfies the parametric requirements
- For each issue, issue quality model can be constructed that is composed of some selected and abstracted functional, structural and parametric requirements extracted in an appropriate manner from the total quality model of the considered system

23

Quality-driven Design - Decision models

- To support the design decision-making in the scope of an issue, the **issue's** decision model can be constructed
- A decision model is a *partial* (reduced to only certain concerns) and *abstract* (reduced to the necessary and/or possible precision level) model of the required quality, *expressed in the decision-theoretical terms*
- *Decision models* and *design parameter estimators* enable application of the *multi-objective decision methods* for construction, improvement and selection of the most promising solutions
- The decision model of a given issue *must account for all system characteristics substantially relevant* to the issue
- It must specify preferences of values for all the characteristics, expressed by hard constraints, objectives, and trade-off information
- For each single characteristic, the preferences of its values can be characterized by specifying a utility (effectiveness or efficiency) function $u_i(x_i)$ for the characteristic x_i

24

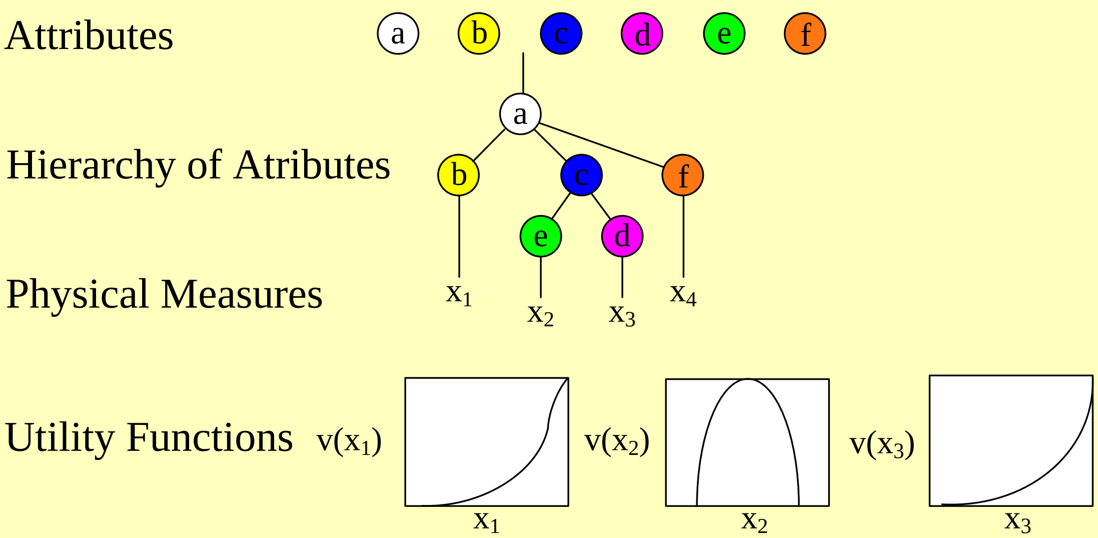
Quality-driven Design - Decision models

- The decision model of a given issue must account for all system characteristics substantially relevant to the issue
- It must specify preferences of values for all the characteristics, expressed by hard constraints, objectives, and trade-off information
- For each single characteristic x_i, the *preferences of x_i values* (i. e. the level of satisfaction from a particular value of x_i) *can be characterized by specifying a utility (effectiveness or efficiency) function $u_i(x_i)$*
- Due to the *multi-aspect nature of systems* and possible *trade-offs*, the *relative importance of different characteristics* or the *reference (aspiration) points in the utility space* should be specified
- With such models the total system quality Q can be modelled as a function of utility levels of all the important system characteristics influencing the systems effectiveness or efficiency
- Such design decision models make it possible to apply the multi-objective decision methods for invention and selection of solutions that are ''totally **optimal"**

25

Example: Modeling quality Q as a (vector) function of utility levels of the system characteristics

$$Q(y)=Q(x_1(y), x_2(y),\ldots, x_n(y))=F(v_1(x_1), v_2(x_2),\ldots, v_n(x_n))$$

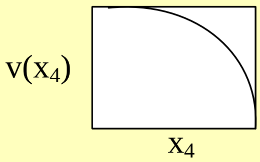


Tradeoffs:

- Relative Importance Among Attributes

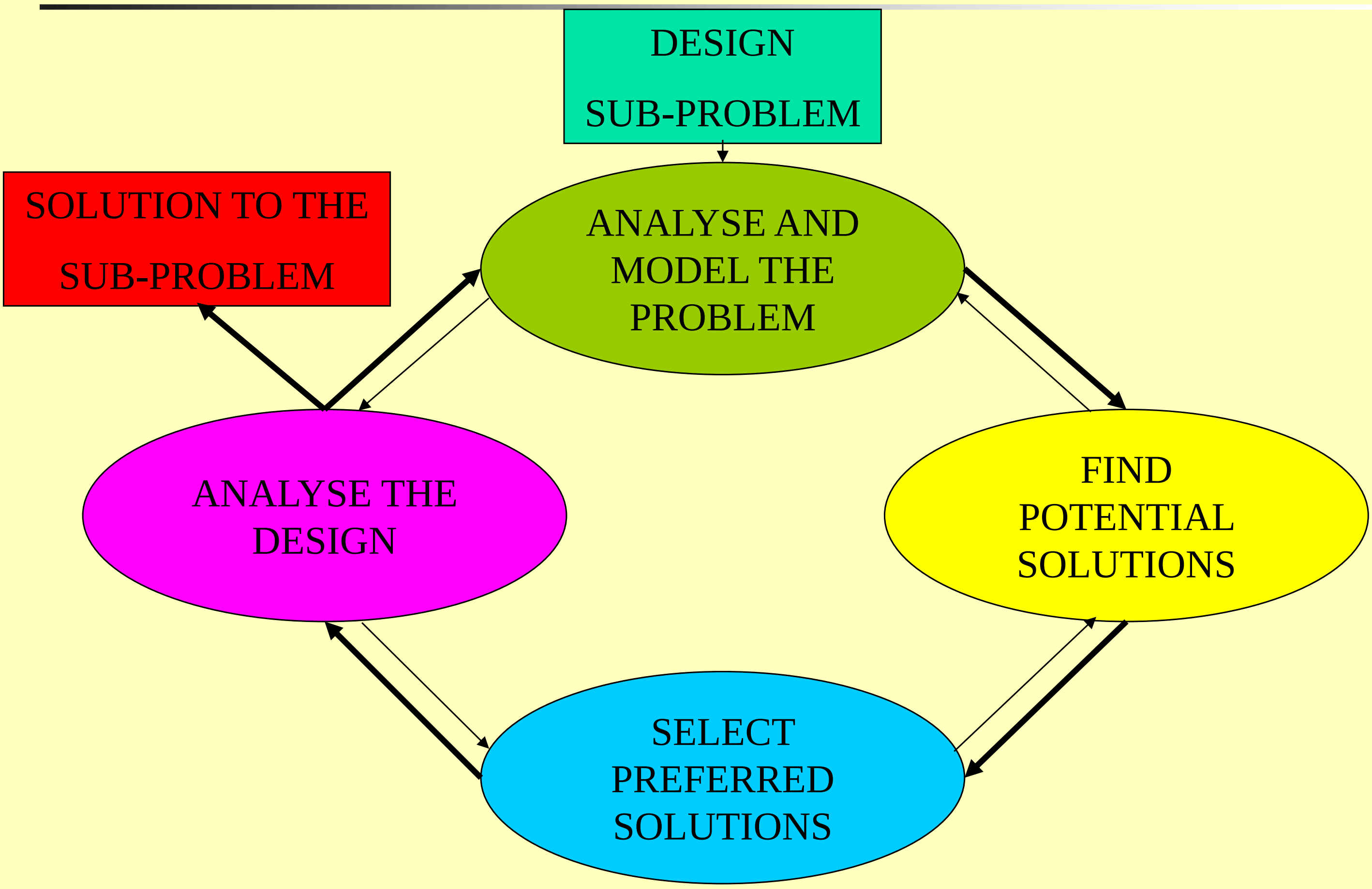
or

- Reference Points in Utility Space (or Parameter Space)



26

Generic model of the quality-driven design space exploration



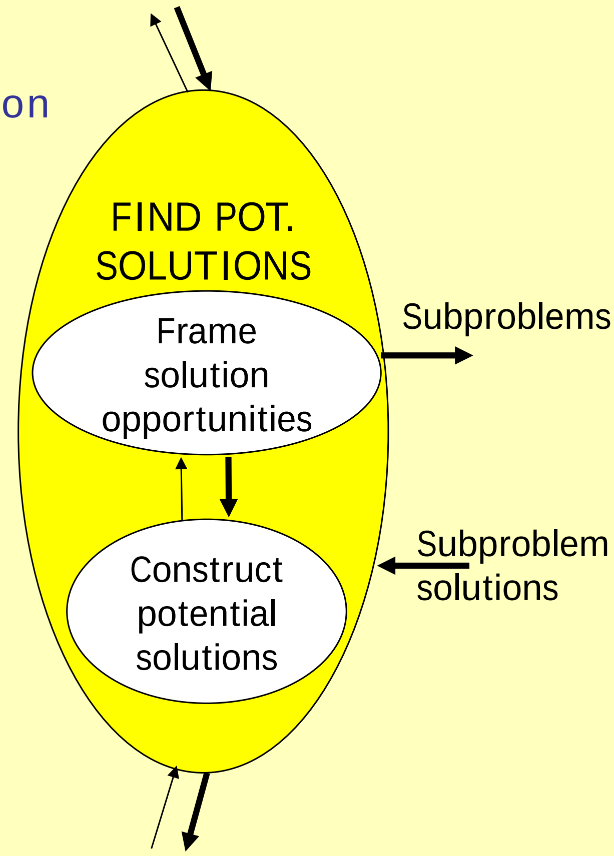
27

Generic model of the quality-driven design space exploration

- The quality-driven design space exploration basically consists of the alternating phases of:
  - *exploration of the space of abstract models of the required quality*

  and

  - *exploration of the space of the more* ***concrete issue's solutions*** obtained with the chosen quality models



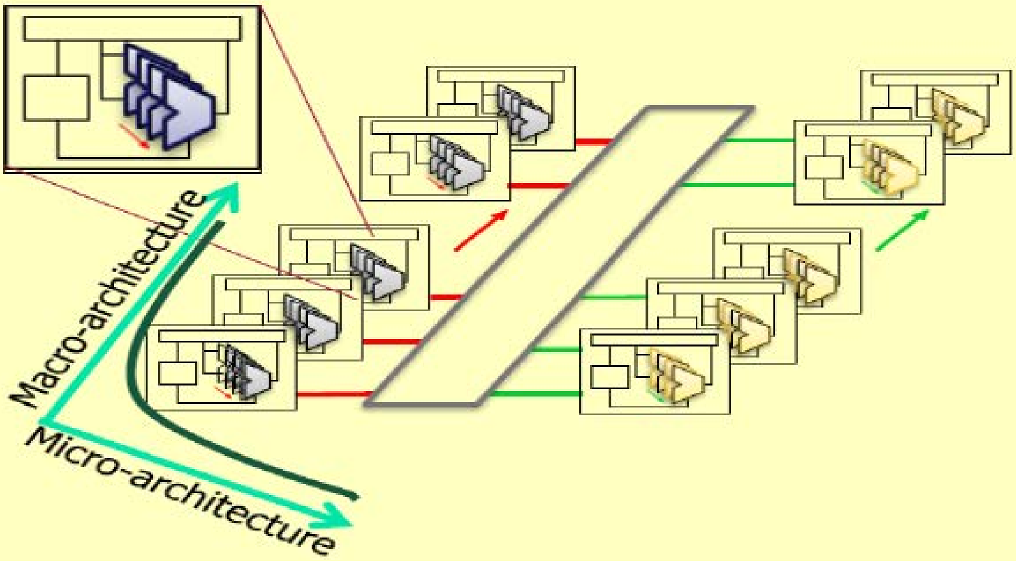
28

Quality-driven design space exploration

- In result of the design space exploration, the considered system is defined as an appropriate *decomposition into a network of sub-systems*
- Each sub-system solves a certain sub-problem
- All *sub-systems cooperating together solve the system design problem* by exposing the external *aggregate behaviour and characteristics* which *match the required behaviour and characteristics*
- The design process breaks down *a complex system* defined in *abstract and non-precise terms* into *a structure of cooperating sub-systems* defined in *more concrete and precise terms*, which are in turn further broken down to the *simpler sub-systems that can be directly implemented with the elements and sub-**systems at the designer's disposal***

29

Example: Quality-driven model-based automated design of heterogeneous massively parallel MPSoCs for CPS: *palallel multi-processor technology*



- Complex massively-parallel multi-processors are necessary, with micro-architectures of elementary processors spanning the full spectrum from serial, through partially-parallel, to fully parallel.
- A very high number of possible macro-architecture/micro-architecture combinations and related computation mappings
- A huge design space of various possible multi-processor architectures with different characteristics.

30

Example: Quality-driven model-based automated design of heterogeneous massively parallel MPSoCs for CPS: *issues and callanges*

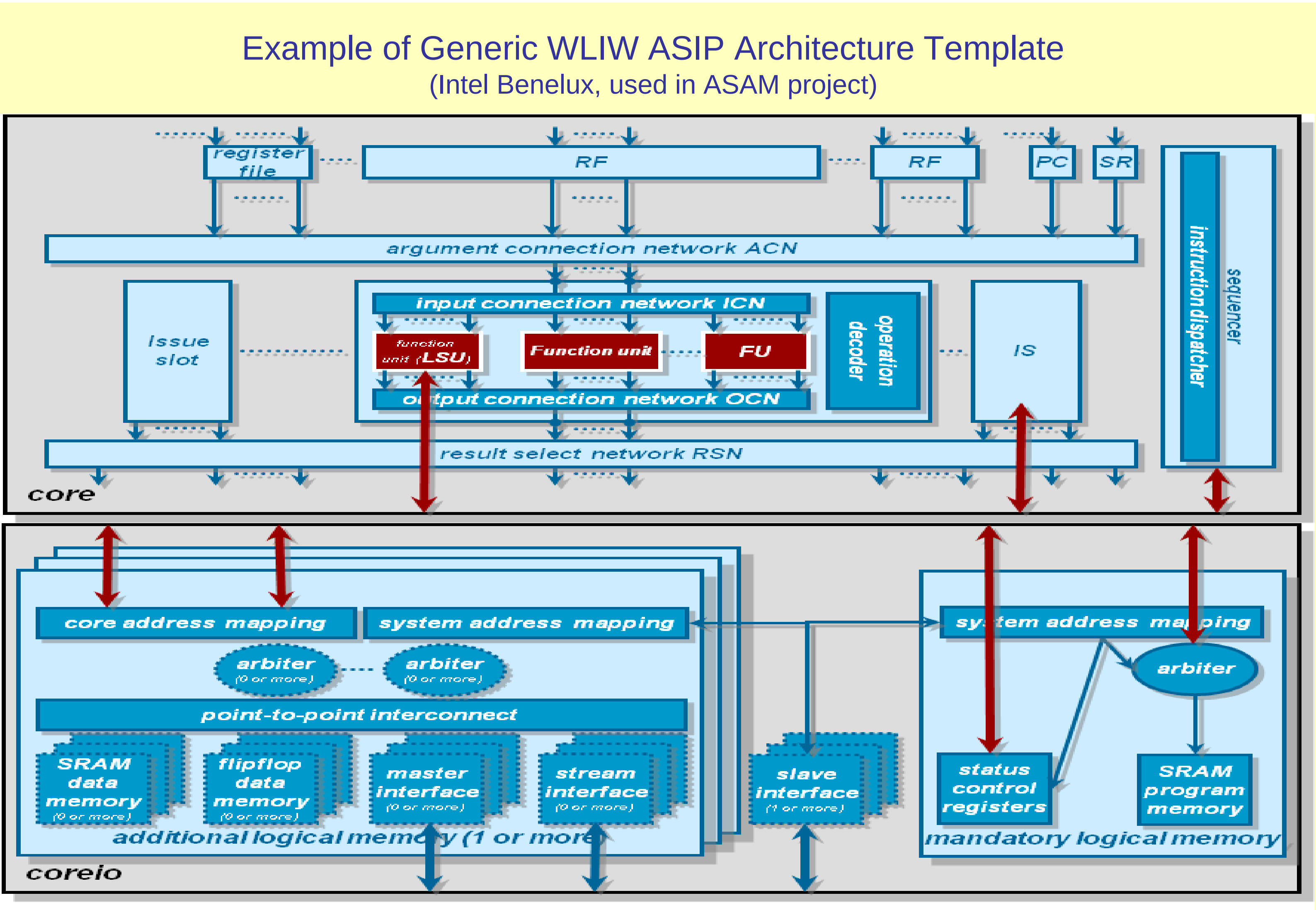
- The application's parallelism has to be exploited at two architecture levels: system macro-architecture and processor micro-architecture level.
- Similar performances can be achieved with:
  - less processors, each being more parallel and better targeted to a particular part of a complex application,
  - more processors, each being less parallel or less application-specific.
- Each of the alternatives can have different physical and economic characteristics, such as power consumption or circuit area.
- This results in the necessity to explore and decide the various possible tradeoffs between the micro-architecture and macro-architecture design.
- Each micro-/macro- architecture combination requires different compatible memory and communication architectures.
- Exploitation of data parallelism in a computing unit micro-architecture usually demands getting the data in parallel for processing.
- This requires simultaneous access to parallel memories and simultaneous data transmission.

31

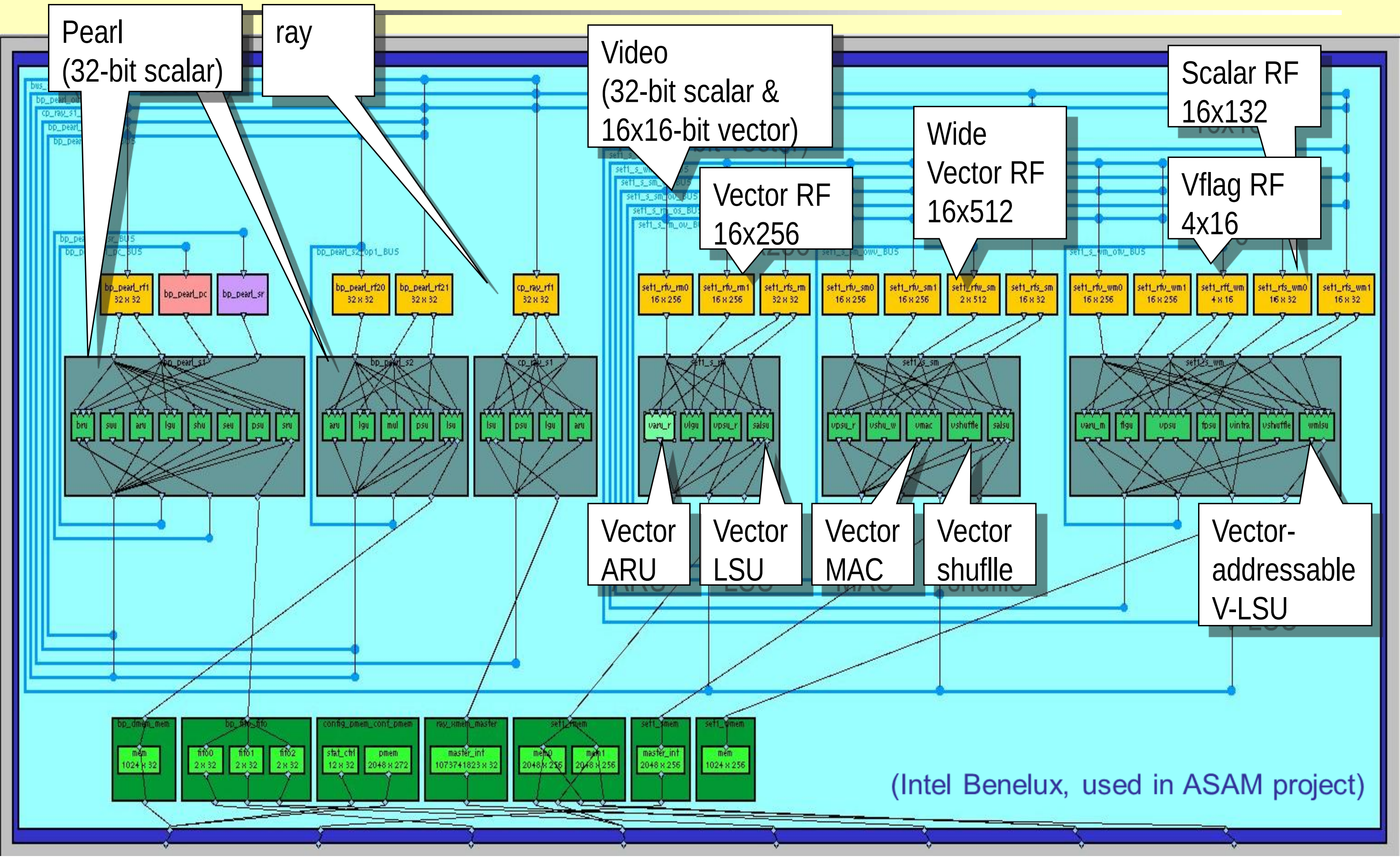
Example: *Quality-driven model-based automated design of heterogeneous massively parallel MPSoCs for CPS:* multi-ASIP case (e. g. ASAM project)

- To develop the complex multi-ASIP MPSoCs, a sophisticated design space exploration is necessary in which only the most promising ASIP and MPSoC architectures will be efficiently constructed, and the best of these architectures will be selected for further analysis, refinement and actual implementation
- The ASAM multi-ASIP MPSoC design-space exploration implements the *quality-driven model-based system design methodology*
- According to this methodology quality has to be modeled, measured, and compared
- The quality of the multi-ASIP MPSoC required is modeled in the form of the:
  - demanded system behavior (application C-code)
  - structural constraints: generic ASIP and MPSoC architecture templates and their pre-characterized generic parts included in the IP library, and
  - parametric constraints and objectives to be satisfied by the MPSoC design
- Based on the analysis of the so modeled required quality, the generic architecture templates are adequately instantiated and used in design space exploration that constructs one or several most promising MPSoC designs supporting the required behavior and satisfying the demanded constraints and objectives

32


Example of Generic WLIW ASIP Architecture Template
(Intel Benelux, used in ASAM project)
register file
RF
RF
PC
SR
argument connection network ACN
issue slot
input connection network ICN
function unit (LSU)
Function unit
FU
operation decoder
output connection network OCN
IS
instruction dispatcher
sequencer
result select network RSN
core
core address mapping
system address mapping
arbiter
(0 or more)
arbiter
(0 or more)
point-to-point interconnect
SRAM data memory
(0 or more)
flipflop data memory
(0 or more)
master interface
(0 or more)
stream interface
(0 or more)
slave interface
(1 or more)
additional logical memory (1 or more)
system address mapping
arbiter
status control registers
SRAM program memory
mandatory logical memory
coreio

Example instance of the generic ASIP template: video processor

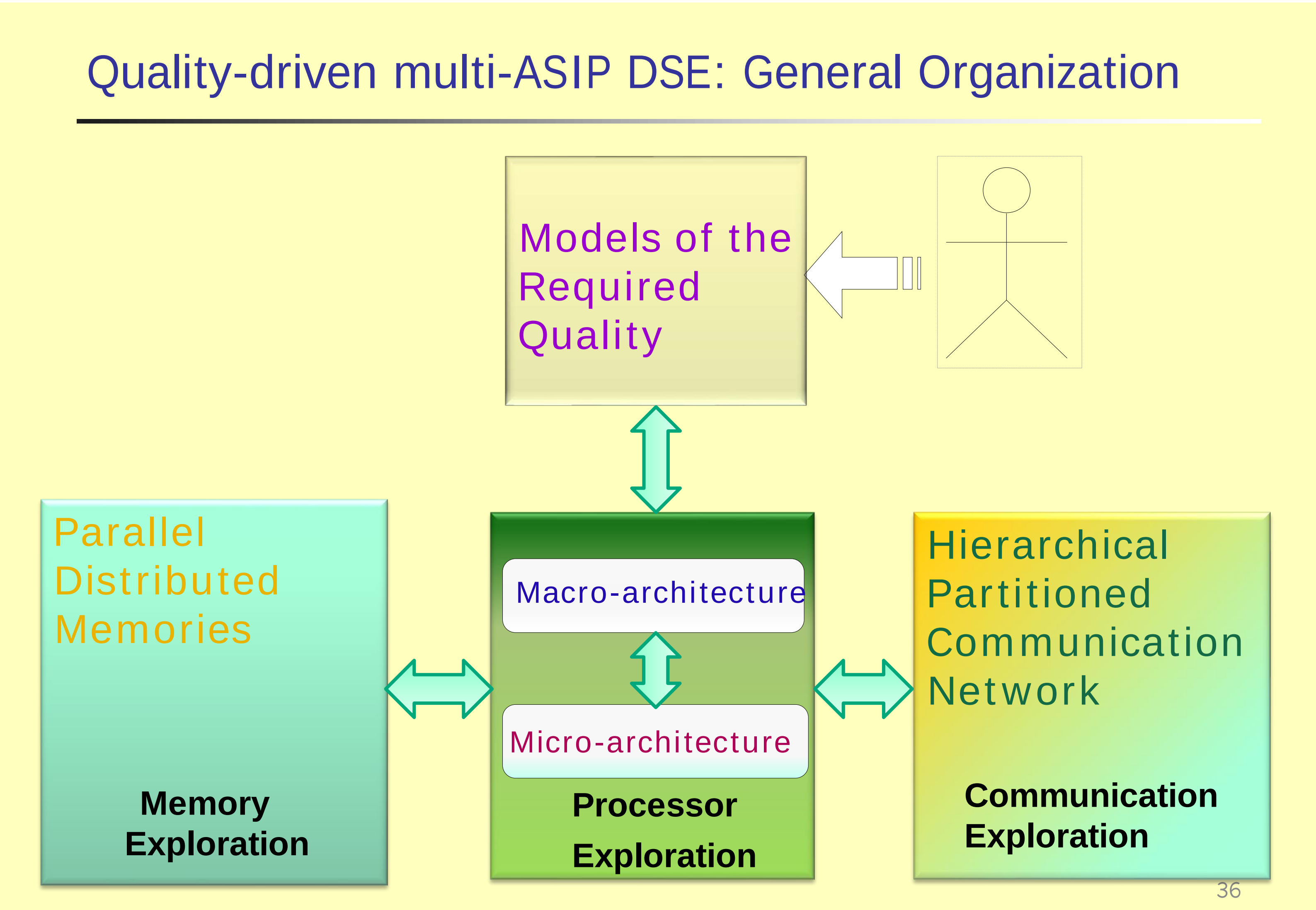


Quality-driven model-based automated design of multi-ASIP MPSoCs: *Quality-driven DSE*

- Based on the analysis of the so modeled required quality, the generic architecture template is adequately instantiated and used in design space exploration (DSE) that aims at:
  - analysis of various architectural choices regarding:
    - processor micro-architectures and multi-processor macro-architecture
    - parallel memories architectures
    - parallel communication architectures
    - macro-/micro-architecture tradeoffs
    - processor, memory and communication tradeoffs,

  and based on this analysis,
  - construction of one or several most promising (sub-)system architectures supporting the required behavior and satisfying the demanded constraints and objectives.

35



Quality-driven multi-ASIP DSE: General Organization
Models of the Required Quality
Parallel Distributed Memories
Memory Exploration
Macro-architecture
Micro-architecture
Processor Exploration
Hierarchical Partitioned Communication Network
Communication Exploration
36

Conclusion

- The extreme complexity, demands and requirements of many modern CPS cannot be addressed without an adequate design methodology and design automation
- More than 20 years ago I proposed the **methodology of quality-driven model-based system design**
- In successive years, my research team and our industrial and academic collaborators were researching the application of this methodology to the design and design automation of embedded processors, MPSoCs and CPS
- This research confirmed the adequacy of the quality-driven model-based design methodology
- For "Outstanding Achievements and Contributions to Quality of Electronic Design" I was awarded the Honorary Fellow Award by the International Society for Quality Electronic Design (San Jose, CA, USA, 2008)

37

Further reading for this tutorial

- L. Jóźwiak: Quality-driven Design in the System-on-a-Chip Era: Why and how?, Journal of Systems Architecture, vol. 47, no. 3-4, Apr. 2001, pp. 201-224
- L. Jóźwiak: Life-inspired Systems and Their Quality-driven Design, Lecture Notes in Computer Science, Vol. 3894, 2006, Springer, pp. 1-16
- L. Jóźwiak and S.-A. Ong: Quality-driven Model-based Architecture Synthesis for Real-time Embedded SoCs, Journal of Systems Architecture, Elsevier Science, Amsterdam, The Netherlands, ISSN 1383-7621, Vol. 54, No 3-4, March-April 2008, pp. 349-368
- Jóźwiak, L.; Lindwer, M.; Corvino, R.; Meloni, P.; Micconi, L.; Madsen, J.; Diken, E.; Gangadharan, D.; Jordans, R.; Pomata, S.; Pop, P.; Tuveri, G.; Raffo, L. and Notarangelo, G.: ASAM: Automatic Architecture Synthesis and Application Mapping, Microprocessors and Microsystems journal, Vol.37, No 8, pp. 1002-1019, 2013
- Jóźwiak, L. and Jan, Y.: Design of Massively Parallel Hardware Multi-Processors for Highly-Demanding Embedded Applications. Microprocessors and Microsystems, Volume 37, Issue 8, November 2013, pp. 1155–1172.
- L. Jóźwiak: Advanced Mobile and Wearable Systems, Microprocessors and Microsystems, Elsevier, Vol. 50, May 2017, pp. 202–221
- Many other papers of myself, and Ph.D. Theses and papers of my former Ph.D. students; many of them referenced in the above listed papers

38

SS-CPSIoT

6th Annual Summer School on
Cyber Physical Systems and Internet of Things

Modelling and Simulation Concepts

Hans Vangheluwe

12 June 2025 – Budva, Montenegro

Universiteit Antwerpen

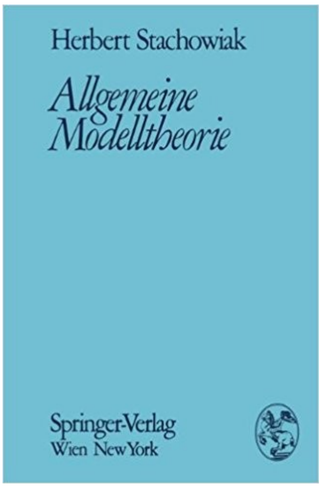
Ansymo
Antwerp Systems & Software Modelling
University of Antwerp

Nexor
Cyber-Physical Systems
University of Antwerp

FLANDERS MAKE
MANUFACTURING INNOVATION NETWORK

MPM4CPS

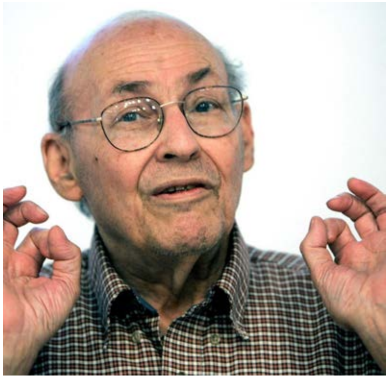
cost



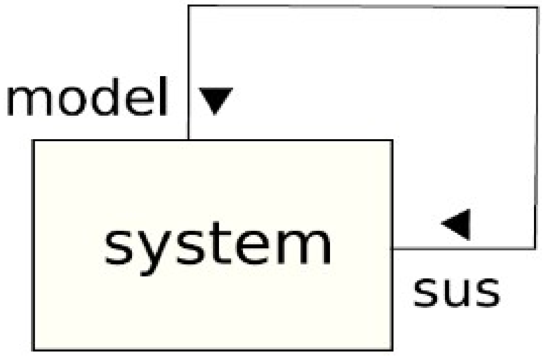
1973

"Model" Features

mapping feature	A model is based on an original.[4]
reduction feature	A model only reflects a (relevant) selection of an original's properties.
pragmatic feature	A model needs to be usable in place of an original with respect to some purpose.

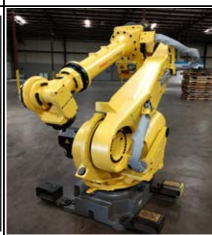
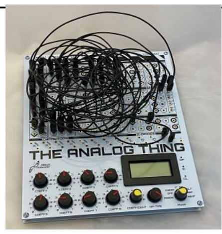
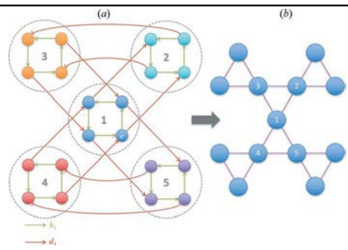


Marvin L. Minsky



To an observer B, an object A* is a model of an object A to the extent that B can use A* to answer questions that interest him about A.
Matter, Mind and Models

System under Study (SuS) vs. Model

	Real-World Model	**Virtual** Model
Real-World SuS	 	 
Virtual SuS	$\frac{dx}{dt} = \alpha x - \beta xy,$ $\frac{dy}{dt} = -\gamma y + \delta xy,$ 	

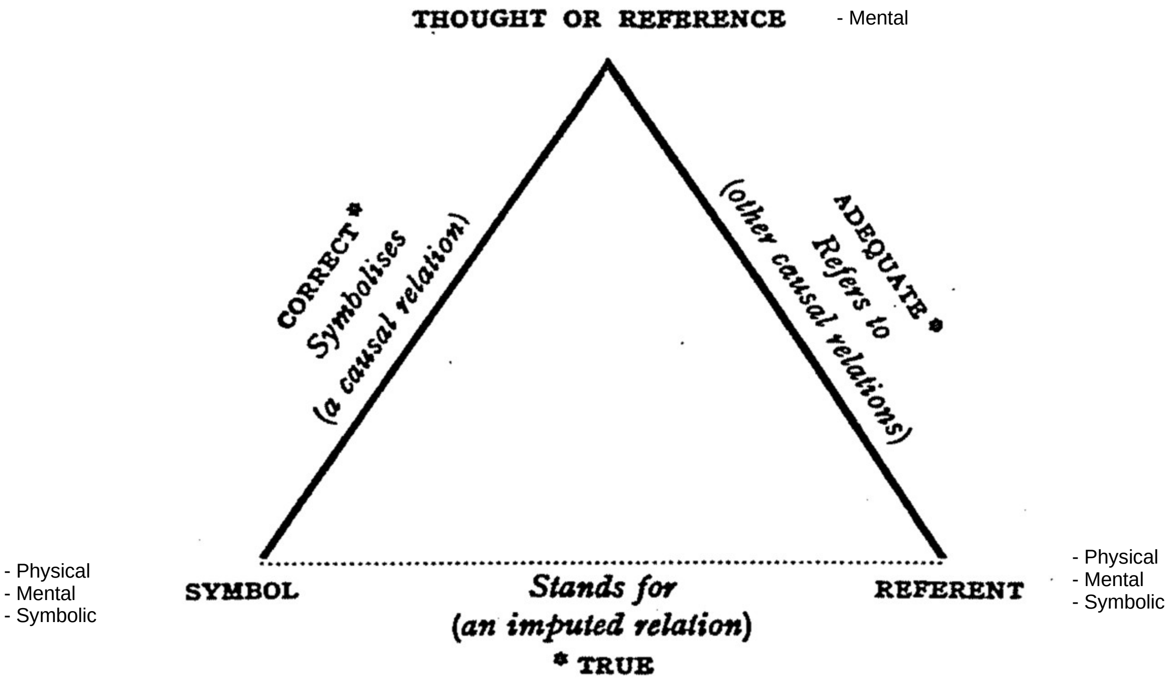
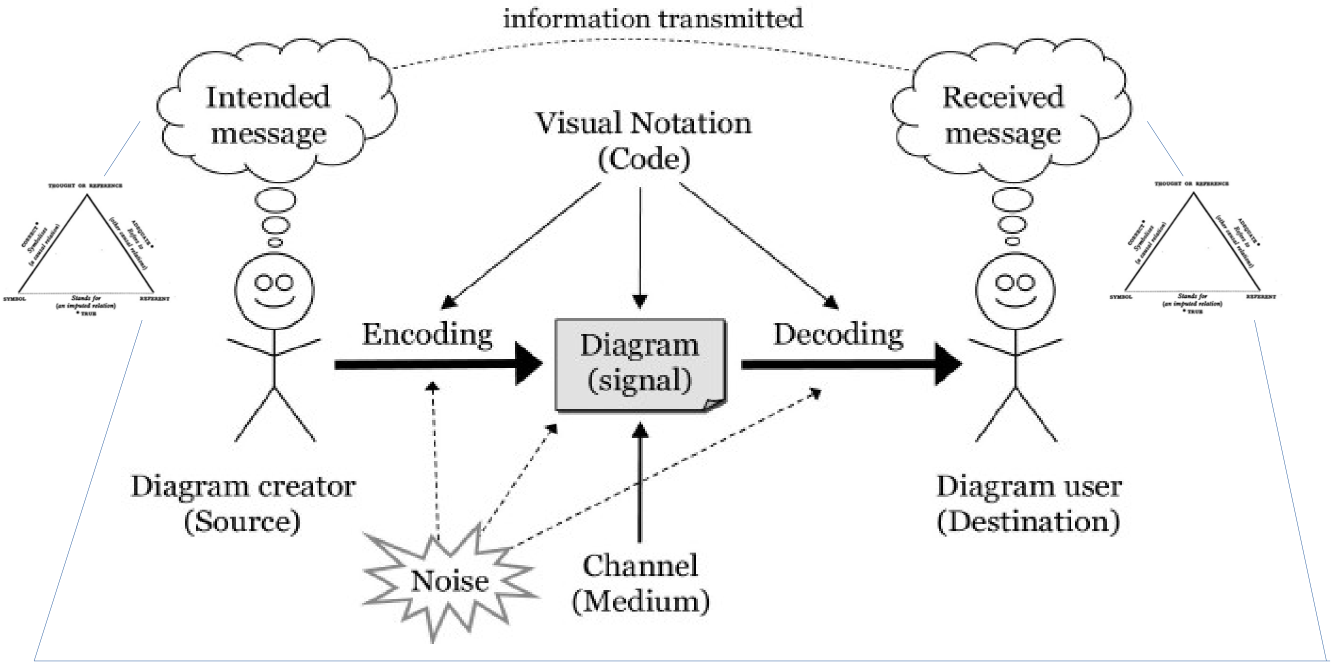
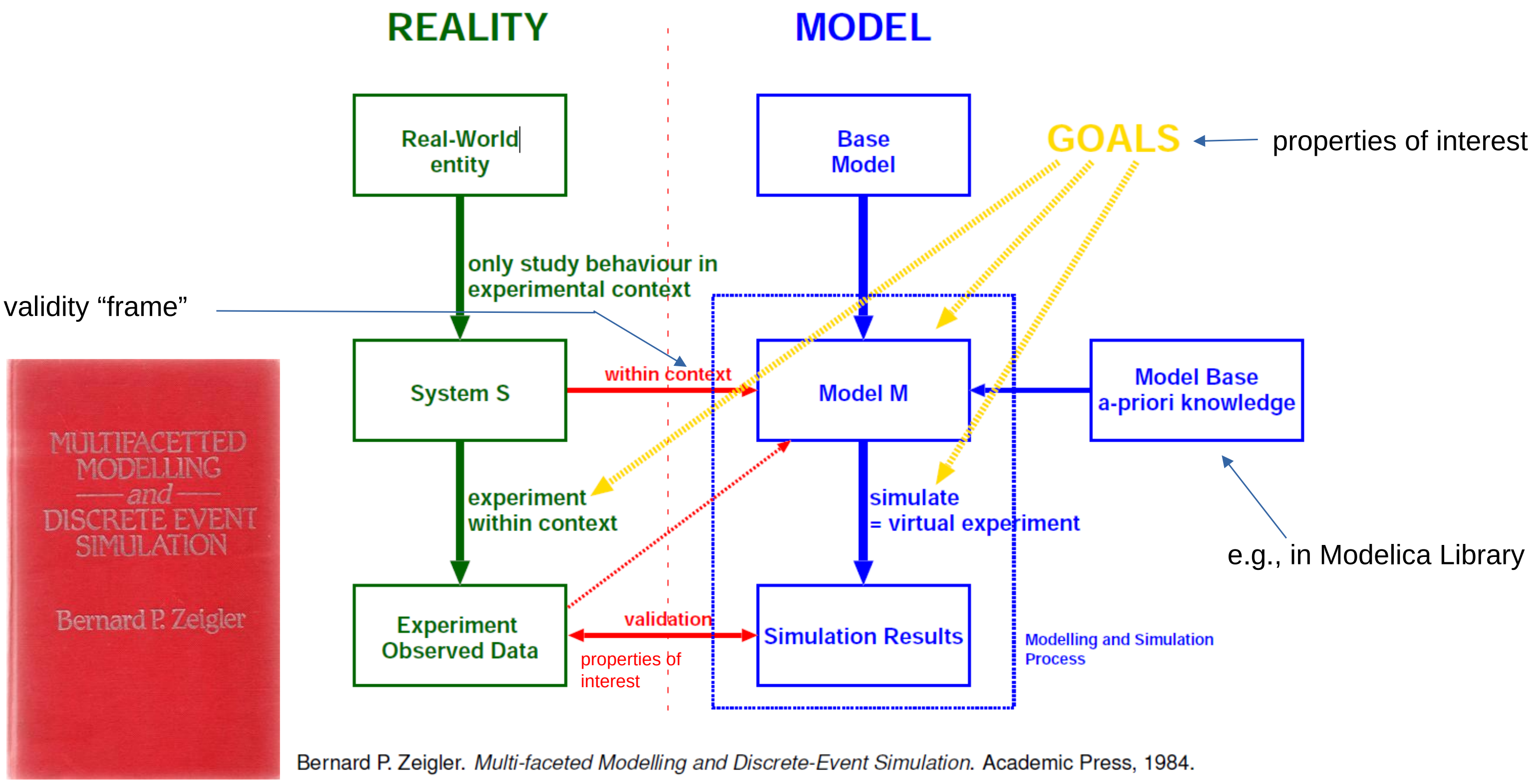


Figure taken from page 11, The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism, 1923, was co-authored by C. K. Ogden and I. A. Richards, Magdalene College, University of Cambridge

Moody "Physics of Notation": communication theory



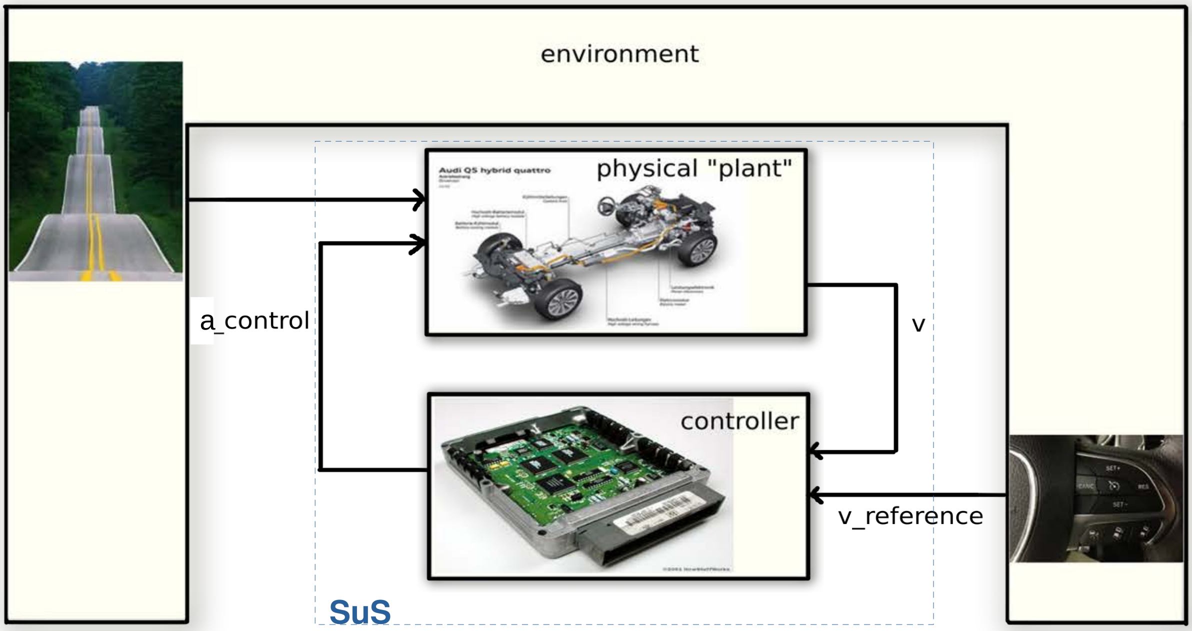
modelling for: substitutability (engineering), explainability (science)



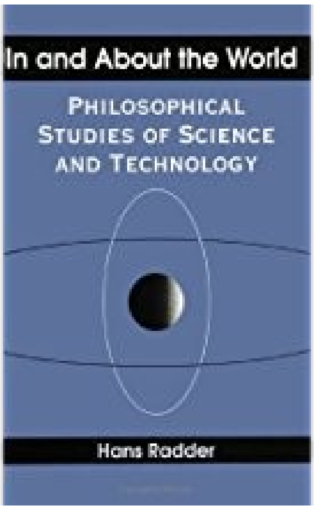
Bernard P. Zeigler. *Multi-faceted Modelling and Discrete-Event Simulation*. Academic Press, 1984.

System under Study (SuS) = "plant" || controller

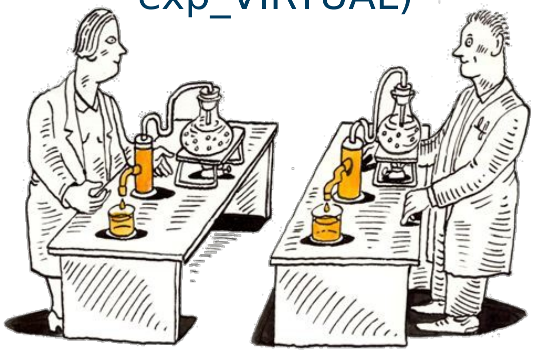
SuS is not studied in isolation: **SuS || environment**



Basis of Science: **Reproducibility** of **Experiments**



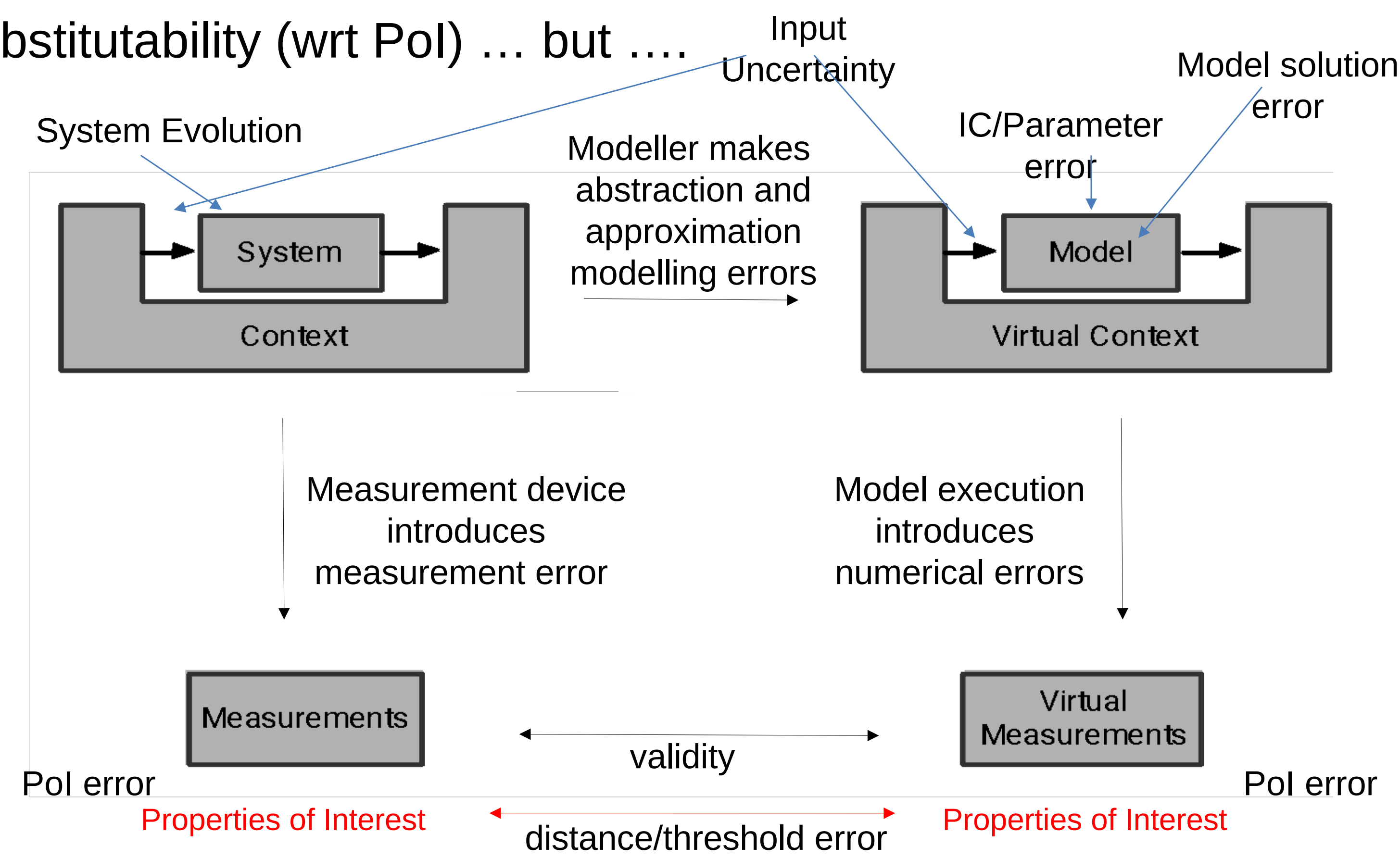
Experiment
(both exp_REAL and exp_VIRTUAL)



Repeatable
Replicable
Reproducible

Plesser, H. E. 2018. *Reproducibility vs. Replicability: A Brief History Of A Confused Terminology.* Frontiers in neuroinformatics 11:76.

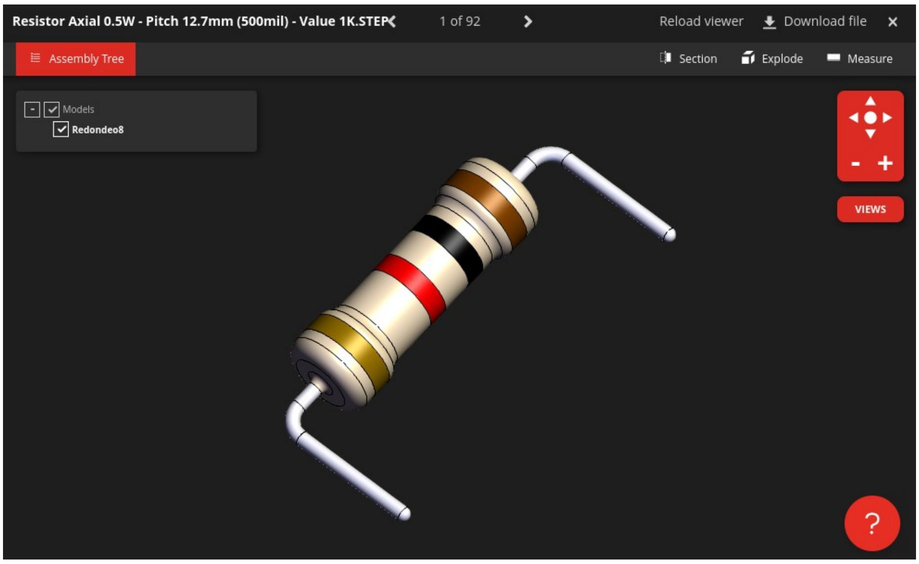
Substitutability (wrt PoI) … but ….



Model Validity Motivating Example: Electrical Resistor

Property of interest (PoI):

geometry

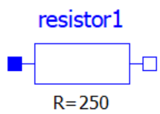


https://grabcad.com/library/resistors-0-5w-pitch-12-7mm-500mil-1

Model Validity Motivating Example: Electrical Resistor

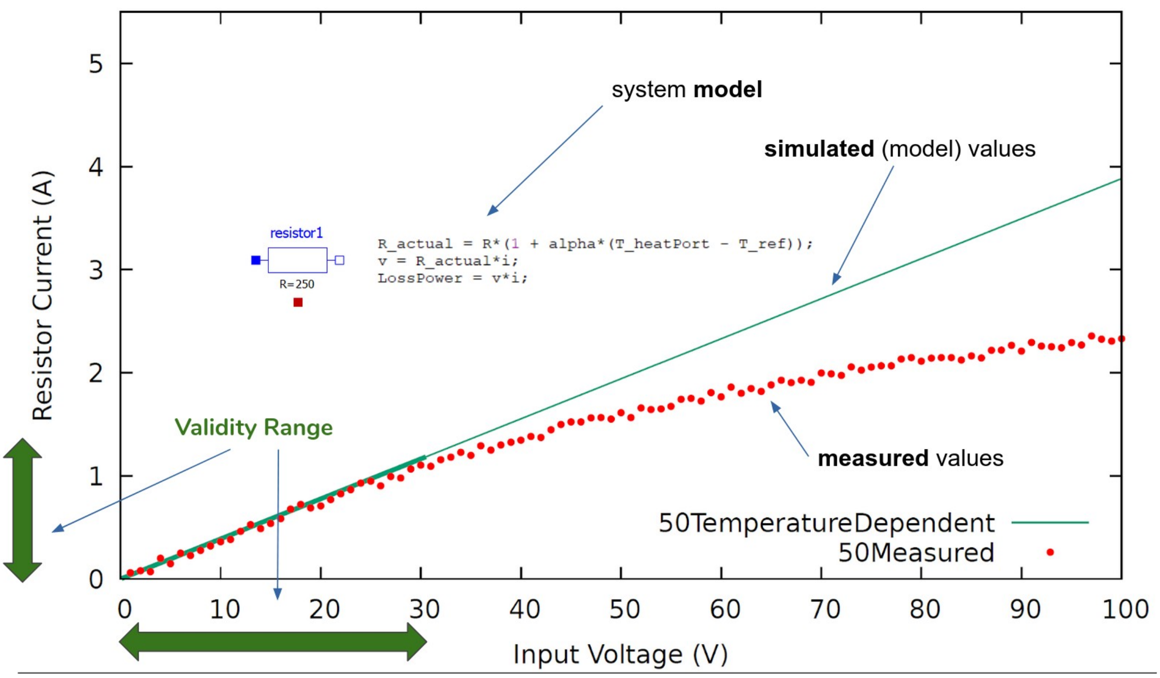
Property of interest (PoI):

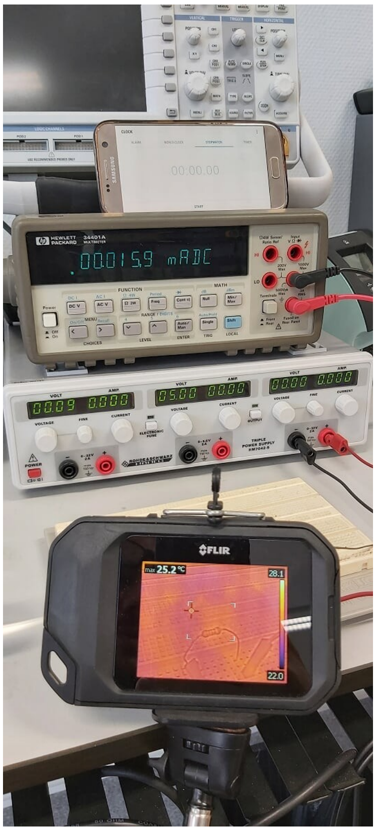
constitutive relation
between voltage drop *V* over resistor and current ***i*** through resistor



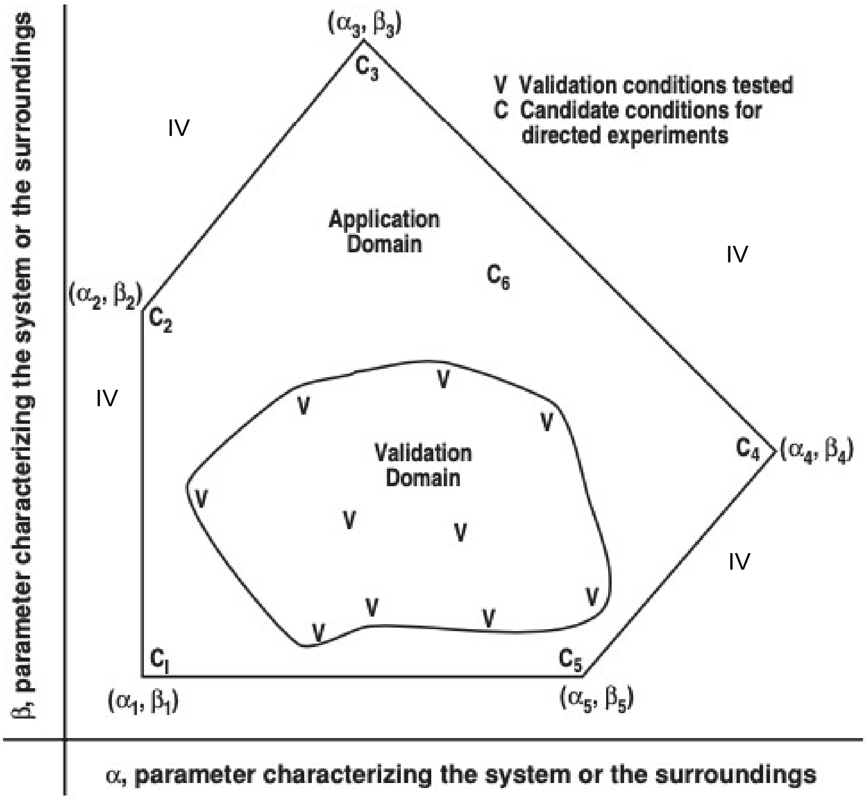

```
R*i = v
```

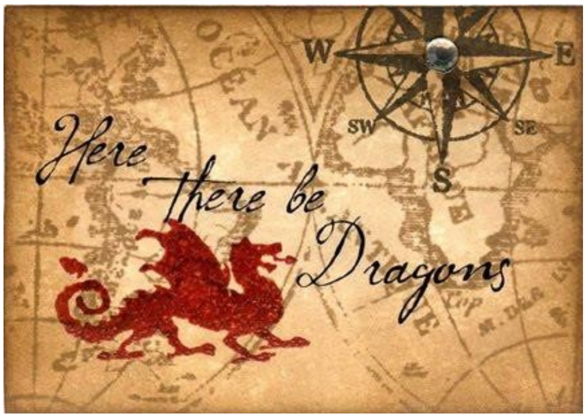
A Resistor Model's **Validity Range**





https://msdl.uantwerpen.be/cloud/public/fcfc42





W. Oberkampf, C. Roy. *Verification and Validation in Scientific Computing.* Cambridge University Press, 2010.

Abstract (In)Validity Frame

The (possibly infinite) **Set of Experiments (descriptions/frames)** ***e*** for which the **Distance *d*** between the obtained (computed) **Properties of Interest PoI** from ***e*** carried out in the **REAL** world and ***e*** carried out in the **VIRTUAL** world is (larger)smaller than a **threshold *Tr***.

$$AVF_{\mu_n} \cup AIF_{\mu_n} = \mathbb{U}_{\mu_n}$$

$$AVF_{\mu_n} \cap AIF_{\mu_n} = \varnothing$$

Thanks to Rhys Goldstein for the notion of abstract frame AUTODESK.

Concrete (In)Validity Frame

- Concrete Validity Frame (CVF)

  The finite set of **specifications (experiment frames) of performed experiments** in which a model is valid

- Concrete Invalidity Frame (CIF)

  The finite set of **specifications (experiment frames) of performed experiments** in which a model is invalid

$$CVF_{\mu_n} \cap CIF_{\mu_n} = \emptyset$$

Rakshit Mittal, Raheleh Eslampanah, Lucas Lima, Hans Vangheluwe and Dominique Blouin. *Towards an Ontological Framework for Validity Frames.* 2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), Västerås, Sweden, pp. 801 - 805.

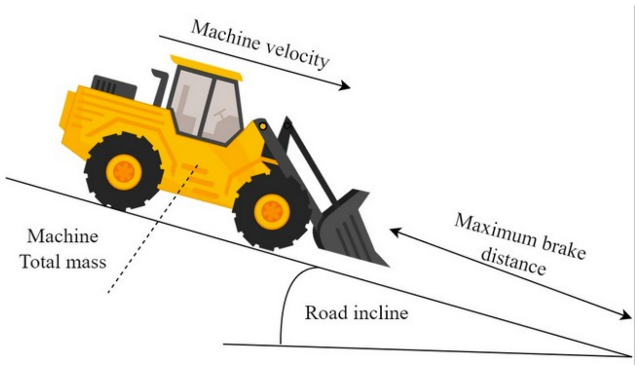
Inferred (In)Validity Frame

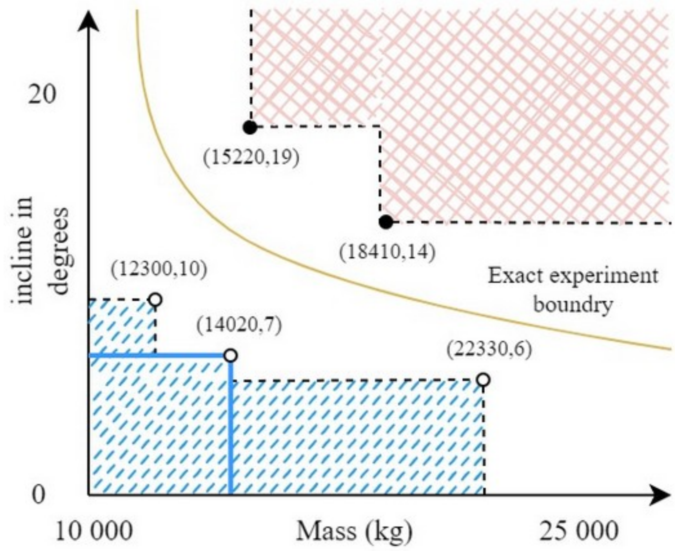
- Inferred Validity Frame (ICVF)

  The finite set of **performed experiments** in which a model is valid **extended** with a possibly infinite set of experiments in which a model is valid. This set is **inferred based on domain knowledge**.

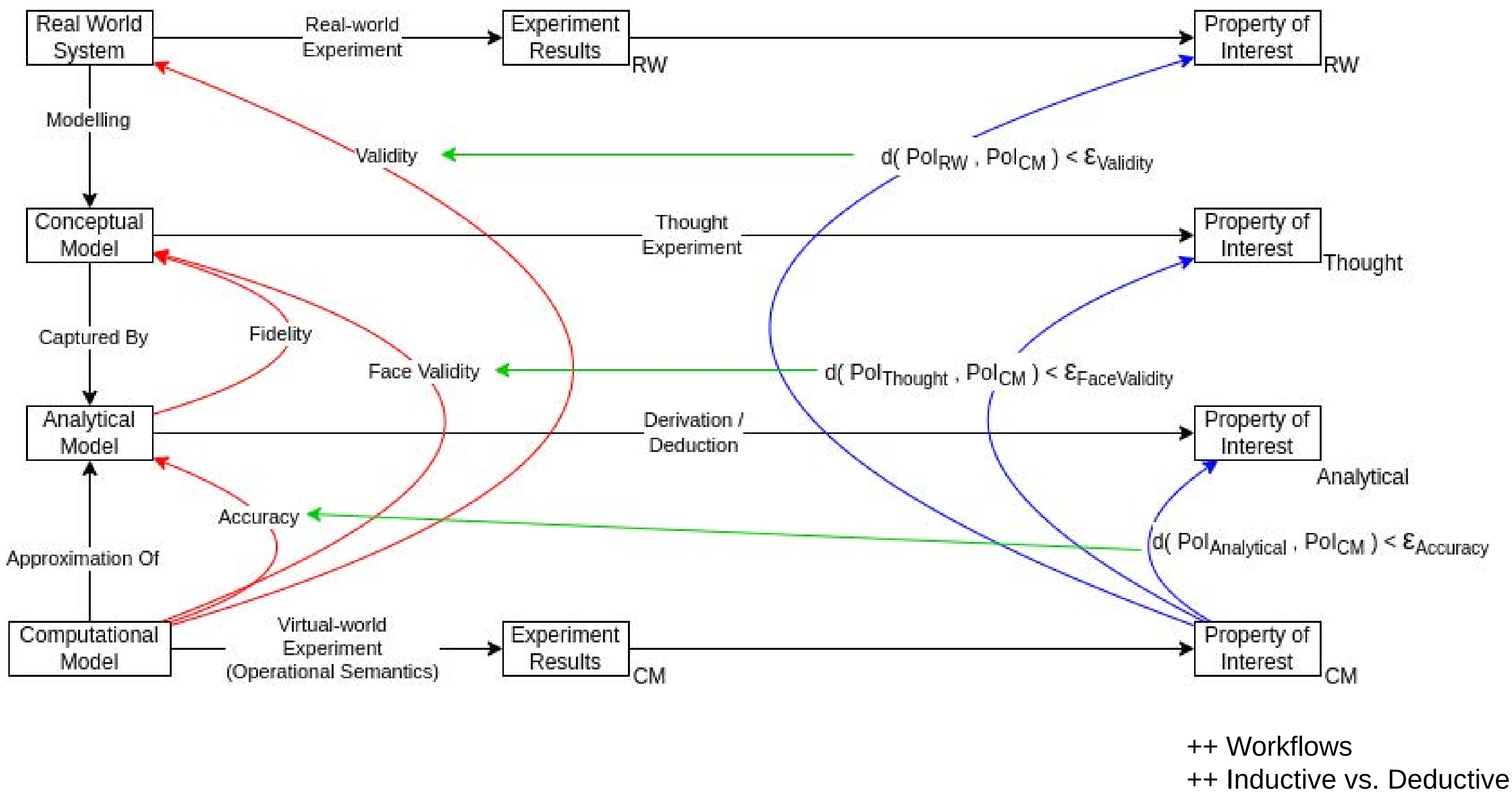
- Inferred Invalidity Frame (ICIF)
  - The finite set of **performed experiments** in which a model is invalid **extended** with a possibly infinite set of experiments in which a model is invalid. This set is **inferred based on domain knowledge**.

Inferred (In)Validity Frame





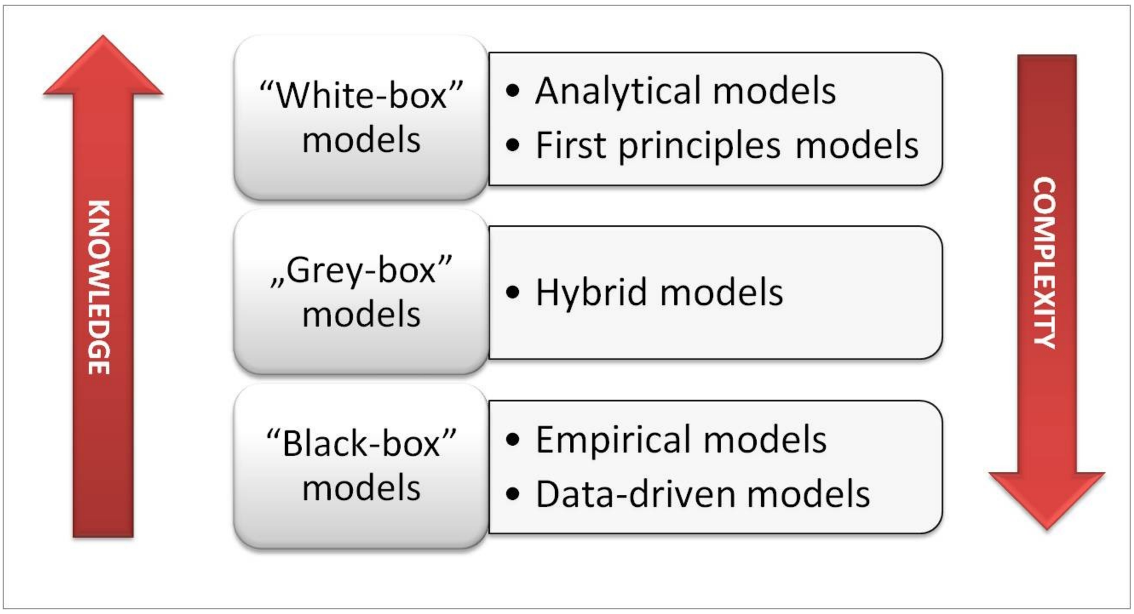
Johan Cederbladh, Loek Cleophas, Eduard Kamburjan, Lucas Lima and Hans Vangheluwe. *Symbolic Reasoning for Early Decision-Making in Model-Based Systems Engineering.* 2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), Västerås, Sweden, pp. 721 – 725.


Real World System
Real-world Experiment
Experiment Results
RW
Property of Interest
RW
Modelling
Validity
d(PoI_{RW} , PoI_{CM}) < $\varepsilon_{Validity}$
Conceptual Model
Thought Experiment
Property of Interest
Thought
Captured By
Fidelity
Face Validity
d($PoI_{Thought}$, PoI_{CM}) < $\varepsilon_{FaceValidity}$
Analytical Model
Derivation / Deduction
Property of Interest
Analytical
Accuracy
Approximation Of
d($PoI_{Analytical}$, PoI_{CM}) < $\varepsilon_{Accuracy}$
Computational Model
Virtual-world Experiment (Operational Semantics)
Experiment Results
CM
Property of Interest
CM
++ Workflows
++ Inductive vs. Deductive

different ways of **obtaining** models

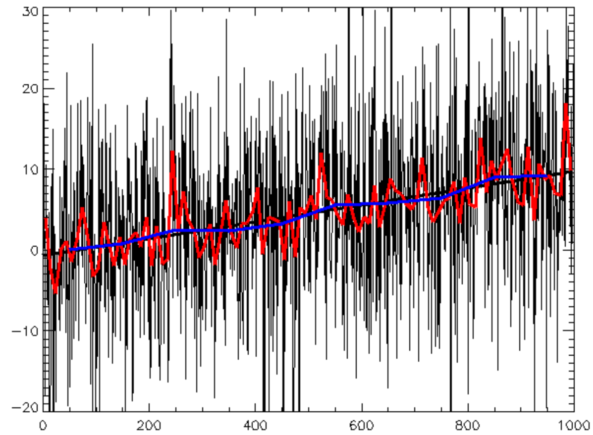
(in same or different formalisms)

→ different DTs



simulatelive.com

Inductive Modelling: models based on (measurements) data



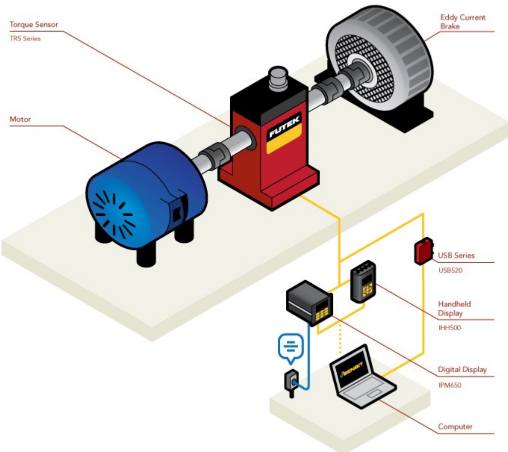
- instance (technology) – specific

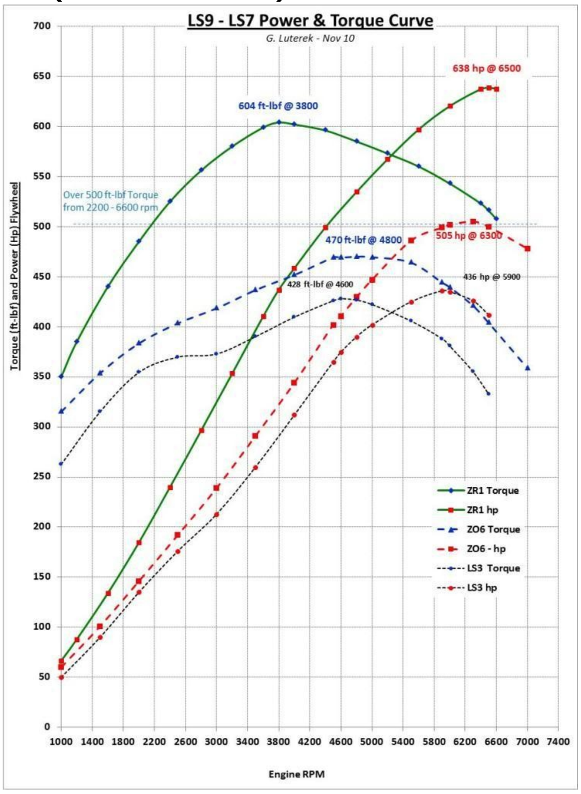
- high (experimentation) cost
- may not even be possible to measure

- allows reproducing data, no extrapolation; no insight/explanation

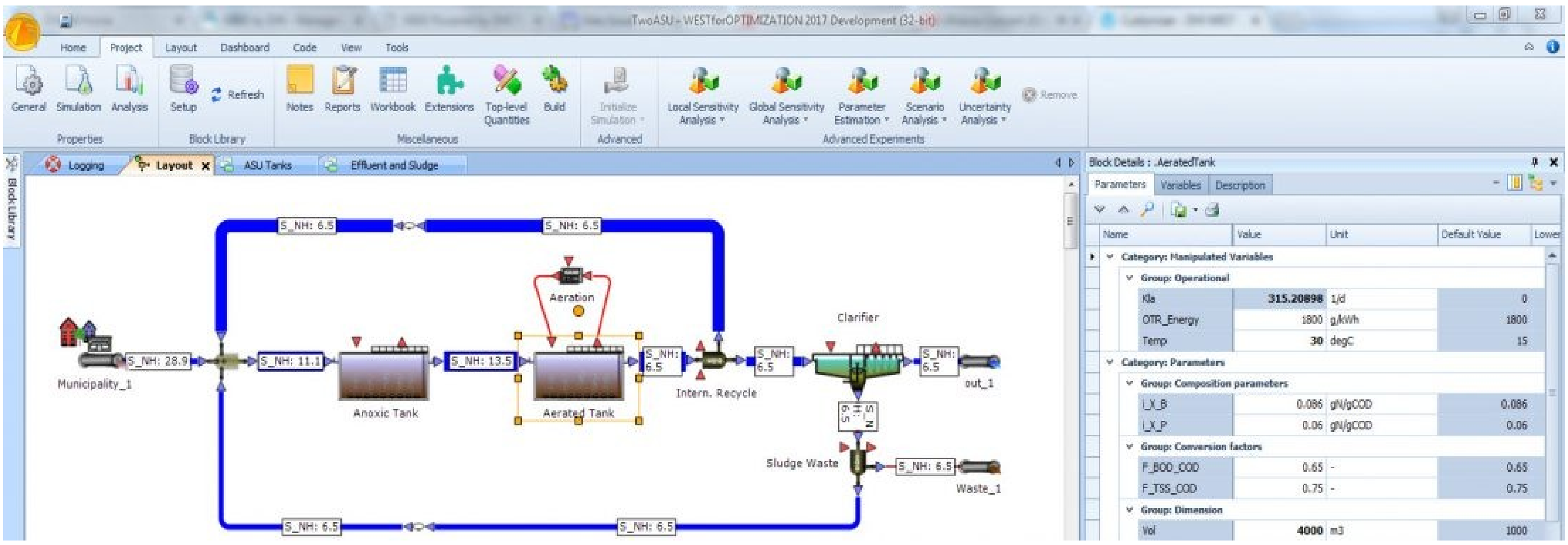
- inductive vs. deductive modelling workflow science vs. engineering, usually combination

Black Box model: “Torque Curve” (measured)



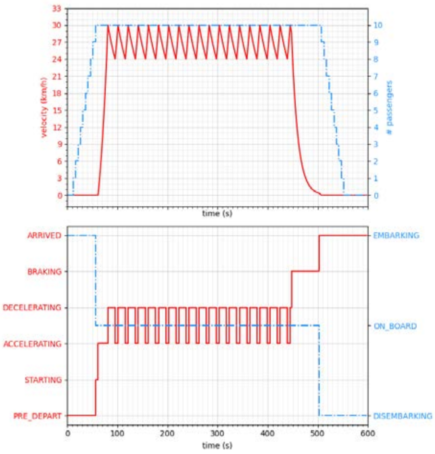


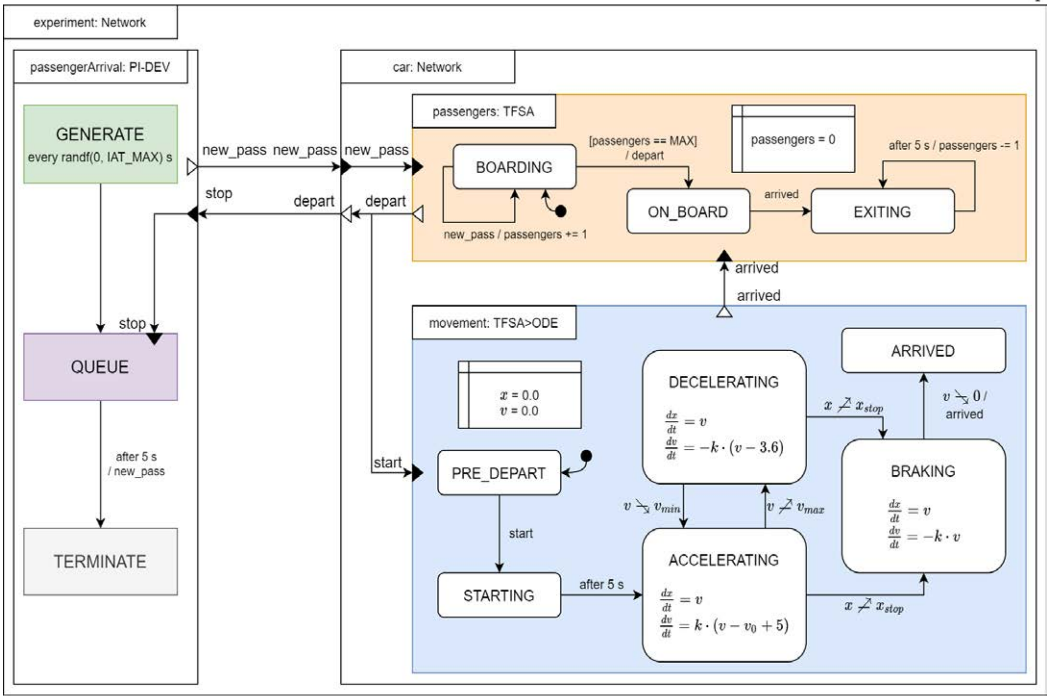
use models in **most appropriate** (domain-specific) **formalism**



http://www.mikebydhi.com/products/west

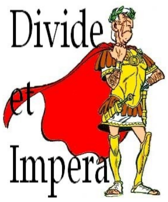
use models in most appropriate formalism**(s)**

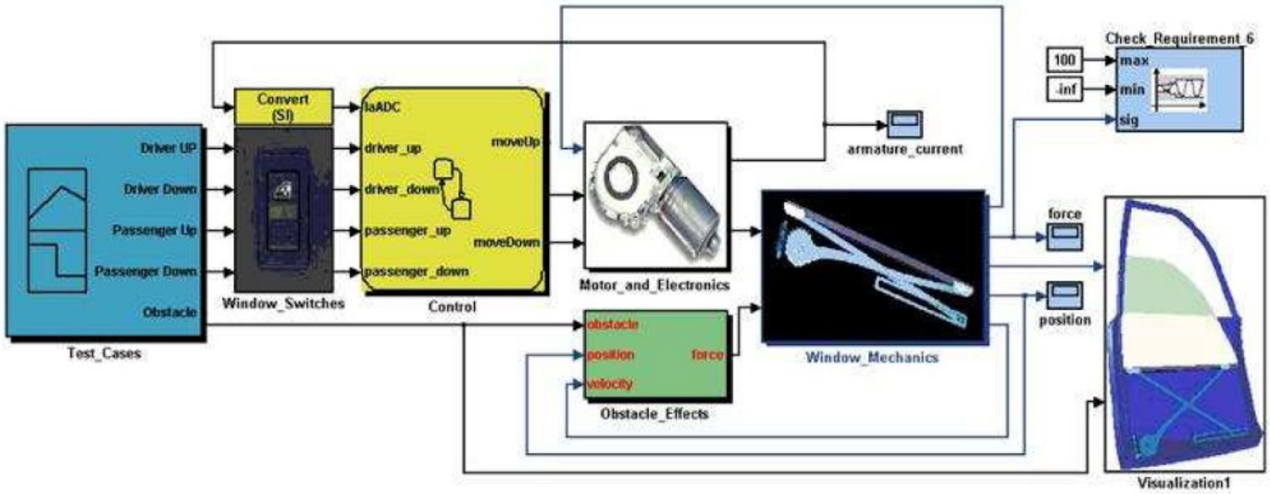




R. Paredis, J. Denil, and H.Vangheluwe. 2021.
"*Specifying and Executing the Combination of Timed Finite State Automata and Causal-Block Diagrams by Mapping onto DEVS*". In Proceedings of WSC 2021.

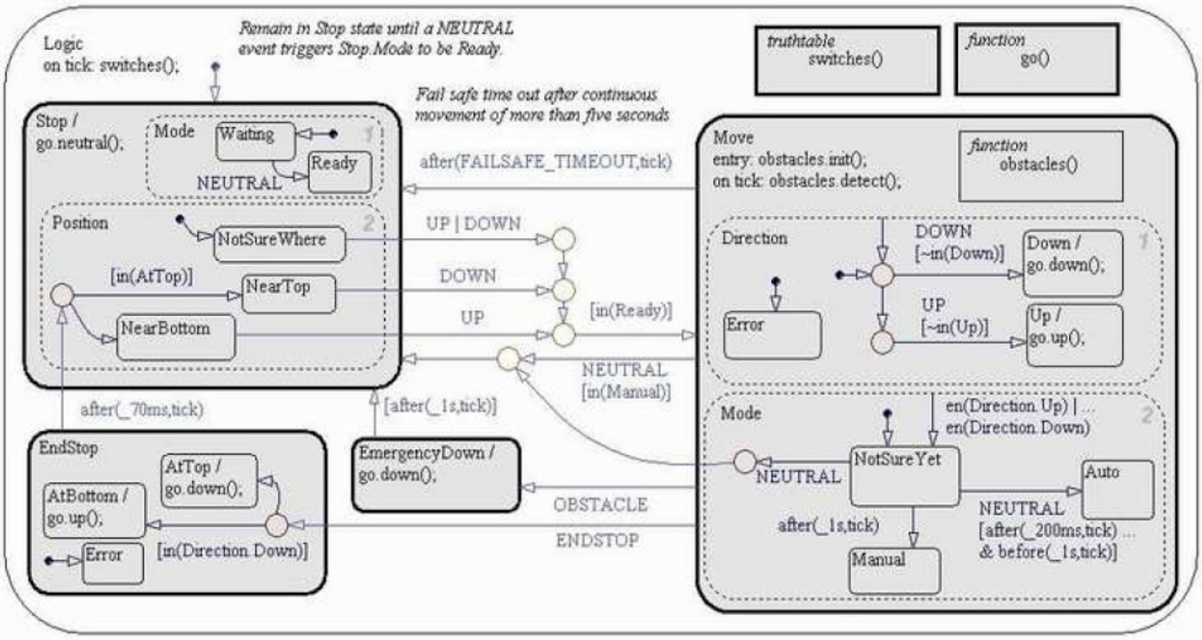
Components in Different Formalisms

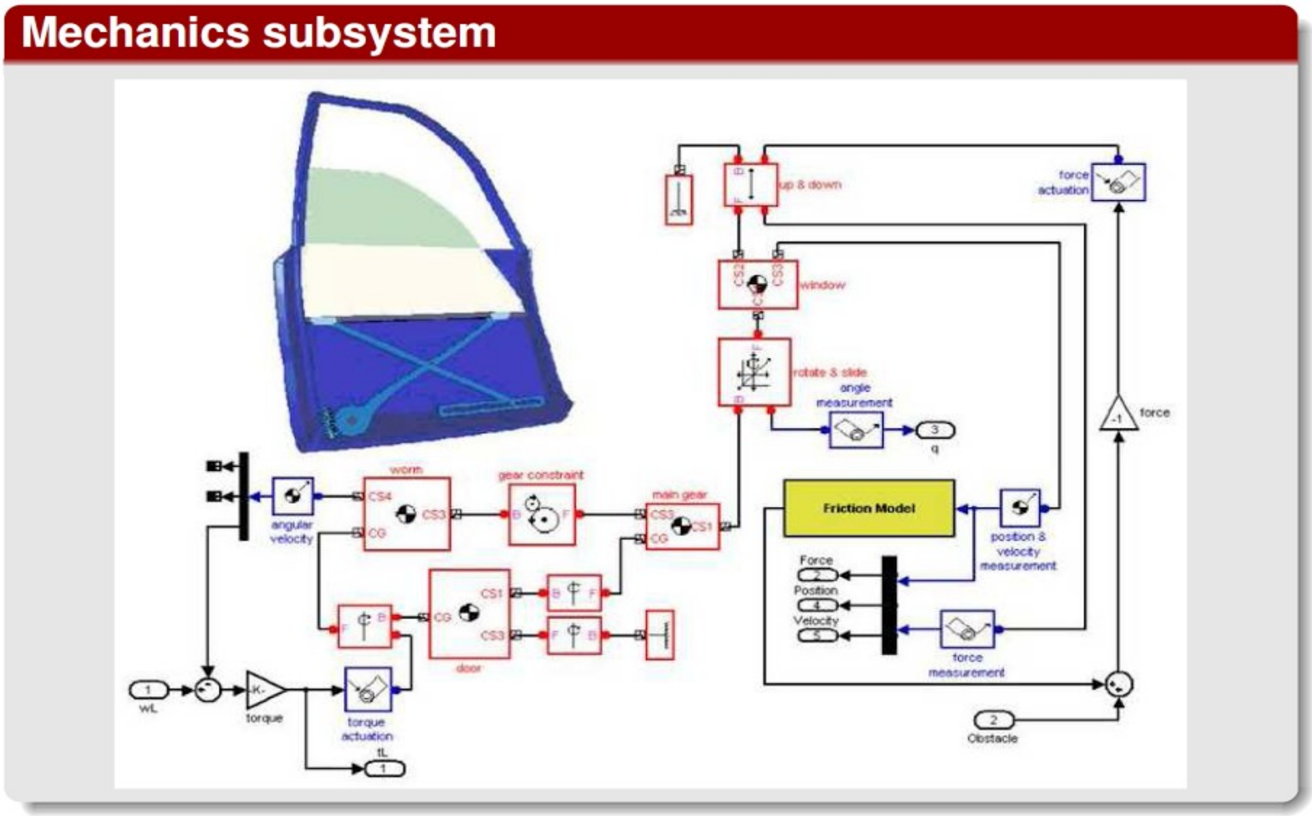




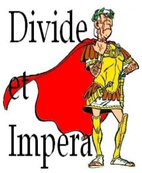
www.mathworks.com/products/demos/simulink/PowerWindow/html/PowerWindow1.html

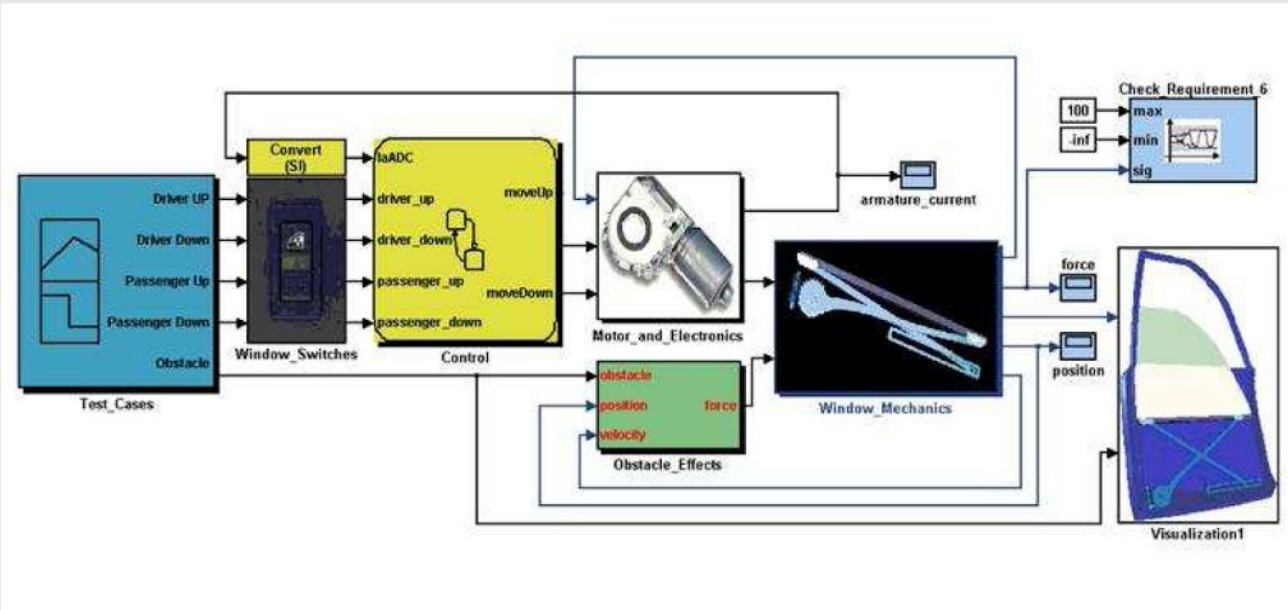
Controller, using Statechart(StateFlow) formalism




Mechanics subsystem
up & down
force actuation
window
rotate & slide
angle measurement
q
force
worm
gear constraint
main gear
angular velocity
Friction Model
position & velocity measurement
Force
Position
Velocity
door
force measurement
wL
torque
torque actuation
tL
Obstacle

Components in Different Formalisms

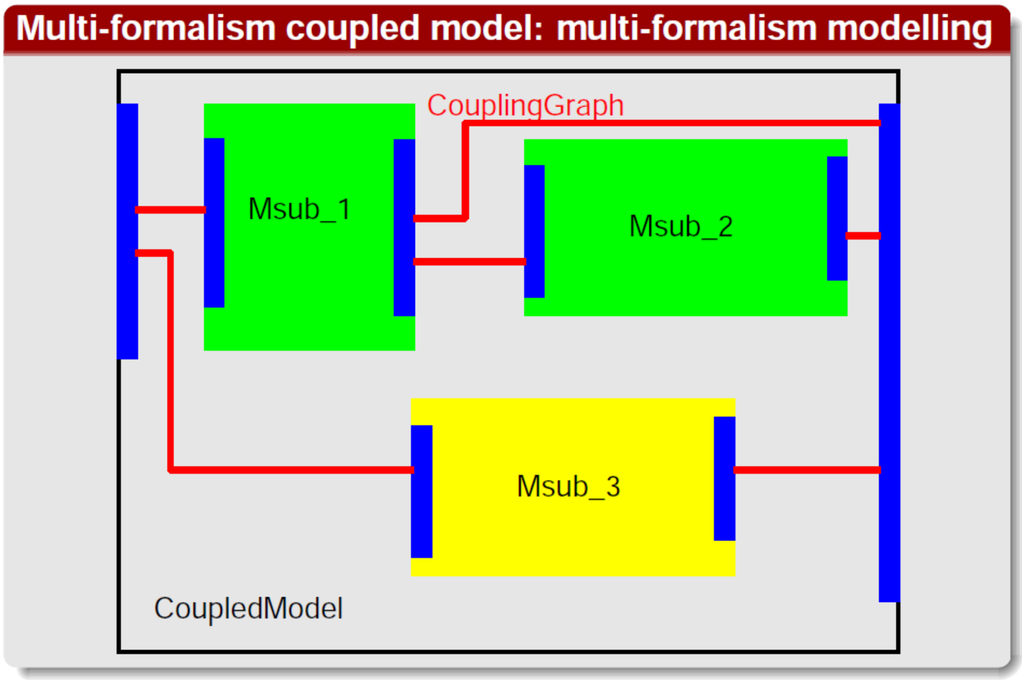


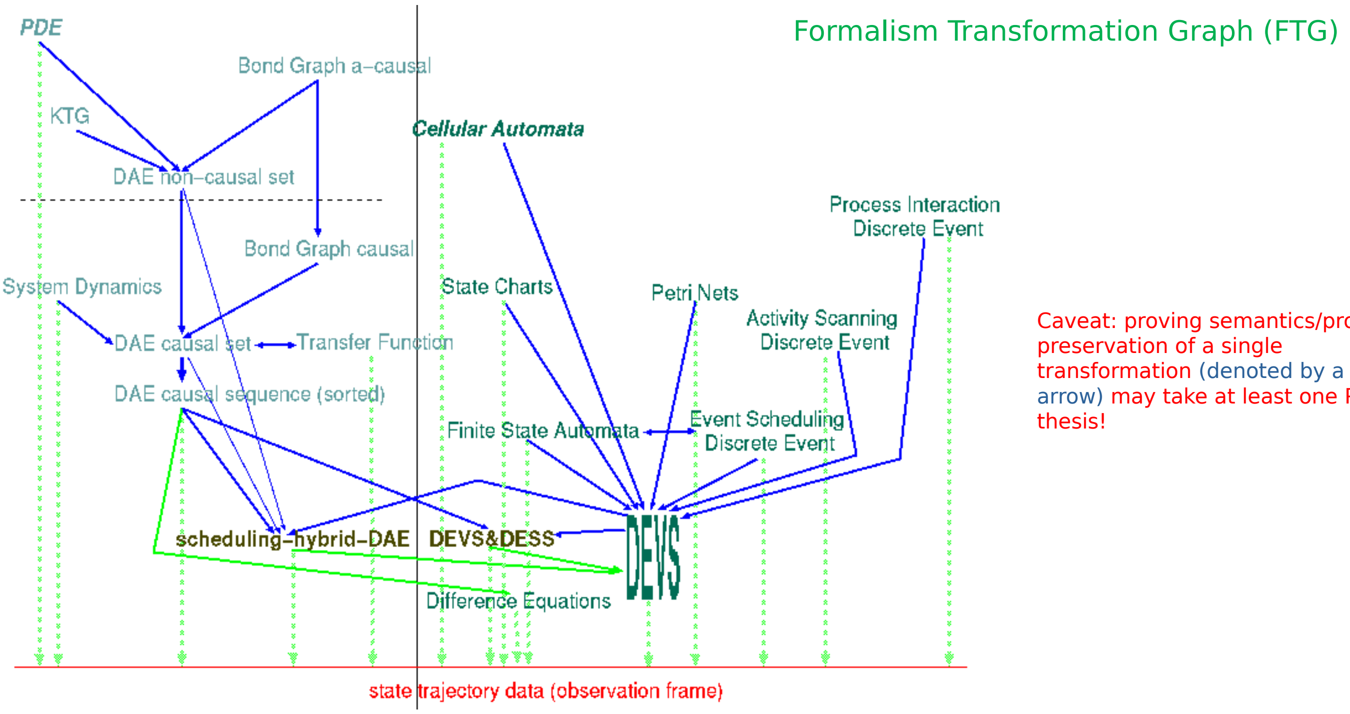


www.mathworks.com/products/demos/simulink/PowerWindow/html/PowerWindow1.html meaning of multi-formalism models?

Formalism Transformation Graph (FTG)

Bran Selić: "fragmentation problem"

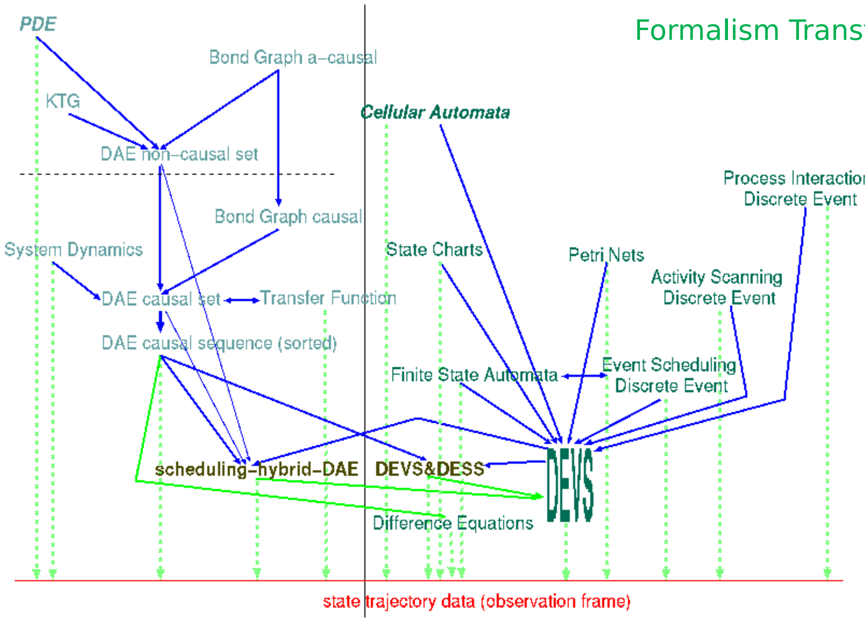


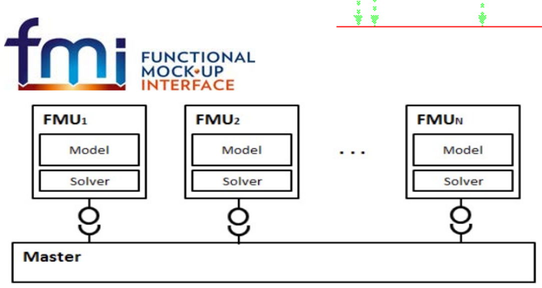


Caveat: proving semantics/property preservation of a single transformation (denoted by a blue arrow) may take at least one PhD thesis!

Hans Vangheluwe and Ghislain C. Vansteenkiste. A multi-paradigm modeling and simulation methodology: Formalisms and languages. In European Simulation Symposium (ESS) , pages 168 – 172. Society for Computer Simulation International (SCS), October 1996. Genoa, Italy.

Formalism Transformation Graph (FTG)
co-simulation

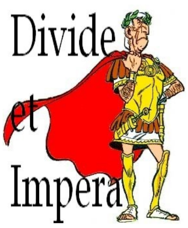




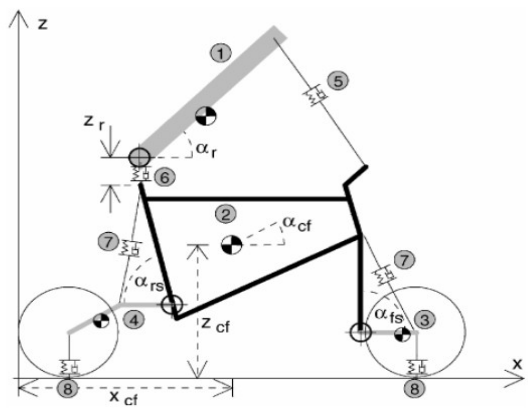
Cláudio Gomes, Casper Thule, David Broman, Peter Gorm Larsen, and Hans Vangheluwe. Co-simulation: A survey. ACM Computing Surveys (CSUR) , 51(3):49:1-49:33, 2018.

different **abstractions** (levels of detail)
(same or different formalisms)

→ different Twin Objects



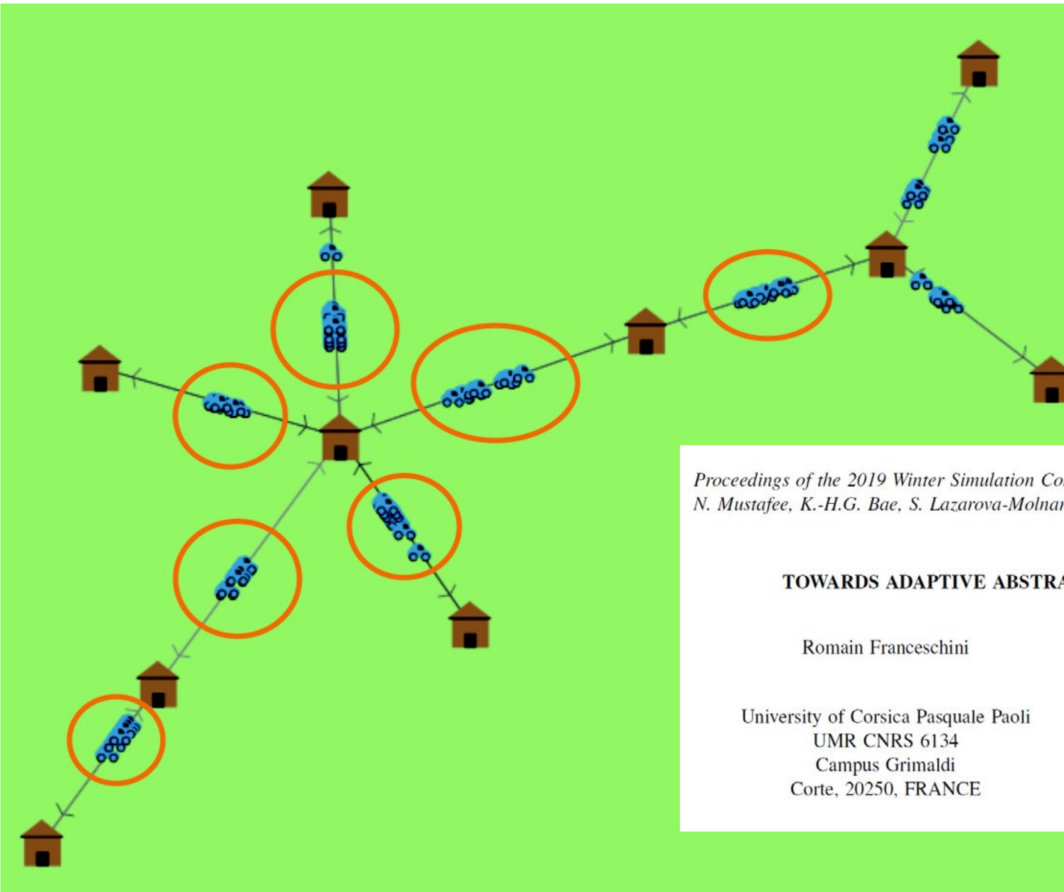
Distributed parameter



Lumped parameter

Abstraction

- For performance (scale-ability)
- For **insight**

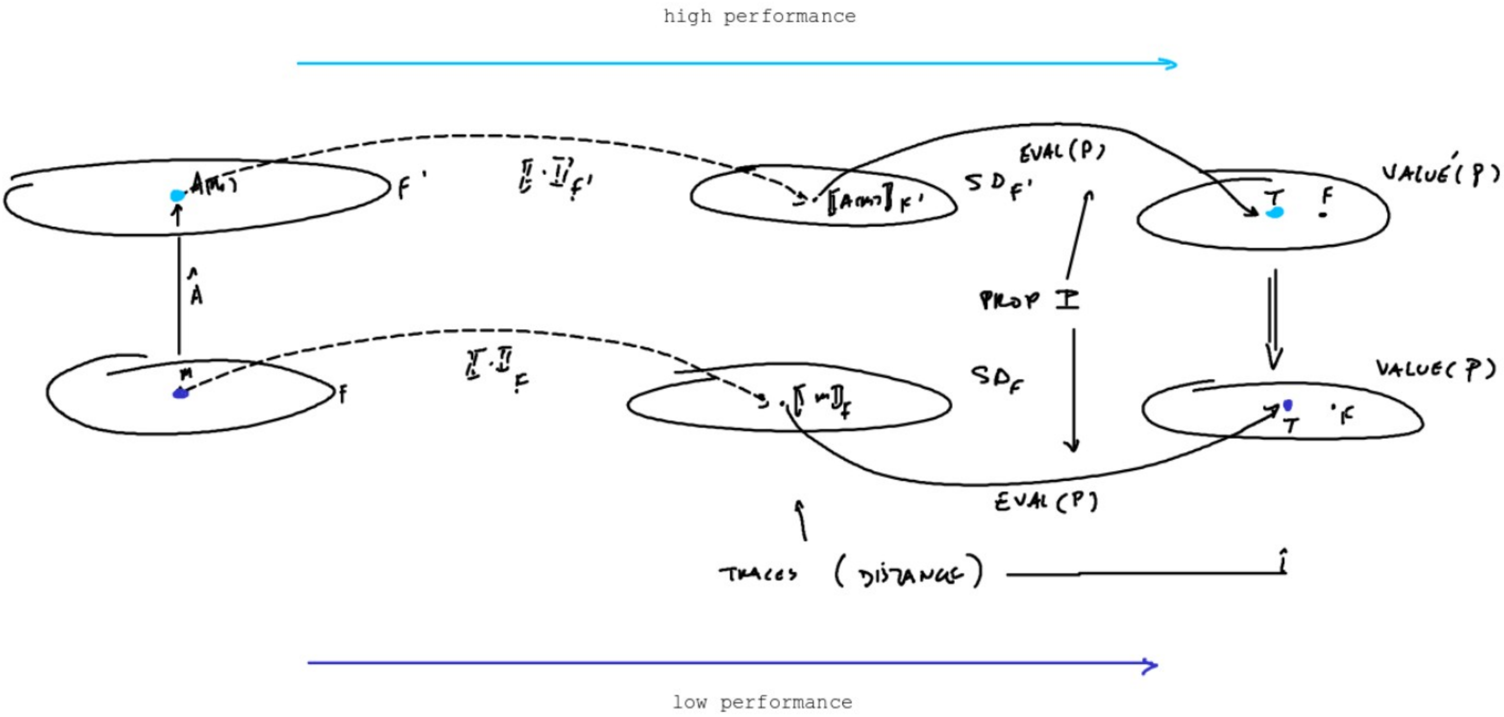


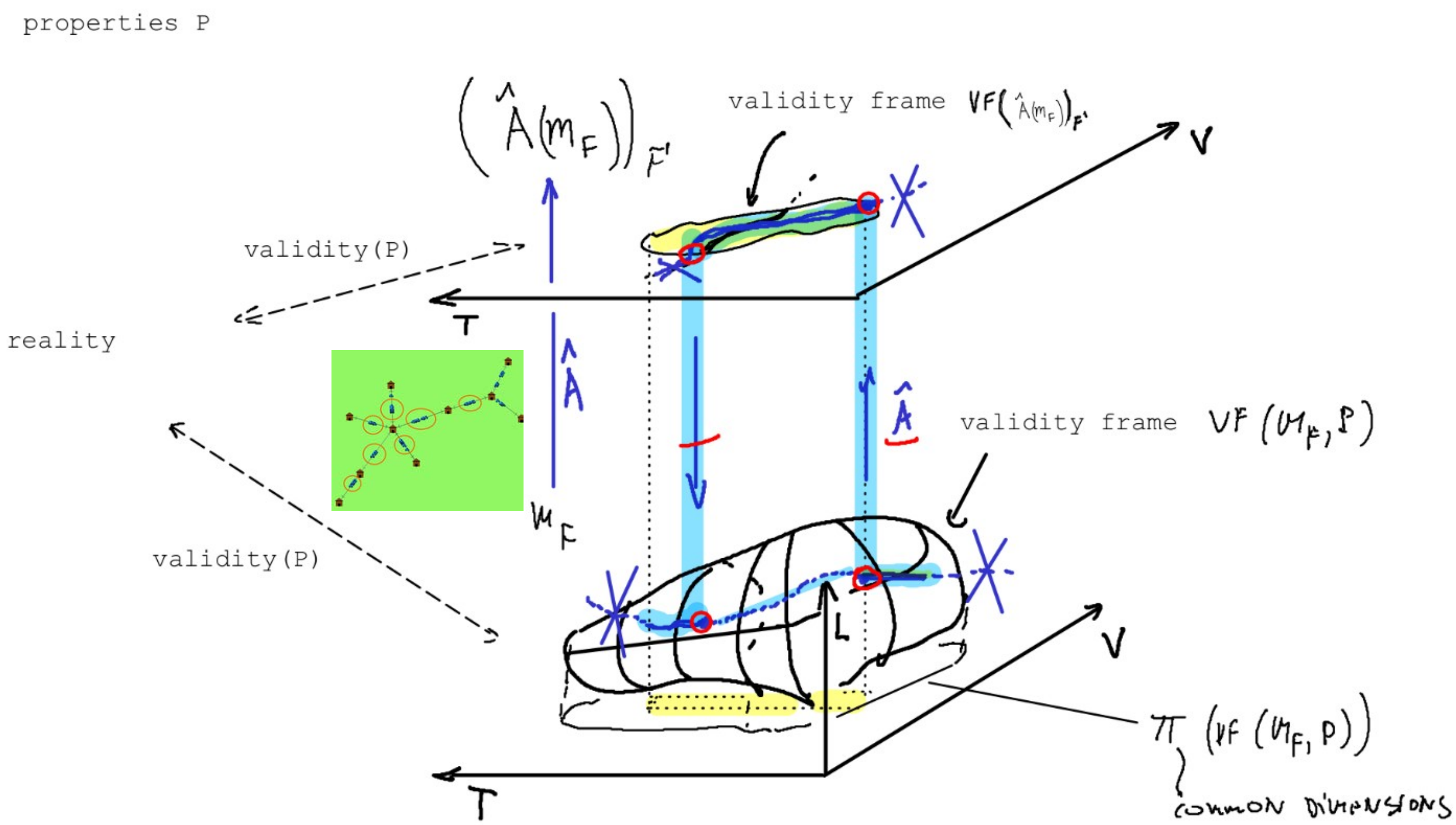
Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

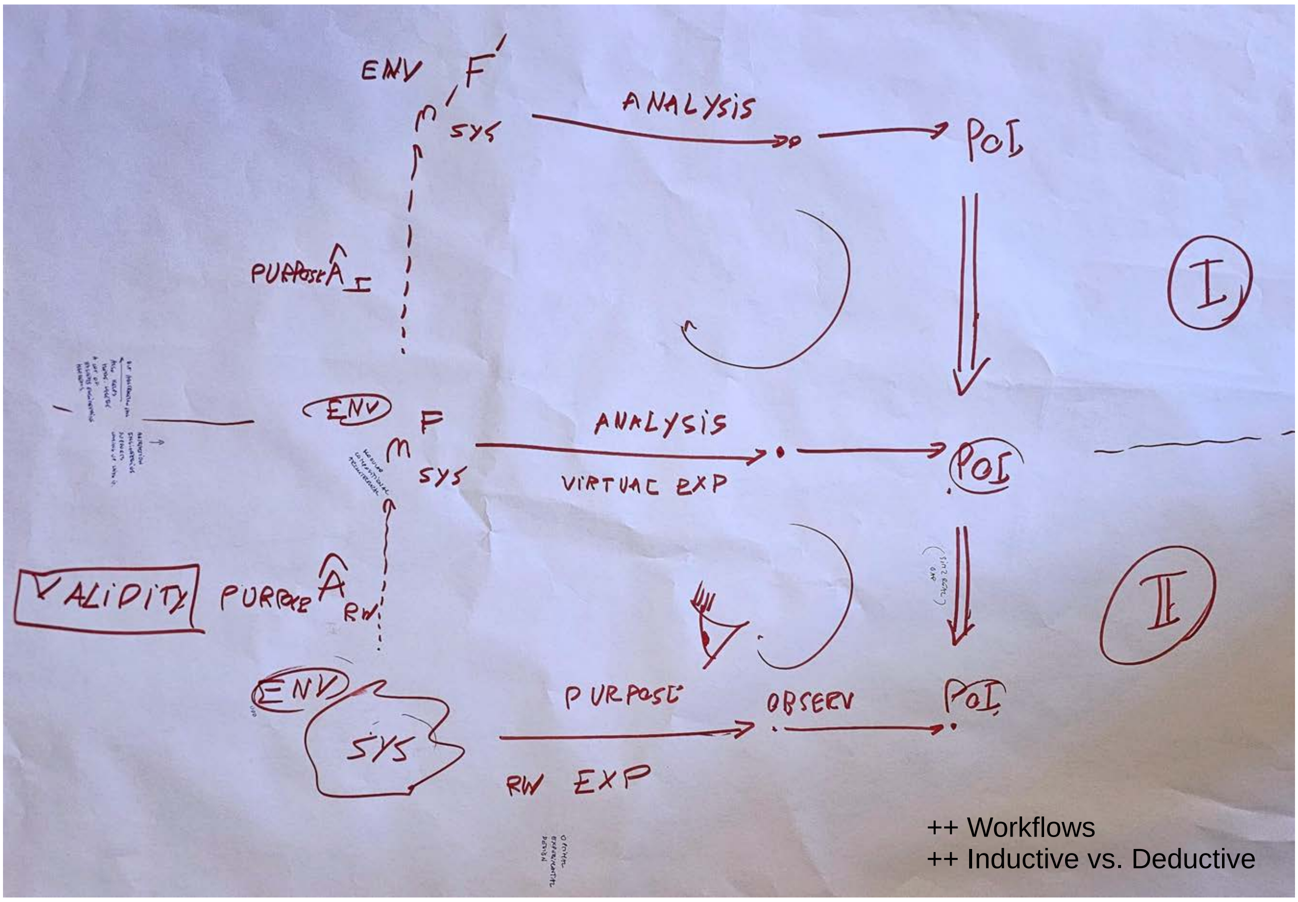
TOWARDS ADAPTIVE ABSTRACTION IN AGENT BASED SIMULATION

Romain Franceschini
University of Corsica Pasquale Paoli
UMR CNRS 6134
Campus Grimaldi
Corte, 20250, FRANCE

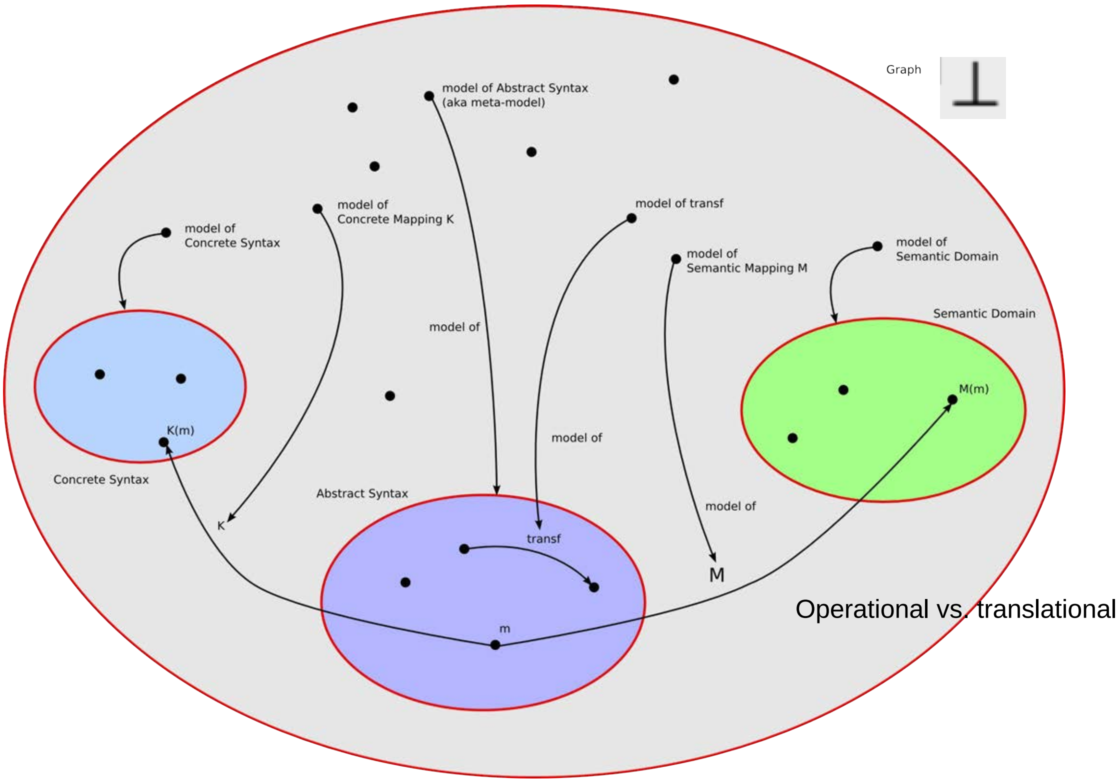
Simon Van Mierlo
Hans Vangheluwe
Department of Mathematics and Computer Science
University of Antwerp - Flanders Make
Middelheimlaan 1
Antwerp, 2020, BELGIUM


high performance
EVAL(P)
VALUE'(P)
PROP I
VALUE(P)
EVAL(P)
TRACES (DISTANCE)
low performance

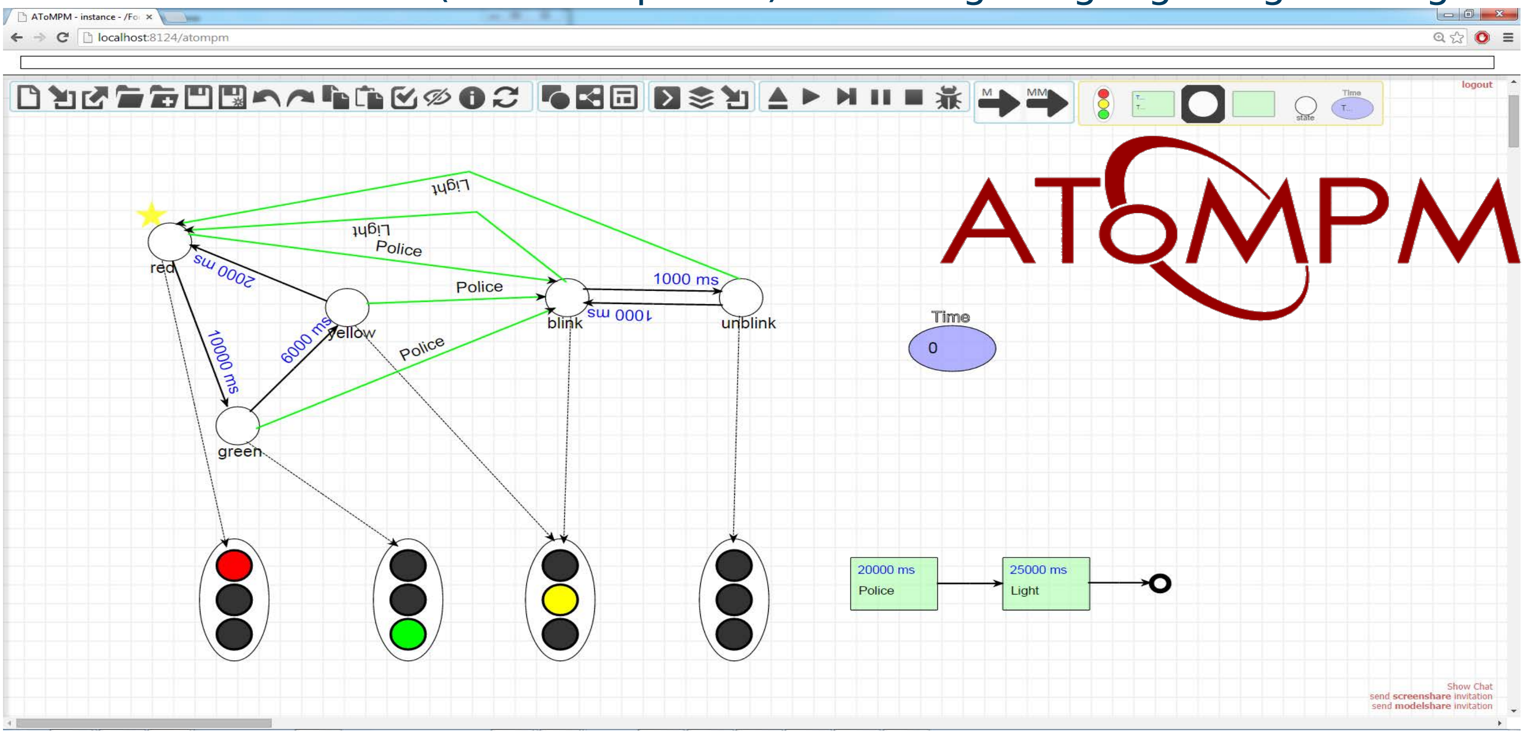

properties P
validity frame
validity(P)
reality
validity(P)
validity frame
T
V
L
T
V
COMMON DIMENSIONS


ENV
F
SYS
ANALYSIS
PoI
PURPOSE
I
ENV
F
SYS
ANALYSIS
VIRTUAL EXP
PoI
II
VALIDITY
PURPOSE
RW
ENV
SYS
PURPOSE
OBSERV
PoI
RW EXP
++ Workflows
++ Inductive vs. Deductive

Explicit Modelling of Modelling Languages/Formalisms (++ debugging)



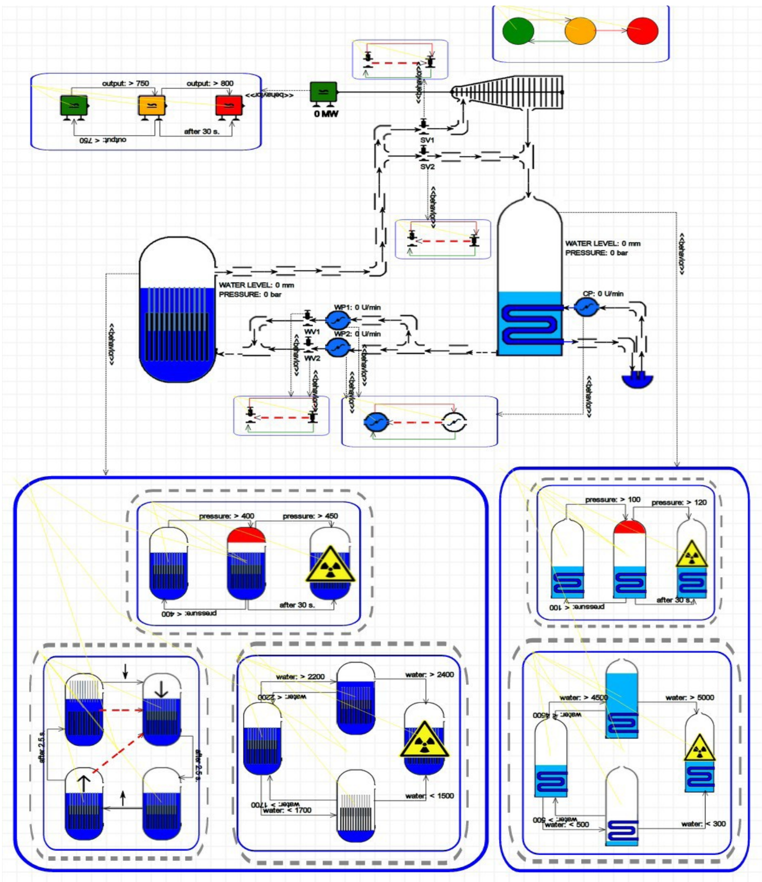
(domain-specific) Modelling Language Engineering



Raphael Mannadiar. A Multi-Paradigm Modelling Approach to the Foundations of Domain-Specific Modelling. PhD thesis, McGill Univ., 2012.

Eugene Syriani, Hans Vangheluwe, Raphael Mannadiar, Conner Hansen, Simon Van Mierlo, and Huseyin Ergin. AToMPM: A web-based modeling environment. In Proceedings of MODELS'13 Demonstration Session co-located with the 16th International Conference on Model Driven Engineering Languages and Systems (MODELS 2013), Miami, USA, pages 21- 25, 2013.

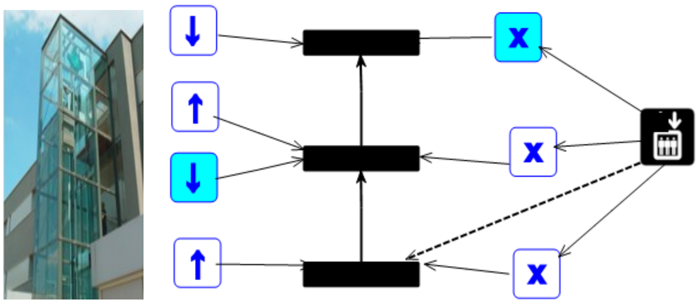
https://www.youtube.com/watch?feature=player_detailpage&v=RYtea2BiQ98


output: > 750
output: > 800
after 30 s.
0 MW
SV1
SV2
WATER LEVEL: 0 mm
PRESSURE: 0 bar
WATER LEVEL: 0 mm
PRESSURE: 0 bar
WP1: 0 U/min
WP2: 0 U/min
WV1
WV2
CP: 0 U/min
pressure: > 400
pressure: > 450
after 30 s.
pressure: > 100
pressure: > 120
water: > 2200
water: > 2400
water: < 1500
water: < 1700
water: > 4500
water: > 5000
water: < 500
water: < 300

Closely Related Languages

Designing Requirements/Property Languages

Design $\models$ Property

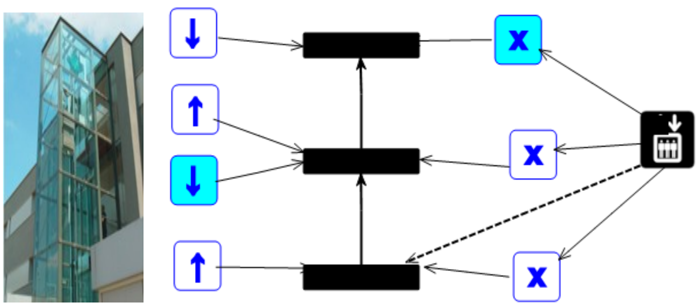


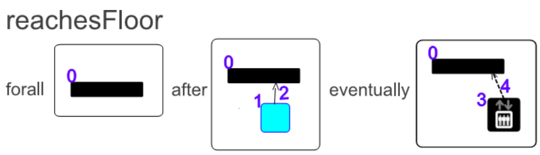
$$
\begin{gathered}
\Box(((go0\wedge up0)\vee\Diamond(floor0\vee idle))\rightarrow((\neg(floor0)\vee\neg(floor0\vee \\
idle))\mathcal{U}((floor0\vee idle)\wedge(((floor0)\vee\neg(floor0\vee idle))\mathcal{U}((floor0\vee \\
idle)\wedge((\neg(floor0)\vee\neg(floor0\vee idle))\mathcal{U}((floor0\vee idle)\wedge \\
(((floor0)\vee\neg(floor0\vee idle))\mathcal{U}((floor0\vee idle)\wedge(\neg(floor0)\mathcal{U}(floor0\vee \\
idle))))))))))\vee\Box(((go1\wedge up1\wedge down1)\vee\Diamond(floor1\vee idle))\rightarrow \\
((\neg(floor1)\vee\neg(floor1\vee idle))\mathcal{U}((floor1\vee idle)\wedge(((floor1)\vee \\
\neg(floor1\vee idle))\mathcal{U}((floor1\vee idle)\wedge((\neg(floor1)\vee\neg(floor1\vee \\
idle))\mathcal{U}((floor1\vee idle)\wedge(((floor1)\vee\neg(floor1\vee idle))\mathcal{U}((floor1\vee \\
idle)\wedge(\neg(floor1)\mathcal{U}(floor1\vee idle))))))))))\vee\Box(((go2\wedge down2)\vee \\
\Diamond(floor2\vee idle))\rightarrow((\neg(floor2)\vee\neg(floor2\vee idle))\mathcal{U}((floor2\vee \\
idle)\wedge(((floor2)\vee\neg(floor2\vee idle))\mathcal{U}((floor2\vee idle)\wedge((\neg(floor2)\vee \\
\neg(floor2\vee idle))\mathcal{U}((floor2\vee idle)\wedge(((floor2)\vee\neg(floor2\vee \\
idle))\mathcal{U}((floor2\vee idle)\wedge(\neg(floor2)\mathcal{U}(floor2\vee idle))))))))))
\end{gathered}
$$

B. Meyers, R. Deshayes, L. Lucio, E. Syriani, H. Vangheluwe, and M. Wimmer. ProMoBox: A Framework for Generating Domain-Specic Property Languages. In Software Language Engineering (SLE), Vasteras, Sweden, LNCS vol. 8706, pp. 1- 20. Springer. September 2014.

Designing Requirements/Property Languages

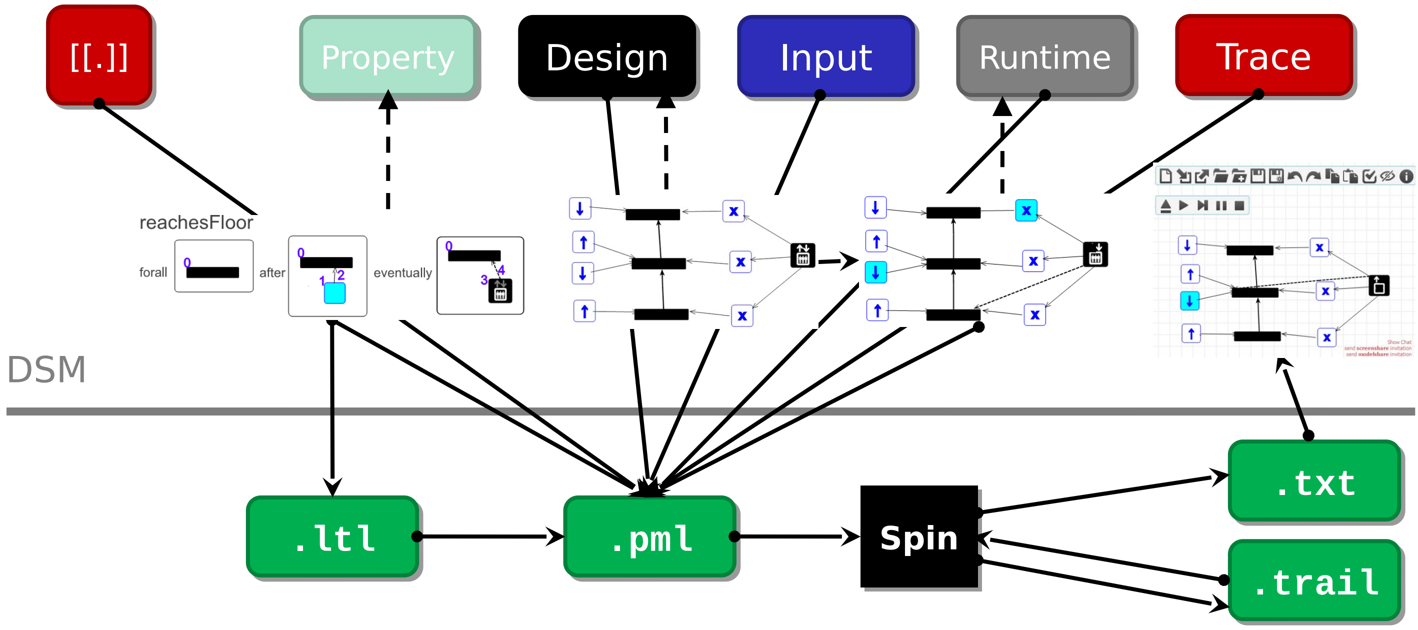
Design ⊨ Property





B. Meyers, R. Deshayes, L. Lucio, E. Syriani, H. Vangheluwe, and M. Wimmer. ProMoBox: A Framework for Generating Domain-Specic Property Languages. In Software Language Engineering (SLE), Vasteras, Sweden, LNCS vol. 8706, pp. 1- 20. Springer. September 2014.

Designing DS Requirements/Property Languages



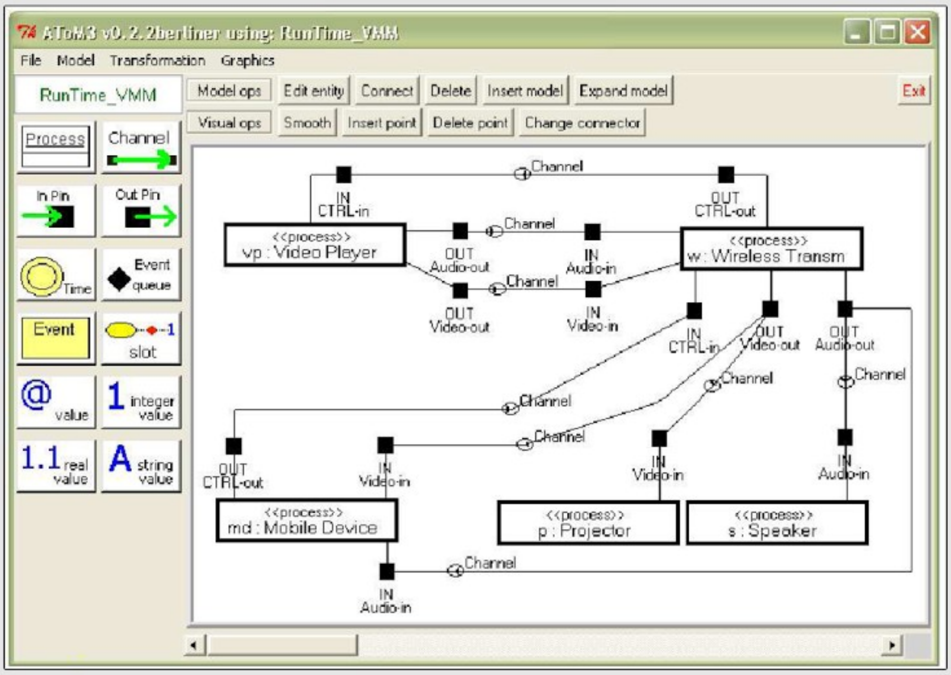
B. Meyers, H. Vangheluwe, J. Denil and R. Salay, "A Framework for Temporal Verification Support in Domain-Specific Modelling," in IEEE Transactions on Software Engineering. doi:10.1109/TSE.2018.2859946

Divide et Impera

Multiple Views

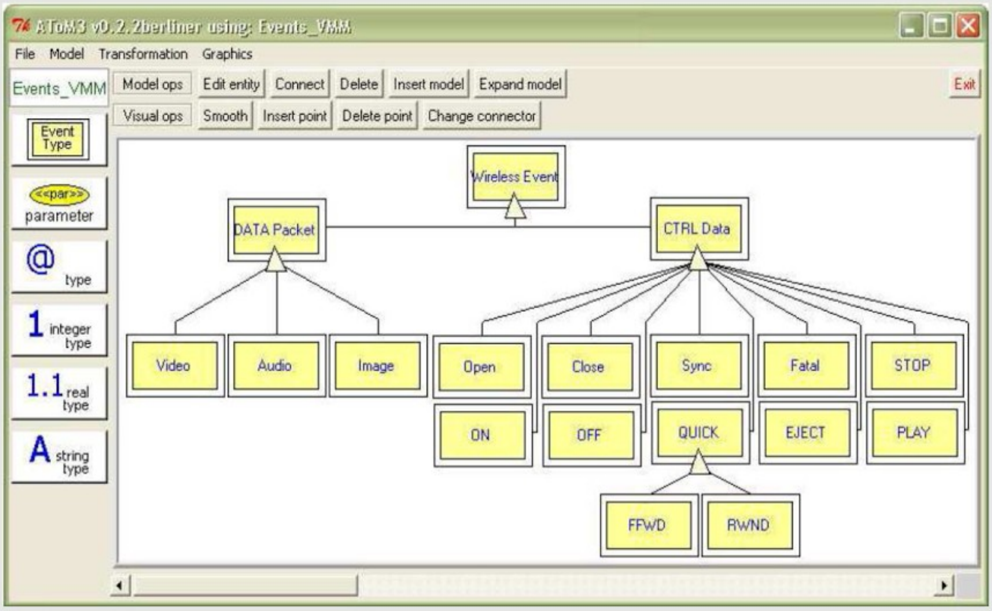
Wireless Home Entertainment System

Multiple (consistent !) Views (in ≠ Formalisms)

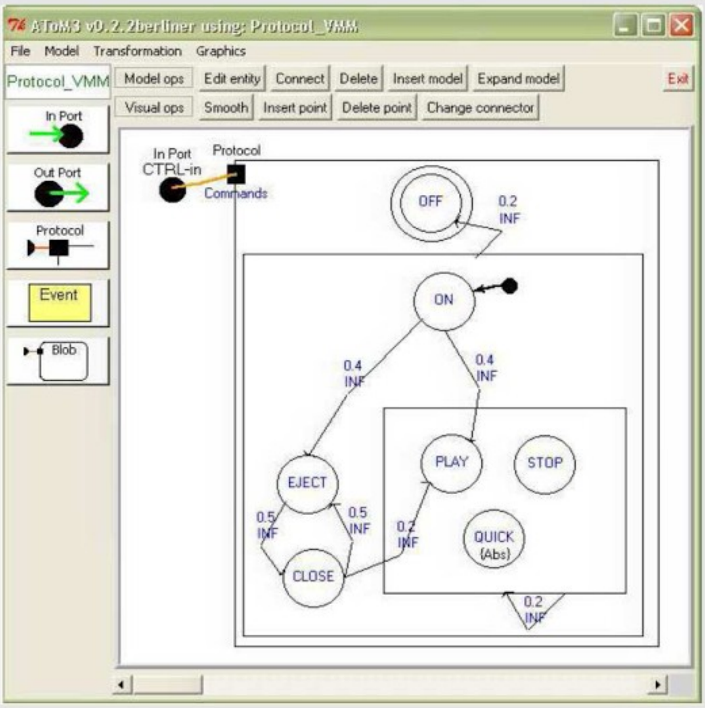


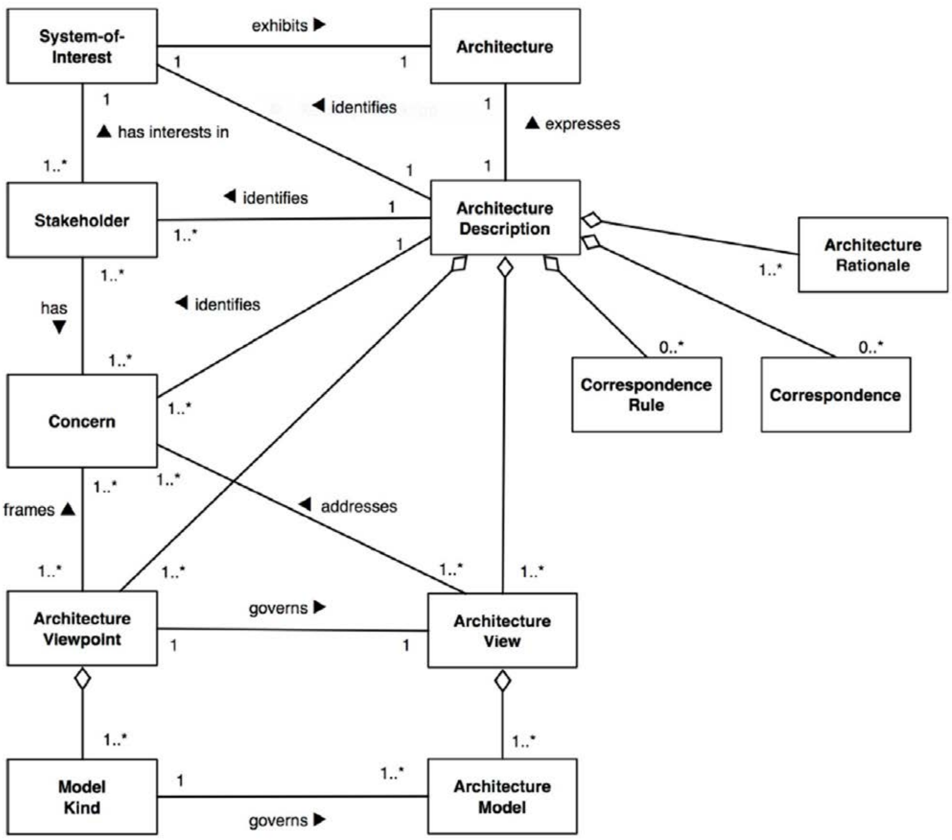
E. Guerra, P. Diaz and J. de Lara, A formal approach to the generation of visual language environments supporting multiple views. 2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05), Dallas, TX, USA, 2005, pp. 284-286, doi: 10.1109/VLHCC.2005.6.

View: Events Diagram



View: Protocol Statechart





INTERNATIONAL STANDARD

ISO/IEC/ IEEE 42010

First edition
2011-12-01

Systems and software engineering — Architecture description

Journal of Computer Languages 76 (2023) 101223

Contents lists available at ScienceDirect

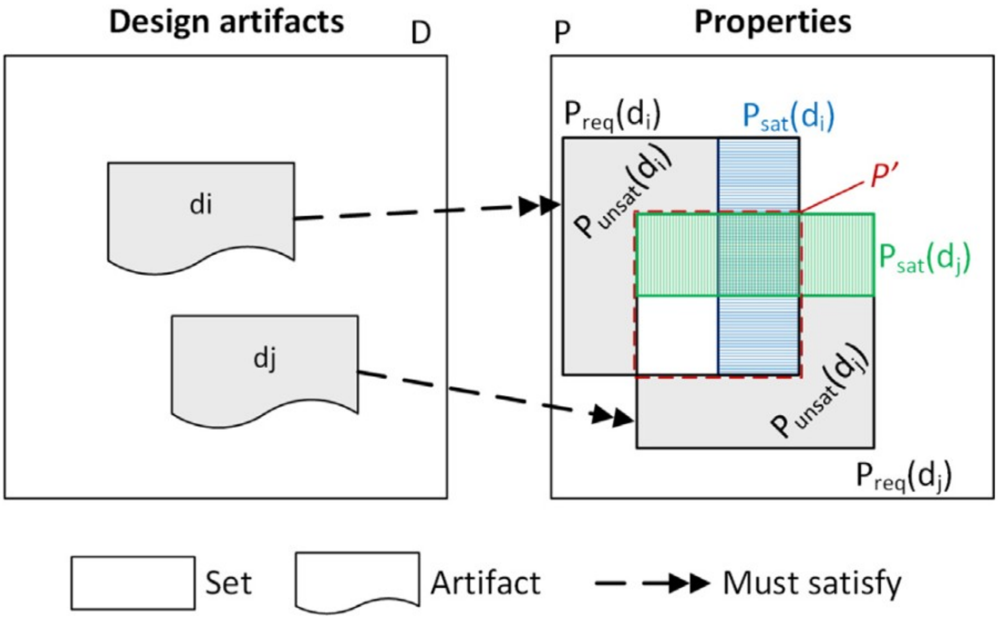
Journal of Computer Languages

journal homepage: www.elsevier.com/locate/cola

Consistency!

Model consistency as a heuristic for eventual correctness

Istvan David [a,*], Hans Vangheluwe [b,c], Eugene Syriani [a]



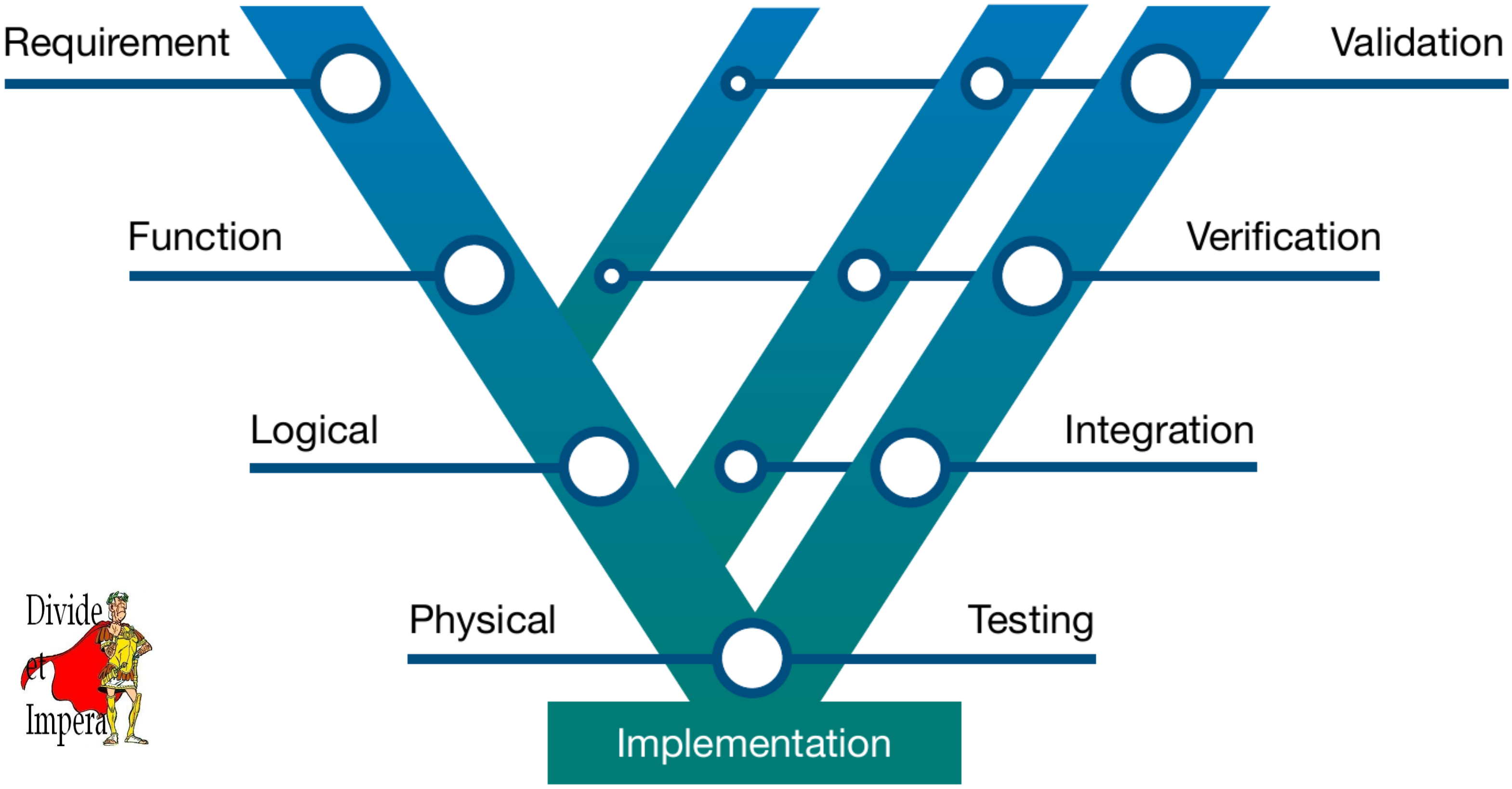
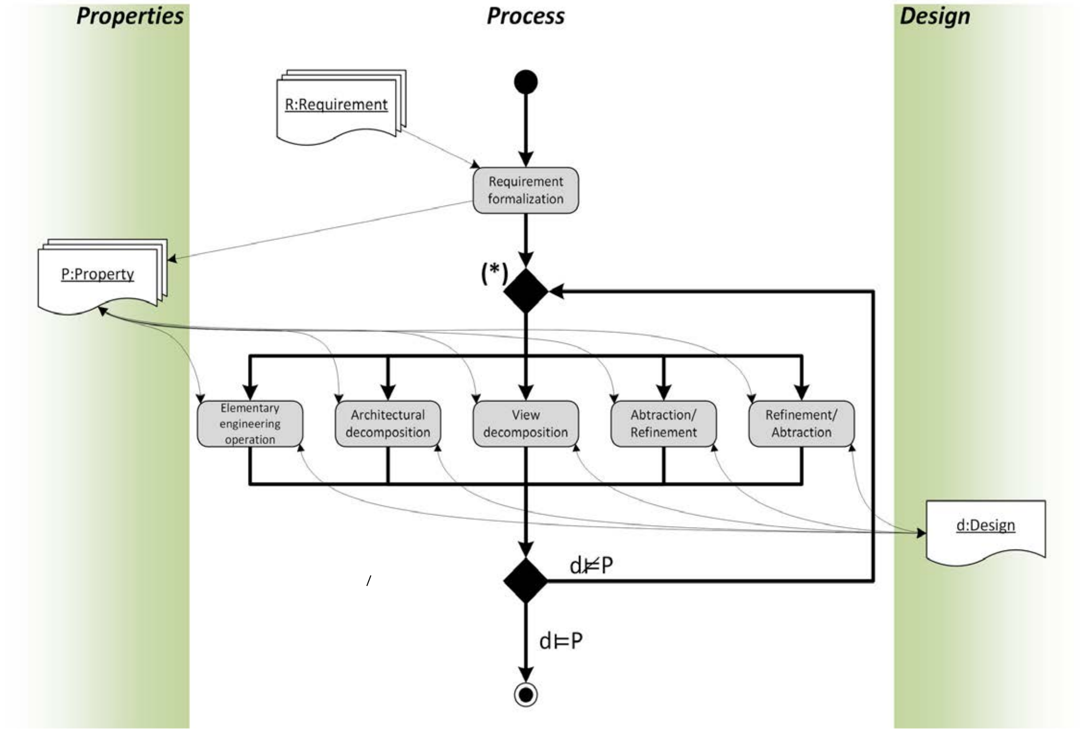
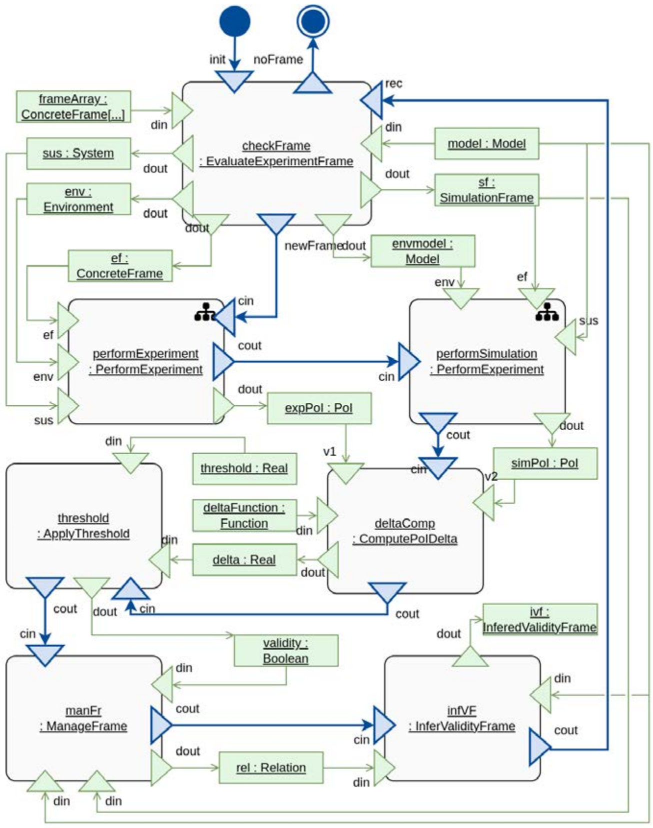


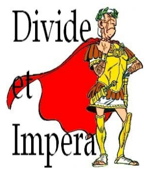
Image courtesy of Cesar Ribeiro dos Santos @ Siemens

Recursive workflow: from Properties to Design

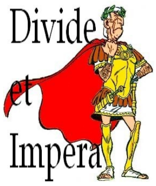


Workflow

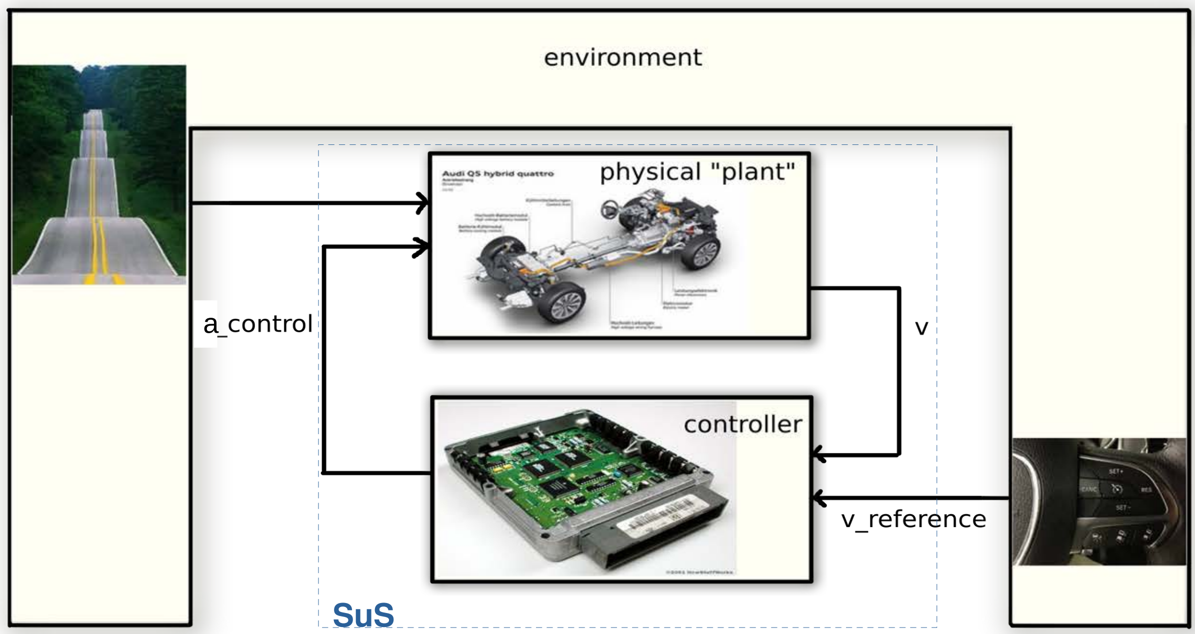




++ enactment

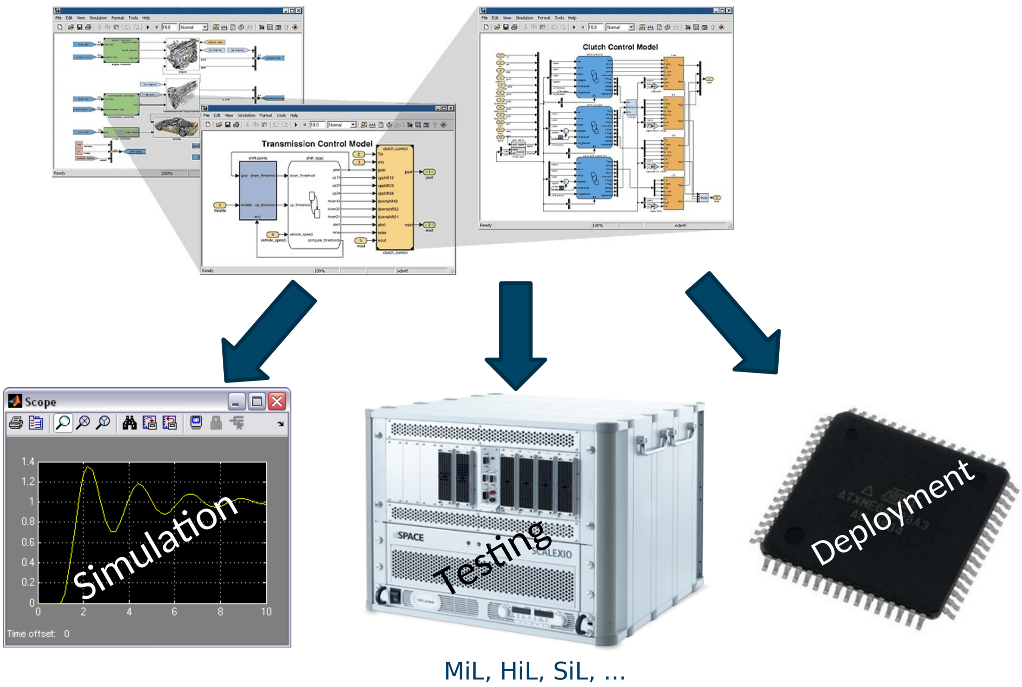


Stepwise realization (deployment for software) of components

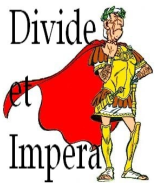


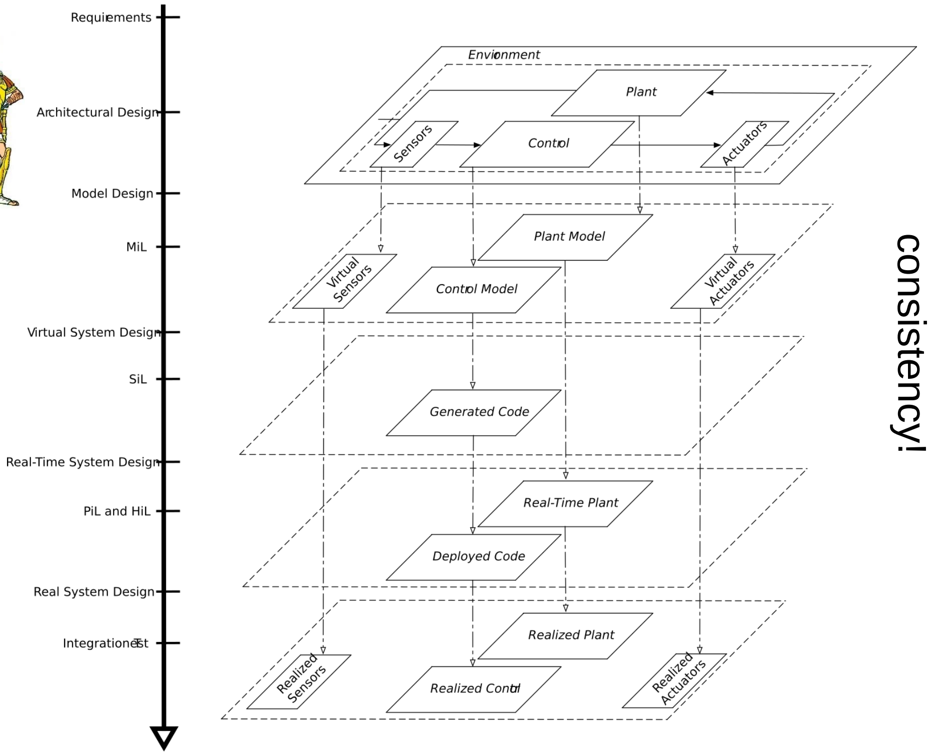
Divide et Impera

Model-Based System Design



XiL: X = Model, Software, Processor, Hardware

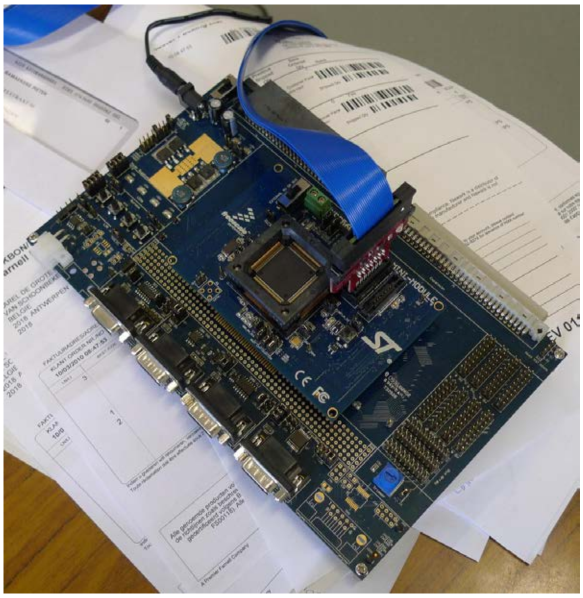




vertical consistency!

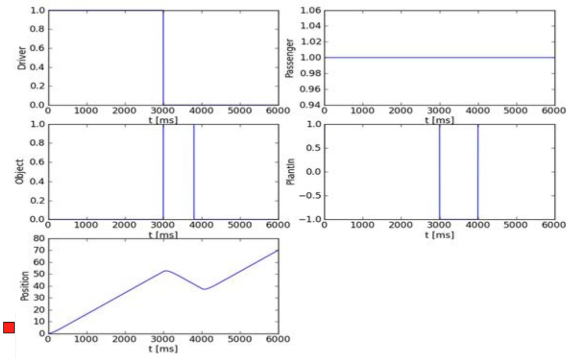
Ken Vanherpen. A contract-based approach for multi-viewpoint consistency in the concurrent design of cyber-physical systems. PhD thesis University of Antwerp. 2018.

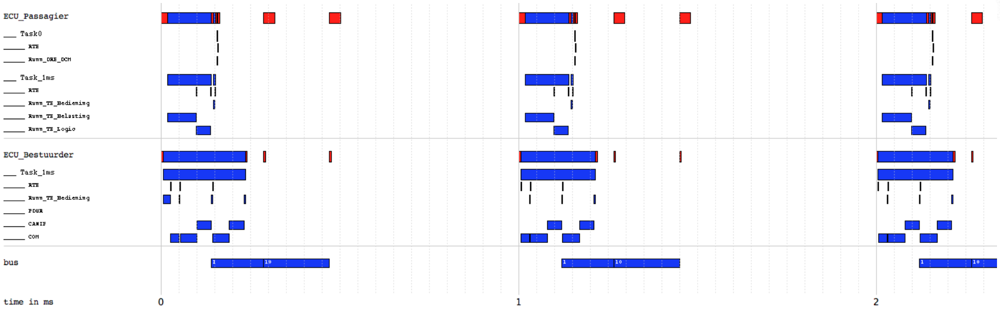
Deployment and Resource-Optimized Execution



Joachim Denil, Paul De Meulenaere, Serge Demeyer, and Hans Vangheluwe. DEVS for AUTOSAR-based system deployment modeling and simulation. SIMULATION: Transactions of the Society for Modeling and Simulation International , 93(6):489 – 513, 2017.

Deployment/Design-Space Exploration (transform to MILP, discrete-event simulation of deployment alternatives)

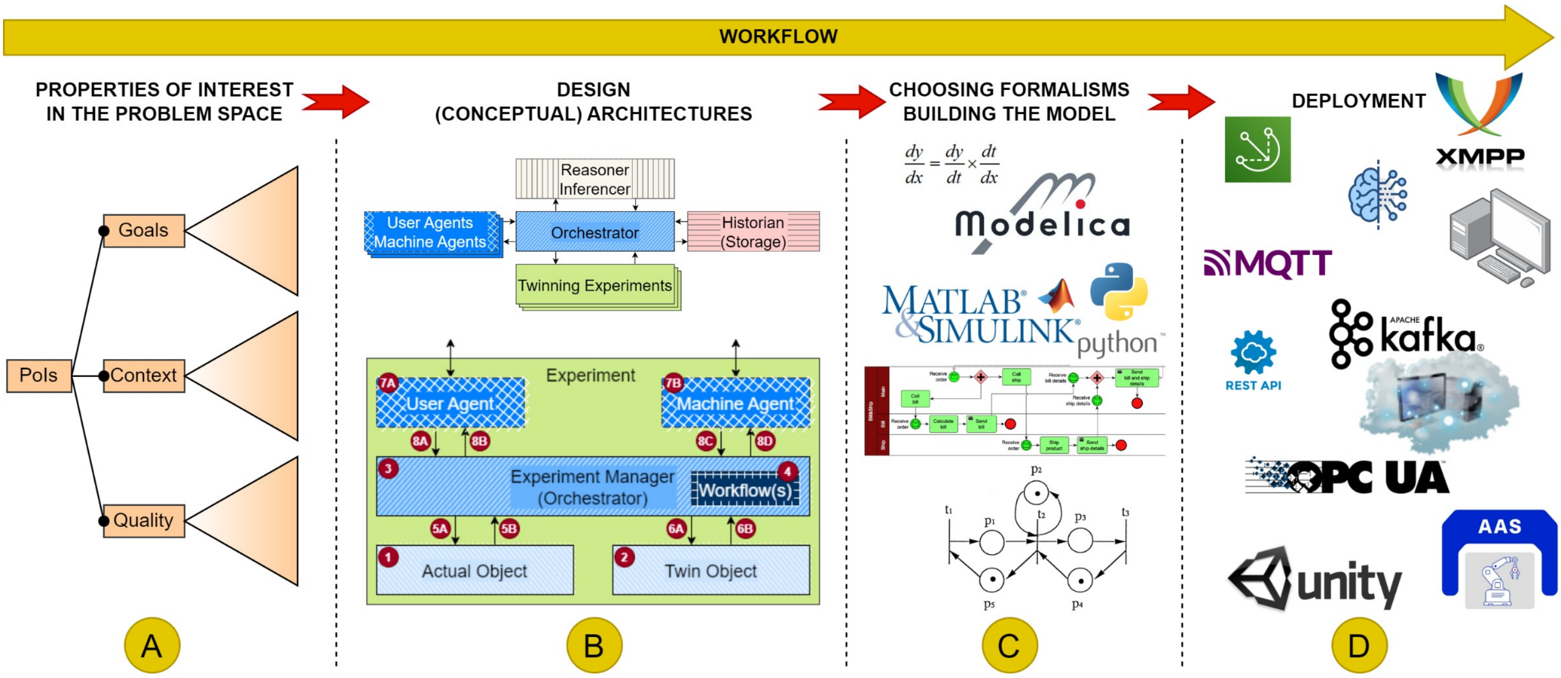




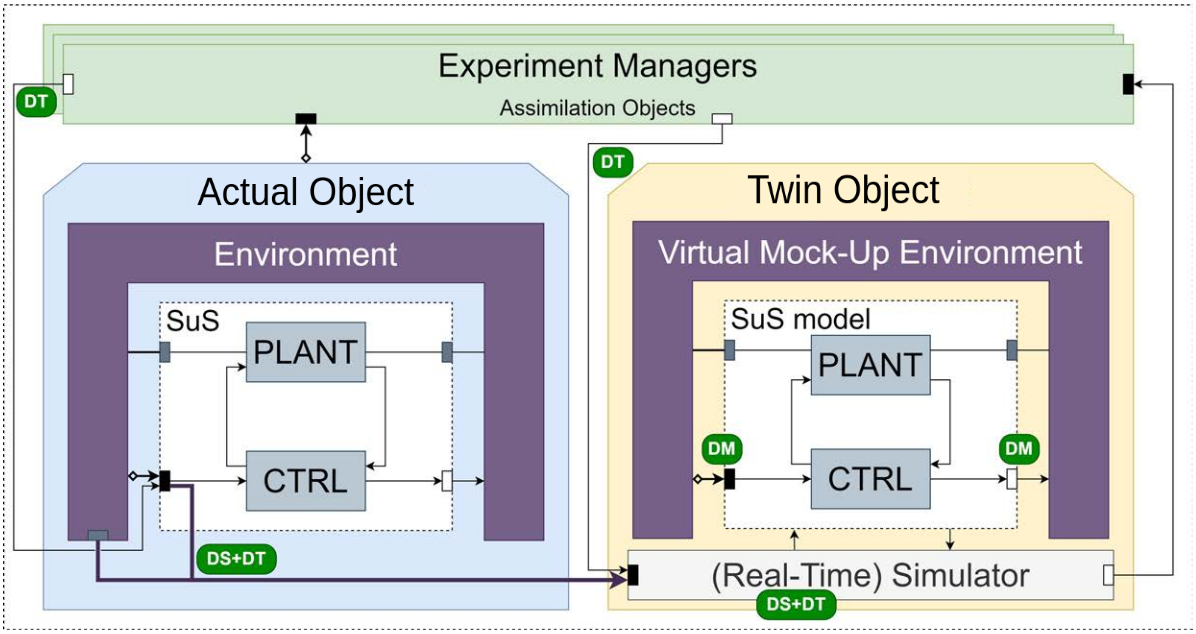
Joachim Denil, Hans Vangheluwe, Pieter Ramaekers, Paul De Meulenaere, and Serge Demeyer. DEVS for AUTOSAR platform modelling. In Spring Simulation Multiconference , pages 67 - 74. Society for Computer Simulation International (SCS), April 2011. Boston, MA, USA.

Foundations for Twinning: workflow

Goals → Conceptual Architecture → Formalism choice/Model building → Deployment

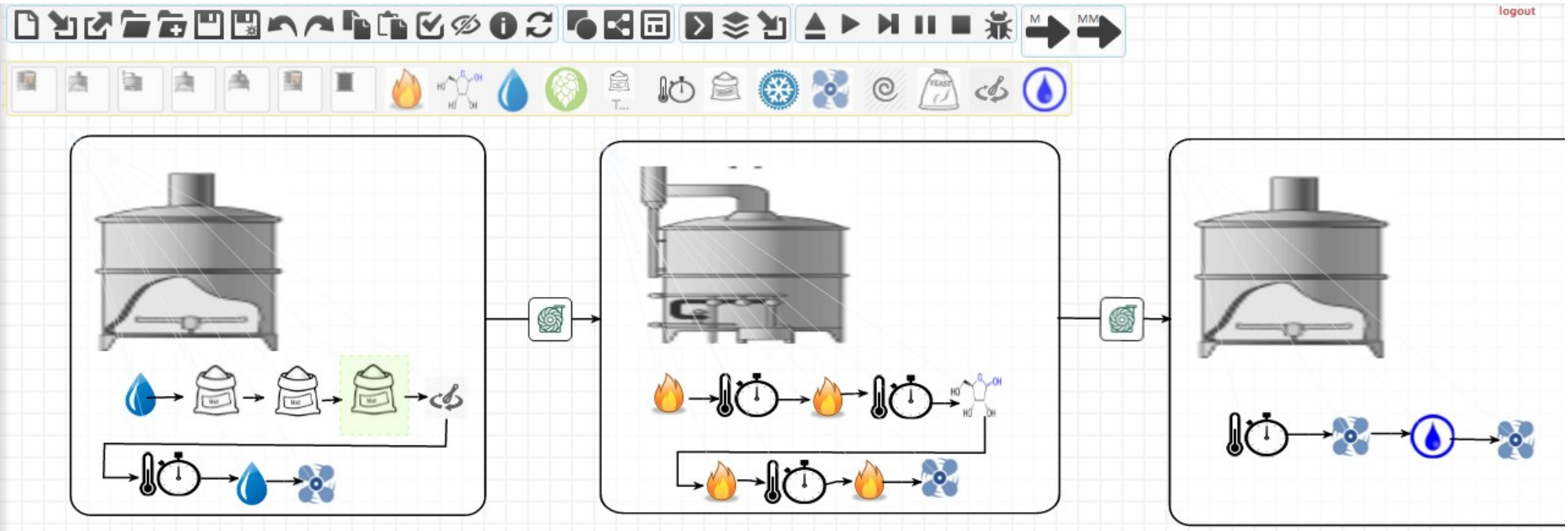


Family of Twinning Architectures



R. Paredis, and H.Vangheluwe. 2022. "*Towards a Digital Z Framework Based on a Family of Architectures and a Virtual Knowledge Graph*". In Proceedings of MoDELS 2022.


MODEL
EVERYTHING!
DSM TP 2014
Theory and Practice
5th International Summer School
on Domain Specific Modeling
Antwerp, Belgium
25 - 29 August
Thomas Kühne



Joachim Denil


MODEL EVERYTHING!
04/04 Bellairs, Barbados
04/05 Bellairs, Barbados
04/06 Bellairs, Barbados
10/06 Genoa, Italy
04/07 Bellairs, Barbados
10/07 Nashville, TN, USA
04/08 Bellairs, Barbados
04/09 Bellairs, Barbados
10/09 Denver, CO, USA
04/10 Bellairs, Barbados
10/10 Oslo, Norway
04/11 Bellairs, Barbados
10/11 Wellington, NZ
04/12 Bellairs, Barbados
10/12 Innsbruck, Austria
05/13 Bellairs, Barbados
09/13 Miami, FL, USA
02/14 Bellairs, Barbados
09/14 Valencia, Spain
01/15 Bellairs, Barbados
09/15 Ottawa, Canada
04/16 Bellairs, Barbados
03/17 Bellairs, Barbados
05/18 Bellairs, Barbados
04/19 Bellairs, Barbados
09/19 Munich, Germany
10/20 Montreal, Canada*
10/21 Fukuoka, Japan*
04/22 Bellairs, Barbados
10/22 Montreal, Canada
03/23 Carghjese, Corsica
05/23 Bellairs, Barbados
10/23 Västerås, Sweden
*virtual event
http://CAMPaM.MPM4CPS.eu

SS-CPSIoT

6th Annual Summer School on
Cyber Physical Systems and Internet of Things

Introducing
Model-based Systems Engineering

Joachim Denil

12JUne 2025

Universiteit Antwerpen · CoSys-Lab Constrained Systems Lab University of Antwerp · Nexor Cyber-Physical Systems University of Antwerp · FLANDERS MAKE MANUFACTURING INNOVATION NETWORK · MPM4CPS · cost

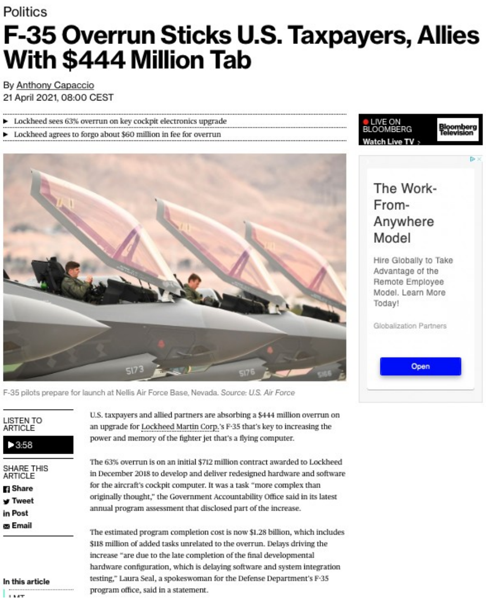
Goals of this lecture

- Why do we need Model-based Systems Engineering (MBSE)?
- How does MBSE work by an example…

Complex Systems

- You cannot build these systems alone…
  - Mechanical engineering
  - Software Engineering
  - Control Engineering
  - Etc.
- Lots of parties are involved:
  - Engineers
  - Stake-holders
  - Project Management
  - Government
  - Etc.

Designing such systems carries risk



Politics

F-35 Overrun Sticks U.S. Taxpayers, Allies With $444 Million Tab

By Anthony Capaccio
21 April 2021, 08:00 CEST

- Lockheed sees 63% overrun on key cockpit electronics upgrade
- Lockheed agrees to forgo about $60 million in fee for overrun

F-35 pilots prepare for launch at Nellis Air Force Base, Nevada. *Source: U.S. Air Force*

LISTEN TO ARTICLE
3:58

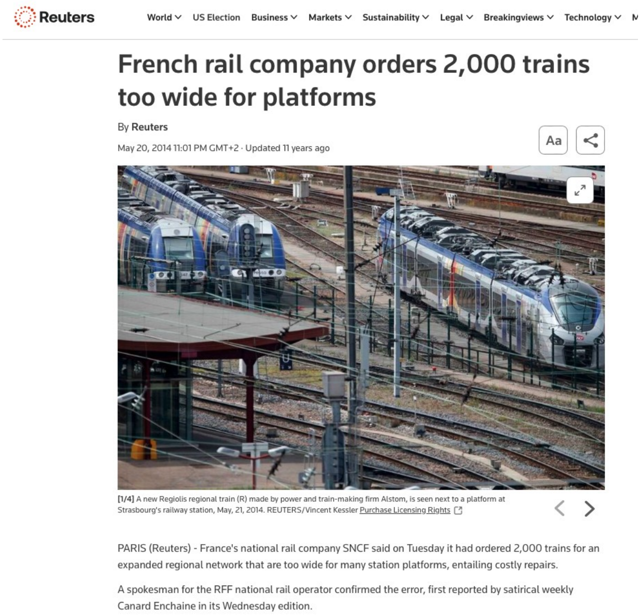
SHARE THIS ARTICLE
Share
Tweet
Post
Email

In this article

U.S. taxpayers and allied partners are absorbing a $444 million overrun on an upgrade for Lockheed Martin Corp.'s F-35 that's key to increasing the power and memory of the fighter jet that's a flying computer.

The 63% overrun is on an initial $712 million contract awarded to Lockheed in December 2018 to develop and deliver redesigned hardware and software for the aircraft's cockpit computer. It was a task "more complex than originally thought," the Government Accountability Office said in its latest annual program assessment that disclosed part of the increase.

The estimated program completion cost is now $1.28 billion, which includes $118 million of added tasks unrelated to the overrun. Delays driving the increase "are due to the late completion of the final developmental hardware configuration, which is delaying software and system integration testing," Laura Seal, a spokeswoman for the Defense Department's F-35 program office, said in a statement.



Reuters
World | US Election | Business | Markets | Sustainability | Legal | Breakingviews | Technology

French rail company orders 2,000 trains too wide for platforms

By Reuters
May 20, 2014 11:01 PM GMT+2 · Updated 11 years ago

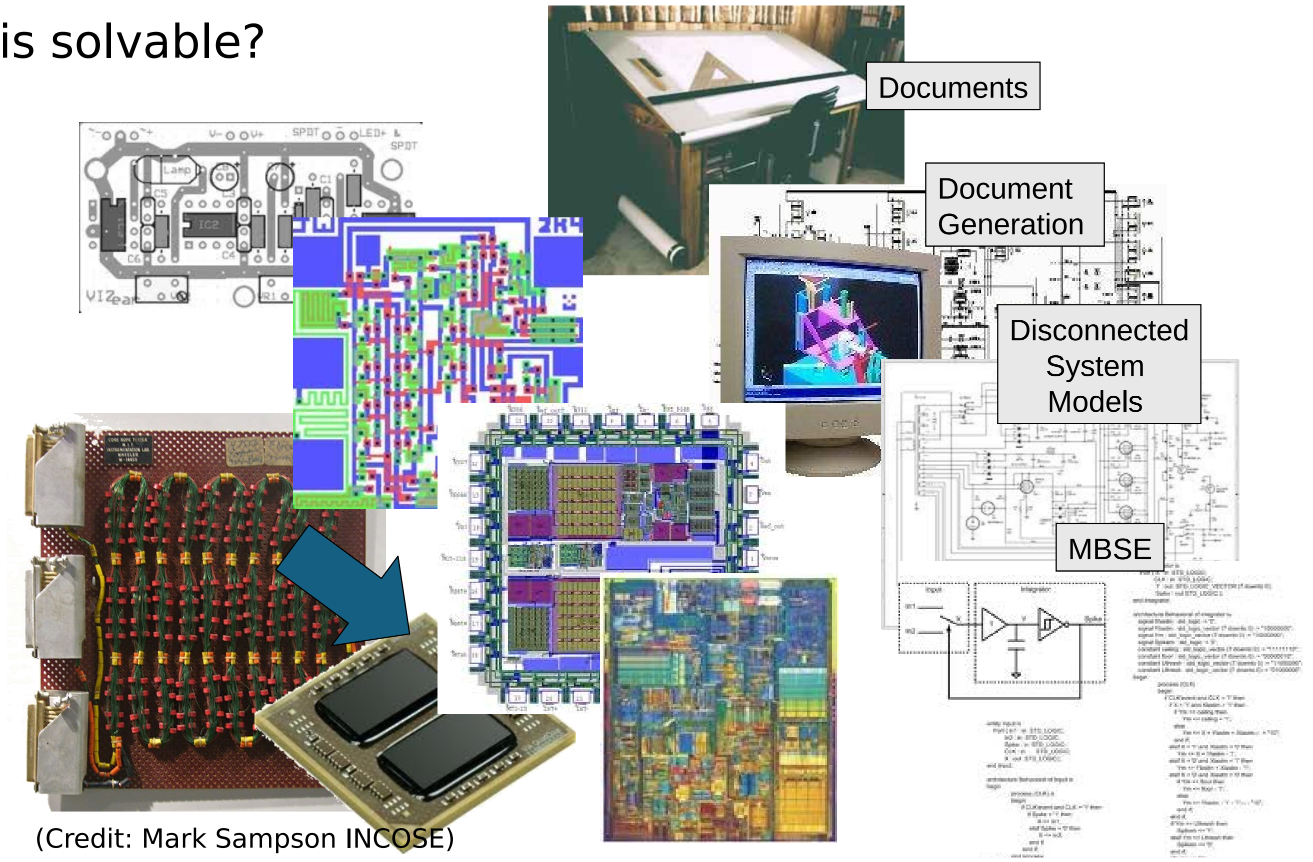
[1/4] A new Regiolis regional train (R) made by power and train-making firm Alstom, is seen next to a platform at Strasbourg's railway station, May, 21, 2014. REUTERS/Vincent Kessler Purchase Licensing Rights

PARIS (Reuters) - France's national rail company SNCF said on Tuesday it had ordered 2,000 trains for an expanded regional network that are too wide for many station platforms, entailing costly repairs.

A spokesman for the RFF national rail operator confirmed the error, first reported by satirical weekly Canard Enchaine in its Wednesday edition.

4

Is this solvable?



(Credit: Mark Sampson INCOSE)

System Engineering

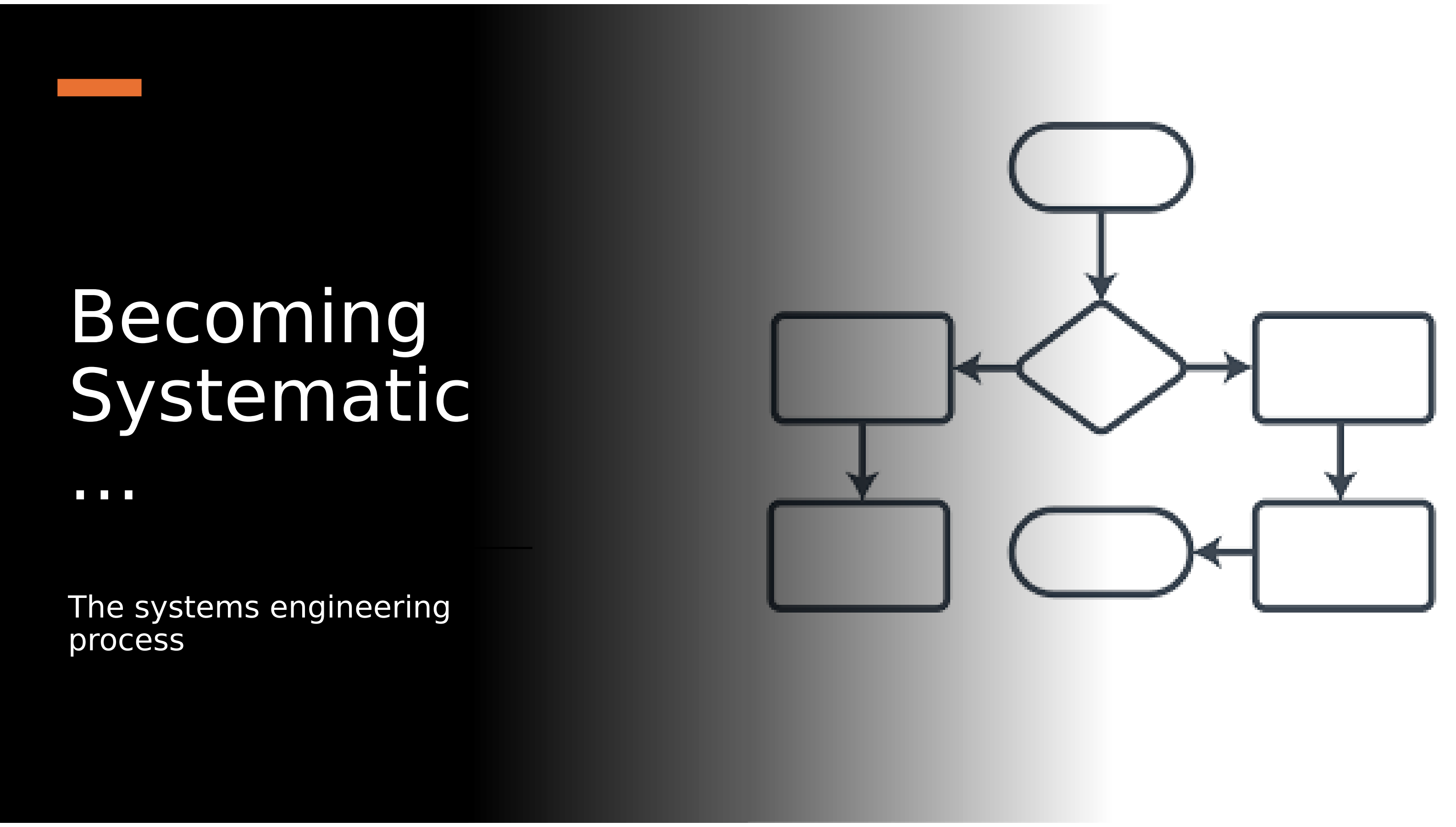
"The multi-disciplined application of analytical, mathematical and scientific principles for the formulation, selection, development and maturing of an optimised solution from a set of viable candidates that has acceptable risk, satisfies the user's operational need(s) and minimises development and life-cycle cost while balancing stakeholders need."

Wasson – Systems Engineering, Wiley

Design and Optimisation are needed

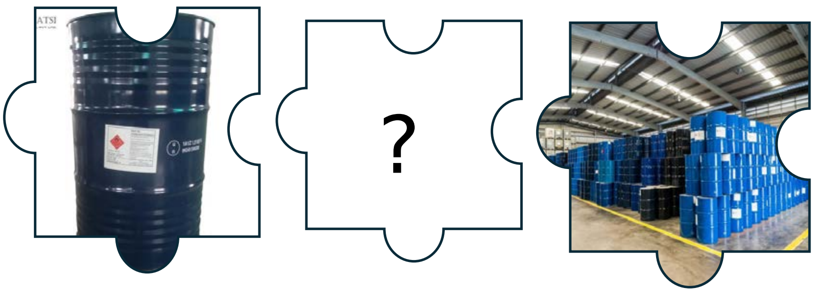
Yes, so we first need to understand what a user wants…

Trade-offs (e.g., you cannot achieve something without any risk)


Becoming Systematic
…
The systems engineering process

We try to reduce risk!

By Being System



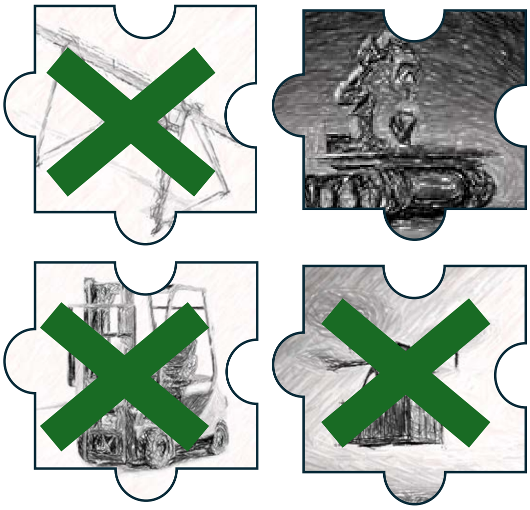
Try and capture the needs pro

Requirement	Specification
payload	500 - 700 kg per barrel
Navigation precision	+- 2 cm
Operational shifts	12-hour
Flexibility for extension	Range + Throughput
Cost (optimize)	< 125 k
Unloading / Loading Time	< 30 min / truck (standard size of 12 m)

8

We try to reduce risk!

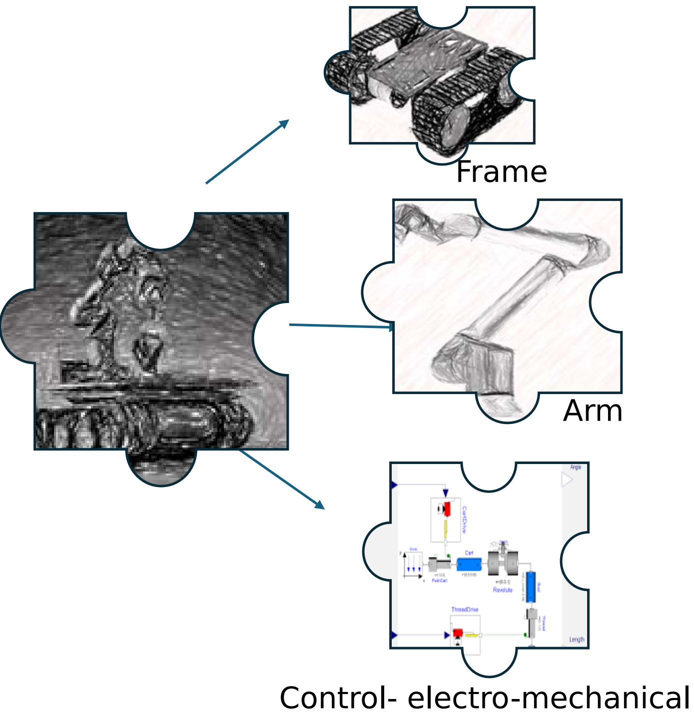
Which concepts supports this?



Requirement	Gantry concept	Forklift concept	UAV concept	Robotic Platform Platoon
Weight	< 5 ton	<3 ton	< 50 kg	< 1000 kg
Precision	< 5 cm	< 10 cm	<25 cm	< 2 cm
Operational shifts	hardwired	Battery swaps	Battery Swaps	Battery Swaps
Flexibity on extension	Fixed	Number + range	Number + Range	Number + Range
Cost	200 k	150 k	100 k	100 k
…				

9

We try to reduce risk!



Decompose (possibly multiple

Decompose:

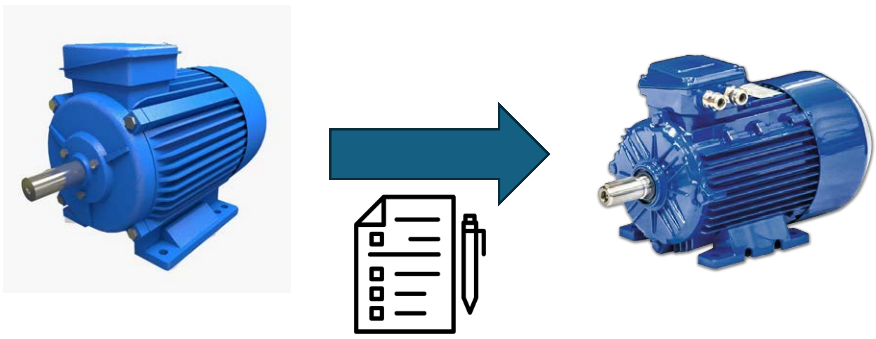
- Views (geometry, safety, electronic, mechanical, control, etc.)
- Components: (Arm, platform, Motors, etc.)
- …

Tyranny of the dominant decompositi

10

We try to reduce risk!

Buy / Design decision



Contracts over the specification of the component

11

We try to reduce risk!

Test individual components.

Integrate and test sub-systems

Are these correct
with respect to the specificatio

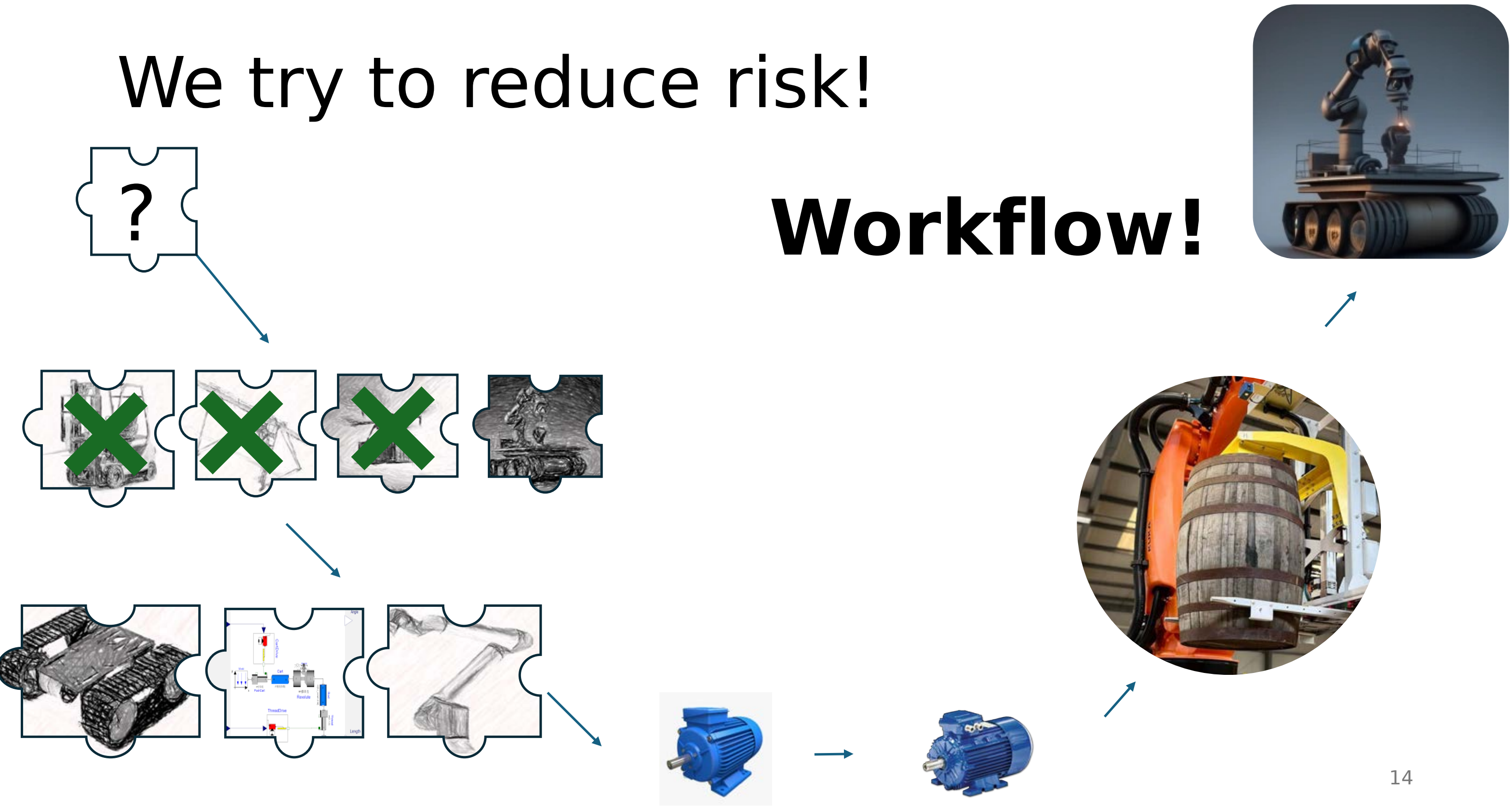
12

We try to reduce risk!

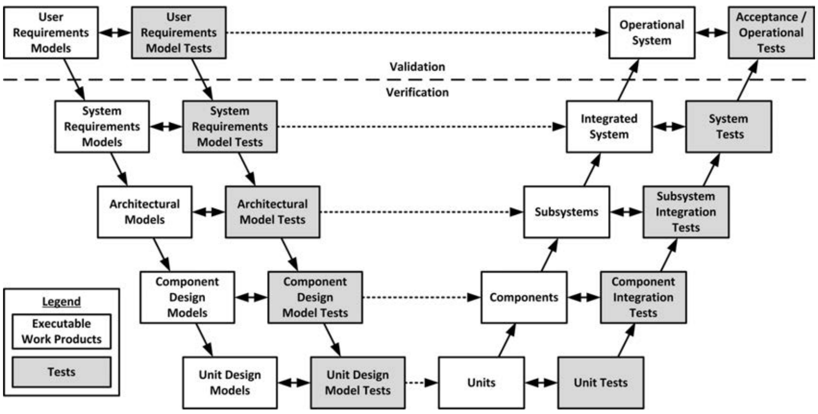
Integrate and test system.

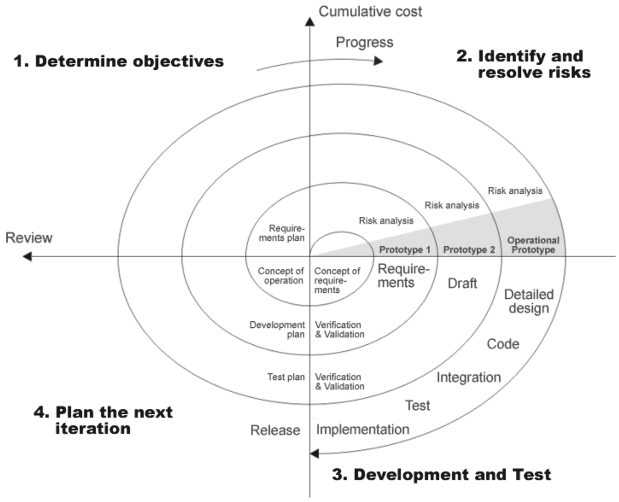
Is this what the customer wanted? (Validate)

13


We try to reduce risk!
?
Workflow!
14

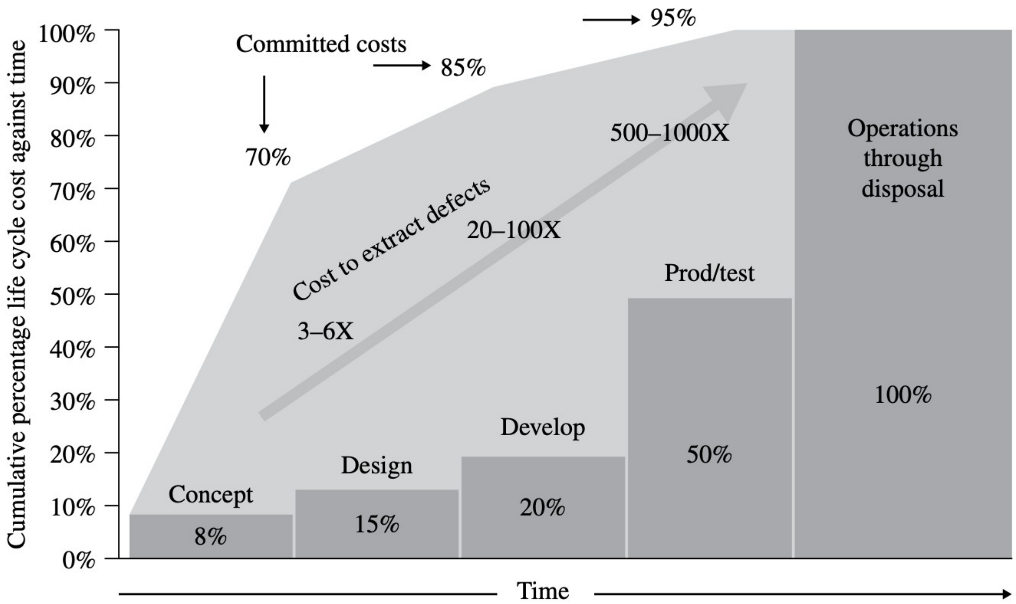
Many possible approaches to "workflow"





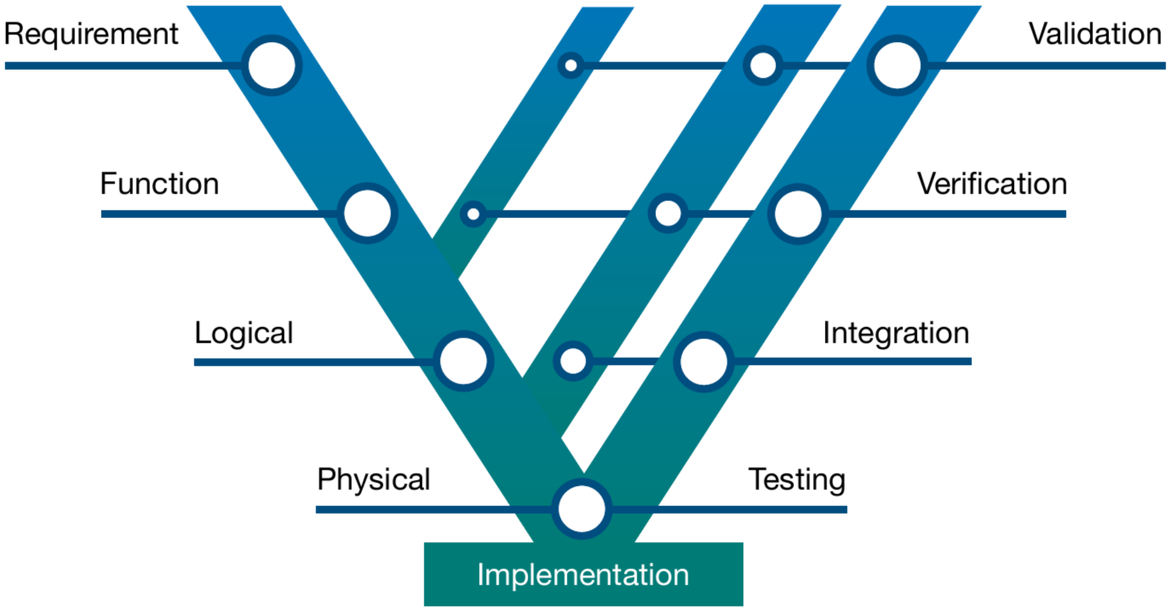
Boehm B, "A Spiral Model of Software Development and Enhancement", IEEE Computer, IEEE, 21(5):61-72, May 1988
Kevin Forsberg and Harold Mooz, "The Relationship of System Engineering to the Project Cycle," in Proceedings of the First Annual Symposium of National Council on System Engineering, October 1991: 57–65.

Life-cycle cost allocation



From:

Modelling and Simulation helps by 'shifting left'



From: Cesar Ribeiro dos Santos

Modelling Systems

Modelling Systems

- Different 'Views', though a single model
  - Structure (Form)
  - Behaviour
  - Functions
  - Requirements
  - Validation and Verification
- Standards
  - Safety (requires traces between elements)
- Allow for trade-off studies (Design-Space Exploration)

Decomposition and Architecture

decompose

integrate

Observing 'structure' of systems

- Platform Subsystem:
  - Chassis
  - Gearbox
  - 2 motors
  - Tracks
  - Track Guiders
- Arm subsystem
  - 4 motors
  - Gear and Linkages
- EE subsystem

Not the only way to decompose a system: Function

- Navigating environment
  - Calculate path
  - Set speed and direction for tracks
  - Execute motion
  - Perceive environment
  - …
- Manipulate Objects
  - Plan arm movement
  - Grasp and release
  - …
- ….

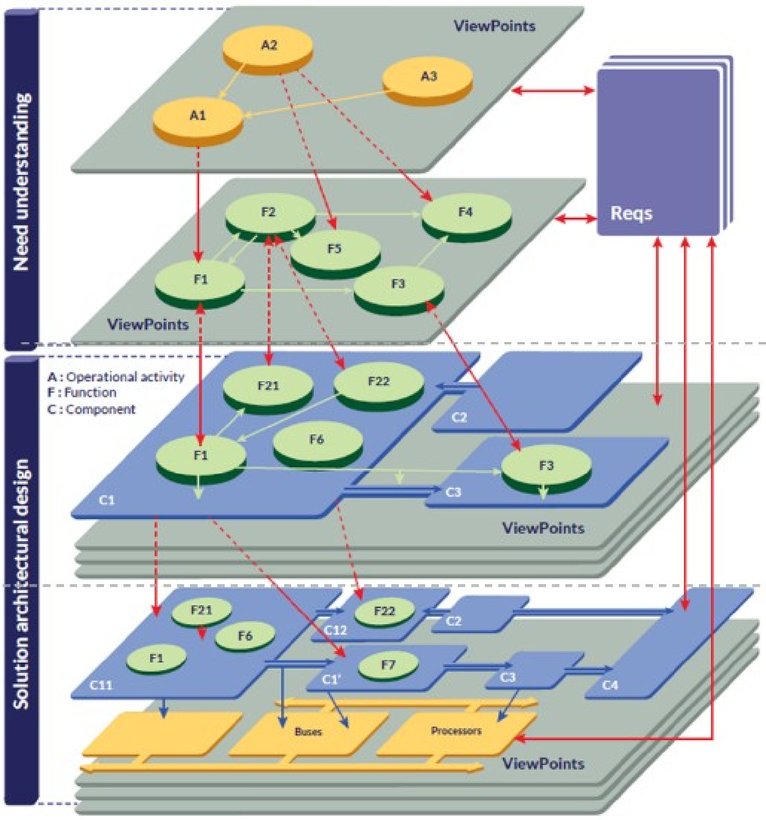
e intuitive to go from requirements to functions to structure to components

Every system (and hierarchical entity)

- Has:
  - Requirements
  - Function
  - Behaviour
  - Physical
- Example: Track motor
  - Function: Convert electrical energy into controlled mechanical energy to move the robot platform, payload and arm
  - Requirements: Provide 30 kW of power output, variable speed control -160 meters per minute, full speed at 3.5 seconds, 40 duty cycles per hour, fail-safe braking within 0.5 seconds of power loss, etc.
  - Behaviour: Idle/standby -> acceleration ->constant speed ->…
  - Physical: Squirrel cage motor design 600 mm * 450 mm * 400 mm; etc.

Different methods and tools exist

- Difference between tool and method
- Methods:
  - Object-Oriented Systems Engineering Method (OOSEM)
    - INCOSE standard
    - Whole lifecycle SE method
    - SysML language supports the method
    - Different commercial tools available
  - Arcadia:
    - Architecture Analysis & Design Integrated Approach
    - Thales (French Engineering Company)
    - Capella language and tool supports the method
    - Focus is on architecture (not whole lifecycle)
  - DODAF
  - MITRE
  - Etc.



Operational Need Analysis

What the users of the system need to accomplish

System Need Analysis

What the system has to accomplish for the users

Logical Architecture (*conceptual solution*)

How the system will work in order to fulfil expectations

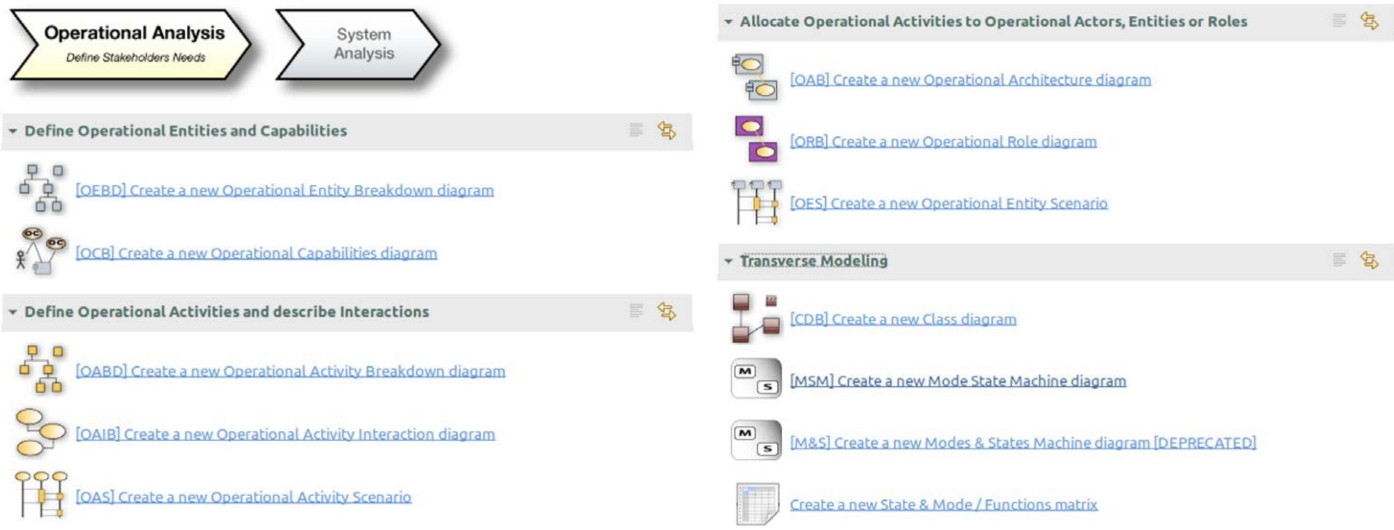
Physical Architecture (finalized solution)

How the system be developed and built

25

From: https://mbse-capella.org/arcadia.html

Operational Need Breakdown



26

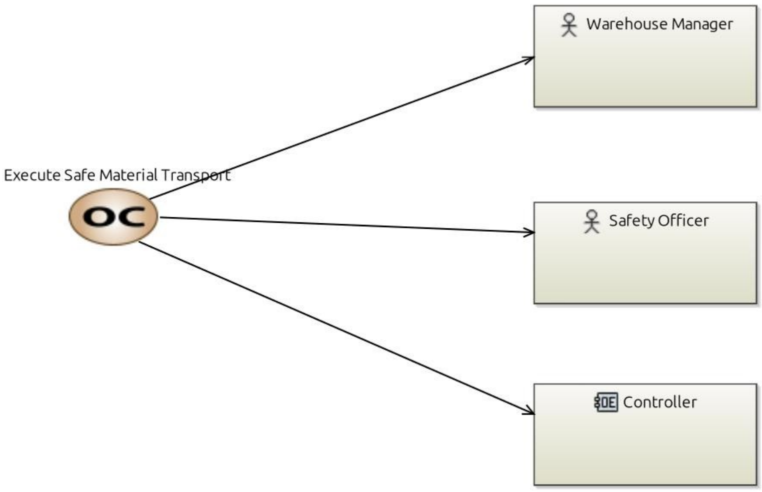
Operational Analysis?

- Understand the stakeholders problem:
  - what are the capabilities that the user expects?
    - Operational **Capabilities**: Capability of an organisation to provide a high-level service leading to the operational objective being reached
    - Operational **Entity**: Entity in the real-world (organisation, excisting system, etc.) whose role is to interact with the system or its users
    - Operational **Actor**: particular entity: human, non-decomposable
- Operational analysis should ideally not mention the system!
  - It could bar potential alternatives.
  - For example: Customer need = hang something on the wall
  - Potential idea would be a pin on the wall with a drill
  - But this excludes, e.g., glue
  - Operational capability = fastening on a specific location on the wall

27

Operational Capabilities Diagram (OCB)

Diagram



Information

- Purpose: Business modelling. Independent of future system. Not focus on the system but on the domain (what are we trying to achieve?)
- Result: What is the operational focus of the business (or higher level system), and how can we bring value?

28

Operational Analysis

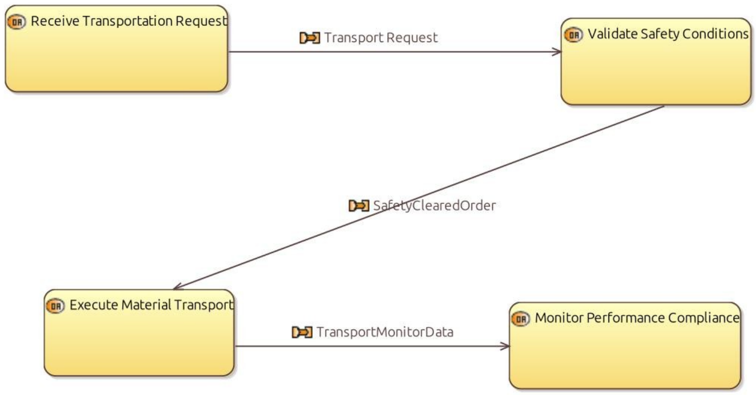
- "What does the user of the future system need to accomplish?"
  - Operational Analysis Terminology:
    - Operational **Activity**: Process steps carried out in order to reach a precise objective by an entity, in which the user might need the future system
    - Operational **Interaction:** exchange of information or mass or energy between operational entities
    - Operational **Process:** Series of activities and interactions that contribute to an operational capability
    - Operational **Scenario:** scenario that describes the behaviour of entities and activities in the context of an operational capability

P. Roques, Systems Architecture Modeling with the Arcadia Method, 2017

29

Operational Activity Interaction Blank

Focus on Data Flow

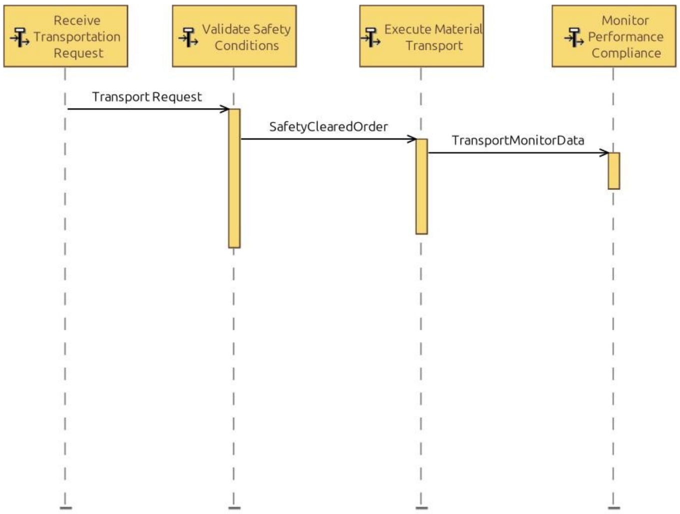


- Purpose: Which activities interact with each other in order to have the capability it describes? Do not consider any allocation to an entity or to an actor!
- Data Flow (not Control Flow)
- Result: Activities to enable capability (black box!)

30

Sequence diagrams

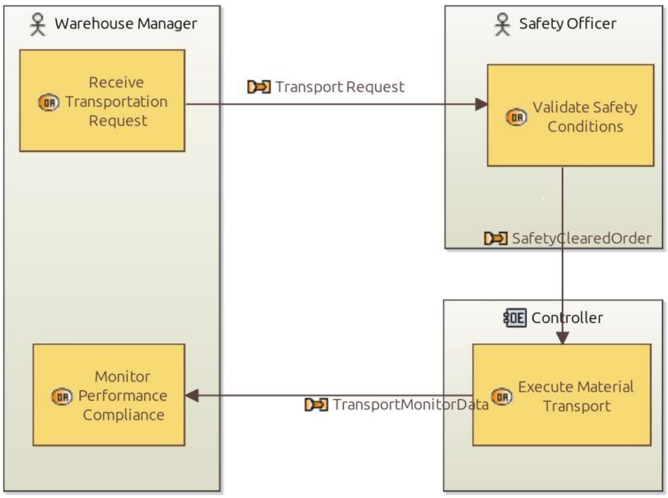
- Behaviour of the system
- Useful for
  - Visualising the flow of 'messages' between objects
  - Scenario modelling
  - Depicting the order of operations in a process
  - Showing how different parts of a system collaborate
  - Identifying potential bottlenecks or inefficiencies in a process
- Benefits include:
  - Explicit representation of complex interactions
  - Easy to understand, even for non-technical stakeholders
  - Useful for identifying and resolving issues in system design



31

Operational Architecture Diagram (OAD)

Diagram

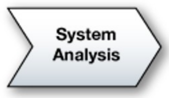


Information

- Purpose: The architecture of the operational decomposition of the system. View on the domain of your system.
- Result: Operational Architecture

32

System Analysis



Formalize System Requirements

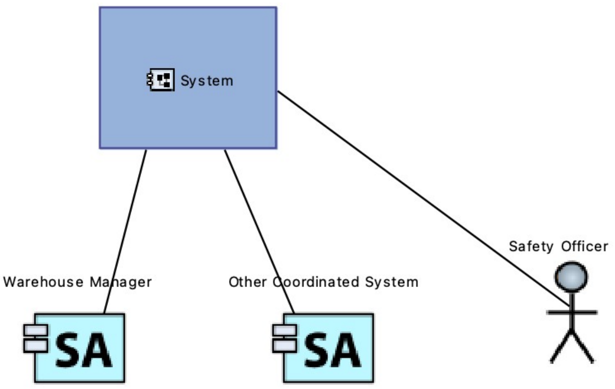
Identify the boundary of the system, consolidate requirements
Define what the system has to accomplish for the users
Model functional dataflows and dynamic behaviour

- Focus on the function of the system we are creating for the user
- Describes: What is the system we need?
- Architecture shows where are the boundaries of the system
- How do the users/actors interact with the system
- How?
  - Functions required by the system (functional requirements)
  - What is the system vs what do actors or other systems do?
  - Functional exchanges with system?
- Trade-offs: Boundaries? Capabilities?
- Need to be extended with textual requirements

33

System Context

- What is the context of the system?
- Boundaries? What is included what not?
- Who is involved in the system? Who interacts with the system?
- Trade-offs!

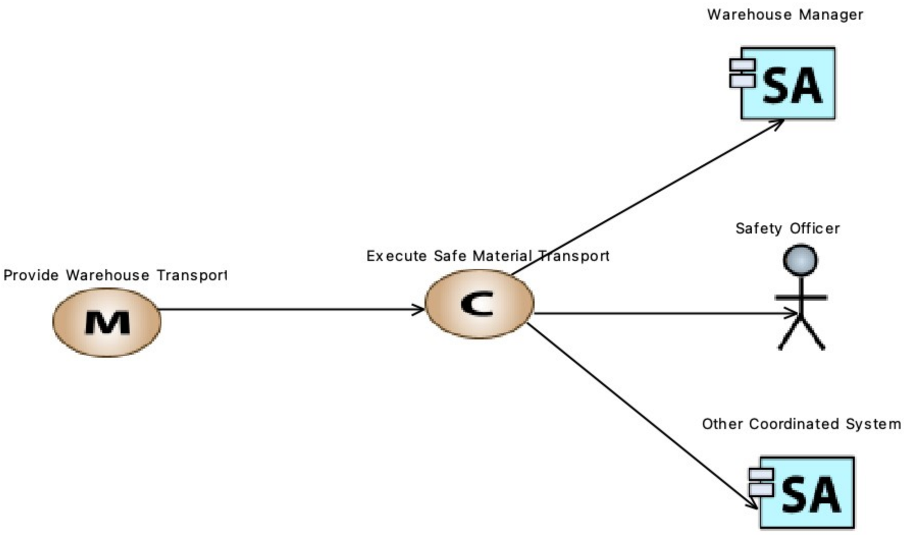


Some operational entities are now replaced by the System

34

Mission Capabilities Blank Diagram

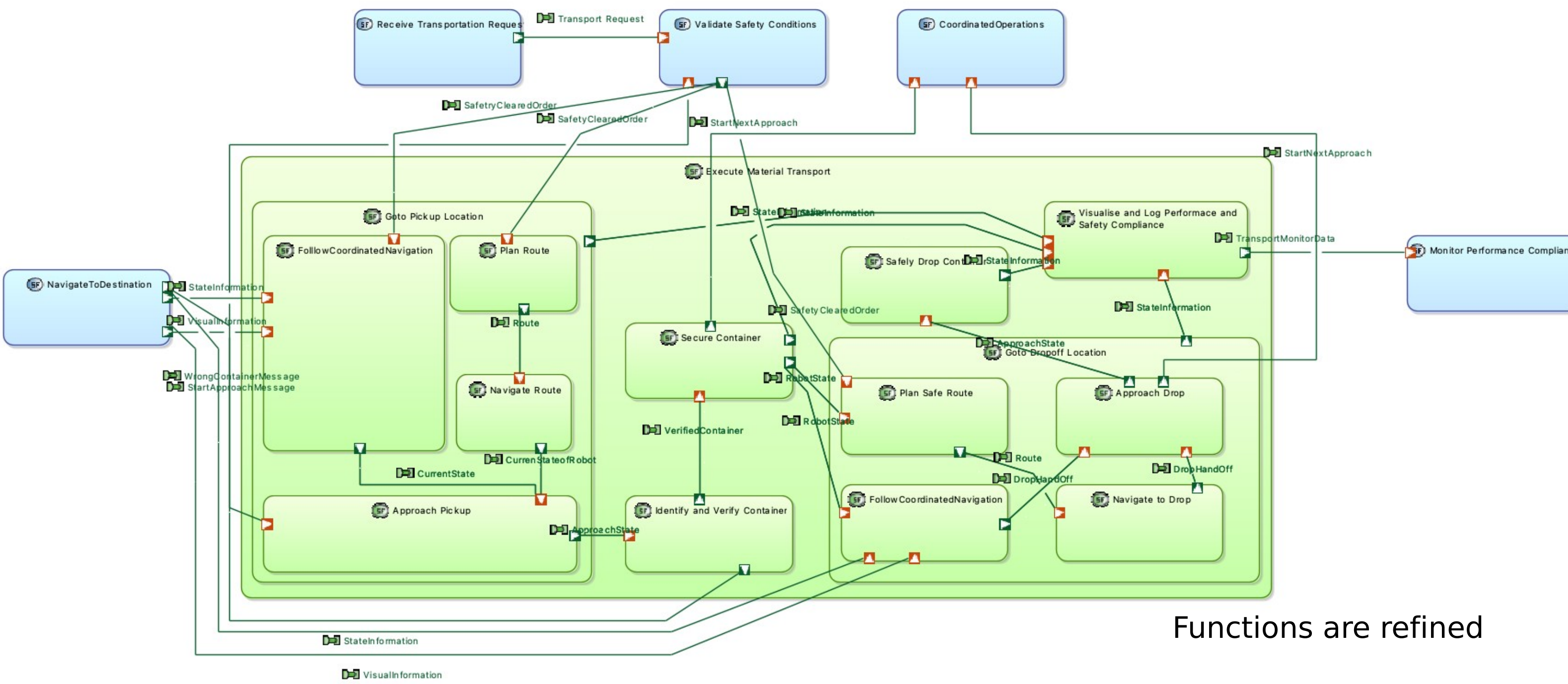
- Identification of System and capabilities with the actors involved.
- What is the mission? (Why?)
- What are the capabilities of the System itself? (What?)
- Trade-offs!



35

System Data Flow Blank Diagram / decompose functions

Functions are appended with requirements



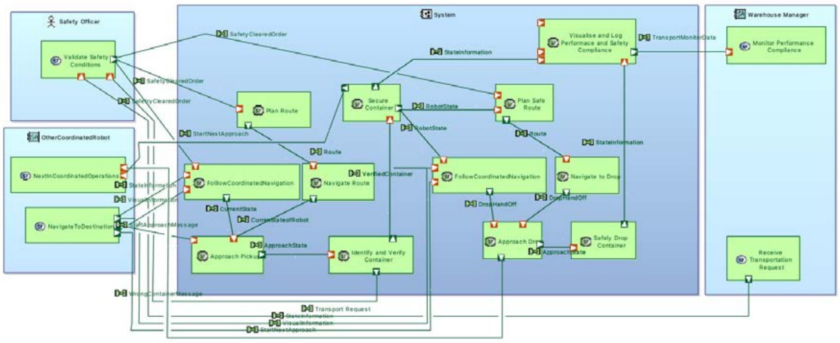
Functions are refined

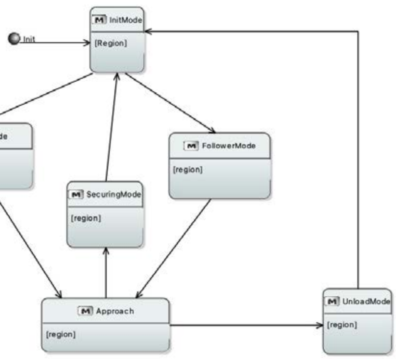
System Analysis result

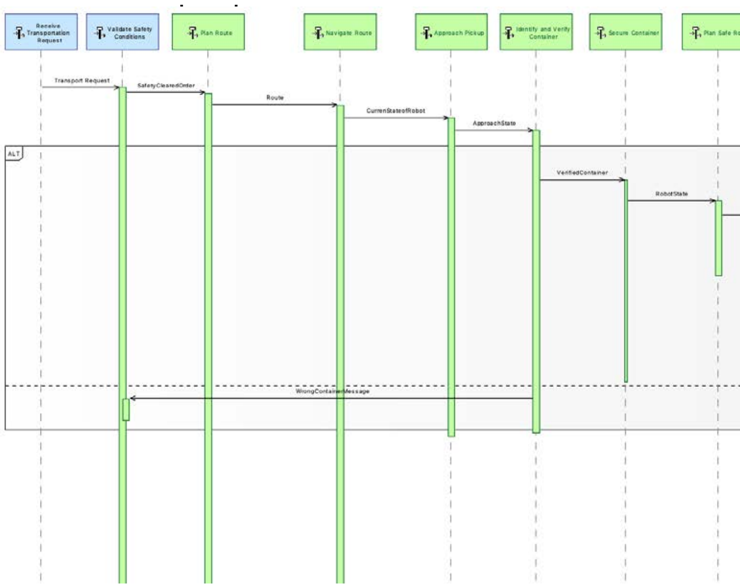
Outcome
- Functional architecture
- Non-functional requirements
- Interface requirements

Þ Representation that helps organise complexity, to fulfil stakeholder requirements.

Þ Might require translation for stakeholder







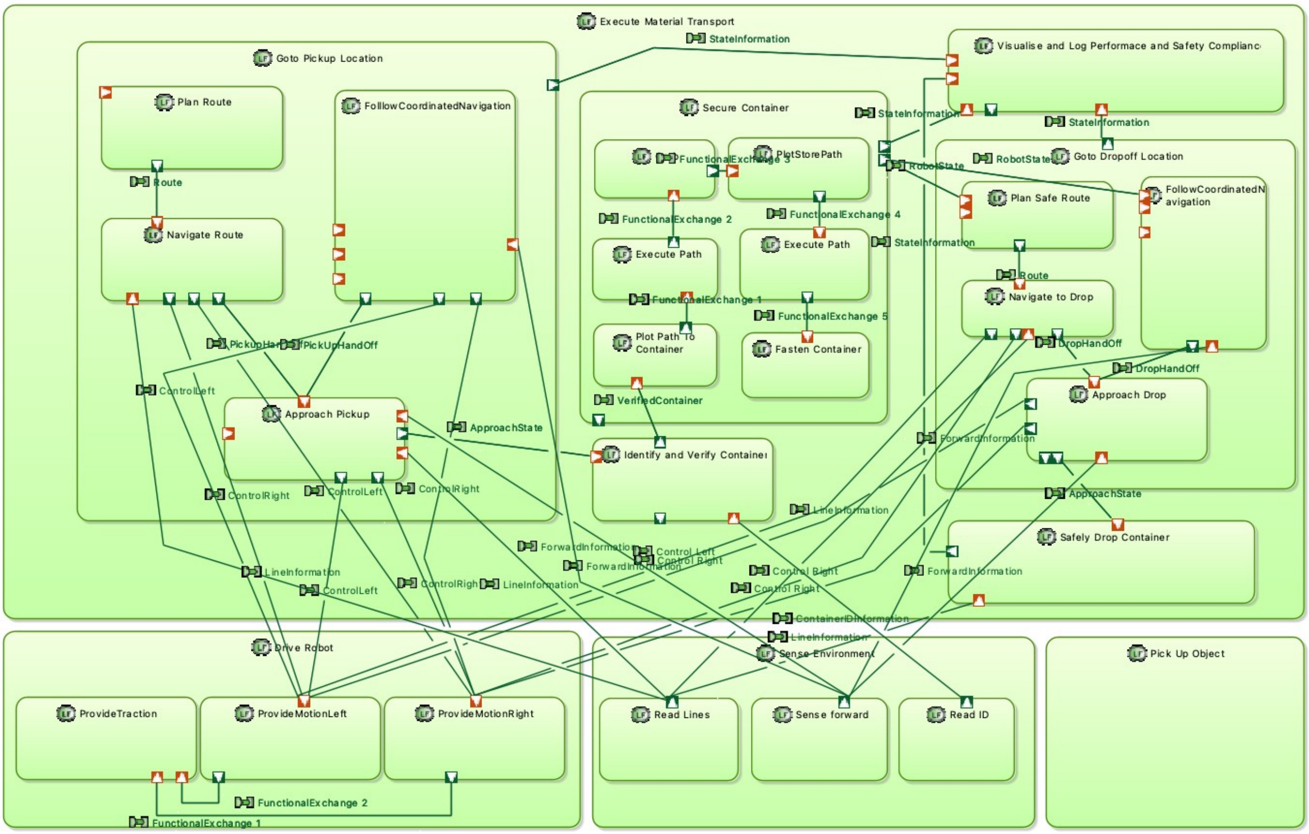
Logical Architectur

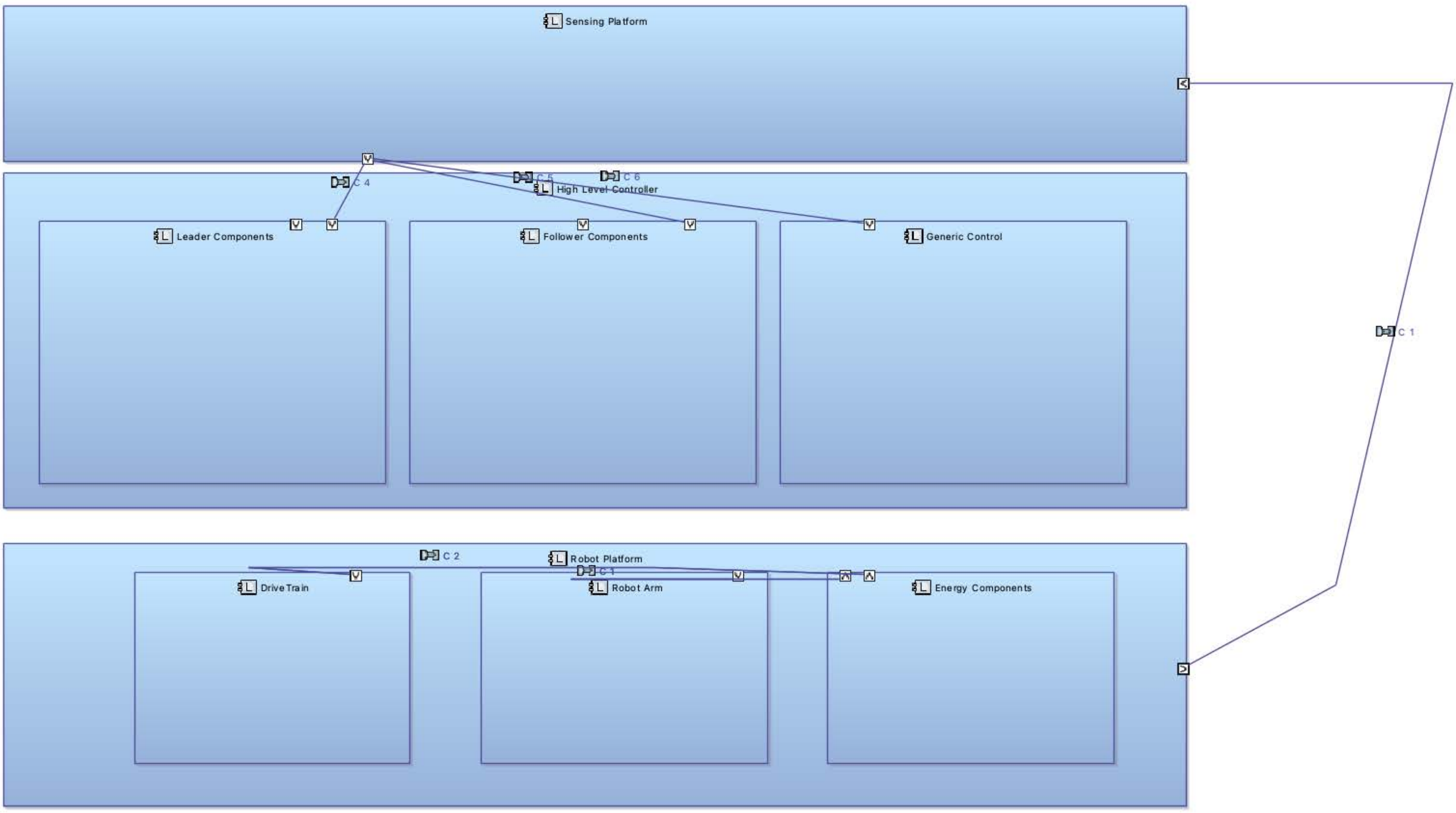
Develop System Logical Architecture

See the system as a white box
Define how the system will work so as to fulfill expectations
Perform a first trade-off analysis

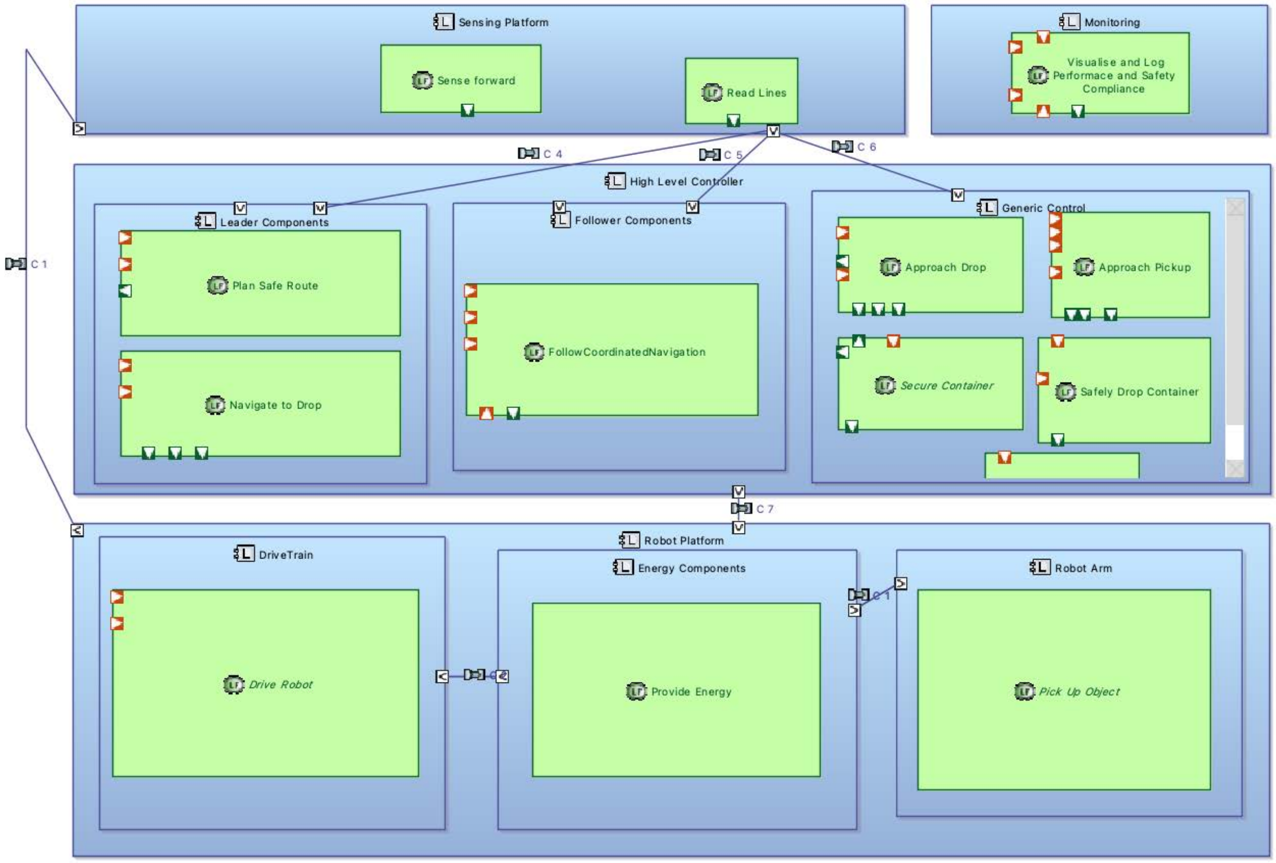
- Breaks open the black box.
- Principle Architecture of the solution. Big decisions of the solution
- Coarse grained level of detail!
  - Hide implementation details (not what specific type of something)
  - Allow for trade-offs between different concepts
  - Less expensive to detail multiple solutions if the level of abstraction is appropriate
- Performance and other non-functional constraints
  - Should be further detailed (e.g. failure rate, energy footprint, etc.)
- Implementation is not for this level
  - Physical architecture can evolve during life-time of the system
  - Logical architecture is expected to stay stable during evolution
- Decision examples:
  - Centralised vs distributed architectures, criticallity patterns (e.g. triple redundancy), behaviours, mechanisms, etc.

38

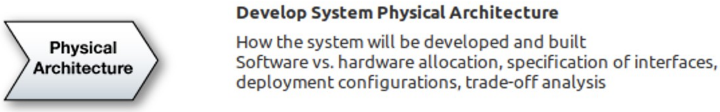




Sensing Platform
C 4
C 5
C 6
High Level Controller
Leader Components
Follower Components
Generic Control
C 1
C 2
Robot Platform
Drive Train
Robot Arm
Energy Components



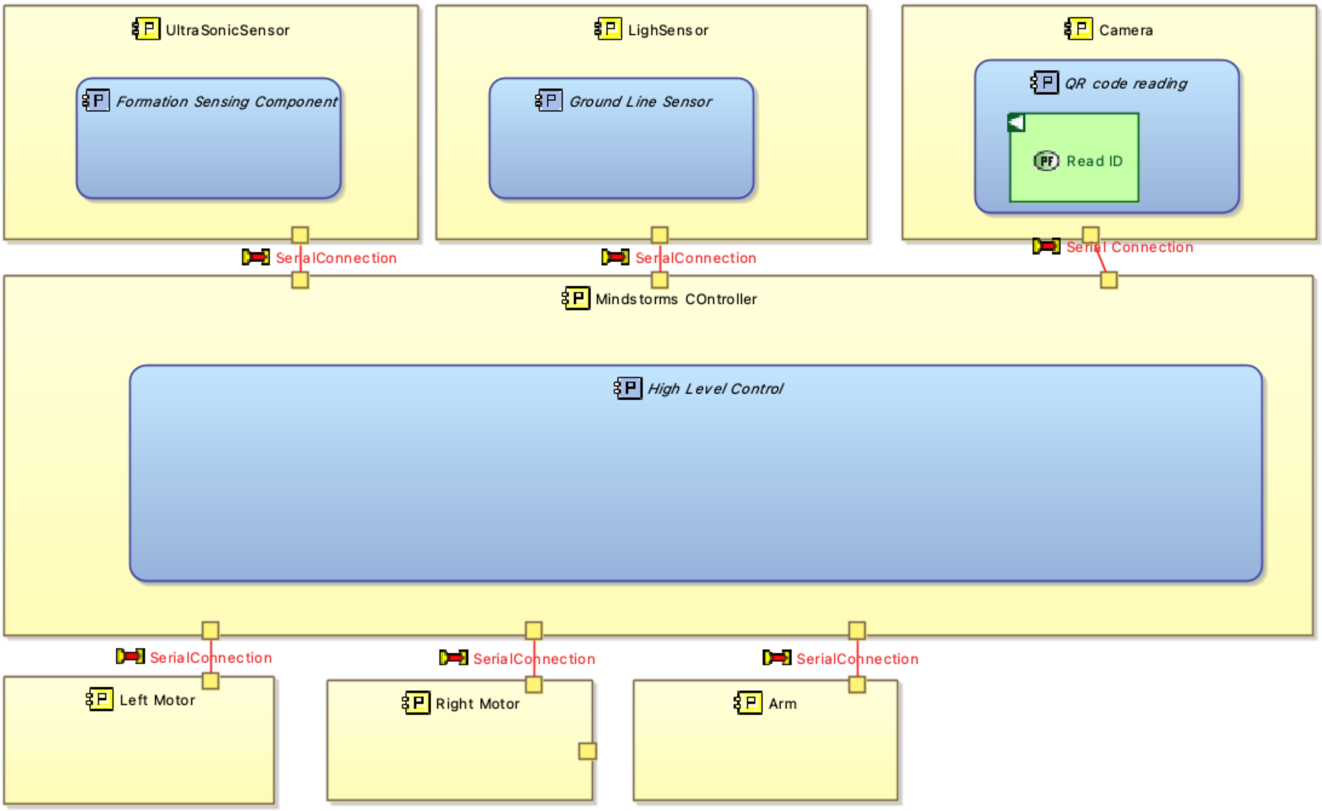
Sensing Platform
Sense forward
Read Lines
Monitoring
Visualise and Log Performace and Safety Compliance
C 1
C 4
C 5
C 6
High Level Controller
Leader Components
Plan Safe Route
Navigate to Drop
Follower Components
FollowCoordinatedNavigation
Generic Control
Approach Drop
Approach Pickup
Secure Container
Safely Drop Container
C 7
Robot Platform
DriveTrain
Drive Robot
Energy Components
Provide Energy
Robot Arm
Pick Up Object

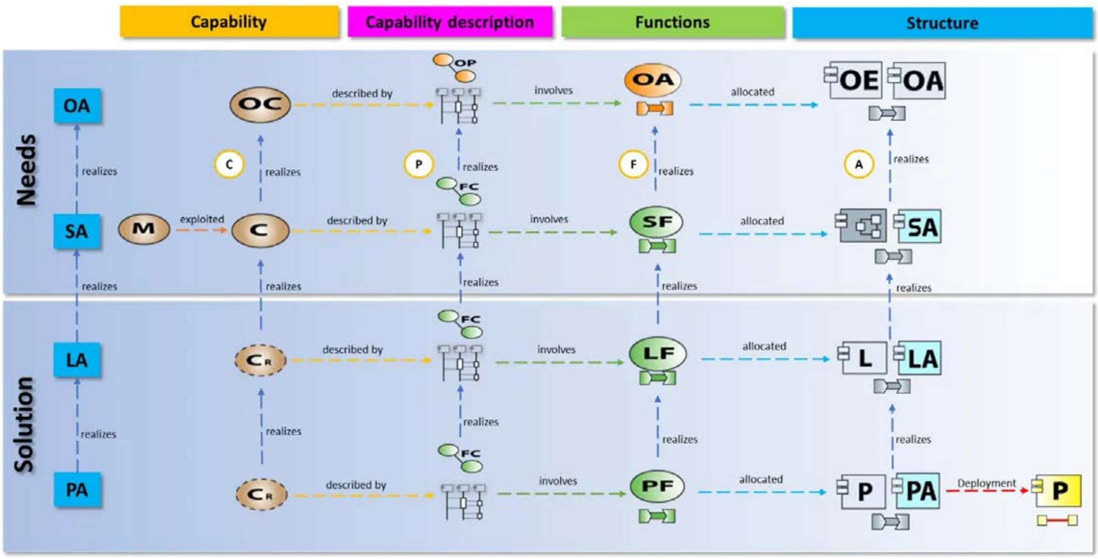


Physical Architecture

- More technology and implementation-oriented view of the system
- Make technological choices
- Refine + decide to create/buy
- 2 types of components:
  - Behaviour physical components -> perform the functions (comes from logical component)
  - Node Physical components -> host the behaviour components (yellow)
  - 1 Node physical component can host many behaviour components:
    - Examples: platforms, ECUs, mechanical parts,
  - 1 Behaviour component can only be mapped to a single node physical component
    - Examples: software component, etc.
  - Physical links (red lines): real means of communication, transport, etc.
  - Physical port

42


UltraSonicSensor
Formation Sensing Component
LighSensor
Ground Line Sensor
Camera
QR code reading
Read ID
SerialConnection
SerialConnection
Serial Connection
Mindstorms COntroller
High Level Control
SerialConnection
SerialConnection
SerialConnection
Left Motor
Right Motor
Arm



Traceability

44

Model-based Systems Engineering

- Going from a document-based to a Model-based approach to Systems Engineering
  - Allows for consistency checking
  - Traceability
  - Model transformations to simulation models
  - Etc.

6th Annual Summer School on
Cyber Physical Systems and Internet of Things

Temporal Logics for Building Systems

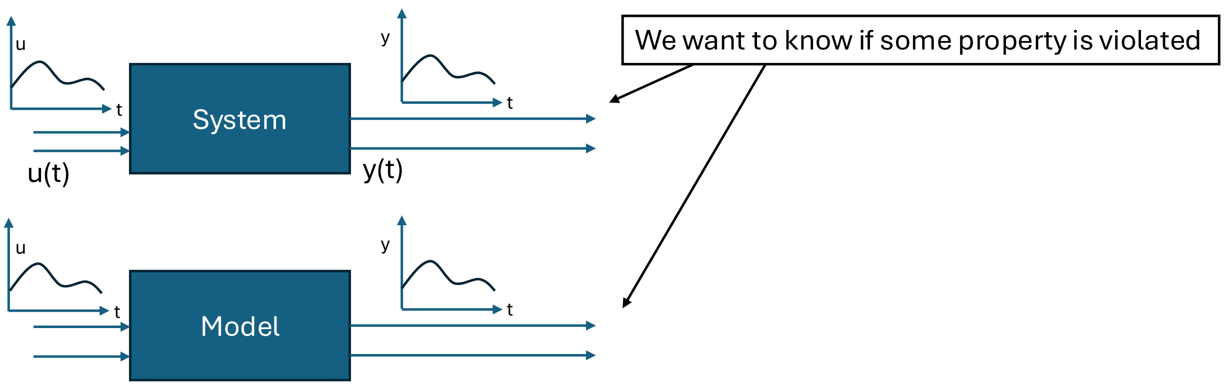
Joachim Denil

12 June 2025

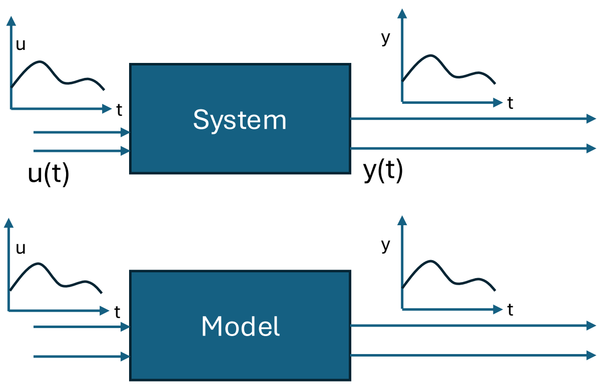
Goal of this lecture

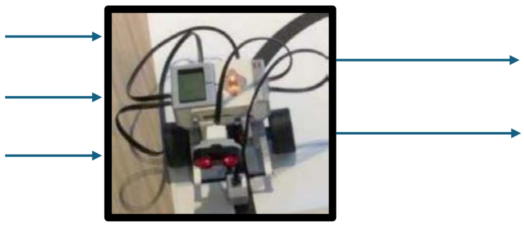
- Understand Linear Temporal Logic
- Understand Signal Temporal Logic
- Why temporal logics?

Dynamic Systems



Dynamic Systems





Requirements of the robot:

- Safety: Distance to the robot in front should not become smaller than x cm.
- Recovery: If the distance between the ego and the robot in front becomes smaller than x cm, after y seconds, the distance should be greater than z cm.
- ....

Types of properties you would like to check

Safety

Something bad should never happen

E.g. Deadlock free, mutual exclusion, Every red phase of traffic should be proceeded by a yellow phase

Liveliness

Something good will happen

E.g. Each process will eventually enter his critical section

Fairness

Guarrantees that there is progress

E.g. Starvation free

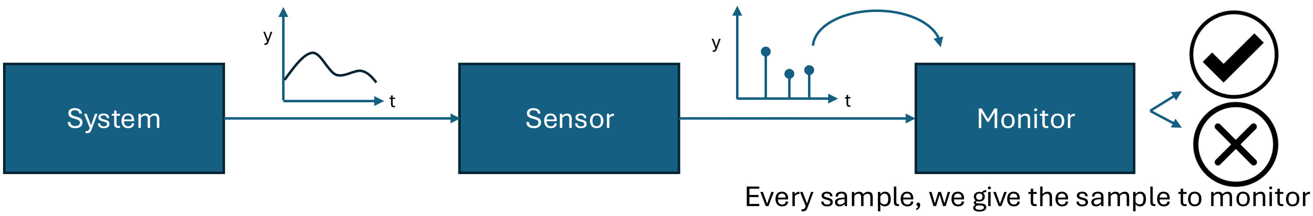
How can I monitor?

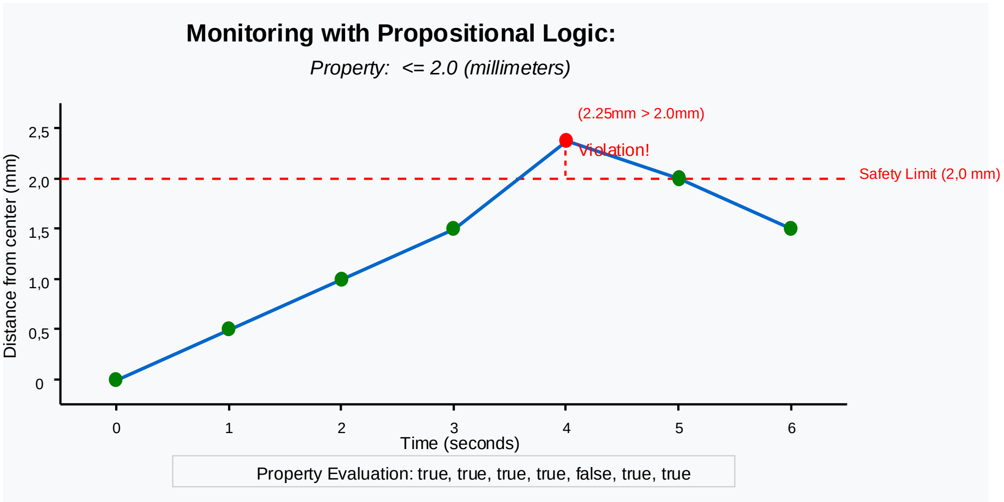
- Some properties are just a check on every value that is sensed from the real system.
  - Example: The robot should not deviate more then 2.0 mm from the center of the line

What about Propositional Logic?

- Propositional variables: Either True or False
  - *P_acceptable_deviation*: x <=2.0 [millimeter]
  - Check at every time that *P_acceptable_deviation* holds

Monitoring using Propositional Logic





Propositional Logic

Property: The robot should not deviate more then 2.0 mm from the center and does not exceed an absolute velocity bigger than 3 m/s

- Propositional variables:
  - P_hoist_height_acceptable: x <= 2.0 [millimeter]
  - P_velocity_acceptable: |velocity|<= 3 [m/s]
- You can create complex properties with operators:

Property:

```
P_hoist_height_acceptable /\
P_velocity_acceptable
```

Formal Name	Nickname	Arity	Symbol
Negation operator	**NOT**	**Unary**	¬
Conjunction operator	**AND**	**Binary**	∧
Disjunction operator	**OR**	**Binary**	∨
Exclusive-OR operator	**XOR**	**Binary**	⊕
Implication operator	**IMPLIES**	**Binary**	→
Biconditional operator	**IFF**	**Binary**	↔

How can I monitor?

- Some properties are just a check on every value that is sensed from the real system
  - Example: *The robot should not deviate more then 2.0 mm from the center of the line*
- But what about temporal orderings of events?
  - *If the robot deviates significantly from the line, then within 2 seconds, it must reduce its deviation to acceptable levels*
  - *After an emergency stop button is pressed, the power to the motors must be cut off in the next state, and the brakes must be engaged.*
- Much more difficult.
- Temporal logics are appropriate

Decomposing a formula

If the emergency button is pressed, then in the next state, motors must be off and brakes engaged.

Propositions in this formula:

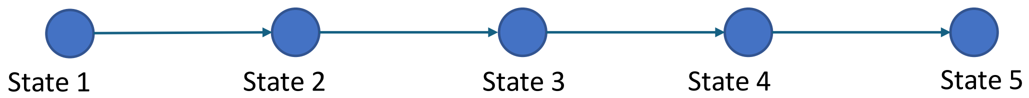
- An emergency stop button is pressed: P_emergency_stop
- The power to the motors is cut off: P_power_off
- The brakes are engaged: P_brakes_engaged

Temporal phenomena:

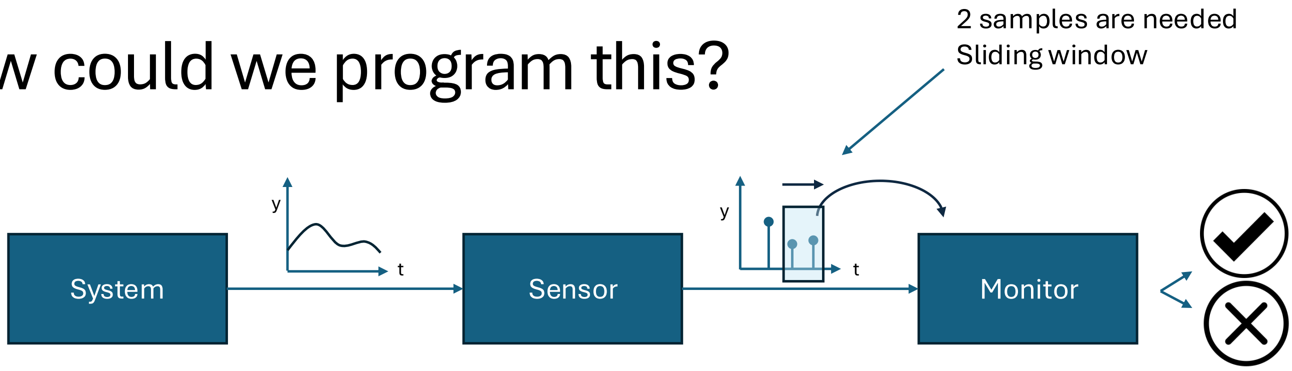
Next state

Time abstraction...

- Look at time as an ordered set of states



How could we program this?




```
monitor (P_emergency_stop, P_power_off, P_brakes_engaged):
    if (P_emergency_stop[0] == true):
            if (P_power_off[1] == true and P_brakes_engaged[1] == true):
                  return true
            else:
                  return false
            endif
    else:
        return true
    endif
```

Mixes up the what and the how..

LTL (Linear Temporal Logic)

More appropriately: Temporal Logic

Linear Temporal Logic: along a single path of states

Each state is defined by a unique assignment of propositional variables

$$\varphi ::= \text{true} \;\Big|\; a \;\Big|\; \varphi_1 \wedge \varphi_2 \;\Big|\; \neg\varphi \;\Big|\; \bigcirc\varphi \;\Big|\; \varphi_1 \,\mathsf{U}\, \varphi_2$$

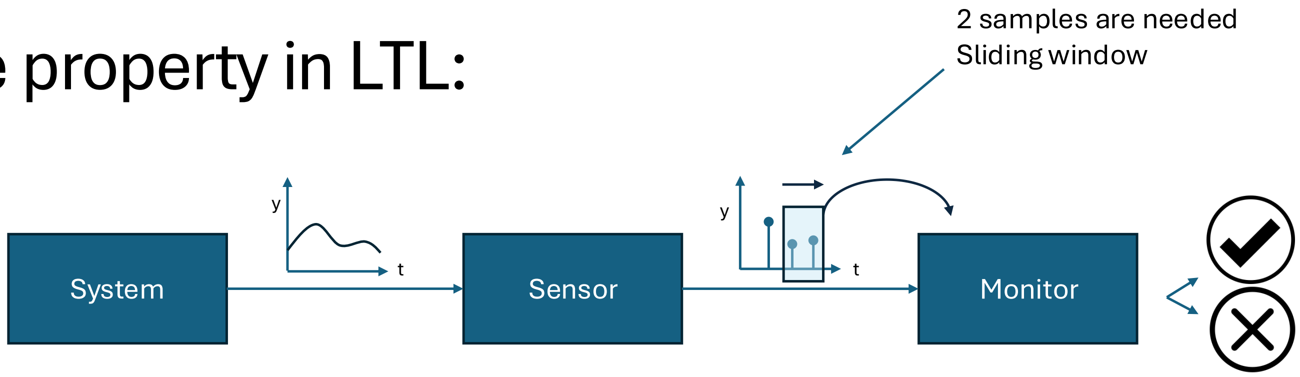
a = atomic proposition (e.g., P_emergency_stop, P_motor_off, etc.)

With temporal operators, it assumes a sequence of states

O = next (sometimes X ϕ): property ϕ holds in the next state

U = Until (ϕ1 U ϕ2): property ϕ1 holds continuously until ϕ2 becomes true

The property in LTL:



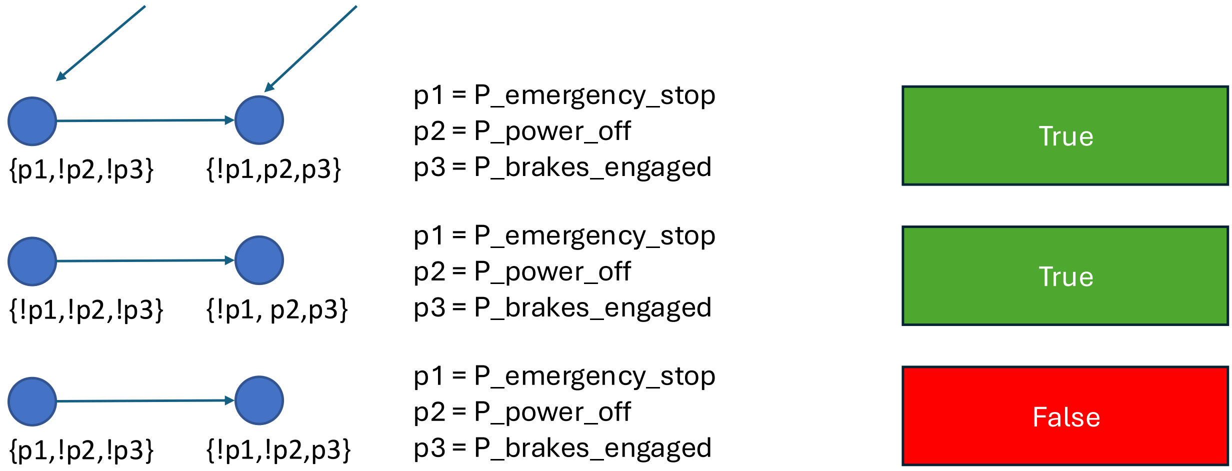

```
Specification: P_emergency_stop -> X(P_power_off /\ P_brake_engaged)
```

More appropriate because:

- Declarative: Specifies what happens, not how to check it
- Formal: Precise meaning
- Compositional: Complex properties from simple ones

Evaluation of a formula

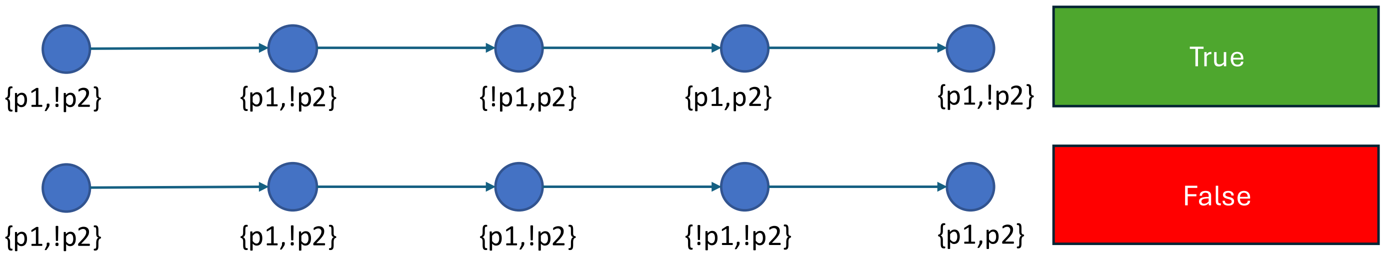
```
P_emergency_stop -> X(P_power_off /\ P_brake_engaged)
```



$$\varphi ::= \text{true} \mid a \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc \varphi \mid \varphi_1 \,\mathsf{U}\, \varphi_2$$

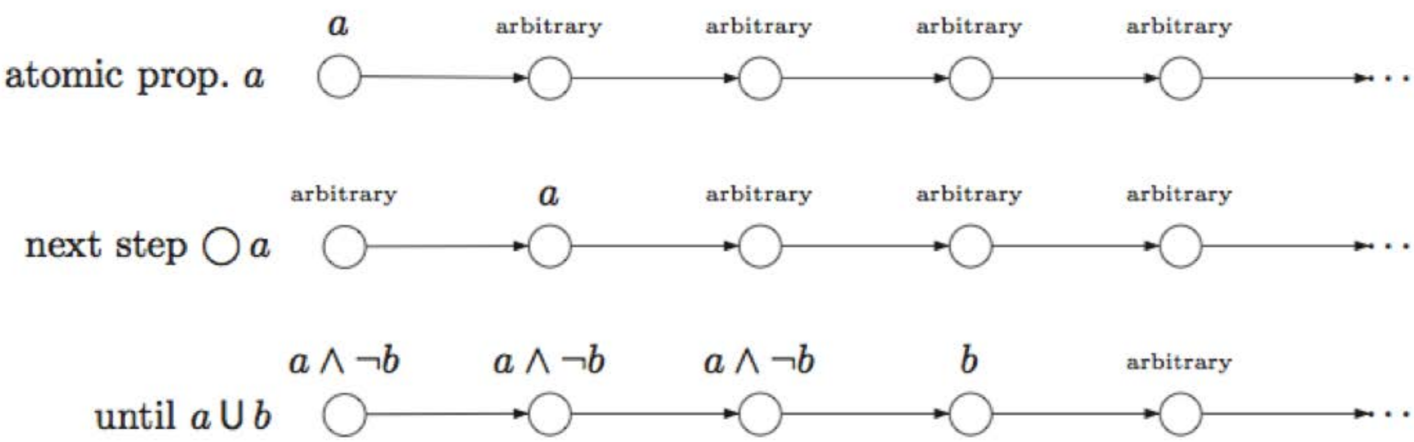
Satisfaction of a formula

`p1 U p2`



$$\varphi ::= \text{true} \;\Big|\; a \;\Big|\; \varphi_1 \wedge \varphi_2 \;\Big|\; \neg\varphi \;\Big|\; \bigcirc \varphi \;\Big|\; \varphi_1 \,\mathsf{U}\, \varphi_2$$

Evaluation of a formula (summary)



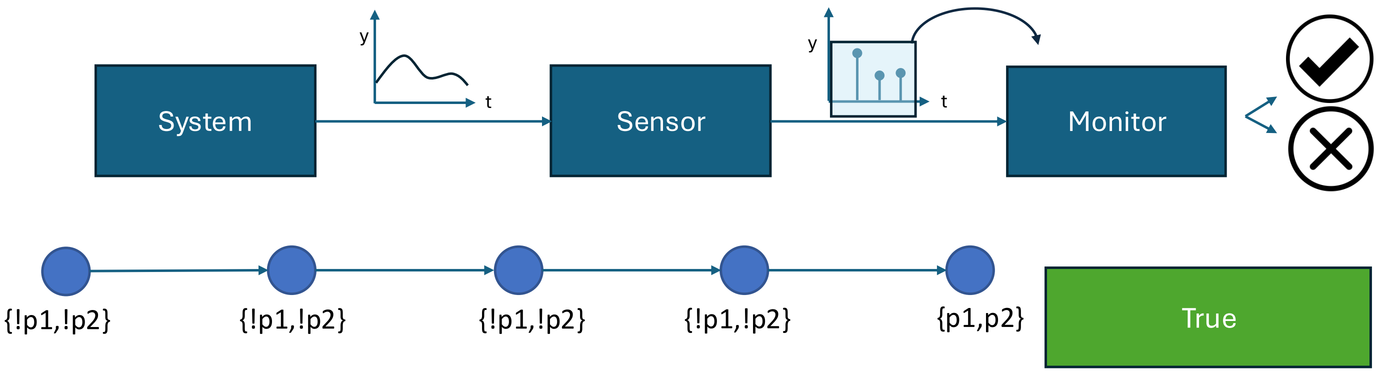
σ ⊨ φ (path "satisfies" specification)

True U p1

Property: By the end of each operational cycle, the crane must eventually come to a complete stop

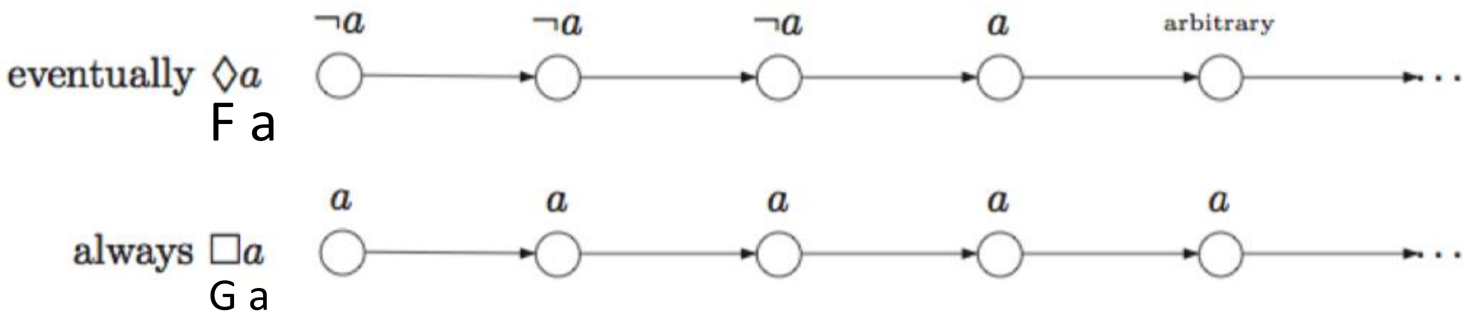
p1 = P_horizontal_no_velocity
p2 = P_vertical_no_velocity

```
true U (p1/\p2)
```



THIS IS OFTEN NEEDED -> OPERATOR: F OR <>

Extra operators (summary)



Binding

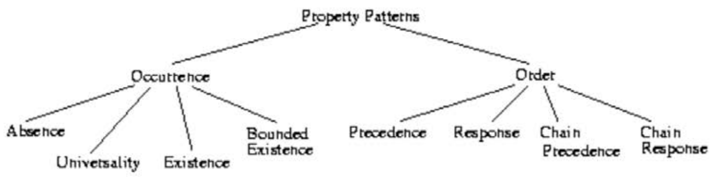
- **Operator precedence**
  - Unary binds stronger than binary
  - Bind from right to left:
  - ☐◊p = (☐(◊p))
    - p U q U r = p U (q U r)
- U takes precedence over ∧, ∨ and →

Dwyer Patterns

https://matthewbdwyer.github.io/psp/patterns/ltl.html

The Patterns

The information in the patterns can be presented in a variety of ways. One organization, illustrated below, is based on classifying the patterns in terms of the kinds of system behaviors they describe.



- Occurrence Patterns talk about the occurrence of a given event/state during system execution.
- Order Patterns talk about relative order in which multiple events/states occur during system execution.
- While not themselves patterns, Pattern Notes discuss common ways to vary the existing patterns to suite your needs.

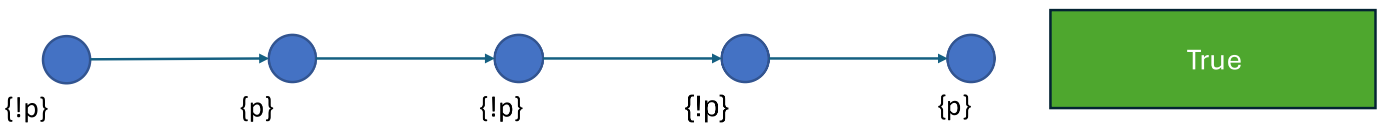
An alternative organization for this information is to group pattern to formalism mappings by specification formalism. The supported formalisms are listed below. Clicking on the formalism will bring you to pages with mappings for each property pattern in that formalisms. We supply the mappings on these formalism-specific pages and you are refered to the complete patterns for information about relationships and example uses.

- Linear Temporal Logic (LTL)
- Computation Tree Logic (CTL)

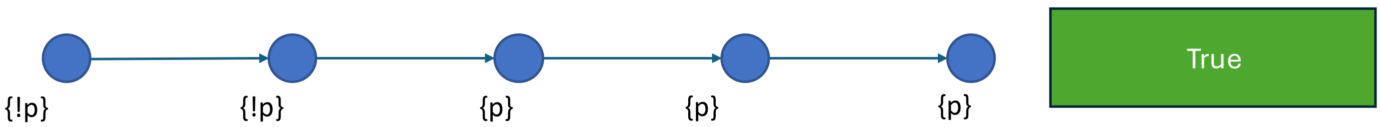
Dwyer Patterns

https://matthewbdwyer.github.io/psp/patterns/ltl.html

- What is G F p



- What is F G p

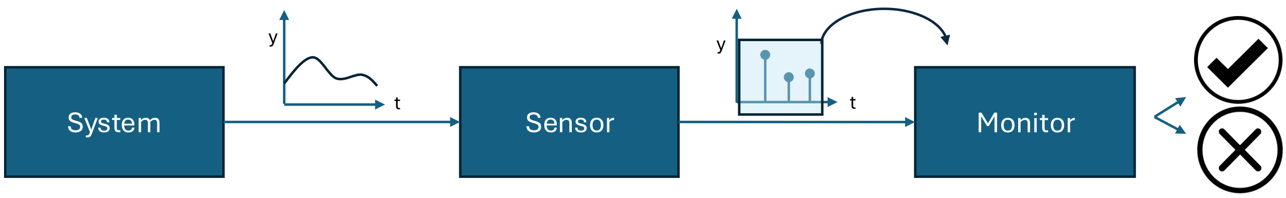


Examples

- *When the robot deviates more than 2 mm from the line, velocity must always be less than 0.2 m/s*
  - Propositions needed:
    - P_high_deviation: line_deviation >2.0
    - P_velocity: velocity < 0.2
  - `G((P_high_deviation) -> (P_velocity))`
- *After losing the line, the robot must stop until the line is detected again.*
  - Propositions needed:
    - P_line_lost: line_sensor_value < threshold
    - P_searching: abs(left_motor_speed + right_motor_speed) < 0.1 && abs(left_motor_speed - right_motor_speed) > 0.2
    - P_line_found: line_sensor_value >= threshold
  - `G(P_line_lost -> (P_searching U P_line_found))`

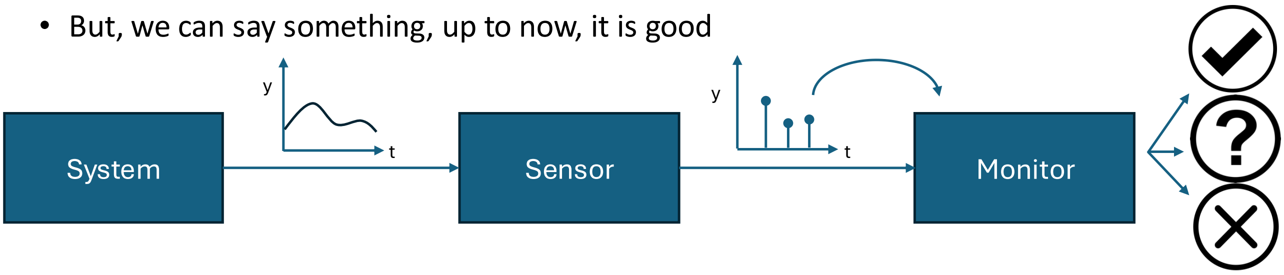
Monitoring Bounded Traces

- What we have shown up to now
- Verification window corresponds to a natural operational cycle
  - For example, one pick-up/move/drop-off of the crane system
- Binary verdict (True or False) as the trace is complete



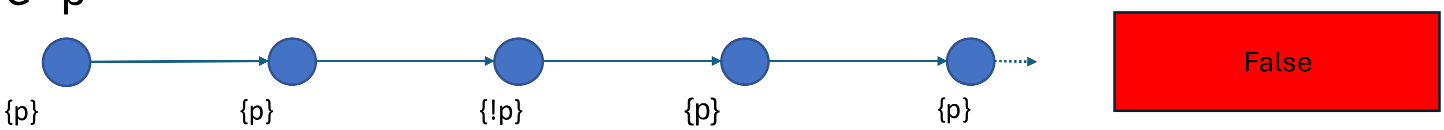
Monitoring Unbounded Traces

- We give the monitor a sample each timestep
- Requires reasoning about potentially infinite future behaviours
- Three possibilities:
  - Property satisfied over the finite trace, independent of future
  - Property falsified over the final trace, for every continuation
  - Finite trace still allows for falsification or satisfaction of the property, but **so far so good**
- Needs three-valued logic (True, False, Inconclusive)
- Watch out: some properties are non-monitorable
  - G(r->Fa): No continuation will ever lead to satisfaction or falsification.
  - But, we can say something, up to now, it is good

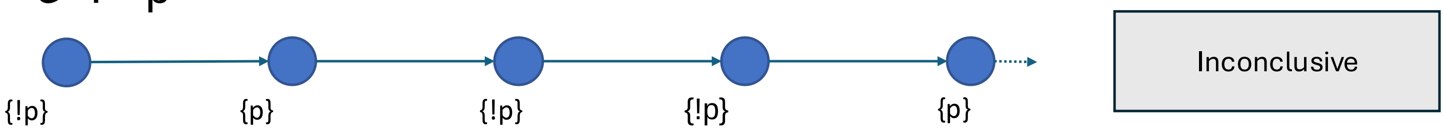


Monitoring Unbounded Traces

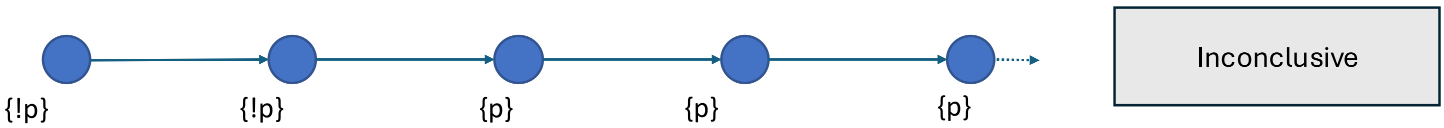
G p



- G F p



- F G p



Towards STL

If the robot deviates significantly from the line, then within 2 seconds, it must reduce its deviation to acceptable levels

LTL `G(P_high_deviation-> F P_low_deviation)`

Boolean predicates, discrete-time: inadequate for our purpose

MTL `G(P_high_deviation->` $F_{[0,2s]}$ `P_low_deviation )`

Boolean predicates, real-time

STL `G((`*line_deviation* `> 2mm) ->` $F_{[0,\ 2s]}$ *line_deviation <= 2 mm*`))`

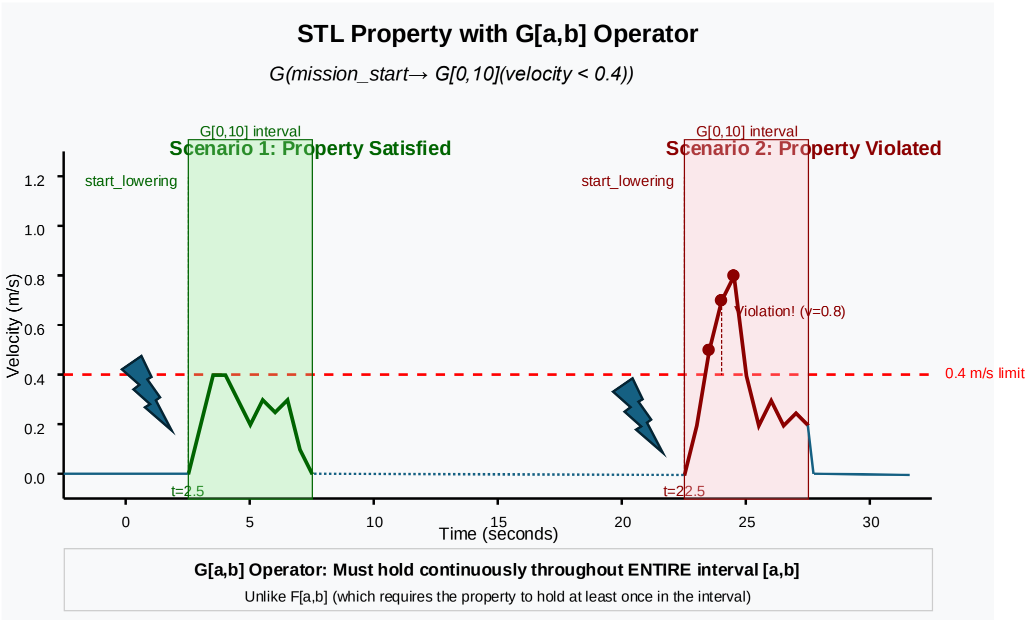
Predicates over real values, real-time

Robustness, how far am I off violating the property?

Examples

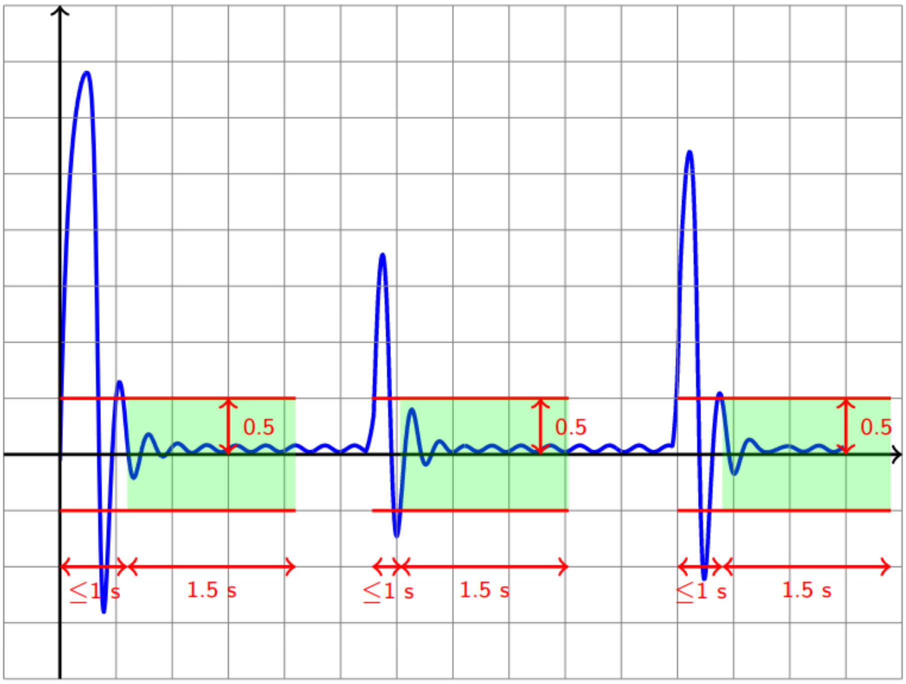
Whenever a mission starts, the robot's velocity must remain below 0.4 m/s for the next 10 seconds

`G(mission_start → G[0,10](velocity < 0.4))`

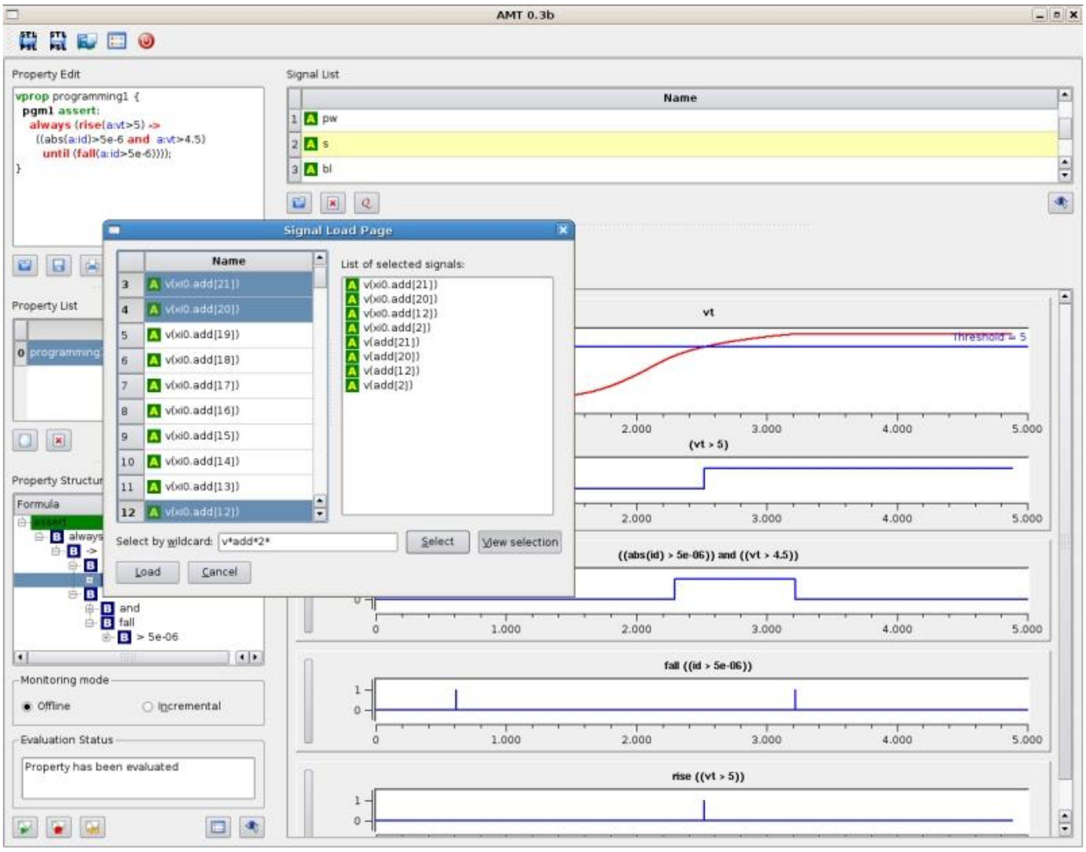


Always $|x| > 0.5 \Rightarrow$ *after 1 s,* $|x|$ *settles under 0.5 for 1.5 s*

$$\varphi := \mathsf{G}(x[t] > .5 \rightarrow \mathsf{F}_{[0,1]} \; (\mathsf{G}_{[0,1.5]} \; x[t] < 0.5))$$



From: A. Donze, On Signal Temporal Logic, 2014



https://www-verimag.imag.fr/DIST-TOOLS/TEMPO/AMT/content.html

International Conference on Computer Aided Verification
CAV 2010: Computer Aided Verification pp 167-170 | Cite as

Breach, A Toolbox for Verification and Parameter Synthesis of Hybrid Systems

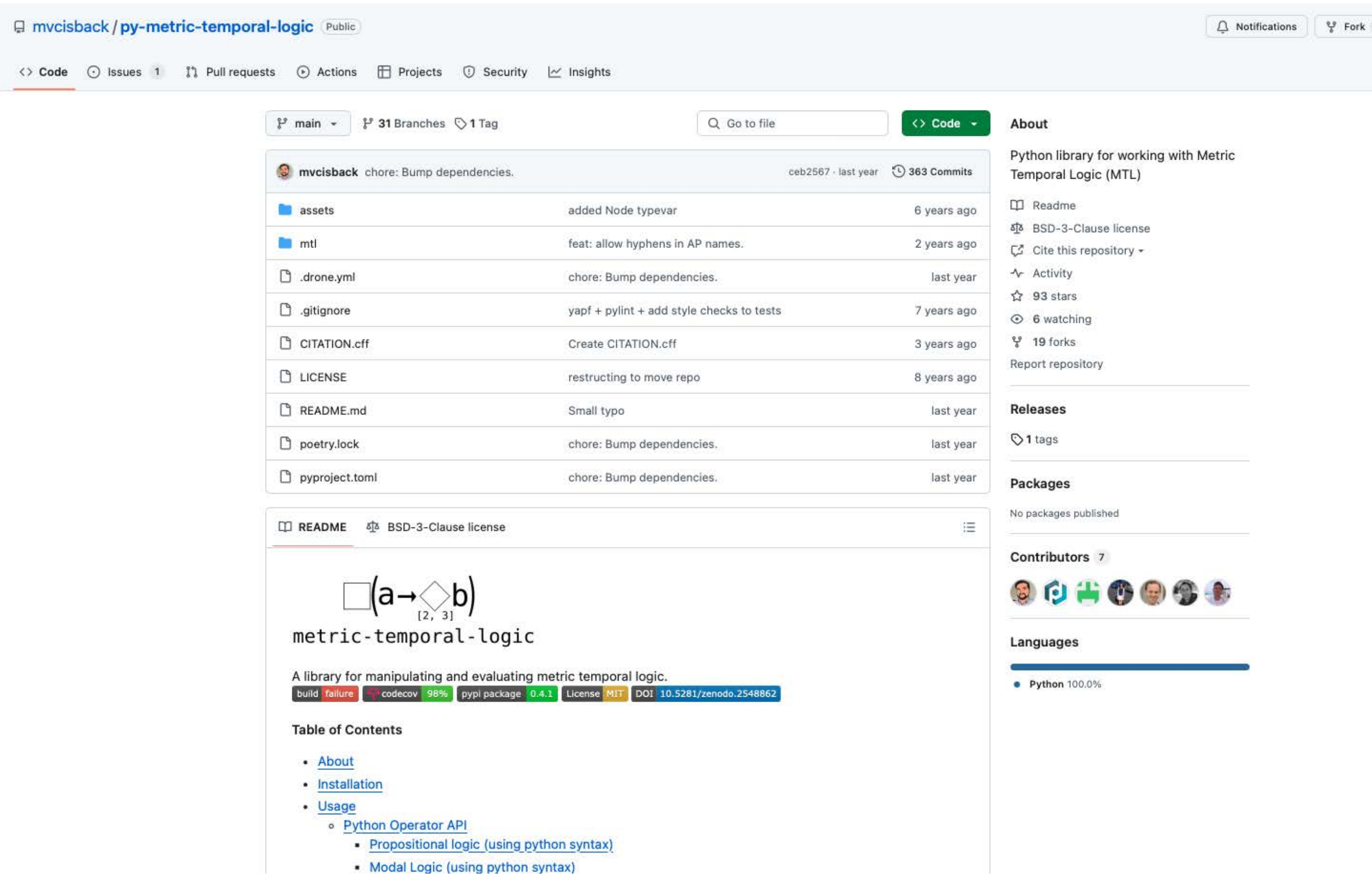
Authors Authors and affiliations

Alexandre Donzé

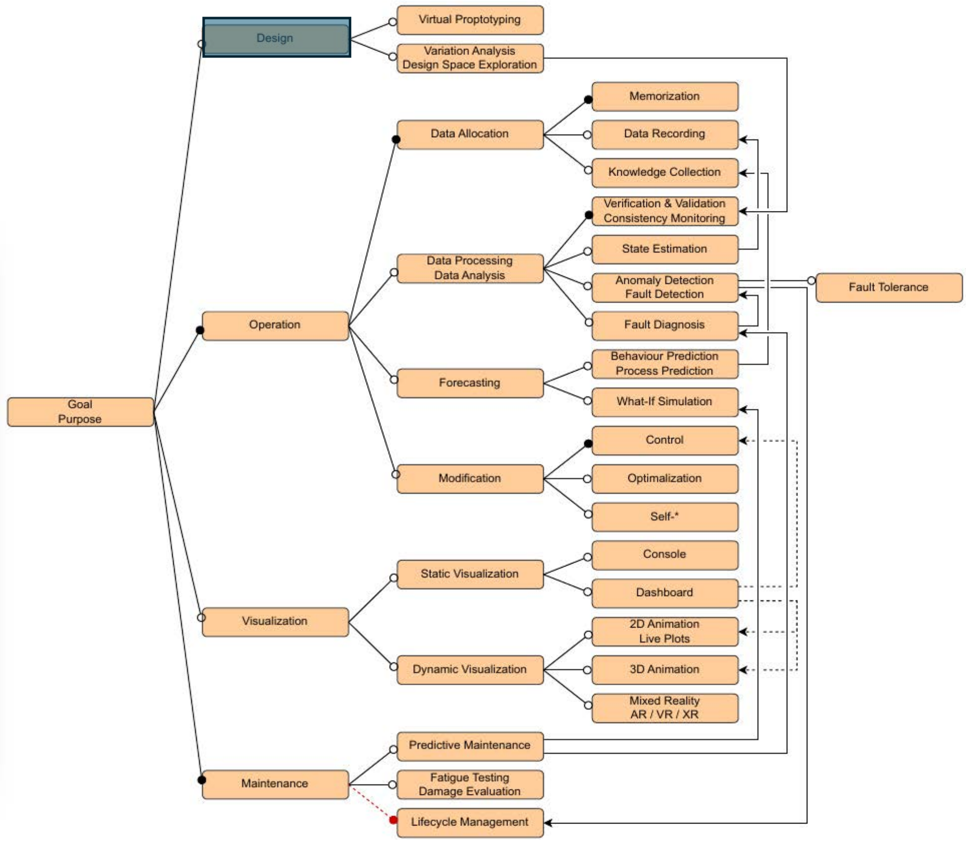
Part of the Lecture Notes in Computer Science book series (LNCS, volume 6174)

Abstract

We describe Breach, a Matlab/C++ toolbox providing a coherent set of simulation-based techniques aimed at the analysis of deterministic models of hybrid dynamical systems. The primary feature of Breach is to facilitate the computation and the property investigation of large sets of trajectories. It relies on an efficient numerical solver of ordinary differential equations that can also provide information about sensitivity with respect to parameters variation. The latter is used to perform approximate reachability analysis and parameter synthesis. A major novel feature is the robust monitoring of metric interval temporal logic (MITL) formulas. The application domain of Breach ranges from embedded systems design to


mvcisback / py-metric-temporal-logic Public
Notifications
Fork
Code
Issues 1
Pull requests
Actions
Projects
Security
Insights
main
31 Branches
1 Tag
Go to file
Code
mvcisback chore: Bump dependencies.
ceb2567 · last year
363 Commits
assets
added Node typevar
6 years ago
mtl
feat: allow hyphens in AP names.
2 years ago
.drone.yml
chore: Bump dependencies.
last year
.gitignore
yapf + pylint + add style checks to tests
7 years ago
CITATION.cff
Create CITATION.cff
3 years ago
LICENSE
restructing to move repo
8 years ago
README.md
Small typo
last year
poetry.lock
chore: Bump dependencies.
last year
pyproject.toml
chore: Bump dependencies.
last year
README
BSD-3-Clause license
metric-temporal-logic
A library for manipulating and evaluating metric temporal logic.
build failure
codecov 98%
pypi package 0.4.1
License MIT
DOI 10.5281/zenodo.2548862
Table of Contents
About
Installation
Usage
Python Operator API
Propositional logic (using python syntax)
Modal Logic (using python syntax)
About
Python library for working with Metric Temporal Logic (MTL)
Readme
BSD-3-Clause license
Cite this repository
Activity
93 stars
6 watching
19 forks
Report repository
Releases
1 tags
Packages
No packages published
Contributors 7
Languages
Python 100.0%

Purpose


Goal
Purpose
Design
Virtual Proptotyping
Variation Analysis
Design Space Exploration
Operation
Data Allocation
Memorization
Data Recording
Knowledge Collection
Data Processing
Data Analysis
Verification & Validation
Consistency Monitoring
State Estimation
Anomaly Detection
Fault Detection
Fault Diagnosis
Fault Tolerance
Forecasting
Behaviour Prediction
Process Prediction
What-If Simulation
Modification
Control
Optimalization
Self-*
Visualization
Static Visualization
Console
Dashboard
Dynamic Visualization
2D Animation
Live Plots
3D Animation
Mixed Reality
AR / VR / XR
Maintenance
Predictive Maintenance
Fatigue Testing
Damage Evaluation
Lifecycle Management

Scenario 1: Proper understanding of requirements

Specification from SpaceWire protocol:

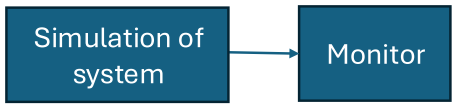
"The *ErrorReset* state shall be entered after a system reset, after link operation has been terminated for any reason or if there is an error during link initialization."

- No mention of when the *ErrorReset* state is to be entered
  - The protocol is synchronous, so after one tick, when one of the three conditions becomes True? Or can it wait for 10 ticks before it should enter that state?

Example from Lee, Seshia; Introduction to embedded systems

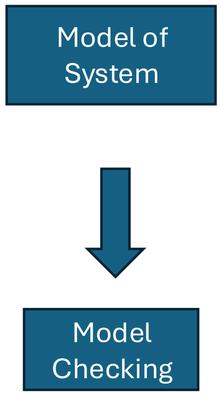
Scenario 2: Monitoring in the design phase of systems

- Testing Systems:
  - Model-in-the-loop
  - Software-in-the-loop
  - Hardware-in-the-loop
- Monitor checks if properties hold:
  - On a set of test scenarios.
  - Safety properties

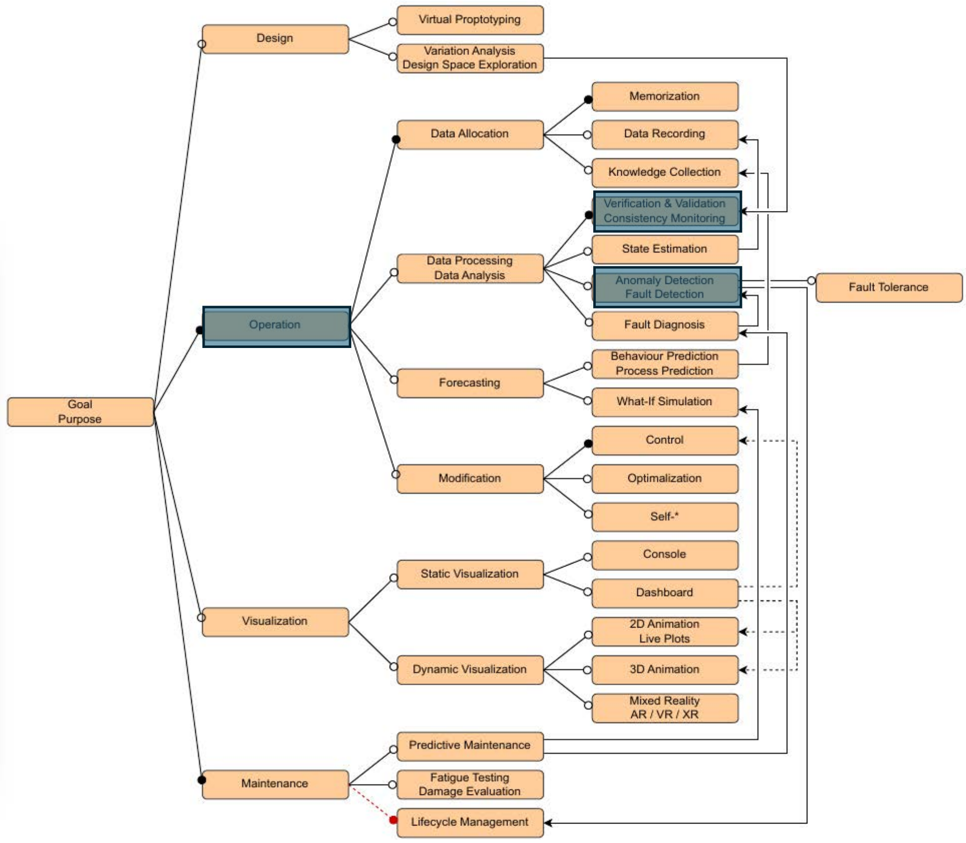


Scenario 3: Model Checking in the design phase

- Check if properties hold in all different circumstances
- Model of the system is used for checking
- *Example: Whenever the emergency stop is pressed, in the very next state, power to all motors must be cut off.*

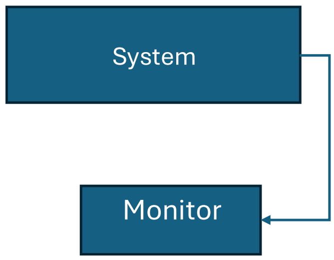


Purpose


Goal
Purpose
Design
Virtual Proptotyping
Variation Analysis
Design Space Exploration
Operation
Data Allocation
Memorization
Data Recording
Knowledge Collection
Data Processing
Data Analysis
Verification & Validation
Consistency Monitoring
State Estimation
Anomaly Detection
Fault Detection
Fault Tolerance
Fault Diagnosis
Forecasting
Behaviour Prediction
Process Prediction
What-If Simulation
Modification
Control
Optimalization
Self-*
Visualization
Static Visualization
Console
Dashboard
Dynamic Visualization
2D Animation
Live Plots
3D Animation
Mixed Reality
AR / VR / XR
Maintenance
Predictive Maintenance
Fatigue Testing
Damage Evaluation
Lifecycle Management

Scenario 4: Monitoring in System Operation

- Check properties during operation of the crane
- Operator can react if a property fails:



Conclusion

- LTL/MTL/STL: Appropriate languages for specifying the dynamic properties of the system

Dimension	LTL	CTL	MTL	STL
Time domain	Discrete	Discrete	Continuous	Continuous
Preposition Domain	Boolean	Boolean	Boolean	Real-valued signal
Primary Application	Desing-time & run-time	Desing-time (sometimes run-time)	Desing-time & run-time	Desing-time & run-time
Path Quantification (in model checking)	Implicit universal	Explicit A,E	N/A (only subset can be used in MC)	N/A
Trance interpretation (monitoring)	Bounded and unbounded linear	finite and infinite branching structures	Bounded and unbounded linear	Bounded and unbounded linear

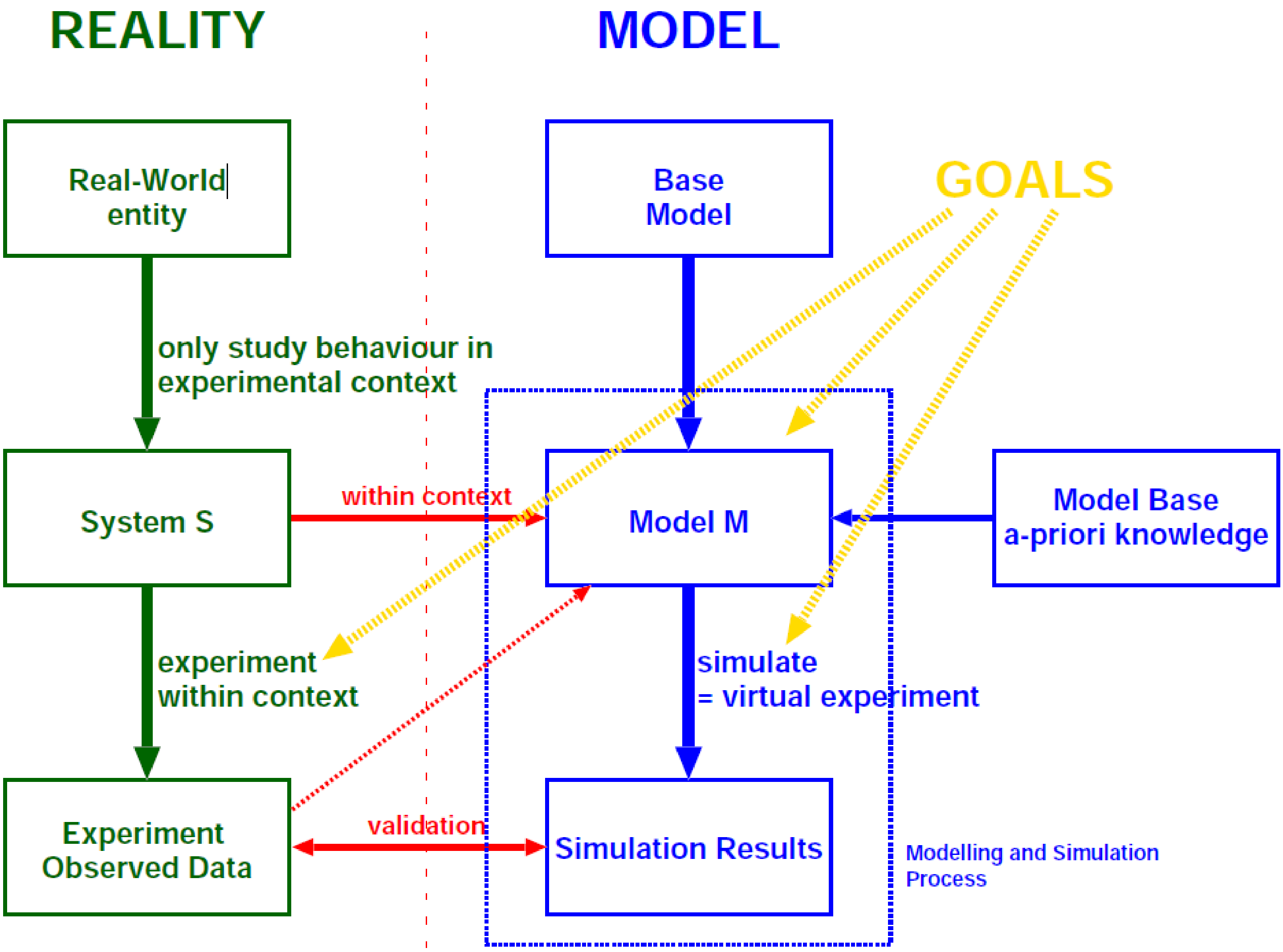
SS-CPSIoT

6th Annual Summer School on
Cyber Physical Systems and Internet of Things

An Introduction to Equation-based Object-Oriented Modelling of Cyber-Physical Systems with (Open)Modelica

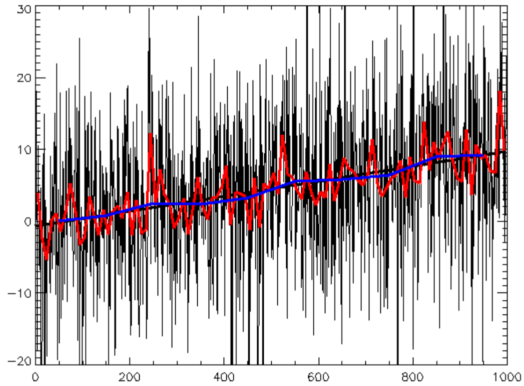
Hans Vangheluwe and Rakshit Mittal

12 June 2025 – Budva, Montenegro



Bernard P. Zeigler. *Multi-faceted Modelling and Discrete-Event Simulation.* Academic Press, 1984.

models based on measurements



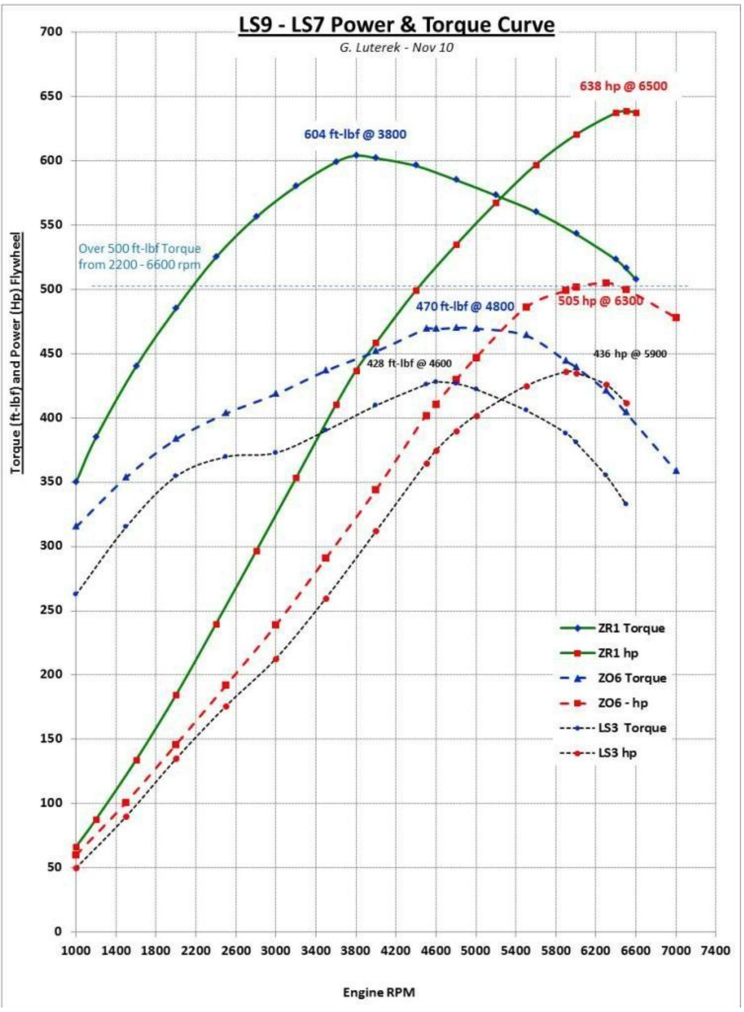
- instance (technology) – specific

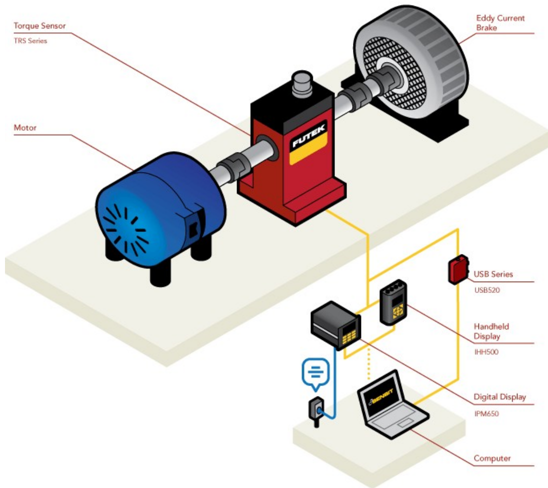
- high (experimentation) cost
- may not even be possible to measure

- allows reproducing data, no extrapolation;
  no insight/explanation

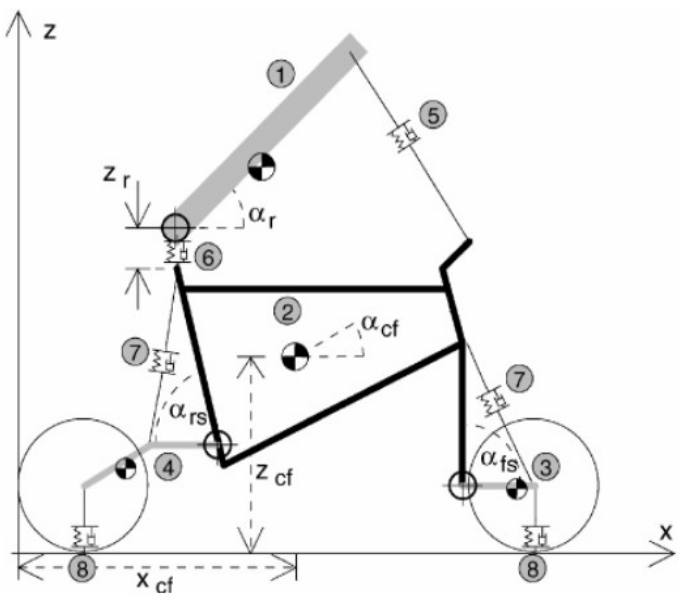
- inductive vs. deductive modelling workflow
  science vs. engineering, usually combination

Torque Curve "model" (measured)





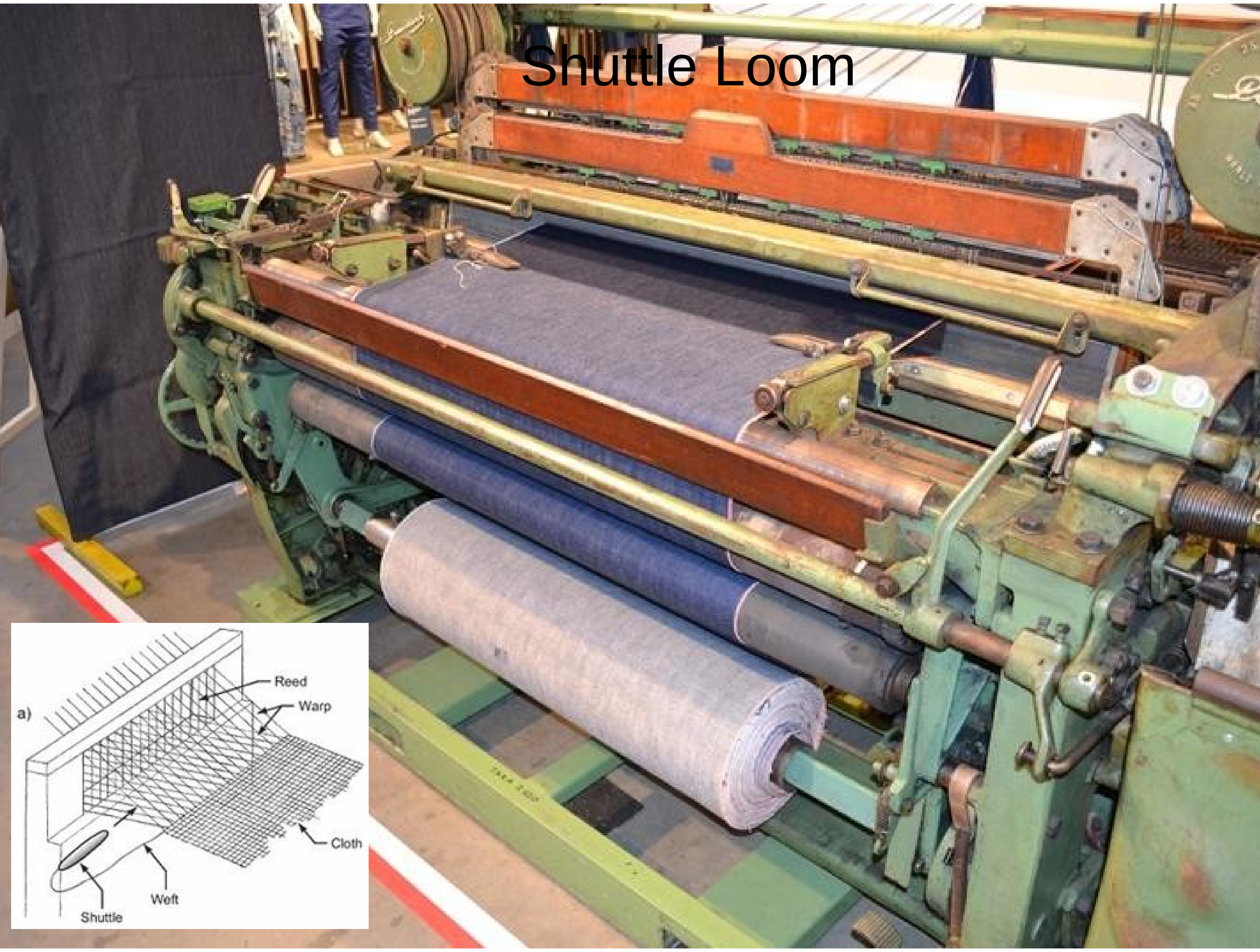
Parameters
(vs. Constants and Variables)



Distributed

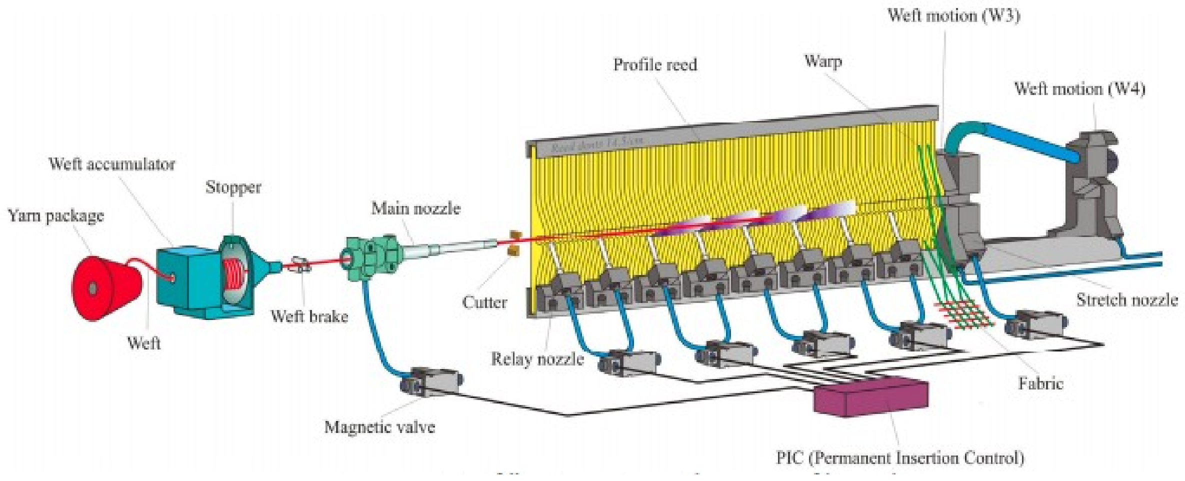
Lumped

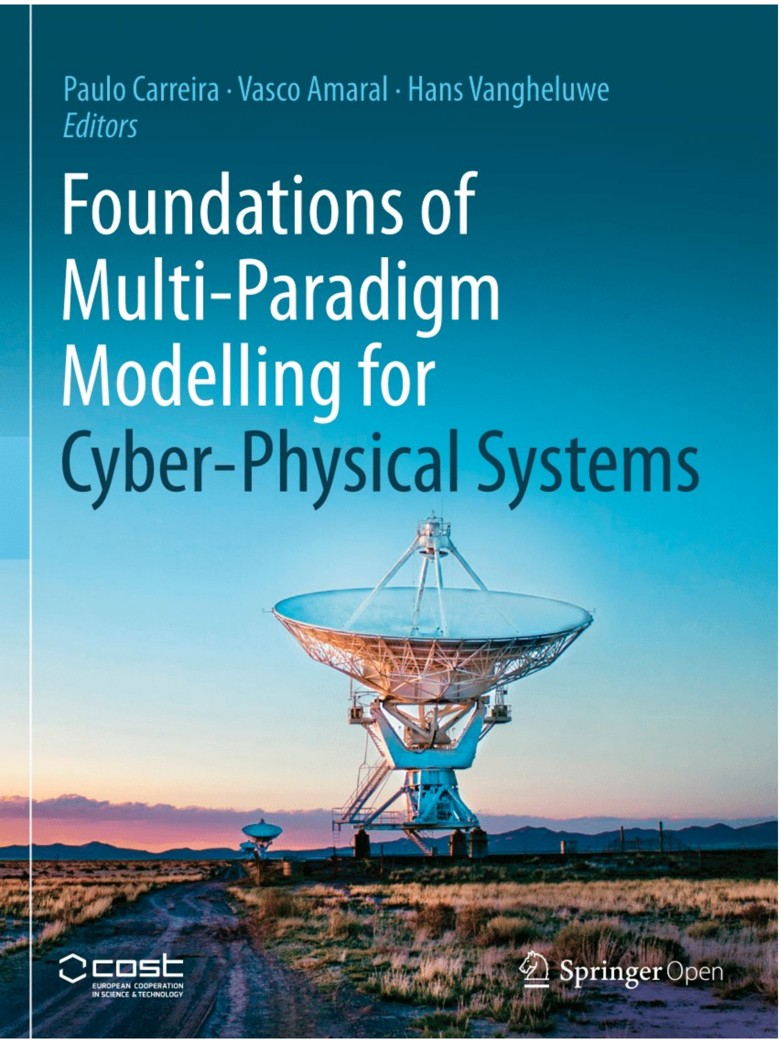
Shuttle Loom



Air Jet Loom

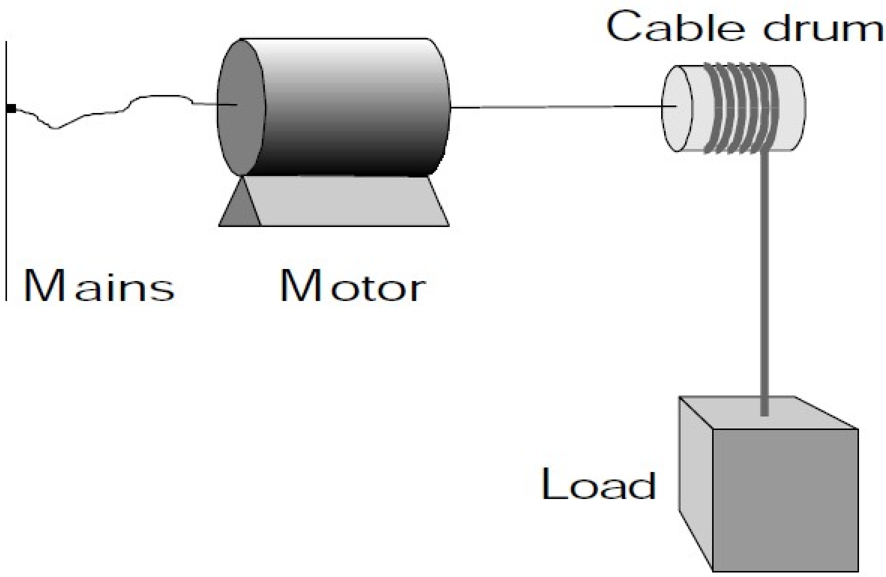

PICANOL


Weft motion (W3)
Profile reed
Warp
Weft motion (W4)
Weft accumulator
Stopper
Yarn package
Main nozzle
Cutter
Weft brake
Weft
Stretch nozzle
Relay nozzle
Fabric
Magnetic valve
PIC (Permanent Insertion Control)

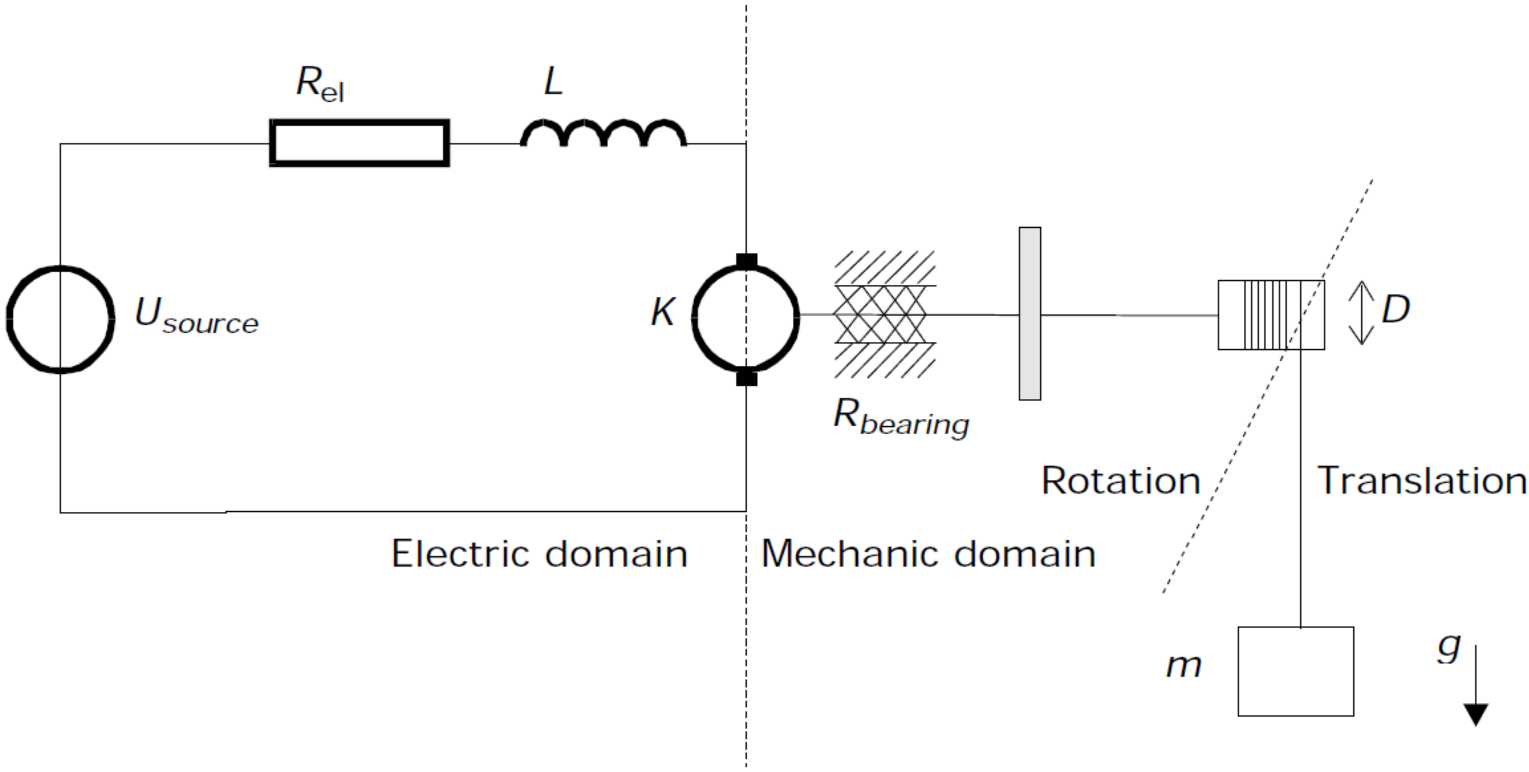


Broenink J.F. (2020) Bond Graphs: A Unifying Framework for Modelling of Physical Systems.
In: Carreira P., Amaral V., Vangheluwe H. (eds) Foundations of Multi-Paradigm Modelling for Cyber-Physical Systems. Springer, Cham.
https://doi.org/10.1007/978-3-030-43946-0_2

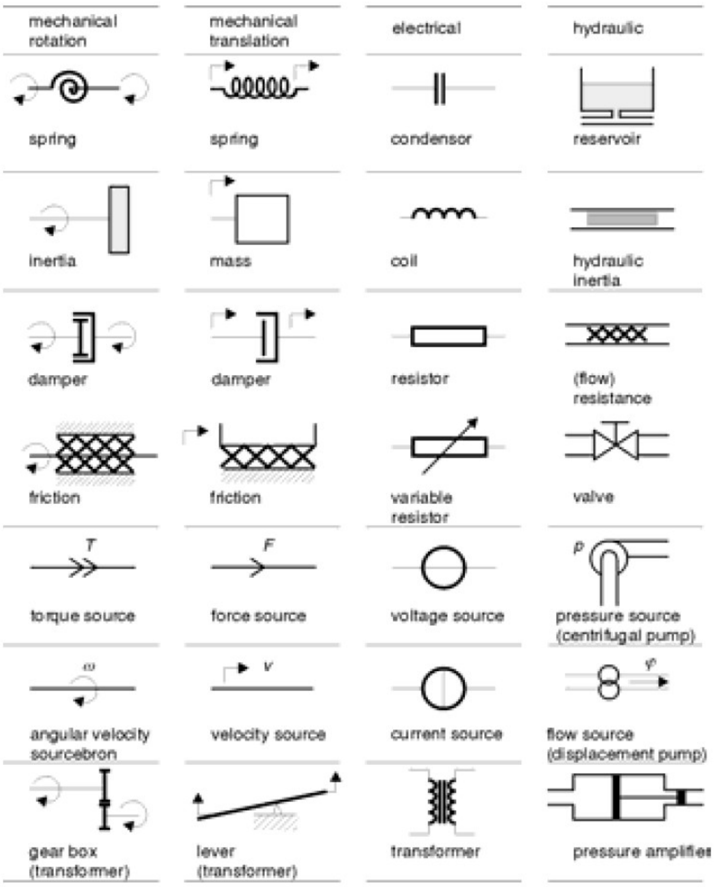
model using domain notation



Idealized Physical Model (IPM)
1D aka "lumped parameter"

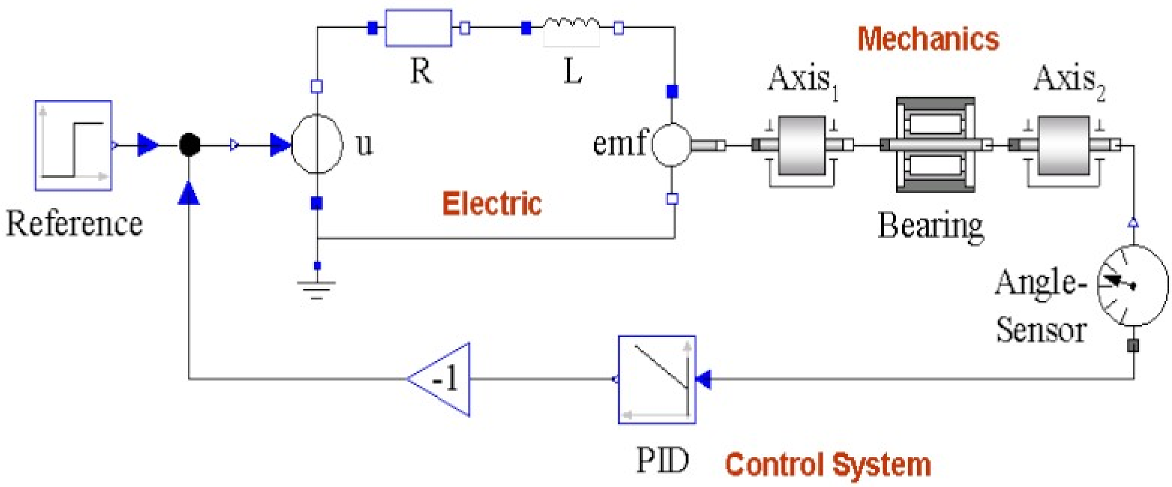


	f *flow*	E *effort*	$q = \int f\,dt$ *generalized displacement*	$p = \int e\,dt$ *generalized momentum*
Electromagnetic	i current	U voltage	$q = \int i dt$ charge	$\lambda = \int u dt$ magnetic flux linkage
mechanical translation	V velocity	F force	$x = \int v dt$ displacement	$p = \int F dt$ momentum
mechanical rotation	ω angular velocity	T torque	$\theta = \int \omega dt$ angular displacement	$b = \int T dt$ angular momentum
hydraulic/ pneumatic	φ volume flow	P pressure	$V = \int \varphi dt$ volume	$\Gamma = \int p dt$ momentum of a flow tube
Thermal	T temperature	F_S entropy flow	$S = \int f_S dt$ entropy	
Chemical	μ chemical potential	F_N molar flow	$N = \int f_N dt$ number of moles	


mechanical rotation
mechanical translation
electrical
hydraulic
spring
spring
condensor
reservoir
inertia
mass
coil
hydraulic inertia
damper
damper
resistor
(flow) resistance
friction
friction
variable resistor
valve
T
torque source
F
force source
voltage source
p
pressure source (centrifugal pump)
ω
angular velocity sourcebron
v
velocity source
current source
flow source (displacement pump)
gear box (transformer)
lever (transformer)
transformer
pressure amplifier

Multi-Domain Modeling

http://www.modelica.org



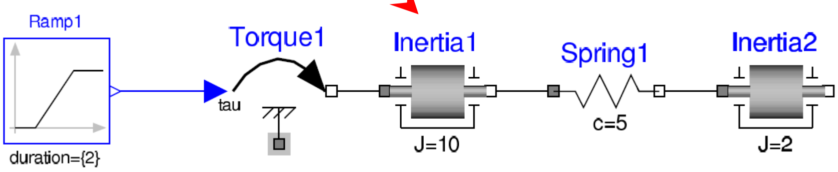
this slide from Peter Fritzson's Modelica tutorial

Multi-Domain Modeling

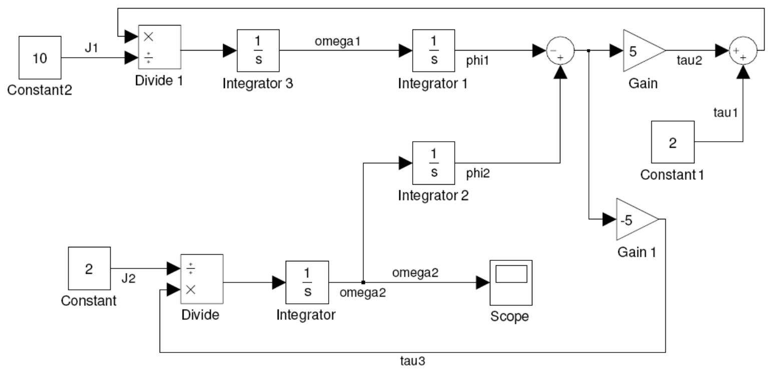
Visual Acausal Hierarchical Component Modeling

Keeps the physical structure

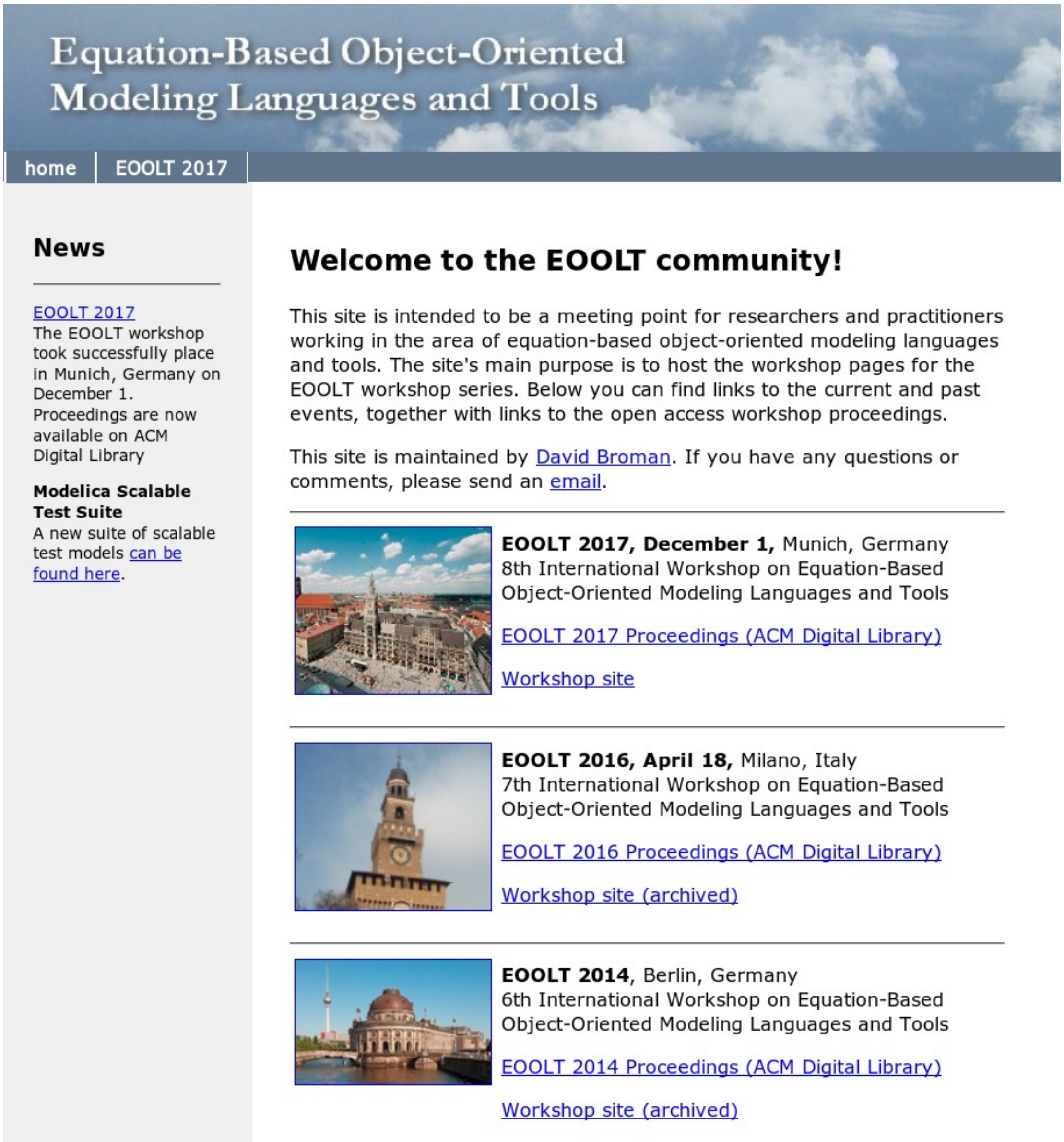
Acausal model (Modelica)



Causal block-based model (Simulink)



this slide from Peter Fritzson's Modelica tutorial



Equation-Based Object-Oriented Modeling Languages and Tools

home | EOOLT 2017

News

EOOLT 2017
The EOOLT workshop took successfully place in Munich, Germany on December 1. Proceedings are now available on ACM Digital Library

Modelica Scalable Test Suite
A new suite of scalable test models can be found here.

Welcome to the EOOLT community!

This site is intended to be a meeting point for researchers and practitioners working in the area of equation-based object-oriented modeling languages and tools. The site's main purpose is to host the workshop pages for the EOOLT workshop series. Below you can find links to the current and past events, together with links to the open access workshop proceedings.

This site is maintained by David Broman. If you have any questions or comments, please send an email.

EOOLT 2017, December 1, Munich, Germany
8th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools

EOOLT 2017 Proceedings (ACM Digital Library)

Workshop site

EOOLT 2016, April 18, Milano, Italy
7th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools

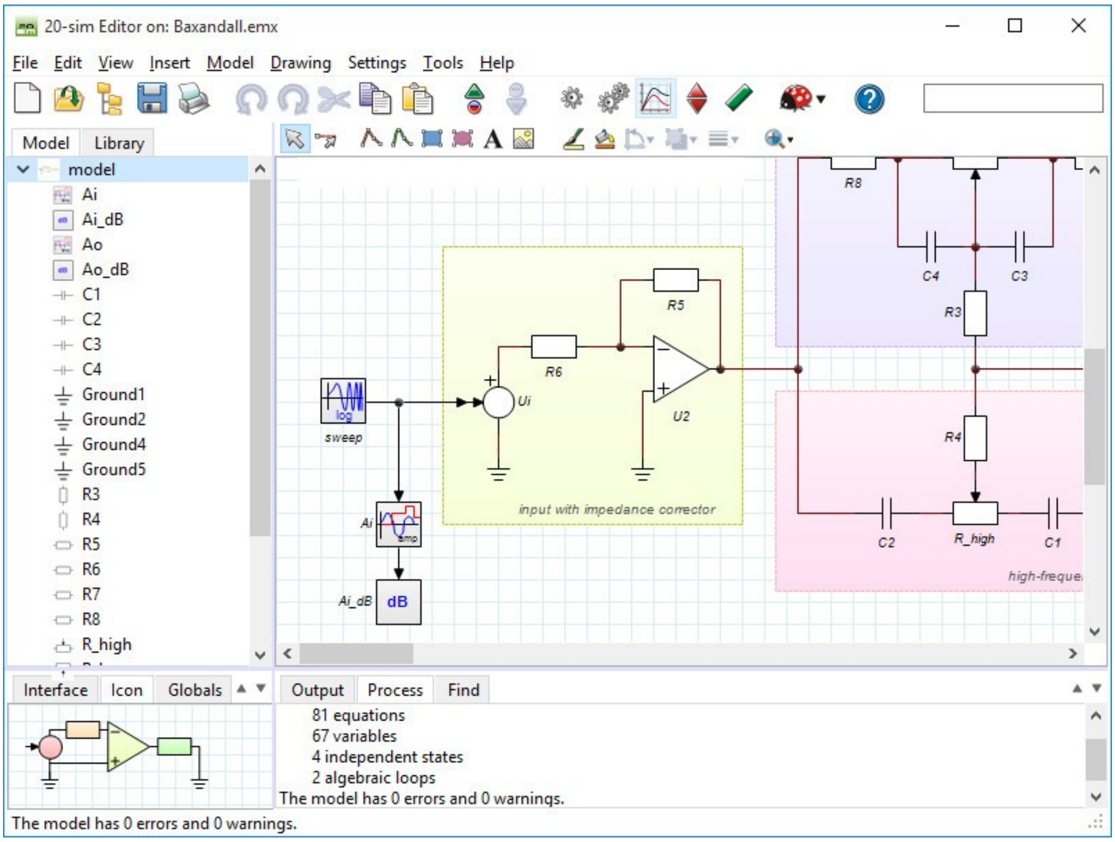
EOOLT 2016 Proceedings (ACM Digital Library)

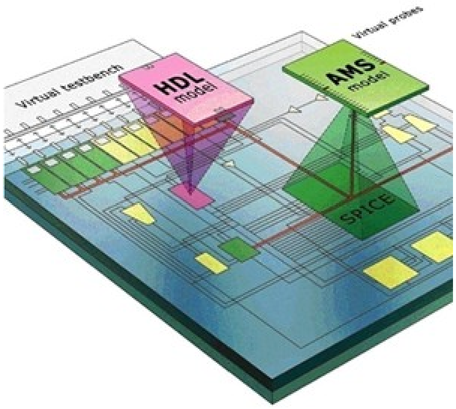
Workshop site (archived)

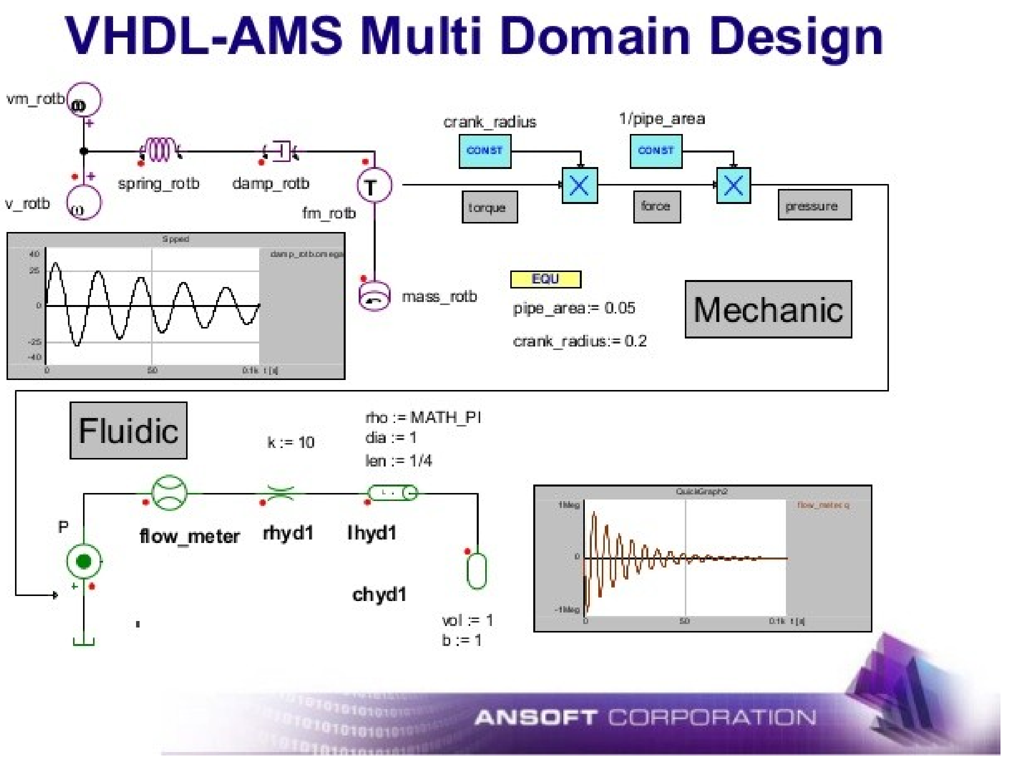
EOOLT 2014, Berlin, Germany
6th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools

EOOLT 2014 Proceedings (ACM Digital Library)

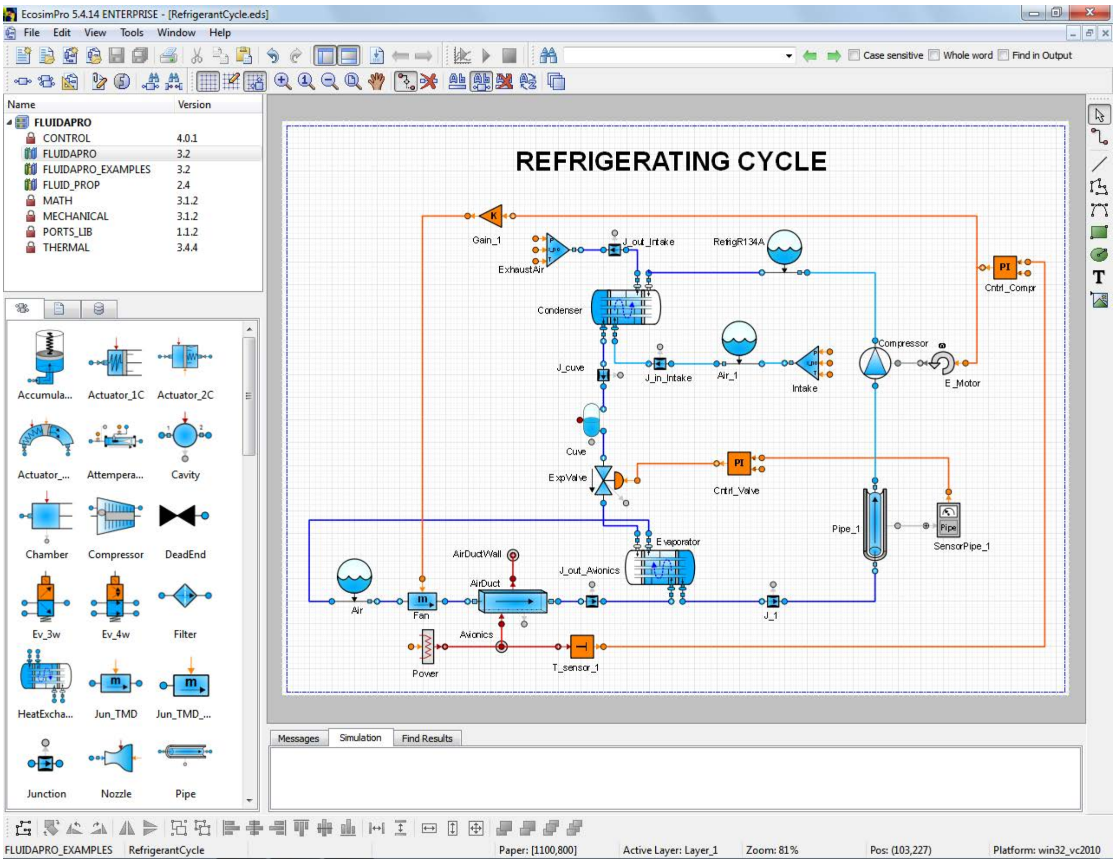
Workshop site (archived)

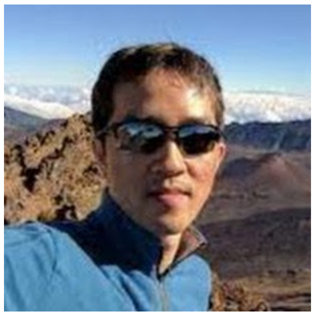

20-sim Editor on: Baxandall.emx
File Edit View Insert Model Drawing Settings Tools Help
Model Library
model
Ai
Ai_dB
Ao
Ao_dB
C1
C2
C3
C4
Ground1
Ground2
Ground4
Ground5
R3
R4
R5
R6
R7
R8
R_high
R8
C4
C3
R3
R5
R6
Ui
U2
sweep
log
R4
Ai
amp
input with impedance corrector
C2
R_high
C1
high-freque
Ai_dB
dB
Interface Icon Globals
Output Process Find
81 equations
67 variables
4 independent states
2 algebraic loops
The model has 0 errors and 0 warnings.
The model has 0 errors and 0 warnings.


Virtual testbench
Virtual probes
HDL model
AMS model
SPICE

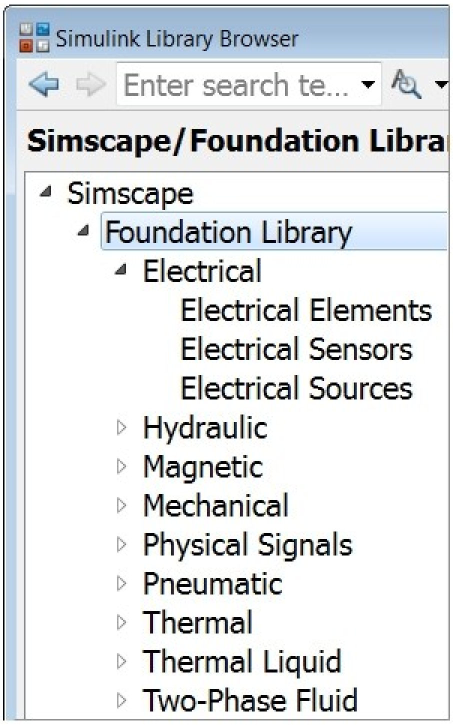

VHDL-AMS Multi Domain Design
vm_rotb
v_rotb
spring_rotb
damp_rotb
fm_rotb
mass_rotb
crank_radius
1/pipe_area
CONST
torque
force
pressure
EQU
pipe_area:= 0.05
crank_radius:= 0.2
Mechanic
Fluidic
k := 10
rho := MATH_PI
dia := 1
len := 1/4
P
flow_meter
rhyd1
lhyd1
chyd1
vol := 1
b := 1
ANSOFT CORPORATION

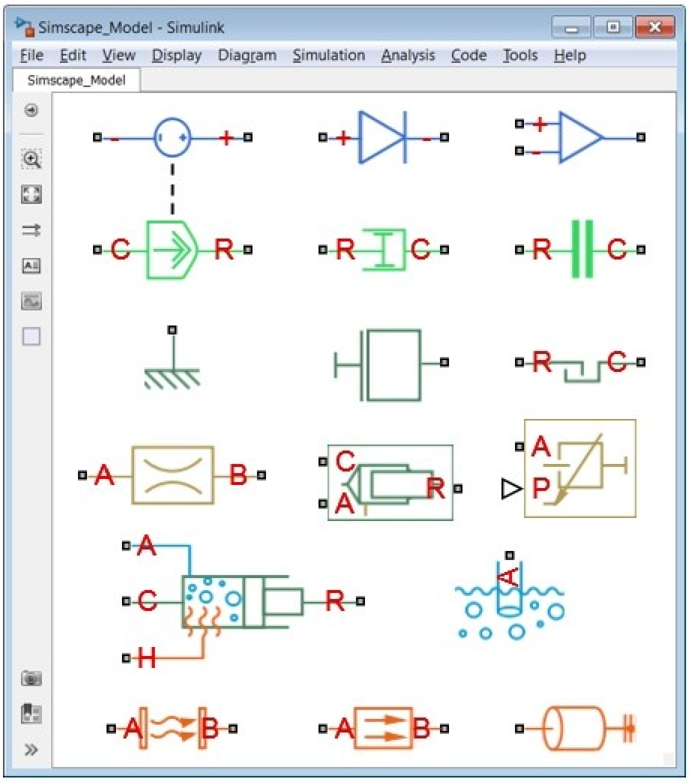
EcosimPro
Modelling and Simulation Software

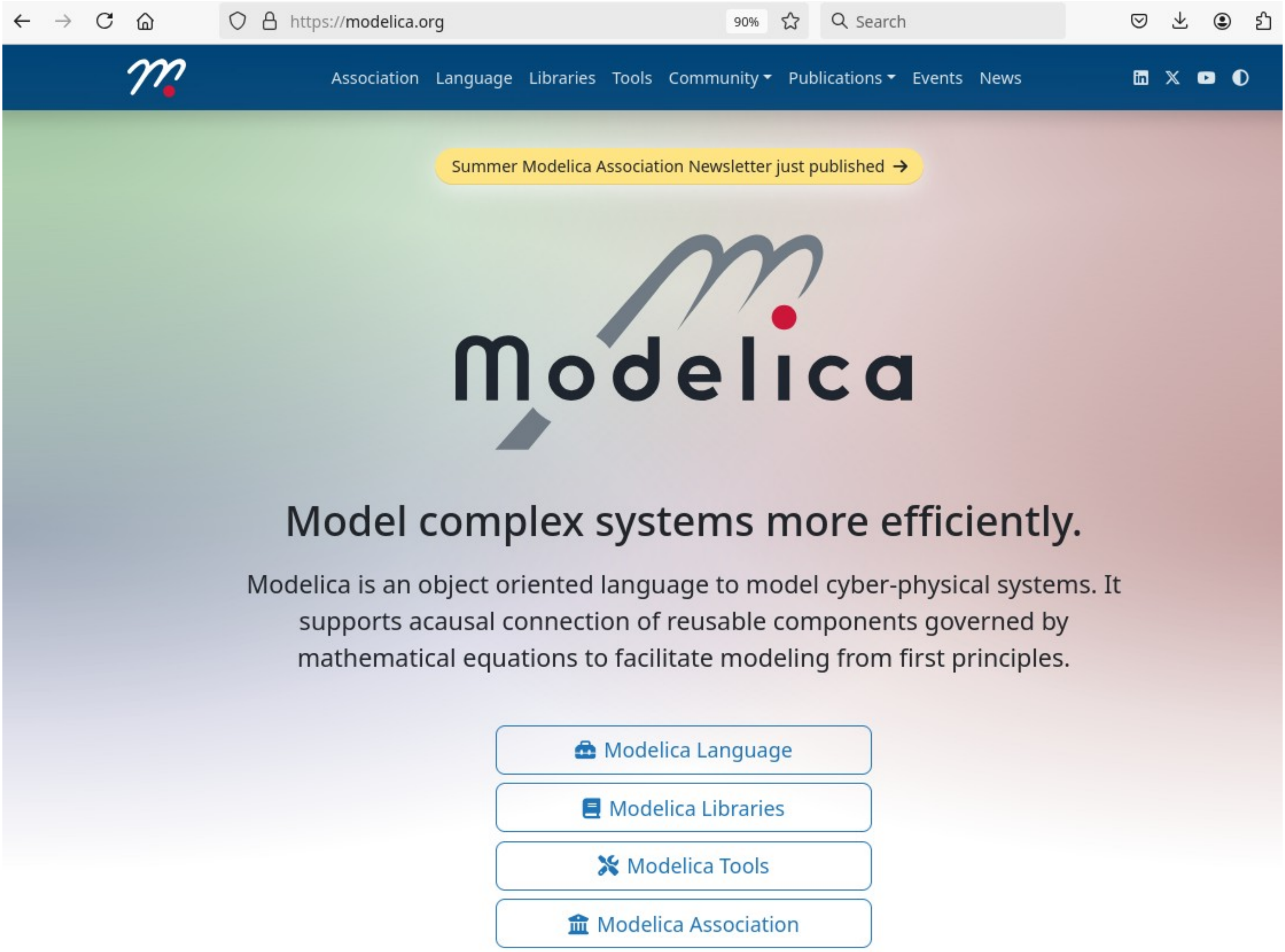




Steven Xu


Simulink Library Browser
Enter search te...
Simscape/Foundation Libra
Simscape
Foundation Library
Electrical
Electrical Elements
Electrical Sensors
Electrical Sources
Hydraulic
Magnetic
Mechanical
Physical Signals
Pneumatic
Thermal
Thermal Liquid
Two-Phase Fluid


Simscape_Model - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
Simscape_Model


https://modelica.org
90%
Search
Association
Language
Libraries
Tools
Community
Publications
Events
News
Summer Modelica Association Newsletter just published
Modelica
Model complex systems more efficiently.
Modelica is an object oriented language to model cyber-physical systems. It supports acausal connection of reusable components governed by mathematical equations to facilitate modeling from first principles.
Modelica Language
Modelica Libraries
Modelica Tools
Modelica Association

Dokumentutgivare
Lund Institute of Technology

Handläggare
Karl Johan Åström

Författare
Hilding Elmqvist

Dokumentnamn
REPORT

Dokumentbeteckning
LUTFD2/(TFRT-1015)/1-226/(1978)

Utgivningsdatum
May 1978

Ärendebeteckning

Dymola

Dokumenttitel och undertitel

A Structured Model Language for Large Continuous Systems

Referat (sammandrag)

A model language, called DYMOLA, for continuous dynamical systems is proposed. Large models are conveniently described hierarchically using a submodel concept. The ordinary differential equations and algebraic equations need not be converted to assignment statements. There is a concept, cut, which corresponds to connection mechanisms of complex types, and there are facilities to describe the connection structure of a system. A model can be manipulated for different purposes such as simulation and static calculations. The model equations are sorted and they are converted to assignment statements using formula manipulation. A translator for the model language is also included.

Referat skrivet av
Author

Förslag till ytterligare nyckelord
nonlinear systems, compiler, permutations, graph theory

Klassifikationssystem och -klass(er)

Indextermer (ange källa)
Mathematical models, Simulation languages, Computerized simulation, Nonlinear systems, Ordinary differential equations, Compilers. (Thesaurus of Engineering and Scientific Terms, Eng. Joint Council,USA)

Omfång
226 pages

Övriga bibliografiska uppgifter

Språk
English

Sekretessuppgifter

ISSN

ISBN

Dokumentet kan erhållas från
Department of Automatic Control
Lund Institute of Technology
P O Box 725, S-220 07 Lund 7, Sweden

Mottagarens uppgifter

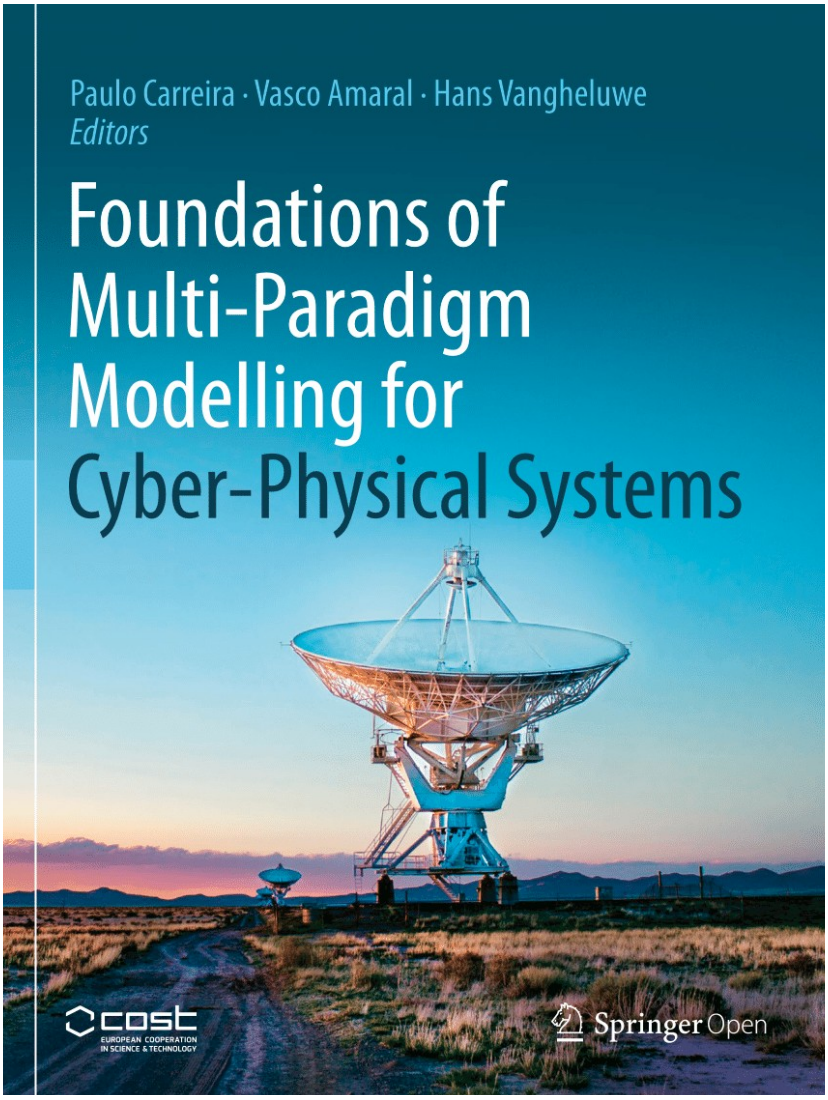
Pris

SIS-DB 1

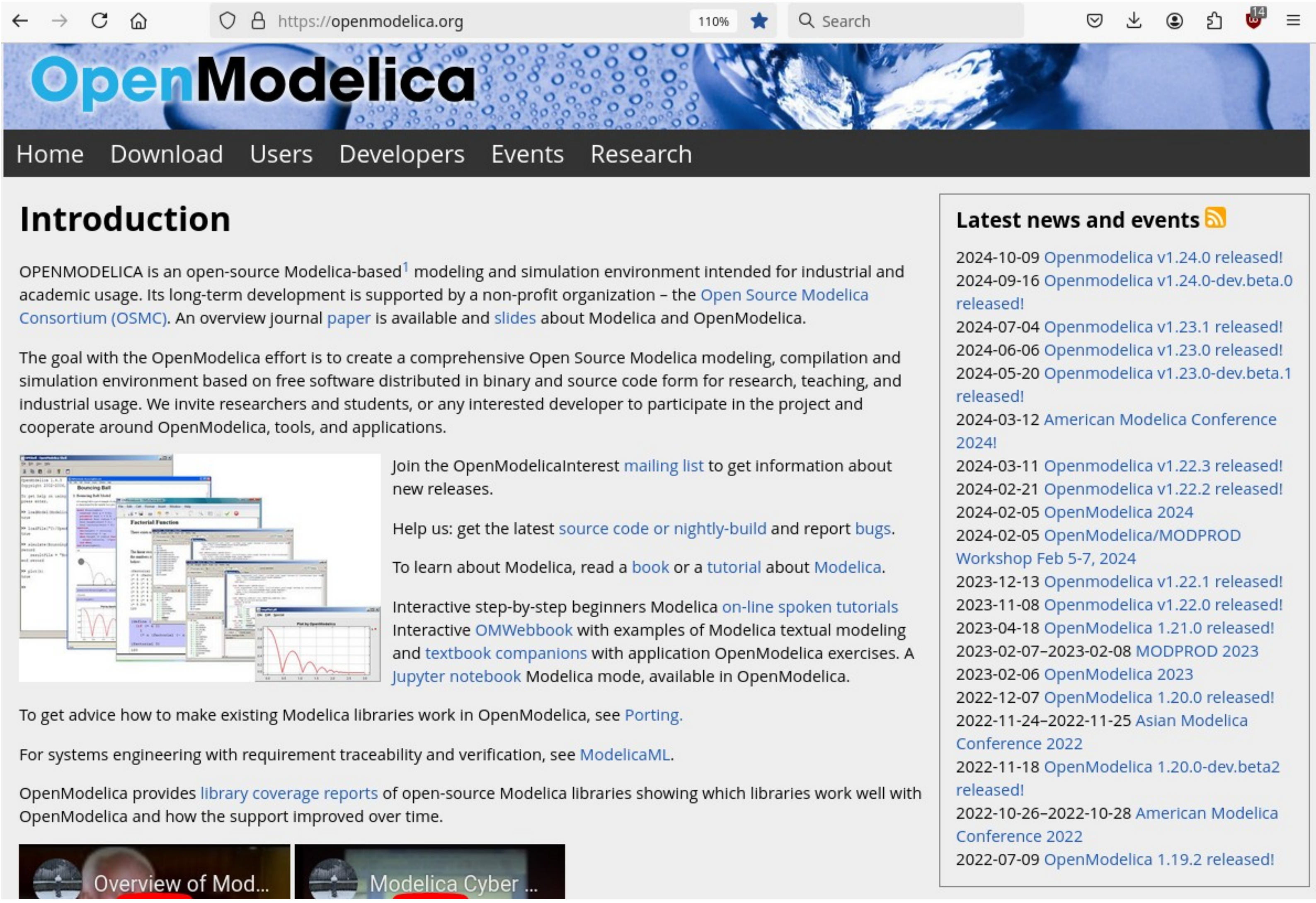
DOKUMENTDATABLAD enligt SIS 62 10 12

Blankett LU 11:25 1976–07

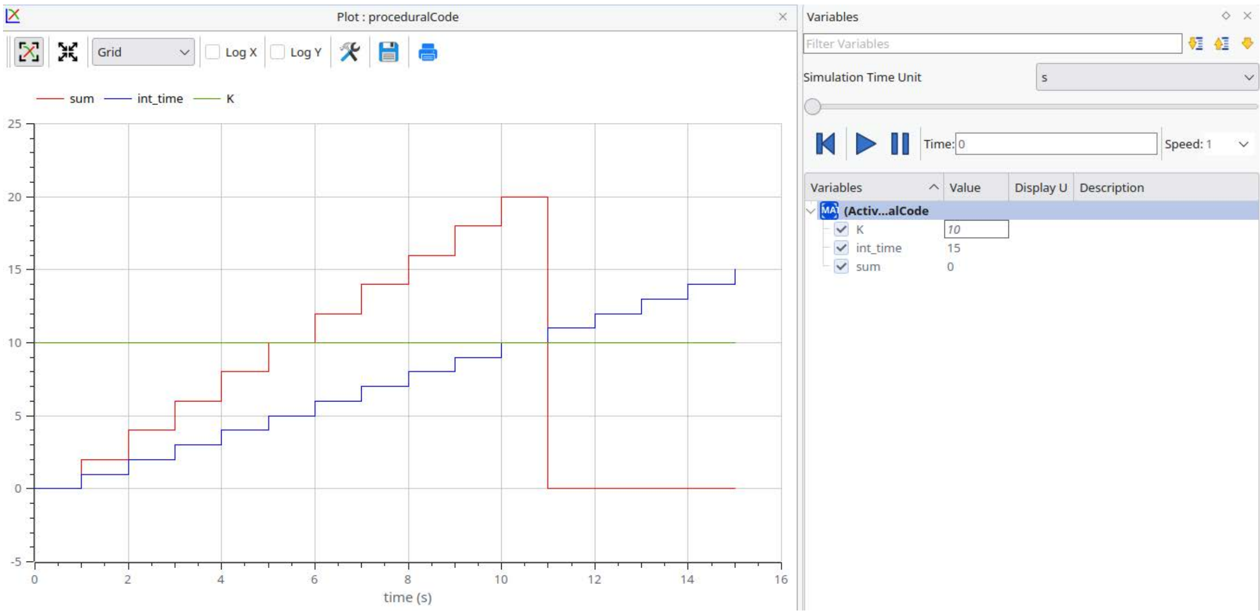

esprit
Simulation in Europe
SiE
ESPRIT Basic Research Working Group 8467
Simulation for the Future: New Concepts, Tools and Applications
Keywords:
simulation technologies, multi-paradigm modelling, solvers, standards, interoperability, industrial deployment, demonstrators, user-simulator interfaces
MODELICA
OpenModelica



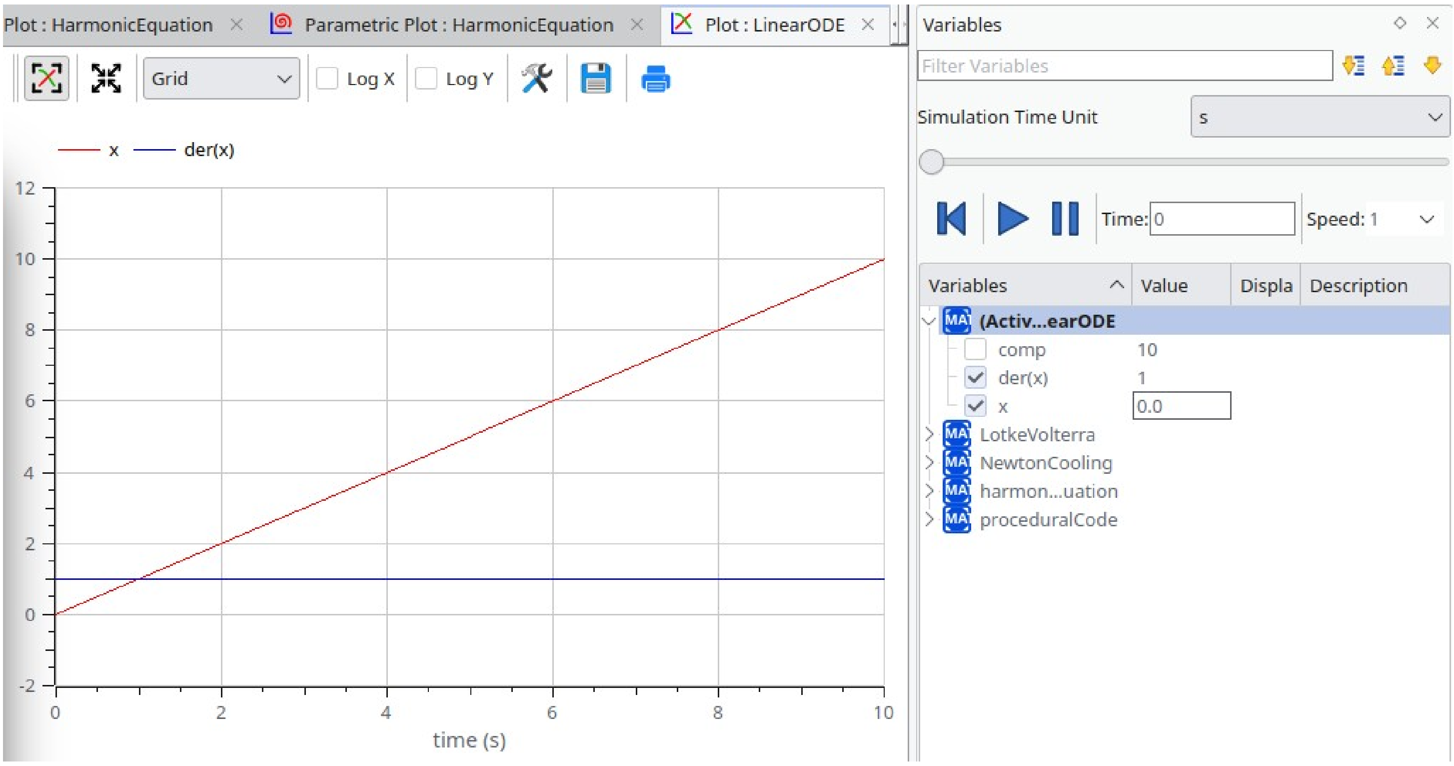
Fritzson P. (2020) Modelica: Equation-Based, Object-Oriented Modelling of Physical Systems.
In: Carreira P., Amaral V., Vangheluwe H. (eds) Foundations of Multi-Paradigm Modelling for Cyber-Physical Systems. Springer, Cham.
https://doi.org/10.1007/978-3-030-43946-0_3


https://openmodelica.org
110%
Search
OpenModelica
Home
Download
Users
Developers
Events
Research
Introduction
OPENMODELICA is an open-source Modelica-based[1] modeling and simulation environment intended for industrial and academic usage. Its long-term development is supported by a non-profit organization – the Open Source Modelica Consortium (OSMC). An overview journal paper is available and slides about Modelica and OpenModelica.
The goal with the OpenModelica effort is to create a comprehensive Open Source Modelica modeling, compilation and simulation environment based on free software distributed in binary and source code form for research, teaching, and industrial usage. We invite researchers and students, or any interested developer to participate in the project and cooperate around OpenModelica, tools, and applications.
Join the OpenModelicaInterest mailing list to get information about new releases.
Help us: get the latest source code or nightly-build and report bugs.
To learn about Modelica, read a book or a tutorial about Modelica.
Interactive step-by-step beginners Modelica on-line spoken tutorials Interactive OMWebbook with examples of Modelica textual modeling and textbook companions with application OpenModelica exercises. A Jupyter notebook Modelica mode, available in OpenModelica.
To get advice how to make existing Modelica libraries work in OpenModelica, see Porting.
For systems engineering with requirement traceability and verification, see ModelicaML.
OpenModelica provides library coverage reports of open-source Modelica libraries showing which libraries work well with OpenModelica and how the support improved over time.
Overview of Mod...
Modelica Cyber ...
Latest news and events
2024-10-09 Openmodelica v1.24.0 released!
2024-09-16 Openmodelica v1.24.0-dev.beta.0 released!
2024-07-04 Openmodelica v1.23.1 released!
2024-06-06 Openmodelica v1.23.0 released!
2024-05-20 Openmodelica v1.23.0-dev.beta.1 released!
2024-03-12 American Modelica Conference 2024!
2024-03-11 Openmodelica v1.22.3 released!
2024-02-21 Openmodelica v1.22.2 released!
2024-02-05 OpenModelica 2024
2024-02-05 OpenModelica/MODPROD Workshop Feb 5-7, 2024
2023-12-13 Openmodelica v1.22.1 released!
2023-11-08 Openmodelica v1.22.0 released!
2023-04-18 OpenModelica 1.21.0 released!
2023-02-07–2023-02-08 MODPROD 2023
2023-02-06 OpenModelica 2023
2022-12-07 OpenModelica 1.20.0 released!
2022-11-24–2022-11-25 Asian Modelica Conference 2022
2022-11-18 OpenModelica 1.20.0-dev.beta2 released!
2022-10-26–2022-10-28 American Modelica Conference 2022
2022-07-09 OpenModelica 1.19.2 released!

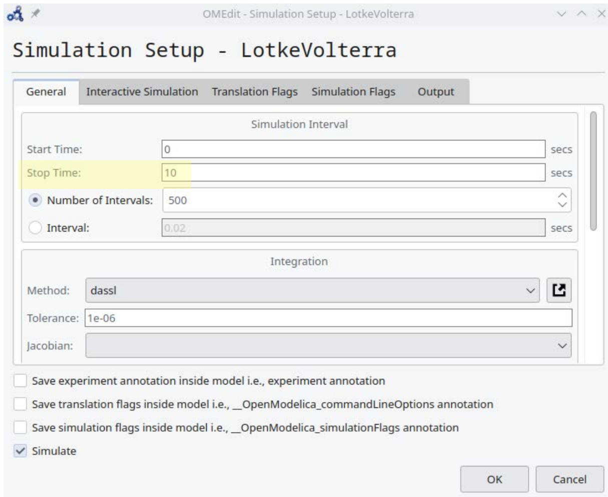
```
package functionExample

function times_two_upto_K "multiplies by two up to K"
  input Integer N  "input";
  input Integer K  "beyond K, result will be 0";
  output Integer result;
algorithm
  result := if N <= K then 2*N else 0;
end times_two_upto_K;

model proceduralCode
  Integer sum(start = 0);
  Integer int_time;
  parameter Integer K = 10;
equation
  int_time = integer(time);
  sum = times_two_upto_K(int_time, K); // or implicit alternative
end proceduralCode;

end functionExample;
```

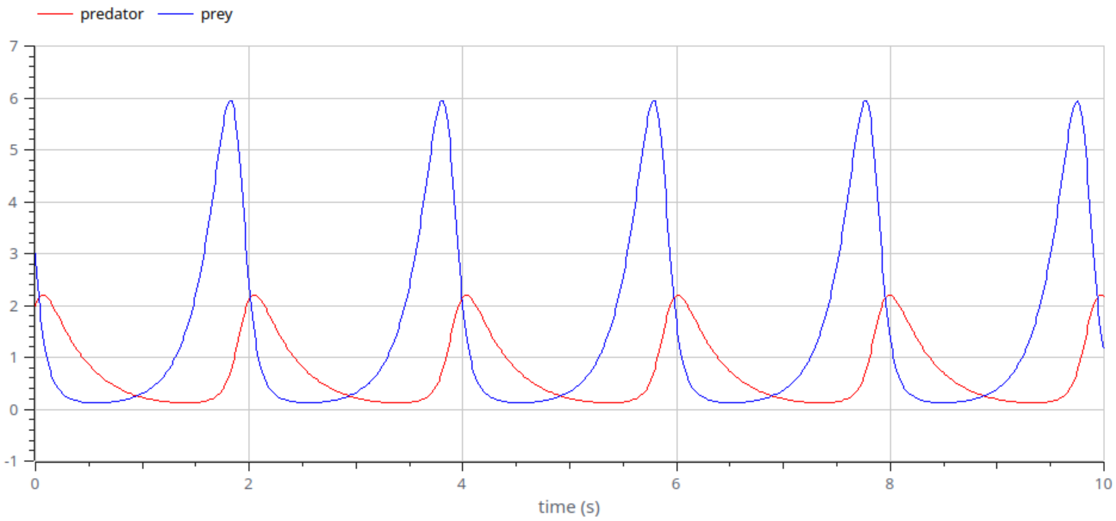


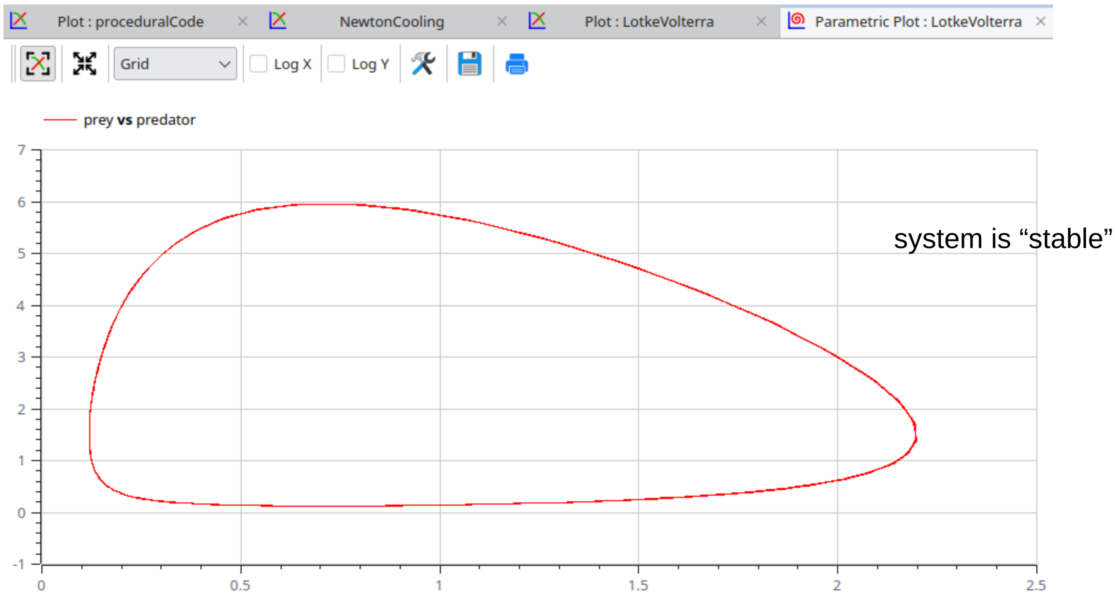
```
model LinearODE
  Real x(start = 0);
  Real comp;
equation
  der(x) =  1;
  comp - x = 0;
end LinearODE;

// x(t) = A * t + B
// x(0) = 0 = B
// dx/dt (0) = A = 1
```



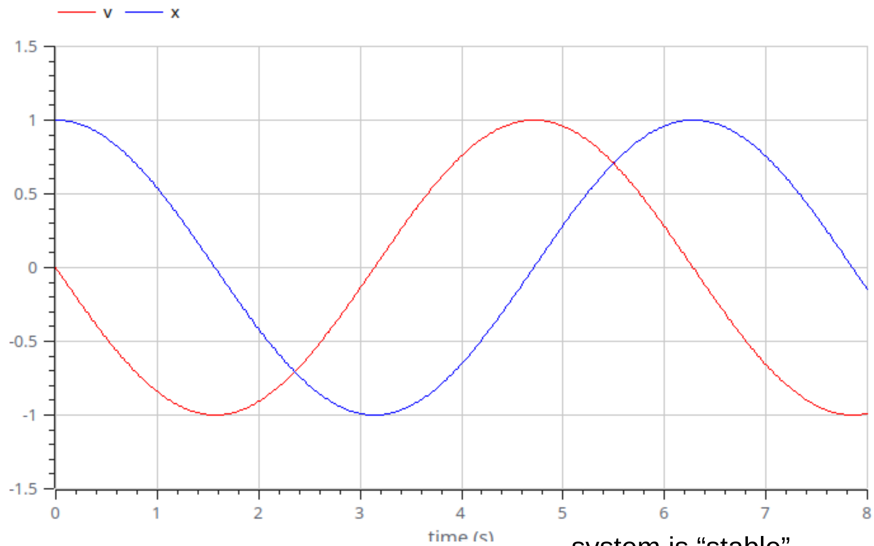
```
model LotkeVolterra "Lotke-Volterra equations modelling a predator-prey system"
  // Types
  type Population=Real(min=0);

  // Parameters
  parameter Population predator_0 = 2 "initial predator population";
  parameter Population prey_0 = 3 "initial prey population";
  parameter Real grazing_factor = 2;
  parameter Real kill_factor    = 7;
  parameter Real excess_death_rate  = 3;
  parameter Real excess_birth_rate  = 5;

  // Variables
  Population predator "predator population";
  Population prey "prey population";

initial equation
  predator = predator_0;
  prey = prey_0;

equation
  der(predator) = -excess_death_rate*predator + grazing_factor*predator*prey;
  der(prey)     =  excess_birth_rate*prey     - kill_factor*predator*prey;
end LotkeVolterra;
```



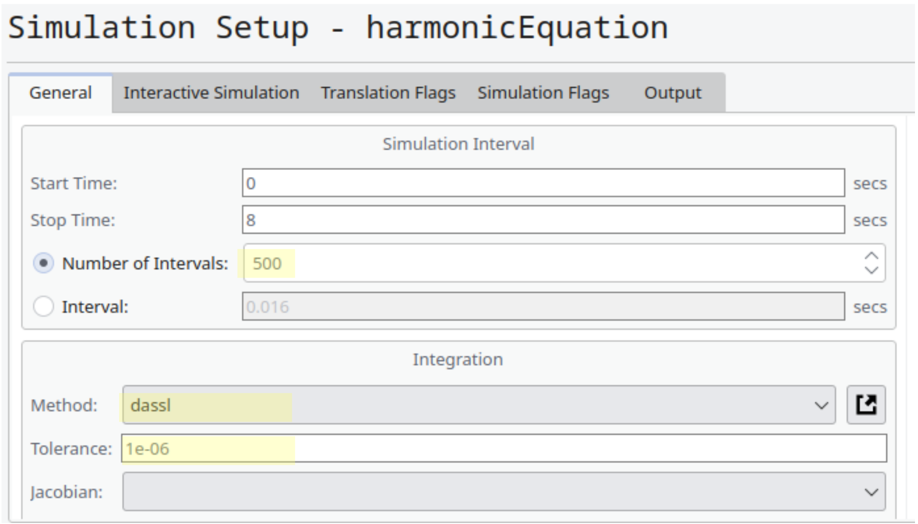

predator
prey
7
6
5
4
3
2
1
0
-1
0
2
4
6
8
10
time (s)

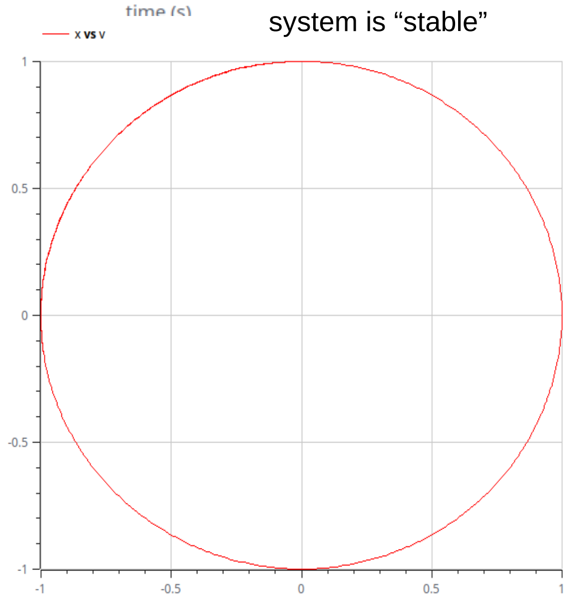

Plot : proceduralCode
NewtonCooling
Plot : LotkeVolterra
Parametric Plot : LotkeVolterra
Grid
Log X
Log Y
prey vs predator
7
6
5
4
3
2
1
0
-1
0
0.5
1
1.5
2
2.5
system is “stable”

```
model harmonicEquation
  Real x(start = 1);
  Real v(start = 0);
  Real comp;
equation
  der(x) =  v;
  der(v) = -x;
  comp - x = 0;
end harmonicEquation;

// x(t) = A * sin(t) + B * cos(t)
// x(0) = 1 = Boolean
// v(t) = A * cos(t) - B * sin(t)
// v(0) = 0 = A
//
// x(t) = cos(t)
```

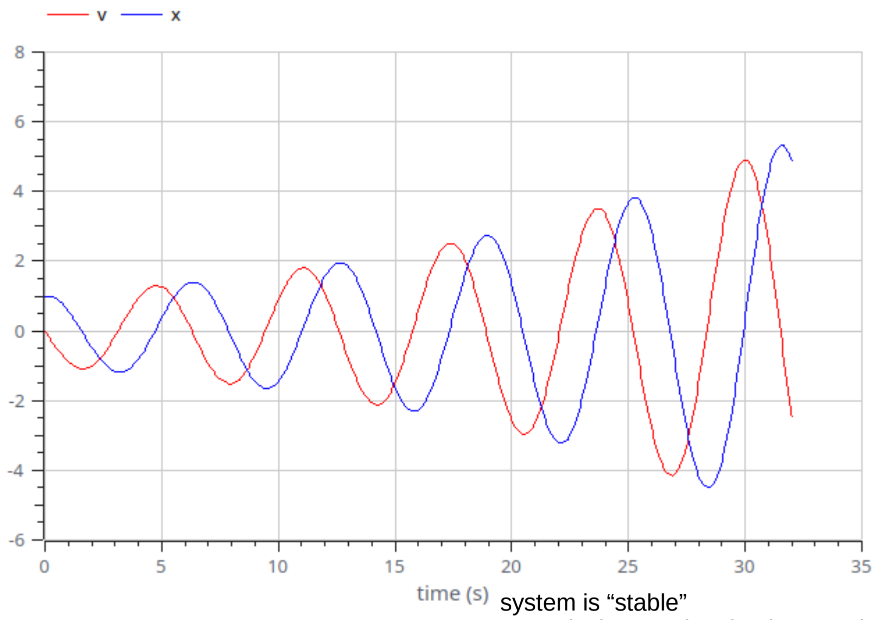


system is “stable”

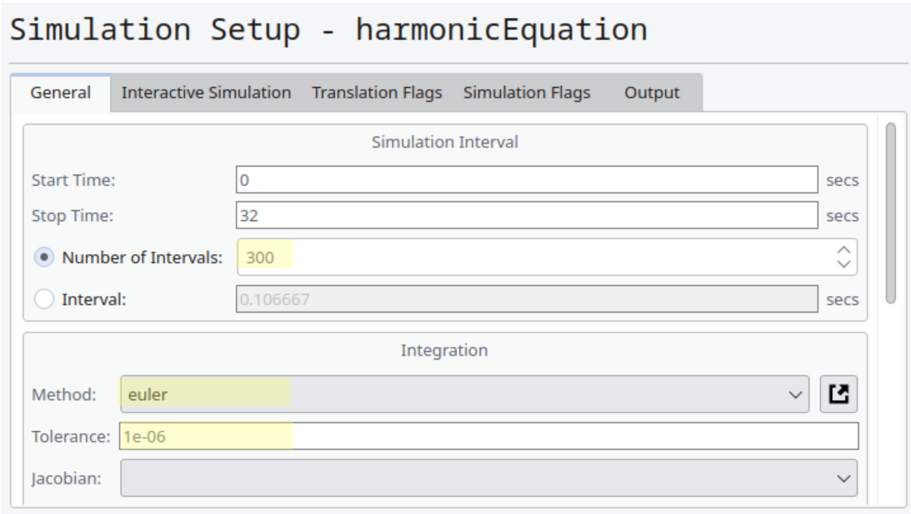


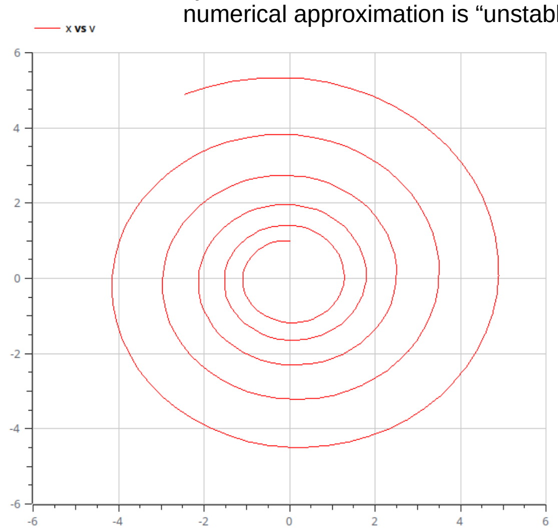


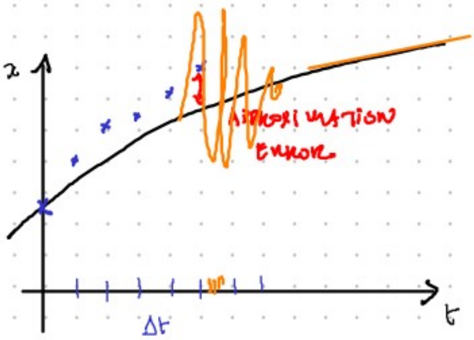
```
model harmonicEquation
  Real x(start = 1);
  Real v(start = 0);
  Real comp;
equation
  der(x) =  v;
  der(v) = -x;
  comp - x = 0;
end harmonicEquation;

// x(t) = A * sin(t) + B * cos(t)
// x(0) = 1 = Boolean
// v(t) = A * cos(t) - B * sin(t)
// v(0) = 0 = A
//
// x(t) = cos(t)
```



system is "stable"
numerical approximation is "unstable"







RATE OF CHANGE

SLOPE OF x(t) at t

$$\begin{cases} \frac{d\bar{x}}{dt} = f(\bar{x}, t) \\ \bar{x}(0) = \bar{x}_0 \end{cases}$$

ODE, IVP

DAE

$\bar{x}(t)$?

$$\frac{dx}{dt} = \lim_{\Delta t \to 0} \frac{x(t+\Delta t) - x(t)}{\Delta t}$$

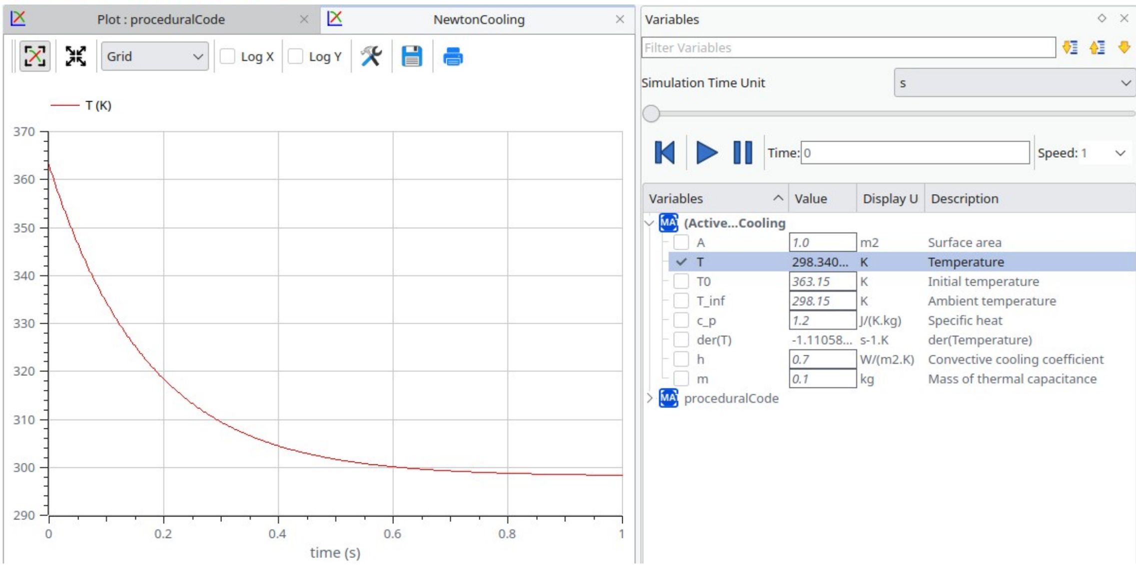
$\Rightarrow$

$$\frac{dx_{APPROX}}{dt} \underset{(\Delta t \ll)}{\approx} \frac{x_A(t+\Delta t) - x_A(t)}{\Delta t}$$

$$x_A(t+\Delta t) = x_A(t) + \Delta t \cdot \underbrace{\frac{dx_A}{dt}}_{} \quad \text{"}f(x_A, t)\text{"}$$

"EULER"

```
model NewtonCooling "Cooling example with physical types"
  // Types
  type Temperature=Real(unit="K", min=0);
  type ConvectionCoefficient=Real(unit="W/(m2.K)", min=0);
  type Area=Real(unit="m2", min=0);
  type Mass=Real(unit="kg", min=0);
  type SpecificHeat=Real(unit="J/(K.kg)", min=0);

  // Parameters
  parameter Temperature T_inf=298.15 "Ambient temperature";
  parameter Temperature T0=363.15 "Initial temperature";
  parameter ConvectionCoefficient h=0.7 "Convective cooling coefficient";
  parameter Area A=1.0 "Surface area";
  parameter Mass m=0.1 "Mass of thermal capacitance";
  parameter SpecificHeat c_p=1.2 "Specific heat";

  // Variables
  Temperature T "Temperature";
initial equation
  T = T0 "Specify initial value for T";
equation
  m*c_p*der(T) = h*A*(T_inf-T) "Newton's law of cooling";
end NewtonCooling;
```



```
hv@sanderling 66% pwd
/home/hv/src/courses/24.BIPtwinning/Modelica/ModelicaTutorial/
generatedCode/NewtonCooling
hv@sanderling 67% ls
NewtonCooling              NewtonCooling_prof.intdata
NewtonCooling_info.json    NewtonCooling_prof.realdata
NewtonCooling_init.xml     NewtonCooling_res.mat
NewtonCooling_JacA.bin     simulate.py
NewtonCooling.log _
```

Calibration / Parameter Estimation

Newton Cooling Model

```
model NewtonCooling "Cooling example with physical types"
  // Types
  type Temperature=Real(unit="K", min=0);
  type ConvectionCoefficient=Real(unit="W/(m2.K)", min=0);
  type Area=Real(unit="m2", min=0);
  type Mass=Real(unit="kg", min=0);
  type SpecificHeat=Real(unit="J/(K.kg)", min=0);

  // Parameters
  parameter Temperature T_inf=298.15 "Ambient temperature";
  parameter Temperature T0=363.15 "Initial temperature";
  parameter ConvectionCoefficient h=0.7 "Convective cooling coefficient";
  parameter Area A=1.0 "Surface area";
  parameter Mass m=0.1 "Mass of thermal capacitance";
  parameter SpecificHeat c_p=1.2 "Specific heat";

  // Variables
  Temperature T "Temperature";
initial equation
  T = T0 "Specify initial value for T";
equation
  m*c_p*der(T) = h*A*(T_inf-T) "Newton's law of cooling";
end NewtonCooling;
```

Parametrized model

Newton Cooling Model

```
model NewtonCooling "Cooling example with physical types"
  // Types
  type Temperature=Real(unit="K", min=0);
  type ConvectionCoefficient=Real(unit="W/(m2.K)", min=0);
  type Area=Real(unit="m2", min=0);
  type Mass=Real(unit="kg", min=0);
  type SpecificHeat=Real(unit="J/(K.kg)", min=0);

  // Parameters
  parameter Temperature T_inf=298.15 "Ambient temperature";
  parameter Temperature T0=363.15 "Initial temperature";
  parameter ConvectionCoefficient h=0.7 "Convective cooling coefficient";
  parameter Area A=1.0 "Surface area";
  parameter Mass m=0.1 "Mass of thermal capacitance";
  parameter SpecificHeat c_p=1.2 "Specific heat";

  // Variables
  Temperature T "Temperature";
initial equation
  T = T0 "Specify initial value for T";
equation
  m*c_p*der(T) = h*A*(T_inf-T) "Newton's law of cooling";
end NewtonCooling;
```

can be measured (T_inf, T0, m)

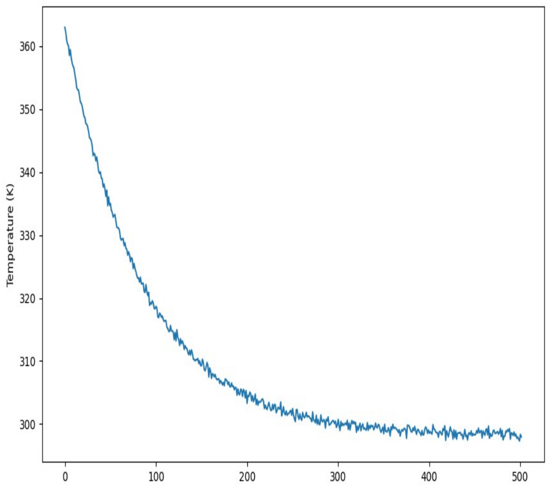
can be calculated (A, c_p)

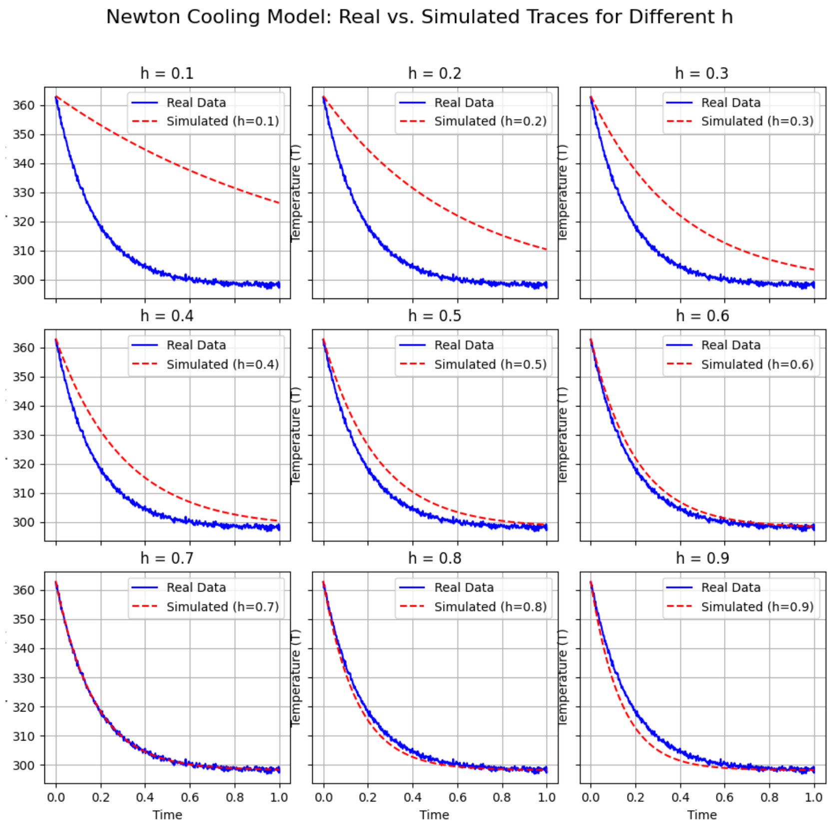
Newton Cooling Model

```
model NewtonCooling "Cooling example with physical types"
  // Types
  type Temperature=Real(unit="K", min=0);
  type ConvectionCoefficient=Real(unit="W/(m2.K)", min=0);
  type Area=Real(unit="m2", min=0);
  type Mass=Real(unit="kg", min=0);
  type SpecificHeat=Real(unit="J/(K.kg)", min=0);

  // Parameters
  parameter Temperature T_inf=298.15 "Ambient temperature";
  parameter Temperature T0=363.15 "Initial temperature";
  parameter ConvectionCoefficient h=0.7 "Convective cooling coefficient";
  parameter Area A=1.0 "Surface area";
  parameter Mass m=0.1 "Mass of thermal capacitance";
  parameter SpecificHeat c_p=1.2 "Specific heat";

  // Variables
  Temperature T "Temperature";
initial equation
  T = T0 "Specify initial value for T";
equation
  m*c_p*der(T) = h*A*(T_inf-T) "Newton's law of cooling";
end NewtonCooling;
```

has to be estimated

Newton Cooling Model



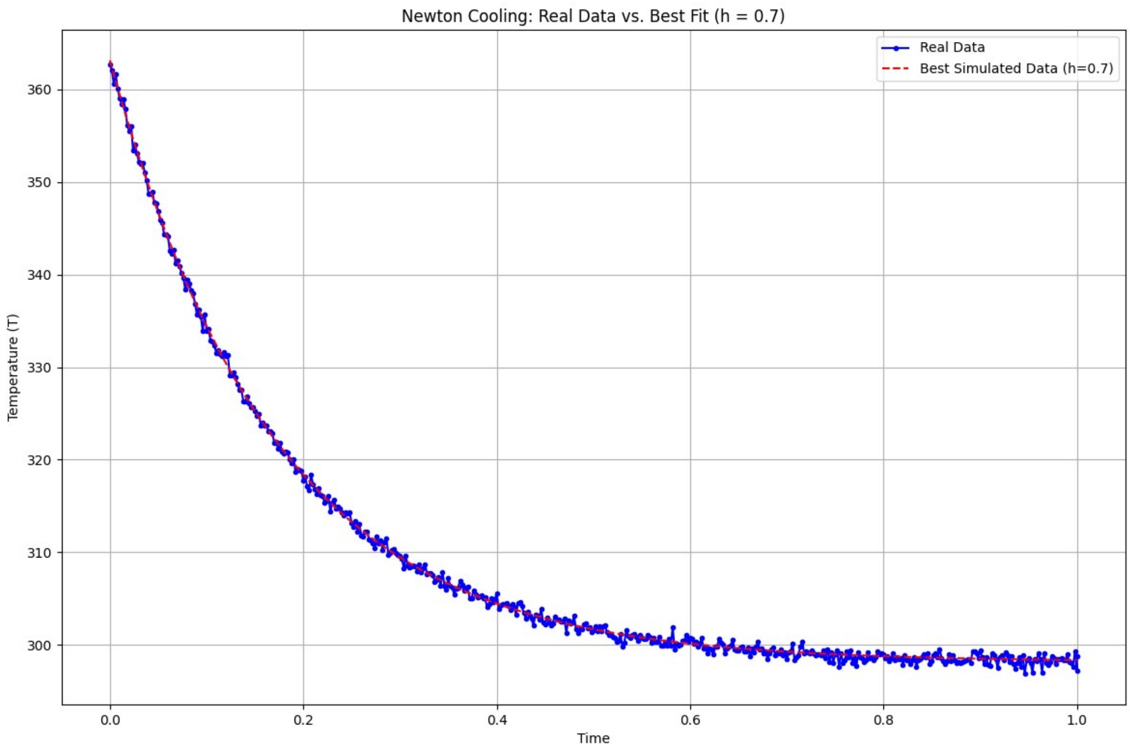


Trace from experiment on real system data with sensor inaccuracies, noise ...

Compare

Multiple traces from simulation with different parameter values for *h*

- distance metric
- search for parameter that yields the smallest distance



Best matching trace →
The corresponding parameter from that simulation →
The best estimate for the parameter value

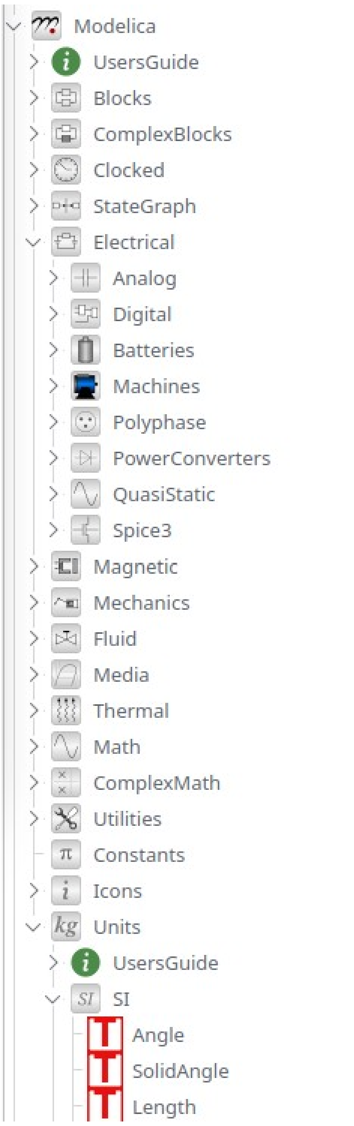
OO Modelling of Physical Systems

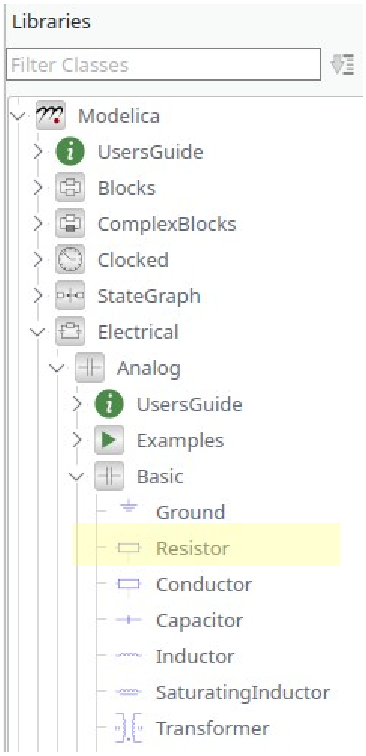
Electrical Types

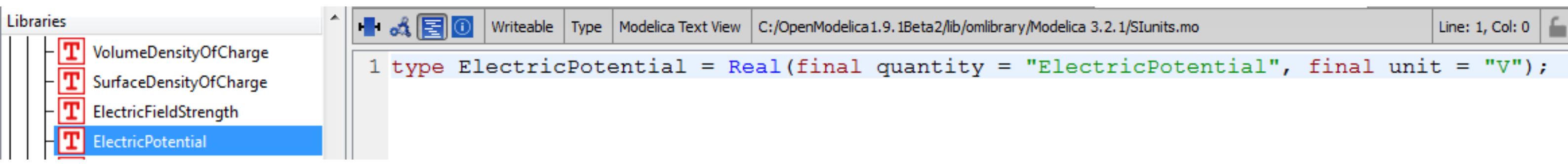
```
type Time = Real (final quantity="Time", final unit="s");
type ElectricPotential = Real (final quantity="ElectricPotential",
                               final unit="V");
type Voltage = ElectricPotential;
type ElectricCurrent = Real (final quantity="ElectricCurrent",
                             final unit="A");
type Current = ElectricCurrent;
```

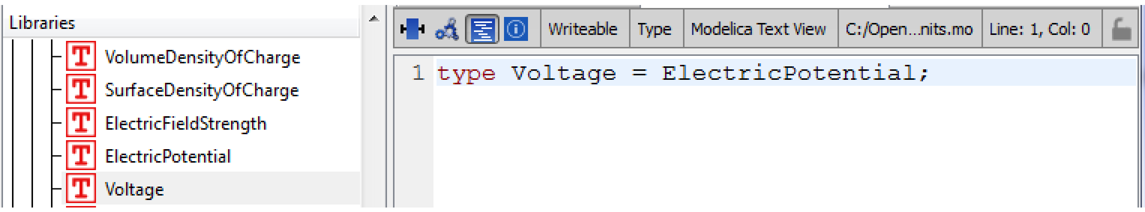
Beware: variables are **signals** (functions of **time**)!

Standard Library (MSL)



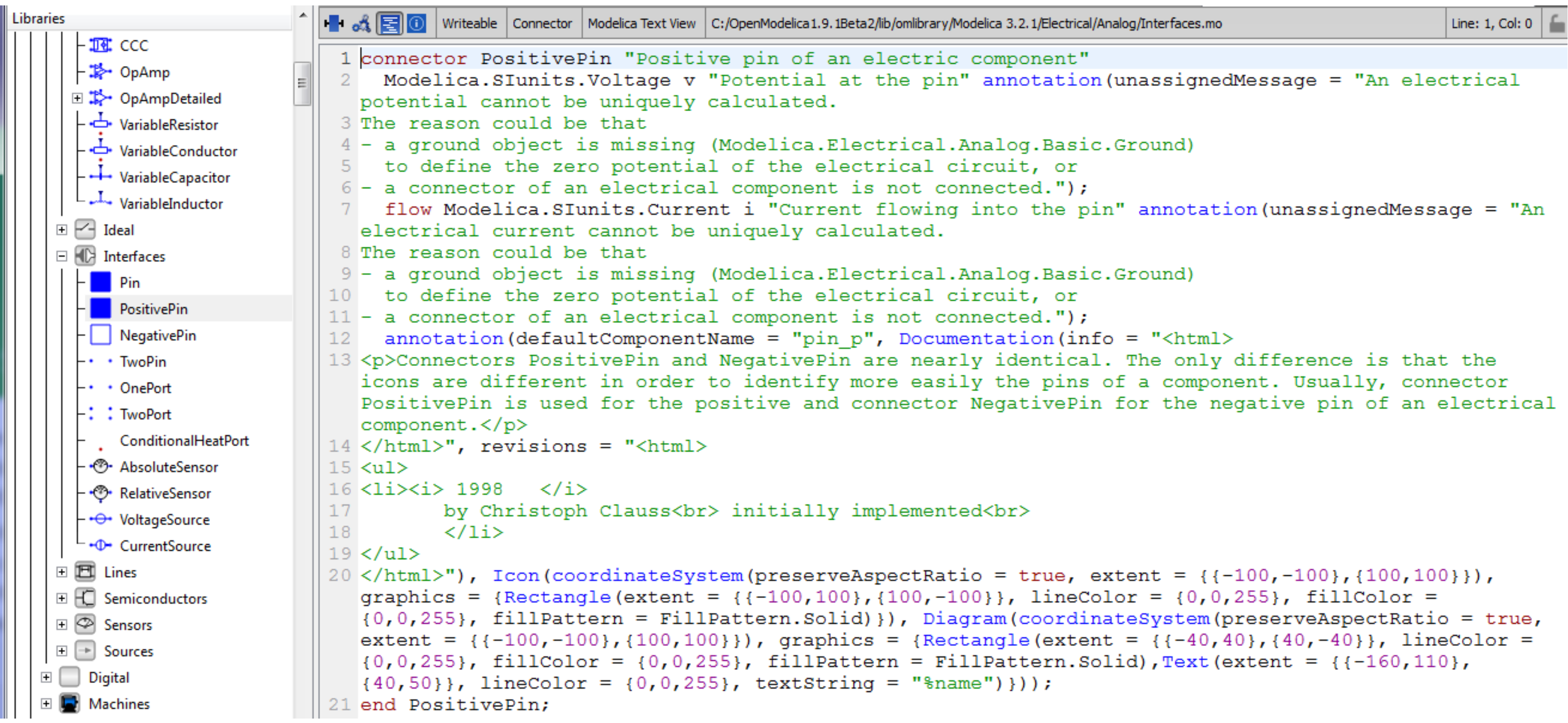



Libraries
VolumeDensityOfCharge
SurfaceDensityOfCharge
ElectricFieldStrength
ElectricPotential
Writeable
Type
Modelica Text View
C:/OpenModelica1.9.1Beta2/lib/omlibrary/Modelica 3.2.1/SIunits.mo
Line: 1, Col: 0
1 type ElectricPotential = Real(final quantity = "ElectricPotential", final unit = "V");


Libraries
VolumeDensityOfCharge
SurfaceDensityOfCharge
ElectricFieldStrength
ElectricPotential
Voltage
Writeable
Type
Modelica Text View
C:/Open...nits.mo
Line: 1, Col: 0
1 type Voltage = ElectricPotential;

Electrical Pin Interface

```
connector PositivePin "Positive pin of an electric component"
       Voltage v "Potential at the pin";
  flow Current i "Current flowing into the pin";
end PositivePin;
```

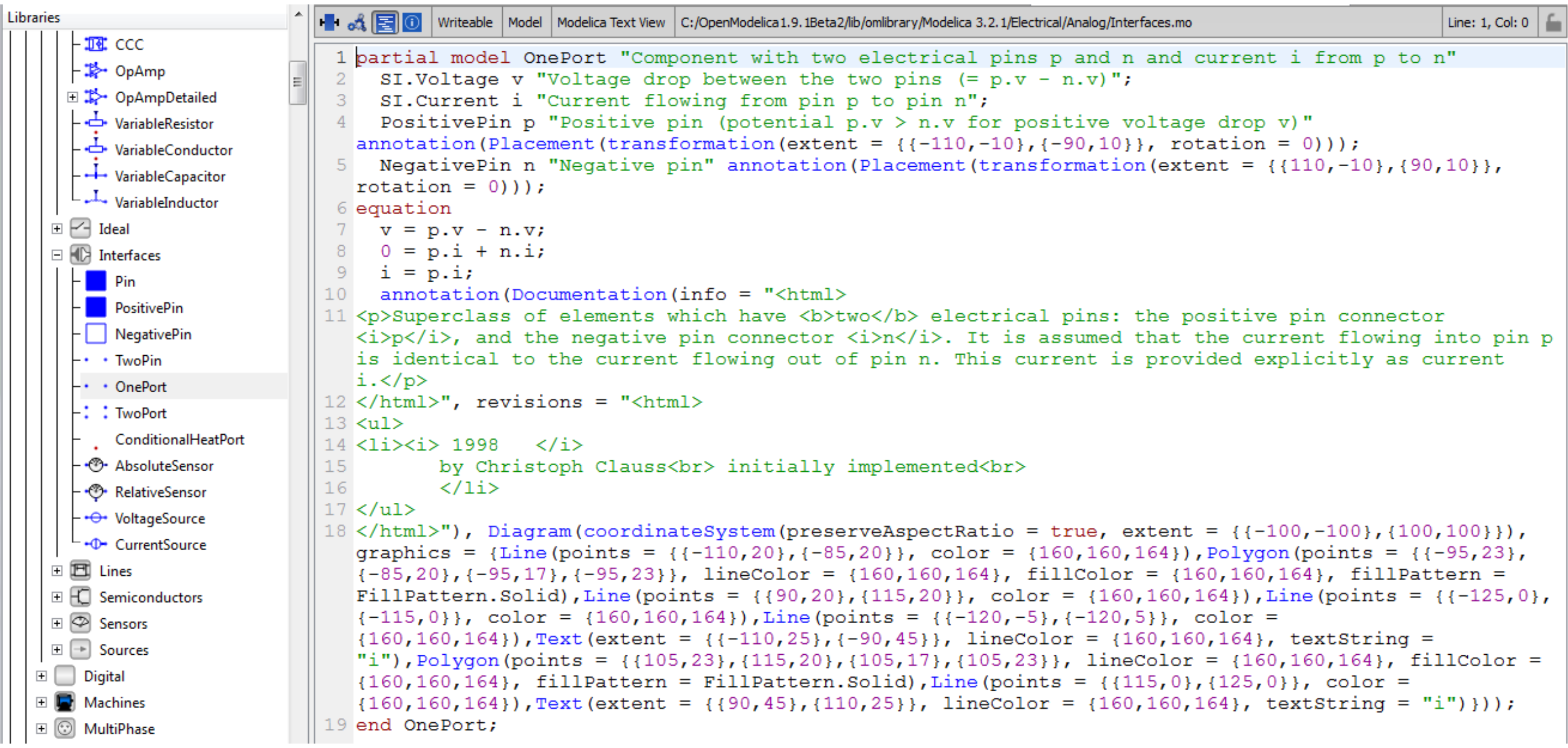

Libraries
CCC
OpAmp
OpAmpDetailed
VariableResistor
VariableConductor
VariableCapacitor
VariableInductor
Ideal
Interfaces
Pin
PositivePin
NegativePin
TwoPin
OnePort
TwoPort
ConditionalHeatPort
AbsoluteSensor
RelativeSensor
VoltageSource
CurrentSource
Lines
Semiconductors
Sensors
Sources
Digital
Machines
Writeable
Connector
Modelica Text View
C:/OpenModelica1.9.1Beta2/lib/omlibrary/Modelica 3.2.1/Electrical/Analog/Interfaces.mo
Line: 1, Col: 0
connector PositivePin "Positive pin of an electric component"
Modelica.SIunits.Voltage v "Potential at the pin" annotation(unassignedMessage = "An electrical potential cannot be uniquely calculated.
The reason could be that
- a ground object is missing (Modelica.Electrical.Analog.Basic.Ground)
to define the zero potential of the electrical circuit, or
- a connector of an electrical component is not connected.");
flow Modelica.SIunits.Current i "Current flowing into the pin" annotation(unassignedMessage = "An electrical current cannot be uniquely calculated.
The reason could be that
- a ground object is missing (Modelica.Electrical.Analog.Basic.Ground)
to define the zero potential of the electrical circuit, or
- a connector of an electrical component is not connected.");
annotation(defaultComponentName = "pin_p", Documentation(info = "<html>
<p>Connectors PositivePin and NegativePin are nearly identical. The only difference is that the icons are different in order to identify more easily the pins of a component. Usually, connector PositivePin is used for the positive and connector NegativePin for the negative pin of an electrical component.</p>
</html>", revisions = "<html>

<i> 1998 </i>
by Christoph Clauss
 initially implemented

</html>"), Icon(coordinateSystem(preserveAspectRatio = true, extent = {{-100,-100},{100,100}}), graphics = {Rectangle(extent = {{-100,100},{100,-100}}, lineColor = {0,0,255}, fillColor = {0,0,255}, fillPattern = FillPattern.Solid)}), Diagram(coordinateSystem(preserveAspectRatio = true, extent = {{-100,-100},{100,100}}), graphics = {Rectangle(extent = {{-40,40},{40,-40}}, lineColor = {0,0,255}, fillColor = {0,0,255}, fillPattern = FillPattern.Solid),Text(extent = {{-160,110},{40,50}}, lineColor = {0,0,255}, textString = "%name")}));
end PositivePin;

Electrical Port

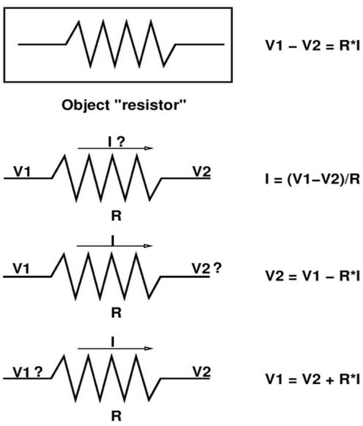
```
partial model OnePort
  "Component with two electrical pins p and n
   and current i from p to n"
  Voltage v "Voltage drop between the two pins (= p.v - n.v)";
  Current i "Current flowing from pin p to pin n";
  PositivePin p;
  NegativePin n;
equation
  v = p.v - n.v;
  0 = p.i + n.i;
  i = p.i;
end OnePort;
```


Libraries
CCC
OpAmp
OpAmpDetailed
VariableResistor
VariableConductor
VariableCapacitor
VariableInductor
Ideal
Interfaces
Pin
PositivePin
NegativePin
TwoPin
OnePort
TwoPort
ConditionalHeatPort
AbsoluteSensor
RelativeSensor
VoltageSource
CurrentSource
Lines
Semiconductors
Sensors
Sources
Digital
Machines
MultiPhase
Writeable
Model
Modelica Text View
C:/OpenModelica1.9.1Beta2/lib/omlibrary/Modelica 3.2.1/Electrical/Analog/Interfaces.mo
Line: 1, Col: 0
partial model OnePort "Component with two electrical pins p and n and current i from p to n"
SI.Voltage v "Voltage drop between the two pins (= p.v - n.v)";
SI.Current i "Current flowing from pin p to pin n";
PositivePin p "Positive pin (potential p.v > n.v for positive voltage drop v)"
annotation(Placement(transformation(extent = {{-110,-10},{-90,10}}, rotation = 0)));
NegativePin n "Negative pin" annotation(Placement(transformation(extent = {{110,-10},{90,10}},
rotation = 0)));
equation
v = p.v - n.v;
0 = p.i + n.i;
i = p.i;
annotation(Documentation(info = "<html>
<p>Superclass of elements which have two electrical pins: the positive pin connector
<i>p</i>, and the negative pin connector <i>n</i>. It is assumed that the current flowing into pin p
is identical to the current flowing out of pin n. This current is provided explicitly as current
i.</p>
</html>", revisions = "<html>

<i> 1998 </i>
by Christoph Clauss
 initially implemented

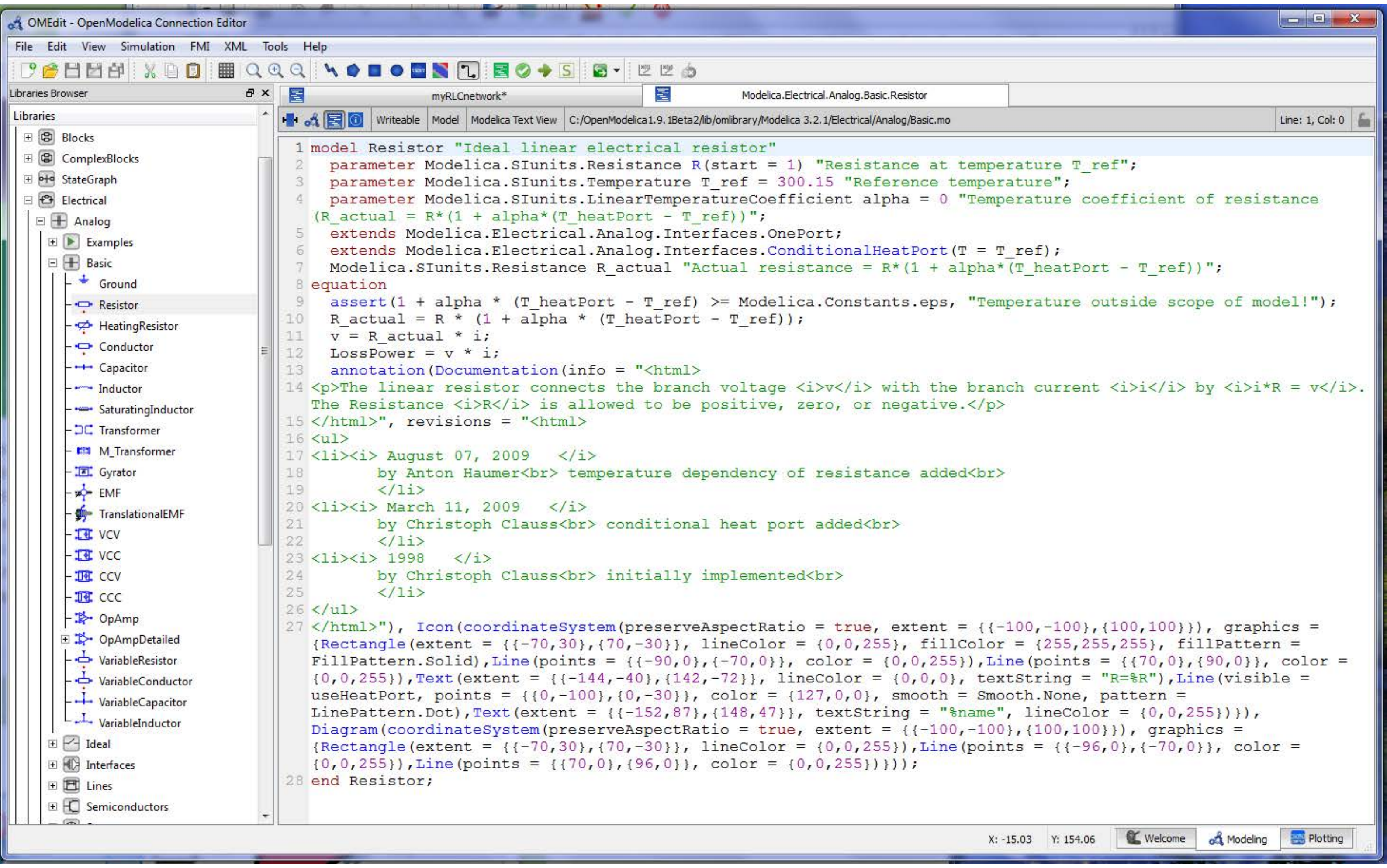
</html>"), Diagram(coordinateSystem(preserveAspectRatio = true, extent = {{-100,-100},{100,100}}),
graphics = {Line(points = {{-110,20},{-85,20}}, color = {160,160,164}),Polygon(points = {{-95,23},
{-85,20},{-95,17},{-95,23}}, lineColor = {160,160,164}, fillColor = {160,160,164}, fillPattern =
FillPattern.Solid),Line(points = {{90,20},{115,20}}, color = {160,160,164}),Line(points = {{-125,0},
{-115,0}}, color = {160,160,164}),Line(points = {{-120,-5},{-120,5}}, color =
{160,160,164}),Text(extent = {{-110,25},{-90,45}}, lineColor = {160,160,164}, textString =
"i"),Polygon(points = {{105,23},{115,20},{105,17},{105,23}}, lineColor = {160,160,164}, fillColor =
{160,160,164}, fillPattern = FillPattern.Solid),Line(points = {{115,0},{125,0}}, color =
{160,160,164}),Text(extent = {{90,45},{110,25}}, lineColor = {160,160,164}, textString = "i")}));
end OnePort;

Object-oriented re-use and causality



Electrical Resistor

```
model Resistor "Ideal linear electrical resistor"
  extends OnePort;
  parameter Resistance R=1 "Resistance";
  equation
    R*i = v;
end Resistor;
```

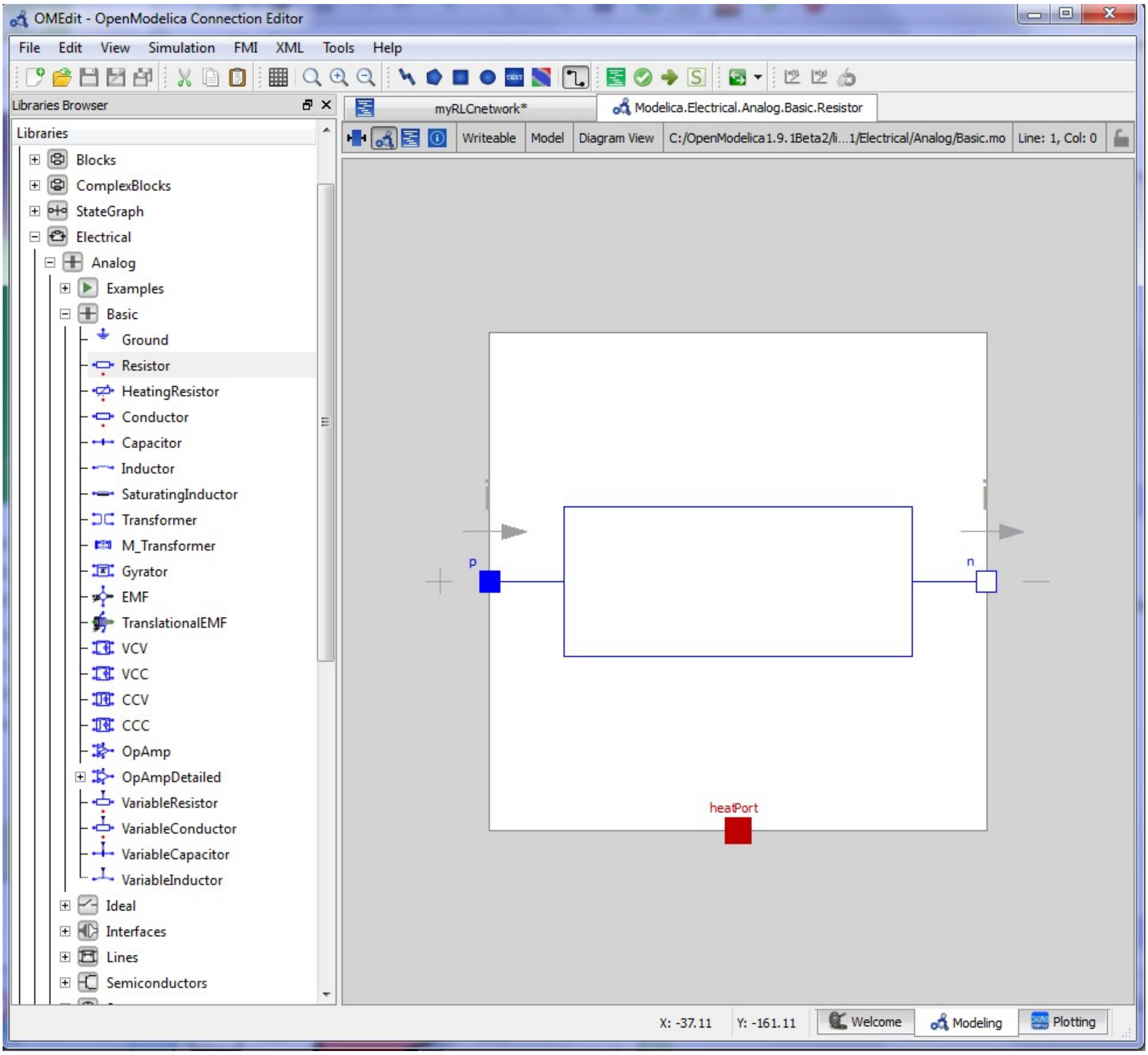

OMEdit - OpenModelica Connection Editor
File Edit View Simulation FMI XML Tools Help
Libraries Browser
Libraries
Blocks
ComplexBlocks
StateGraph
Electrical
Analog
Examples
Basic
Ground
Resistor
HeatingResistor
Conductor
Capacitor
Inductor
SaturatingInductor
Transformer
M_Transformer
Gyrator
EMF
TranslationalEMF
VCV
VCC
CCV
CCC
OpAmp
OpAmpDetailed
VariableResistor
VariableConductor
VariableCapacitor
VariableInductor
Ideal
Interfaces
Lines
Semiconductors
myRLCnetwork*
Modelica.Electrical.Analog.Basic.Resistor
Writeable
Model
Modelica Text View
C:/OpenModelica1.9.1Beta2/lib/omlibrary/Modelica 3.2.1/Electrical/Analog/Basic.mo
Line: 1, Col: 0
model Resistor "Ideal linear electrical resistor"
parameter Modelica.SIunits.Resistance R(start = 1) "Resistance at temperature T_ref";
parameter Modelica.SIunits.Temperature T_ref = 300.15 "Reference temperature";
parameter Modelica.SIunits.LinearTemperatureCoefficient alpha = 0 "Temperature coefficient of resistance (R_actual = R*(1 + alpha*(T_heatPort - T_ref))";
extends Modelica.Electrical.Analog.Interfaces.OnePort;
extends Modelica.Electrical.Analog.Interfaces.ConditionalHeatPort(T = T_ref);
Modelica.SIunits.Resistance R_actual "Actual resistance = R*(1 + alpha*(T_heatPort - T_ref))";
equation
assert(1 + alpha * (T_heatPort - T_ref) >= Modelica.Constants.eps, "Temperature outside scope of model!");
R_actual = R * (1 + alpha * (T_heatPort - T_ref));
v = R_actual * i;
LossPower = v * i;
annotation(Documentation(info = "<html>
<p>The linear resistor connects the branch voltage <i>v</i> with the branch current <i>i</i> by <i>i*R = v</i>. The Resistance <i>R</i> is allowed to be positive, zero, or negative.</p>
</html>", revisions = "<html>

<i> August 07, 2009 </i>
by Anton Haumer
 temperature dependency of resistance added

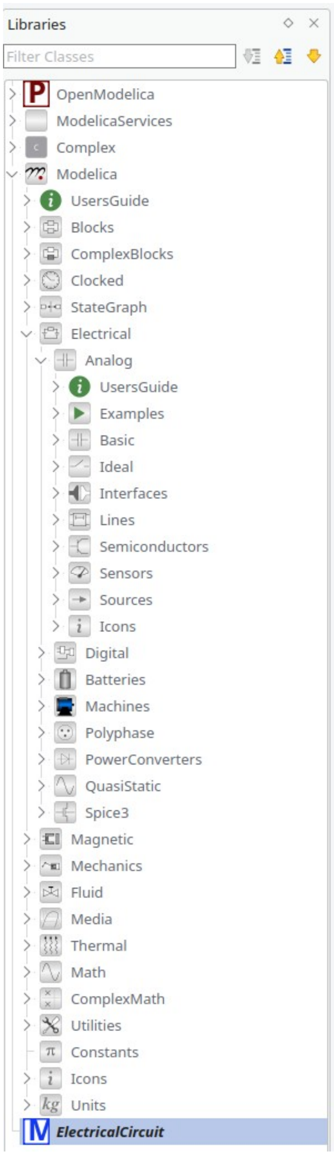
<i> March 11, 2009 </i>
by Christoph Clauss
 conditional heat port added

<i> 1998 </i>
by Christoph Clauss
 initially implemented

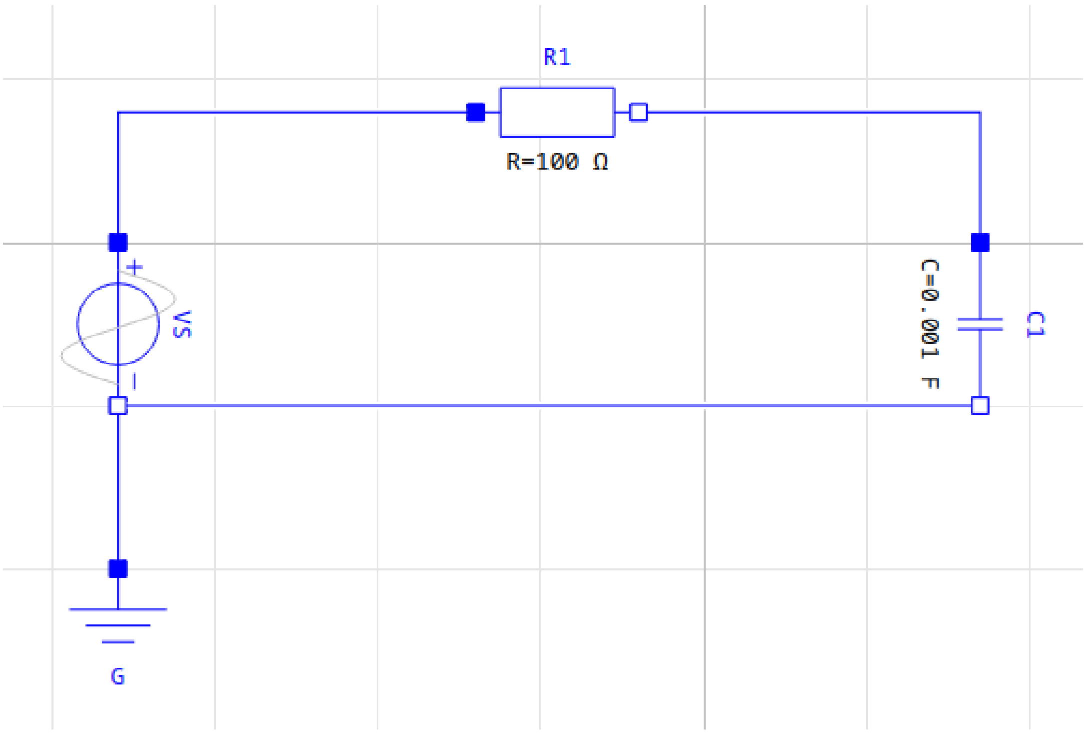
</html>"), Icon(coordinateSystem(preserveAspectRatio = true, extent = {{-100,-100},{100,100}}), graphics = {Rectangle(extent = {{-70,30},{70,-30}}, lineColor = {0,0,255}, fillColor = {255,255,255}, fillPattern = FillPattern.Solid),Line(points = {{-90,0},{-70,0}}, color = {0,0,255}),Line(points = {{70,0},{90,0}}, color = {0,0,255}),Text(extent = {{-144,-40},{142,-72}}, lineColor = {0,0,0}, textString = "R=%R"),Line(visible = useHeatPort, points = {{0,-100},{0,-30}}, color = {127,0,0}, smooth = Smooth.None, pattern = LinePattern.Dot),Text(extent = {{-152,87},{148,47}}, textString = "%name", lineColor = {0,0,255})}), Diagram(coordinateSystem(preserveAspectRatio = true, extent = {{-100,-100},{100,100}}), graphics = {Rectangle(extent = {{-70,30},{70,-30}}, lineColor = {0,0,255}),Line(points = {{-96,0},{-70,0}}, color = {0,0,255}),Line(points = {{70,0},{96,0}}, color = {0,0,255})}));
end Resistor;
X: -15.03
Y: 154.06
Welcome
Modeling
Plotting



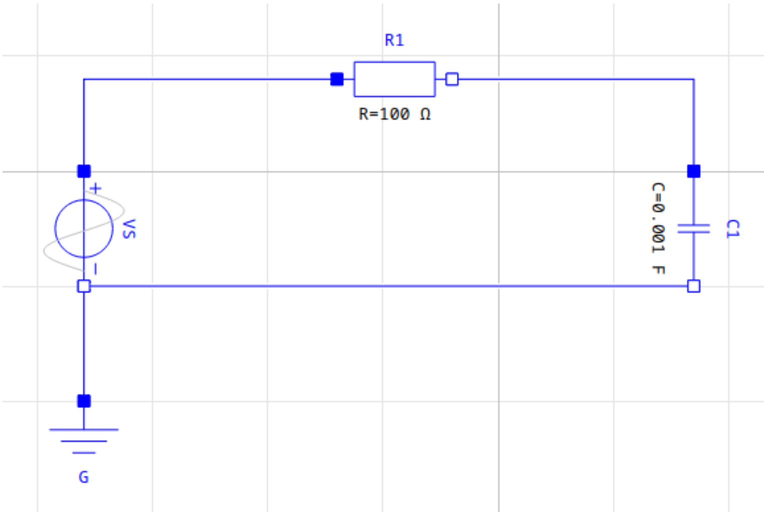
OMEdit - OpenModelica Connection Editor
File
Edit
View
Simulation
FMI
XML
Tools
Help
Libraries Browser
Libraries
Blocks
ComplexBlocks
StateGraph
Electrical
Analog
Examples
Basic
Ground
Resistor
HeatingResistor
Conductor
Capacitor
Inductor
SaturatingInductor
Transformer
M_Transformer
Gyrator
EMF
TranslationalEMF
VCV
VCC
CCV
CCC
OpAmp
OpAmpDetailed
VariableResistor
VariableConductor
VariableCapacitor
VariableInductor
Ideal
Interfaces
Lines
Semiconductors
myRLCnetwork*
Modelica.Electrical.Analog.Basic.Resistor
Writeable
Model
Diagram View
C:/OpenModelica1.9.1Beta2/li...1/Electrical/Analog/Basic.mo
Line: 1, Col: 0
p
n
heatPort
X: -37.11
Y: -161.11
Welcome
Modeling
Plotting



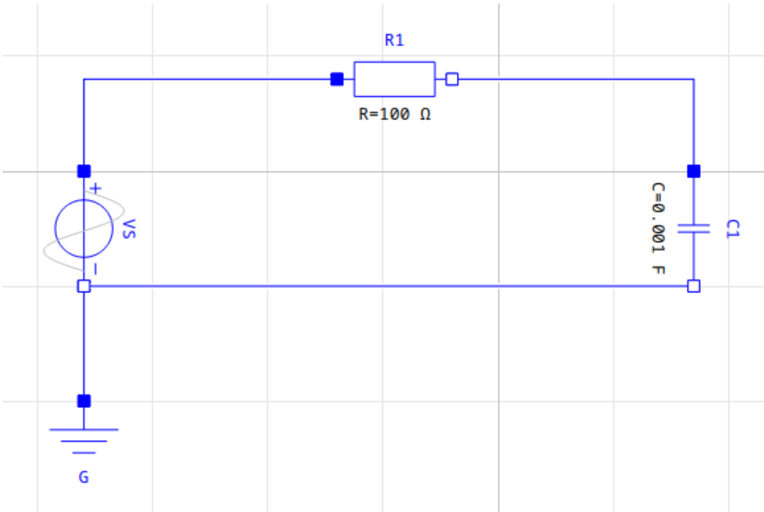
"low (frequency) pass filter"



```
model ElectricalCircuit
  Modelica.Electrical.Analog.Basic.Ground G annotation( ...);
  Modelica.Electrical.Analog.Basic.Resistor R1(R = 100)  annotation( ...);
  Modelica.Electrical.Analog.Basic.Capacitor C1(C = 0.001)  annotation( ...);
  Modelica.Electrical.Analog.Sources.SineVoltage VS(V = 220, f (displayUnit = "Hz")= 50)
annotation( ...);
equation
  connect(R1.p, VS.p) annotation( ...);
  connect(VS.n, G.p) annotation( ...);
  connect(VS.n, C1.n) annotation( ...);
  connect(R1.n, C1.p) annotation( ...);

annotation( ...);
end ElectricalCircuit;
```

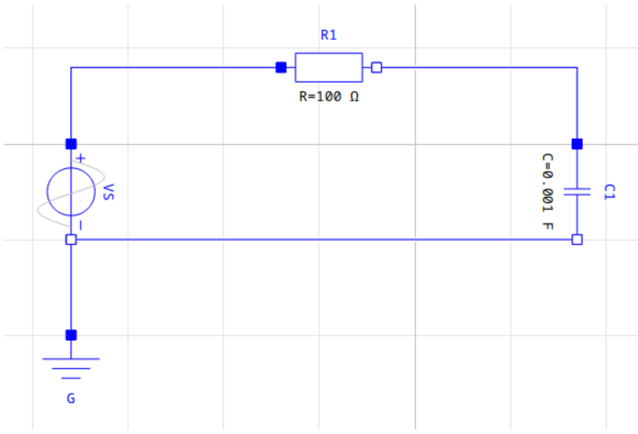


```
model ElectricalCircuit
  Modelica.Electrical.Analog.Basic.Ground G annotation(
    Placement(visible = true, transformation(origin = {-72, -50}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));
  Modelica.Electrical.Analog.Basic.Resistor R1(R = 100)  annotation(
    Placement(transformation(origin = {-18, 16}, extent = {{-10, -10}, {10, 10}})));
  Modelica.Electrical.Analog.Basic.Capacitor C1(C = 0.001)  annotation(
    Placement(visible = true, transformation(origin = {34, -10}, extent = {{-10, -10}, {10, 10}}, rotation = -90)));
  Modelica.Electrical.Analog.Sources.SineVoltage VS(V = 220, f (displayUnit = "Hz")= 50)  annotation(
    Placement(visible = true, transformation(origin = {-72, -10}, extent = {{-10, -10}, {10, 10}}, rotation = -90)));
equation
  connect(R1.p, VS.p) annotation(
    Line(points = {{-28, 16}, {-72, 16}, {-72, 0}}, color = {0, 0, 255}));
  connect(VS.n, G.p) annotation(
    Line(points = {{-72, -20}, {-72, -20}, {-72, -40}, {-72, -40}}, color = {0, 0, 255}));
  connect(VS.n, C1.n) annotation(
    Line(points = {{-72, -20}, {34, -20}}, color = {0, 0, 255}));
  connect(R1.n, C1.p) annotation(
    Line(points = {{-8, 16}, {34, 16}, {34, 0}}, color = {0, 0, 255}));

annotation(
    uses(Modelica(version = "4.0.0")));
end ElectricalCircuit;
```



MODELICA

```
model ElectricalCircuit
  Modelica.Electrical.Analog.Basic.Ground G annotation( ...);
  Modelica.Electrical.Analog.Basic.Resistor R1(R = 100)  annotation( ...);
  Modelica.Electrical.Analog.Basic.Capacitor C1(C = 0.001)  annotation( ...);
  Modelica.Electrical.Analog.Sources.SineVoltage VS(V = 220, f (displayUnit = "Hz")= 50)
annotation( ...);
equation
  connect(R1.p, VS.p) annotation( ...);
  connect(VS.n, G.p) annotation( ...);
  connect(VS.n, C1.n) annotation( ...);
  connect(R1.n, C1.p) annotation( ...);

annotation( ...);
end ElectricalCircuit;
```



Meaning: set of Differential Algebraic Equations (DAEs) obtained by

1.a. expanding inheritance
1.b. instantiation of classes
2. flattening hierarchy, constructing unique names
3. expanding connect() into equations (across vs. flow)

Meaning: set of Differential Algebraic Equations (DAEs) obtained by

1.a. expanding inheritance
1.b. instantiation of classes
2. flattening hierarchy, constructing unique names
3. expanding connect() into equations (across vs. flow)

[1] 00:29:59 Scripting Notification
Check of ElectricalCircuit completed successfully.
Class ElectricalCircuit has 24 equation(s) and 24 variable(s).
16 of these are trivial equation(s).

```
class ElectricalCircuit
  Real G.p.v(quantity = "ElectricPotential", unit = "V") "Potential at the pin";
  Real G.p.i(quantity = "ElectricCurrent", unit = "A") "Current flowing into the pin";
  parameter Real R1.R(quantity = "Resistance", unit = "Ohm", start = 1.0) = 100.0 "Resistance at temperature T_ref";
  parameter Real R1.T_ref(quantity = "ThermodynamicTemperature", unit = "K", displayUnit = "degC", min = 0.0, start = 288.15, nominal = 300.0) = 300.15 "Reference temperature";
  parameter Real R1.alpha(quantity = "LinearTemperatureCoefficient", unit = "1/K") = 0.0 "Temperature coefficient of resistance (R_actual = R*(1 + alpha*(T_heatPort - T_ref))";
  Real R1.v(quantity = "ElectricPotential", unit = "V") "Voltage drop of the two pins (= p.v - n.v)";
  Real R1.p.v(quantity = "ElectricPotential", unit = "V") "Potential at the pin";
  Real R1.p.i(quantity = "ElectricCurrent", unit = "A") "Current flowing into the pin";
  Real R1.n.v(quantity = "ElectricPotential", unit = "V") "Potential at the pin";
  Real R1.n.i(quantity = "ElectricCurrent", unit = "A") "Current flowing into the pin";
  Real R1.i(quantity = "ElectricCurrent", unit = "A") "Current flowing from pin p to pin n";
  final parameter Boolean R1.useHeatPort = false "= true, if heatPort is enabled";
  parameter Real R1.T(quantity = "ThermodynamicTemperature", unit = "K", displayUnit = "degC", min = 0.0, start = 288.15, nominal = 300.0) = R1.T_ref "Fixed device temperature if useHeatPort = false";
  Real R1.LossPower(quantity = "Power", unit = "W") "Loss power leaving component via heatPort";
  Real R1.T_heatPort(quantity = "ThermodynamicTemperature", unit = "K", displayUnit = "degC", min = 0.0, start = 288.15, nominal = 300.0) "Temperature of heatPort";
  Real R1.R_actual(quantity = "Resistance", unit = "Ohm") "Actual resistance = R*(1 + alpha*(T_heatPort - T_ref))";
  Real C1.v(quantity = "ElectricPotential", unit = "V", start = 0.0) "Voltage drop of the two pins (= p.v - n.v)";
  Real C1.p.v(quantity = "ElectricPotential", unit = "V") "Potential at the pin";
  Real C1.p.i(quantity = "ElectricCurrent", unit = "A") "Current flowing into the pin";
  Real C1.n.v(quantity = "ElectricPotential", unit = "V") "Potential at the pin";
  Real C1.n.i(quantity = "ElectricCurrent", unit = "A") "Current flowing into the pin";
  Real C1.i(quantity = "ElectricCurrent", unit = "A") "Current flowing from pin p to pin n";
  parameter Real C1.C(quantity = "Capacitance", unit = "F", min = 0.0, start = 1.0) = 0.001 "Capacitance";
  parameter Real VS.V(quantity = "ElectricPotential", unit = "V", start = 1.0) = 220.0 "Amplitude of sine wave";
  parameter Real VS.phase(quantity = "Angle", unit = "rad", displayUnit = "deg") = 0.0 "Phase of sine wave";
  parameter Real VS.f(quantity = "Frequency", unit = "Hz", displayUnit = "Hz", start = 1.0) = 50.0 "Frequency of sine wave";
  Real VS.v(quantity = "ElectricPotential", unit = "V") "Voltage drop of the two pins (= p.v - n.v)";
  Real VS.p.v(quantity = "ElectricPotential", unit = "V") "Potential at the pin";
  Real VS.p.i(quantity = "ElectricCurrent", unit = "A") "Current flowing into the pin";
  Real VS.n.v(quantity = "ElectricPotential", unit = "V") "Potential at the pin";
  Real VS.n.i(quantity = "ElectricCurrent", unit = "A") "Current flowing into the pin";
  Real VS.i(quantity = "ElectricCurrent", unit = "A") "Current flowing from pin p to pin n";
  final parameter Real VS.signalSource.amplitude = VS.V "Amplitude of sine wave";
  final parameter Real VS.signalSource.f(quantity = "Frequency", unit = "Hz", start = 1.0) = VS.f "Frequency of sine wave";
  final parameter Real VS.signalSource.phase(quantity = "Angle", unit = "rad", displayUnit = "deg") = VS.phase "Phase of sine wave";
  Real VS.signalSource.y "Connector of Real output signal";
  final parameter Real VS.signalSource.offset = VS.offset "Offset of output signal y";
  final parameter Real VS.signalSource.startTime(quantity = "Time", unit = "s") = VS.startTime "Output y = offset for time < startTime";
  parameter Real VS.offset(quantity = "ElectricPotential", unit = "V") = 0.0 "Voltage offset";
  parameter Real VS.startTime(quantity = "Time", unit = "s") = 0.0 "Time offset";
```

Meaning: set of Differential Algebraic Equations (DAEs) obtained by

1.a. expanding inheritance
1.b. instantiation of classes
2. flattening hierarchy, constructing unique names
3. expanding connect() into equations (across vs. flow)

[1] 00:29:59 Scripting Notification
Check of ElectricalCircuit completed successfully.
Class ElectricalCircuit has 24 equation(s) and 24 variable(s).
16 of these are trivial equation(s).

```
equation
  R1.p.v = VS.p.v;
  VS.n.v = C1.n.v;
  VS.n.v = G.p.v;
  R1.n.v = C1.p.v;
  VS.n.i + C1.n.i + G.p.i = 0.0;
  VS.p.i + R1.p.i = 0.0;
  C1.p.i + R1.n.i = 0.0;
  G.p.v = 0.0;
  assert(1.0 + R1.alpha * (R1.T_heatPort - R1.T_ref) >= 1e-15, "Temperature outside scope of model!");
  R1.R_actual = R1.R * (1.0 + R1.alpha * (R1.T_heatPort - R1.T_ref));
  R1.v = R1.R_actual * R1.i;
  R1.LossPower = R1.v * R1.i;
  R1.T_heatPort = R1.T;
  0.0 = R1.p.i + R1.n.i;
  R1.i = R1.p.i;
  R1.v = R1.p.v - R1.n.v;
  C1.i = C1.C * der(C1.v);
  0.0 = C1.p.i + C1.n.i;
  C1.i = C1.p.i;
  C1.v = C1.p.v - C1.n.v;
  VS.signalSource.y = VS.signalSource.offset + (if time < VS.signalSource.startTime then 0.0 else VS.signalSource.amplitude *
sin(6.283185307179586 * VS.signalSource.f * (time - VS.signalSource.startTime) + VS.signalSource.phase));
  VS.v = VS.signalSource.y;
  0.0 = VS.p.i + VS.n.i;
  VS.i = VS.p.i;
  VS.v = VS.p.v - VS.n.v;
end ElectricalCircuit;
```

"across" variables are equal in a connection node

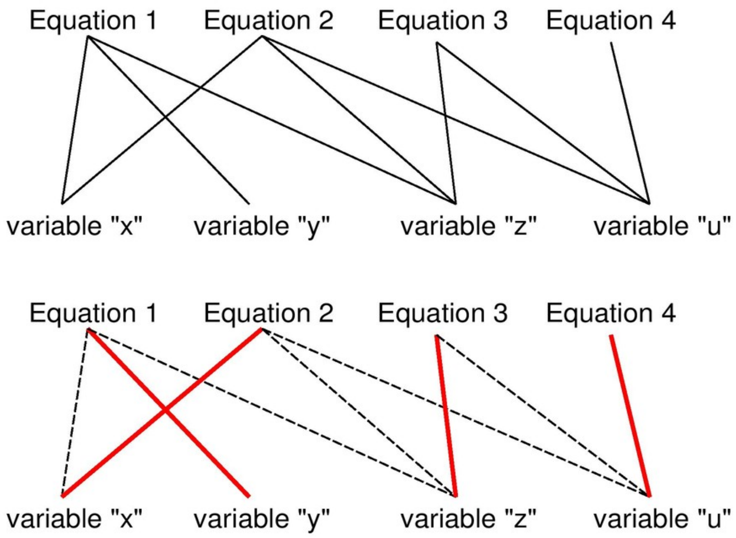
"through/flow" variables sum to 0 in a connection node

equations with unique variable names, after flattening

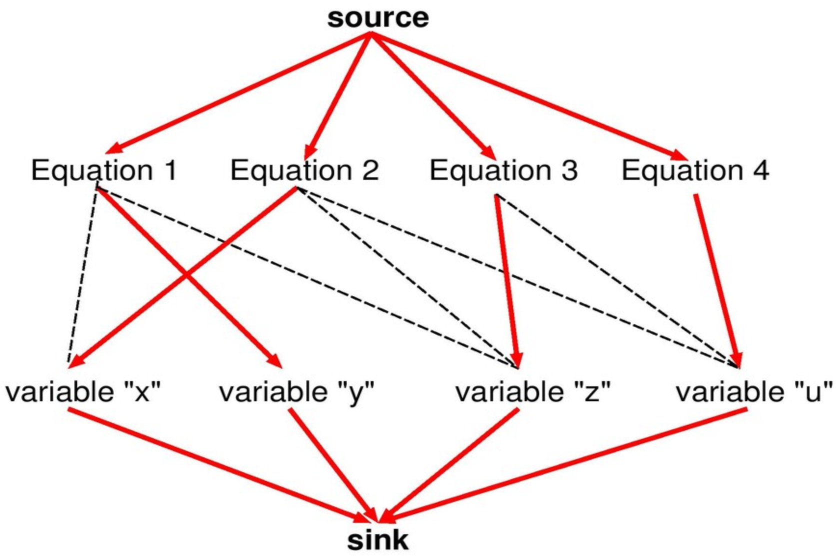
Non-causal model
(*e.g.*, from physical conservation laws)

$$\begin{cases} x + y + z &= 0 \quad \text{Equation 1} \\ x + 3z + u^2 &= 0 \quad \text{Equation 2} \\ z - u - 16 &= 0 \quad \text{Equation 3} \\ u - 5 &= 0 \quad \text{Equation 4} \end{cases}$$

Causality assignment: bipartite graph, maximum cardinality matching



Causality assignment: network flow



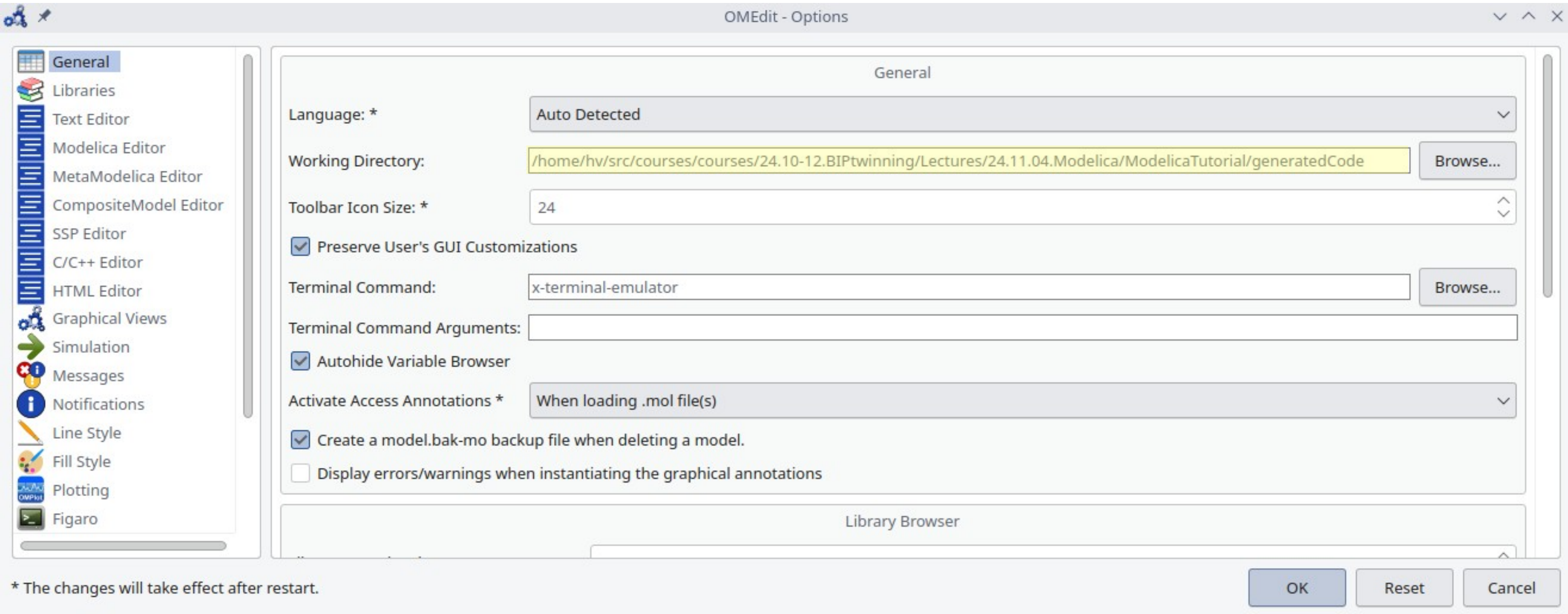
+ weights for "bad inverses"

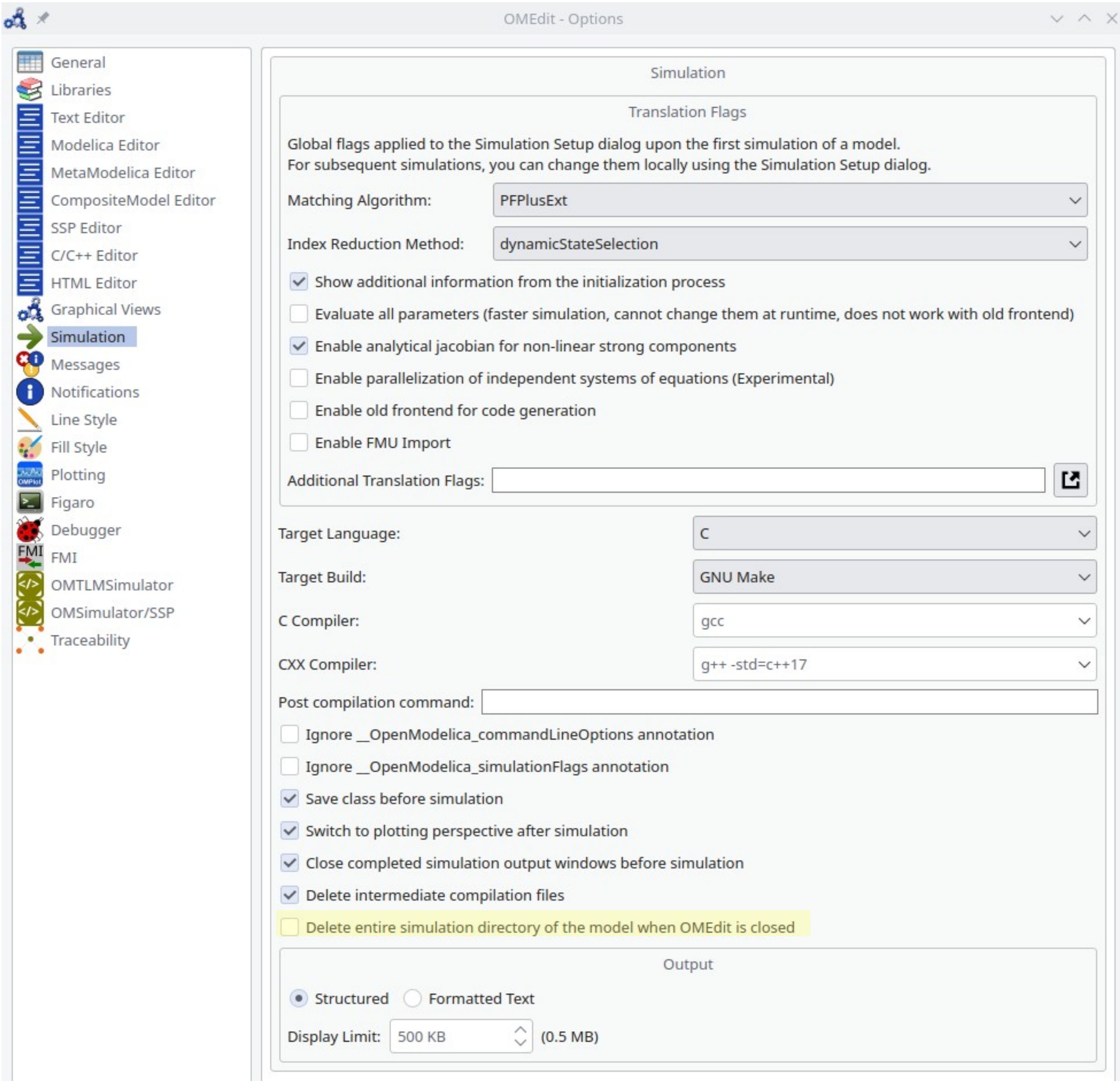
Causality assigned

$$\begin{cases} x + \underline{y} + z &= 0 \quad \text{Equation 1} \\ \underline{x} + 3z + u^2 &= 0 \quad \text{Equation 2} \\ \underline{z} - u - 16 &= 0 \quad \text{Equation 3} \\ \underline{u} - 5 &= 0 \quad \text{Equation 4} \end{cases}$$

re-write in causal form (symbolically, using Computer Algebra)

$$\begin{cases} \underline{y} &= -x - z \\ \underline{x} &= -3z - u^2 \\ \underline{z} &= u + 16 \\ \underline{u} &= 5 \end{cases}$$


OMEdit - Options
General
Libraries
Text Editor
Modelica Editor
MetaModelica Editor
CompositeModel Editor
SSP Editor
C/C++ Editor
HTML Editor
Graphical Views
Simulation
Messages
Notifications
Line Style
Fill Style
Plotting
Figaro
General
Language: *
Auto Detected
Working Directory:
/home/hv/src/courses/courses/24.10-12.BIPtwinning/Lectures/24.11.04.Modelica/ModelicaTutorial/generatedCode
Browse...
Toolbar Icon Size: *
24
Preserve User's GUI Customizations
Terminal Command:
x-terminal-emulator
Browse...
Terminal Command Arguments:
Autohide Variable Browser
Activate Access Annotations *
When loading .mol file(s)
Create a model.bak-mo backup file when deleting a model.
Display errors/warnings when instantiating the graphical annotations
Library Browser
* The changes will take effect after restart.
OK
Reset
Cancel


OMEdit - Options
General
Libraries
Text Editor
Modelica Editor
MetaModelica Editor
CompositeModel Editor
SSP Editor
C/C++ Editor
HTML Editor
Graphical Views
Simulation
Messages
Notifications
Line Style
Fill Style
Plotting
Figaro
Debugger
FMI
OMTLMSimulator
OMSimulator/SSP
Traceability
Simulation
Translation Flags
Global flags applied to the Simulation Setup dialog upon the first simulation of a model.
For subsequent simulations, you can change them locally using the Simulation Setup dialog.
Matching Algorithm:
PFPlusExt
Index Reduction Method:
dynamicStateSelection
Show additional information from the initialization process
Evaluate all parameters (faster simulation, cannot change them at runtime, does not work with old frontend)
Enable analytical jacobian for non-linear strong components
Enable parallelization of independent systems of equations (Experimental)
Enable old frontend for code generation
Enable FMU Import
Additional Translation Flags:
Target Language:
C
Target Build:
GNU Make
C Compiler:
gcc
CXX Compiler:
g++ -std=c++17
Post compilation command:
Ignore __OpenModelica_commandLineOptions annotation
Ignore __OpenModelica_simulationFlags annotation
Save class before simulation
Switch to plotting perspective after simulation
Close completed simulation output windows before simulation
Delete intermediate compilation files
Delete entire simulation directory of the model when OMEdit is closed
Output
Structured
Formatted Text
Display Limit:
500 KB
(0.5 MB)

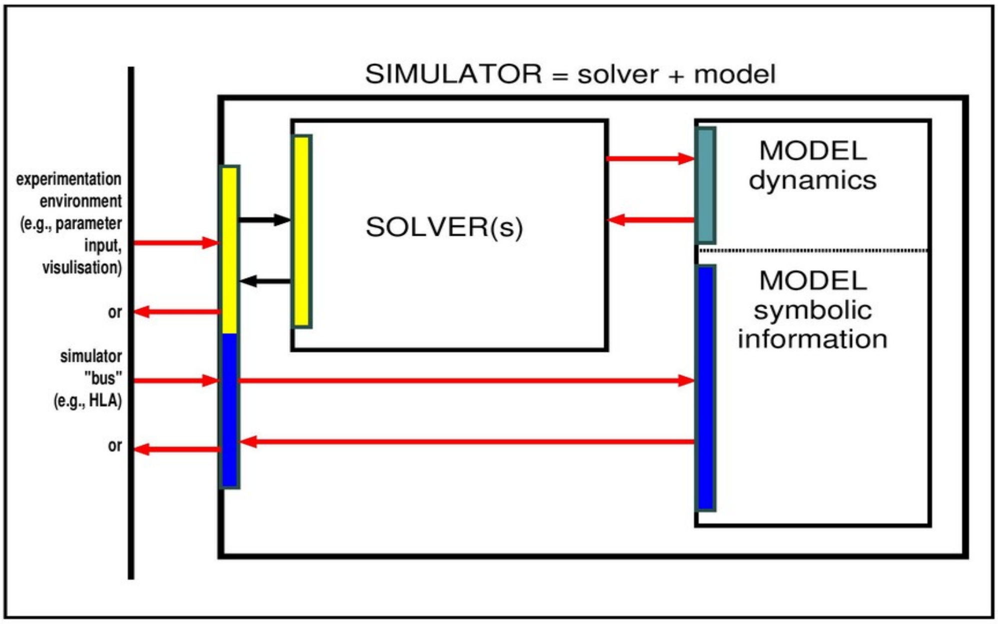
Compilation | Output

```
make -j4 -f ElectricalCircuit.makefile
gcc  -Os -DOM_HAVE_PTHREADS -fPIC -falign-functions -mfpmath=sse -fno-dollars-in-identifiers -Wno-parentheses-equality     -I"/opt/openmodelica-nightly/bin/../include/omc/c" -I"/opt/
openmodelica-nightly/bin/../include/omc" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -DOMC_MODEL_PREFIX=ElectricalCircuit -DOMC_NUM_MIXED_SYSTEMS=0 -DOMC_NUM_LINEAR_SYSTEMS=0 -
DOMC_NUM_NONLINEAR_SYSTEMS=0 -DOMC_NDELAY_EXPRESSIONS=0 -DOMC_NVAR_STRING=0  -c -o ElectricalCircuit.o ElectricalCircuit.c
gcc  -Os -DOM_HAVE_PTHREADS -fPIC -falign-functions -mfpmath=sse -fno-dollars-in-identifiers -Wno-parentheses-equality     -I"/opt/openmodelica-nightly/bin/../include/omc/c" -I"/opt/
openmodelica-nightly/bin/../include/omc" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -DOMC_MODEL_PREFIX=ElectricalCircuit -DOMC_NUM_MIXED_SYSTEMS=0 -DOMC_NUM_LINEAR_SYSTEMS=0 -
DOMC_NUM_NONLINEAR_SYSTEMS=0 -DOMC_NDELAY_EXPRESSIONS=0 -DOMC_NVAR_STRING=0  -c -o ElectricalCircuit_functions.o ElectricalCircuit_functions.c
gcc  -Os -DOM_HAVE_PTHREADS -fPIC -falign-functions -mfpmath=sse -fno-dollars-in-identifiers -Wno-parentheses-equality     -I"/opt/openmodelica-nightly/bin/../include/omc/c" -I"/opt/
openmodelica-nightly/bin/../include/omc" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -DOMC_MODEL_PREFIX=ElectricalCircuit -DOMC_NUM_MIXED_SYSTEMS=0 -DOMC_NUM_LINEAR_SYSTEMS=0 -
DOMC_NUM_NONLINEAR_SYSTEMS=0 -DOMC_NDELAY_EXPRESSIONS=0 -DOMC_NVAR_STRING=0  -c -o ElectricalCircuit_records.o ElectricalCircuit_records.c
gcc  -Os -DOM_HAVE_PTHREADS -fPIC -falign-functions -mfpmath=sse -fno-dollars-in-identifiers -Wno-parentheses-equality     -I"/opt/openmodelica-nightly/bin/../include/omc/c" -I"/opt/
openmodelica-nightly/bin/../include/omc" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -DOMC_MODEL_PREFIX=ElectricalCircuit -DOMC_NUM_MIXED_SYSTEMS=0 -DOMC_NUM_LINEAR_SYSTEMS=0 -
DOMC_NUM_NONLINEAR_SYSTEMS=0 -DOMC_NDELAY_EXPRESSIONS=0 -DOMC_NVAR_STRING=0  -c -o ElectricalCircuit_01exo.o ElectricalCircuit_01exo.c
gcc  -Os -DOM_HAVE_PTHREADS -fPIC -falign-functions -mfpmath=sse -fno-dollars-in-identifiers -Wno-parentheses-equality     -I"/opt/openmodelica-nightly/bin/../include/omc/c" -I"/opt/
openmodelica-nightly/bin/../include/omc" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -DOMC_MODEL_PREFIX=ElectricalCircuit -DOMC_NUM_MIXED_SYSTEMS=0 -DOMC_NUM_LINEAR_SYSTEMS=0 -
DOMC_NUM_NONLINEAR_SYSTEMS=0 -DOMC_NDELAY_EXPRESSIONS=0 -DOMC_NVAR_STRING=0  -c -o ElectricalCircuit_02nls.o ElectricalCircuit_02nls.c
gcc  -Os -DOM_HAVE_PTHREADS -fPIC -falign-functions -mfpmath=sse -fno-dollars-in-identifiers -Wno-parentheses-equality     -I"/opt/openmodelica-nightly/bin/../include/omc/c" -I"/opt/
openmodelica-nightly/bin/../include/omc" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -DOMC_MODEL_PREFIX=ElectricalCircuit -DOMC_NUM_MIXED_SYSTEMS=0 -DOMC_NUM_LINEAR_SYSTEMS=0 -
DOMC_NUM_NONLINEAR_SYSTEMS=0 -DOMC_NDELAY_EXPRESSIONS=0 -DOMC_NVAR_STRING=0  -c -o ElectricalCircuit_03lsy.o ElectricalCircuit_03lsy.c
gcc  -Os -DOM_HAVE_PTHREADS -fPIC -falign-functions -mfpmath=sse -fno-dollars-in-identifiers -Wno-parentheses-equality     -I"/opt/openmodelica-nightly/bin/../include/omc/c" -I"/opt/
openmodelica-nightly/bin/../include/omc" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -DOMC_MODEL_PREFIX=ElectricalCircuit -DOMC_NUM_MIXED_SYSTEMS=0 -DOMC_NUM_LINEAR_SYSTEMS=0 -
DOMC_NUM_NONLINEAR_SYSTEMS=0 -DOMC_NDELAY_EXPRESSIONS=0 -DOMC_NVAR_STRING=0  -c -o ElectricalCircuit_04set.o ElectricalCircuit_04set.c
```

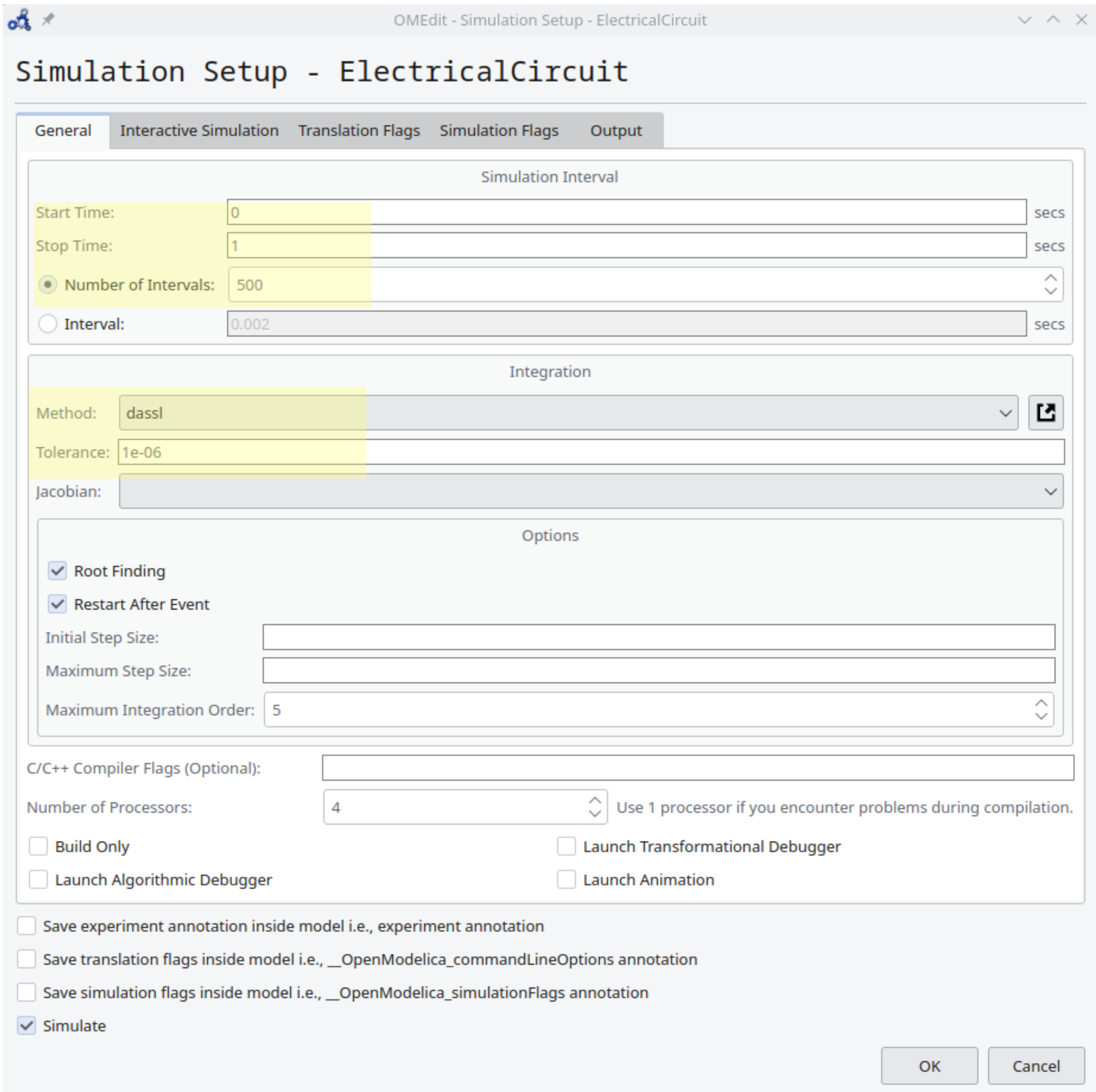
```
DOMC_NUM_NONLINEAR_SYSTEMS=0 -DOMC_NDELAY_EXPRESSIONS=0 -DOMC_NVAR_STRING=0  -c -o ElectricalCircuit_13opt.o ElectricalCircuit_13opt.c
gcc  -Os -DOM_HAVE_PTHREADS -fPIC -falign-functions -mfpmath=sse -fno-dollars-in-identifiers -Wno-parentheses-equality     -I"/opt/openmodelica-nightly/bin/../include/omc/c" -I"/opt/
openmodelica-nightly/bin/../include/omc" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -DOMC_MODEL_PREFIX=ElectricalCircuit -DOMC_NUM_MIXED_SYSTEMS=0 -DOMC_NUM_LINEAR_SYSTEMS=0 -
DOMC_NUM_NONLINEAR_SYSTEMS=0 -DOMC_NDELAY_EXPRESSIONS=0 -DOMC_NVAR_STRING=0  -c -o ElectricalCircuit_14lnz.o ElectricalCircuit_14lnz.c
gcc  -Os -DOM_HAVE_PTHREADS -fPIC -falign-functions -mfpmath=sse -fno-dollars-in-identifiers -Wno-parentheses-equality     -I"/opt/openmodelica-nightly/bin/../include/omc/c" -I"/opt/
openmodelica-nightly/bin/../include/omc" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -DOMC_MODEL_PREFIX=ElectricalCircuit -DOMC_NUM_MIXED_SYSTEMS=0 -DOMC_NUM_LINEAR_SYSTEMS=0 -
DOMC_NUM_NONLINEAR_SYSTEMS=0 -DOMC_NDELAY_EXPRESSIONS=0 -DOMC_NVAR_STRING=0  -c -o ElectricalCircuit_15syn.o ElectricalCircuit_15syn.c
gcc  -Os -DOM_HAVE_PTHREADS -fPIC -falign-functions -mfpmath=sse -fno-dollars-in-identifiers -Wno-parentheses-equality     -I"/opt/openmodelica-nightly/bin/../include/omc/c" -I"/opt/
openmodelica-nightly/bin/../include/omc" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -DOMC_MODEL_PREFIX=ElectricalCircuit -DOMC_NUM_MIXED_SYSTEMS=0 -DOMC_NUM_LINEAR_SYSTEMS=0 -
DOMC_NUM_NONLINEAR_SYSTEMS=0 -DOMC_NDELAY_EXPRESSIONS=0 -DOMC_NVAR_STRING=0  -c -o ElectricalCircuit_16dae.o ElectricalCircuit_16dae.c
gcc  -Os -DOM_HAVE_PTHREADS -fPIC -falign-functions -mfpmath=sse -fno-dollars-in-identifiers -Wno-parentheses-equality     -I"/opt/openmodelica-nightly/bin/../include/omc/c" -I"/opt/
openmodelica-nightly/bin/../include/omc" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -DOMC_MODEL_PREFIX=ElectricalCircuit -DOMC_NUM_MIXED_SYSTEMS=0 -DOMC_NUM_LINEAR_SYSTEMS=0 -
DOMC_NUM_NONLINEAR_SYSTEMS=0 -DOMC_NDELAY_EXPRESSIONS=0 -DOMC_NVAR_STRING=0  -c -o ElectricalCircuit_17inl.o ElectricalCircuit_17inl.c
gcc  -Os -DOM_HAVE_PTHREADS -fPIC -falign-functions -mfpmath=sse -fno-dollars-in-identifiers -Wno-parentheses-equality     -I"/opt/openmodelica-nightly/bin/../include/omc/c" -I"/opt/
openmodelica-nightly/bin/../include/omc" -I. -DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -DOMC_MODEL_PREFIX=ElectricalCircuit -DOMC_NUM_MIXED_SYSTEMS=0 -DOMC_NUM_LINEAR_SYSTEMS=0 -
DOMC_NUM_NONLINEAR_SYSTEMS=0 -DOMC_NDELAY_EXPRESSIONS=0 -DOMC_NVAR_STRING=0  -c -o ElectricalCircuit_18spd.o ElectricalCircuit_18spd.c
gcc -I. -o ElectricalCircuit ElectricalCircuit.o ElectricalCircuit_functions.o ElectricalCircuit_records.o ElectricalCircuit_01exo.o ElectricalCircuit_02nls.o ElectricalCircuit_03lsy.o
ElectricalCircuit_04set.o ElectricalCircuit_05evt.o ElectricalCircuit_06inz.o ElectricalCircuit_07dly.o ElectricalCircuit_08bnd.o ElectricalCircuit_09alg.o ElectricalCircuit_10asr.o
ElectricalCircuit_11mix.o ElectricalCircuit_12jac.o ElectricalCircuit_13opt.o ElectricalCircuit_14lnz.o ElectricalCircuit_15syn.o ElectricalCircuit_16dae.o ElectricalCircuit_17inl.o
ElectricalCircuit_18spd.o -L"/home/hv/src/courses/24.10-12.BIPtwinning/Lectures/24.11.04.Modelica/ModelicaTutorial/OOEquationBased"    -Os -DOM_HAVE_PTHREADS -fPIC -falign-functions -
mfpmath=sse -fno-dollars-in-identifiers -Wno-parentheses-equality     -I"/opt/openmodelica-nightly/bin/../include/omc/c" -I"/opt/openmodelica-nightly/bin/../include/omc" -I. -
DOPENMODELICA_XML_FROM_FILE_AT_RUNTIME -DOMC_MODEL_PREFIX=ElectricalCircuit -DOMC_NUM_MIXED_SYSTEMS=0 -DOMC_NUM_LINEAR_SYSTEMS=0 -DOMC_NUM_NONLINEAR_SYSTEMS=0 -DOMC_NDELAY_EXPRESSIONS=0 -
DOMC_NVAR_STRING=0 -L"/opt/openmodelica-nightly/bin/../lib/x86_64-linux-gnu/omc" -L"/opt/openmodelica-nightly/bin/../lib" -Wl,-rpath,"/opt/openmodelica-nightly/bin/../lib/x86_64-linux-gnu/omc"
-Wl,-rpath,"/opt/openmodelica-nightly/bin/../lib"   -Wl,--no-as-needed -Wl,--disable-new-dtags -lSimulationRuntimeC -llapack -lblas -lm -lomcgc -lryu -lpthread -rdynamic -Wl,--no-undefined
Compilation process finished successfully.
```

```
hv@sanderling 59% pwd
/home/hv/src/courses/24.10-12.BIPtwinning/Lectures/24.11.04.Modelica/ModelicaTutorial/generatedCode/ElectricalCircuit
hv@sanderling 60% ls
ElectricalCircuit           ElectricalCircuit_08bnd.c  ElectricalCircuit_14lnz.c      ElectricalCircuit_includes.h
ElectricalCircuit_01exo.c   ElectricalCircuit_08bnd.o  ElectricalCircuit_14lnz.o      ElectricalCircuit_info.json
ElectricalCircuit_01exo.o   ElectricalCircuit_09alg.c  ElectricalCircuit_15syn.c      ElectricalCircuit_init.xml
ElectricalCircuit_02nls.c   ElectricalCircuit_09alg.o  ElectricalCircuit_15syn.o      ElectricalCircuit_JacA.bin
ElectricalCircuit_02nls.o   ElectricalCircuit_10asr.c  ElectricalCircuit_16dae.c      ElectricalCircuit_literals.h
ElectricalCircuit_03lsy.c   ElectricalCircuit_10asr.o  ElectricalCircuit_16dae.h      ElectricalCircuit.log
ElectricalCircuit_03lsy.o   ElectricalCircuit_11mix.c  ElectricalCircuit_16dae.o      ElectricalCircuit.makefile
ElectricalCircuit_04set.c   ElectricalCircuit_11mix.h  ElectricalCircuit_17inl.c      ElectricalCircuit_model.h
ElectricalCircuit_04set.o   ElectricalCircuit_11mix.o  ElectricalCircuit_17inl.o      ElectricalCircuit.o
ElectricalCircuit_05evt.c   ElectricalCircuit_12jac.c  ElectricalCircuit_18spd.c      ElectricalCircuit_prof.intdata
ElectricalCircuit_05evt.o   ElectricalCircuit_12jac.h  ElectricalCircuit_18spd.o      ElectricalCircuit_prof.realdata
ElectricalCircuit_06inz.c   ElectricalCircuit_12jac.o  ElectricalCircuit.c            ElectricalCircuit_records.c
ElectricalCircuit_06inz.o   ElectricalCircuit_13opt.c  ElectricalCircuit_functions.c  ElectricalCircuit_records.o
ElectricalCircuit_07dly.c   ElectricalCircuit_13opt.h  ElectricalCircuit_functions.h  ElectricalCircuit_res.mat
ElectricalCircuit_07dly.o   ElectricalCircuit_13opt.o  ElectricalCircuit_functions.o
```

Model-Solver Interface
Simulator-Environment Interface



see Claudio's lecture on co-simulation
(and the FMI standard)

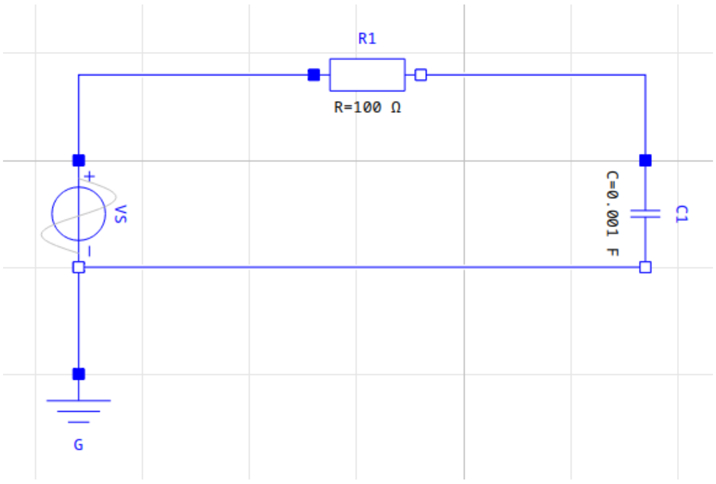


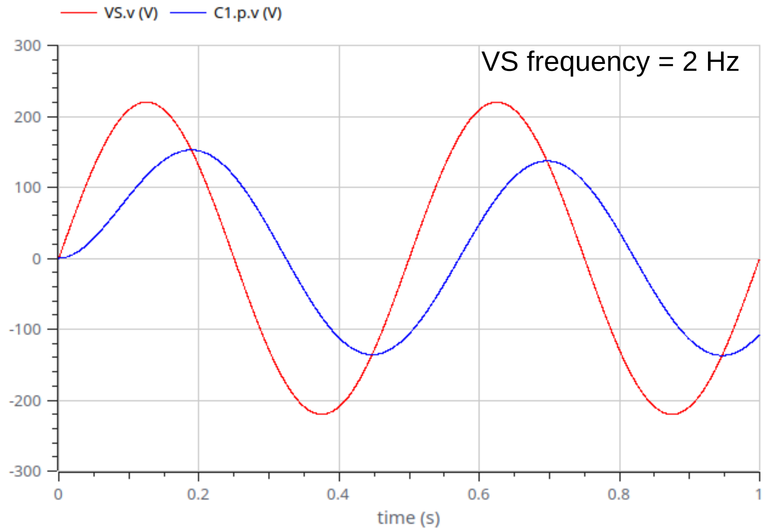
OMEdit - Simulation Setup - ElectricalCircuit
Simulation Setup - ElectricalCircuit
General
Interactive Simulation
Translation Flags
Simulation Flags
Output
Simulation Interval
Start Time: 0 secs
Stop Time: 1 secs
Number of Intervals: 500
Interval: 0.002 secs
Integration
Method: dassl
Tolerance: 1e-06
Jacobian:
Options
Root Finding
Restart After Event
Initial Step Size:
Maximum Step Size:
Maximum Integration Order: 5
C/C++ Compiler Flags (Optional):
Number of Processors: 4
Use 1 processor if you encounter problems during compilation.
Build Only
Launch Transformational Debugger
Launch Algorithmic Debugger
Launch Animation
Save experiment annotation inside model i.e., experiment annotation
Save translation flags inside model i.e., __OpenModelica_commandLineOptions annotation
Save simulation flags inside model i.e., __OpenModelica_simulationFlags annotation
Simulate
OK
Cancel

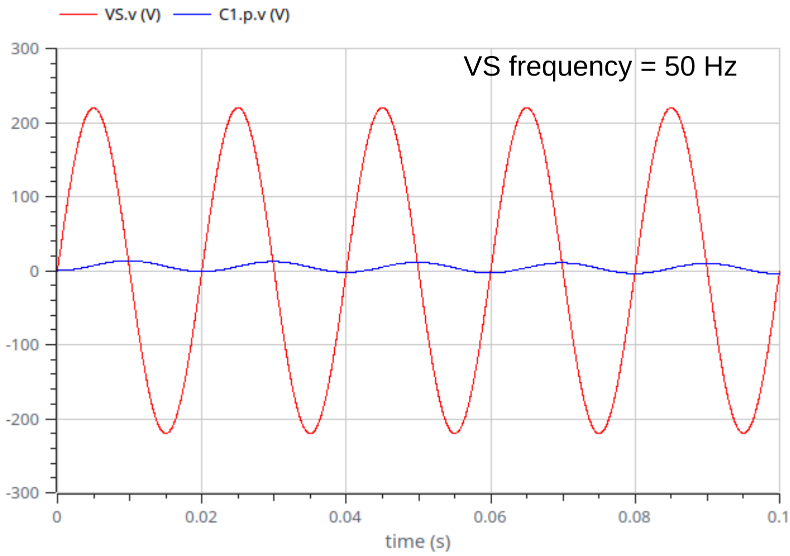
simulation run statistics

```
/home/hv/src/courses/courses/24.10-12.BIPtwinning/Lectures/24.11.04.Modelica/ModelicaTutorial/generatedCode/
ElectricalCircuit/ElectricalCircuit -port=46007 -logFormat=xmltcp -
override=startTime=0,stopTime=1,stepSize=0.002,tolerance=1e-06,solver=dassl,outputFormat=mat,variableFilter=.* -
r=/home/hv/src/courses/courses/24.10-12.BIPtwinning/Lectures/24.11.04.Modelica/ModelicaTutorial/generatedCode/
ElectricalCircuit/ElectricalCircuit_res.mat -w -lv=LOG_STDOUT,LOG_ASSERT,LOG_STATS -inputPath=/home/hv/src/
courses/courses/24.10-12.BIPtwinning/Lectures/24.11.04.Modelica/ModelicaTutorial/generatedCode/ElectricalCircuit -
outputPath=/home/hv/src/courses/courses/24.10-12.BIPtwinning/Lectures/24.11.04.Modelica/ModelicaTutorial/
generatedCode/ElectricalCircuit
The initialization finished successfully without homotopy method.
### STATISTICS ###
timer
 0.000811424s          reading init.xml
  9.6951e-05s          reading info.xml
 0.000222972s [  3.7%] pre-initialization
  6.1091e-05s [  1.0%] initialization
    7.91e-06s [  0.1%] steps
    0.001742s [ 28.8%] solver (excl. callbacks)
 0.000352641s [  5.8%] creating output-file
 0.000344505s [  5.7%] event-handling
  8.0611e-05s [  1.3%] overhead
  0.00324135s [ 53.5%] simulation
  0.00605308s [100.0%] total
events
    0 state events
    0 time events
solver: dassl
 2938 steps taken
 3255 calls of functionODE
   44 evaluations of jacobian
   28 error test failures
    0 convergence test failures
1.8071e-05s time of jacobian evaluation
The simulation finished successfully.
```

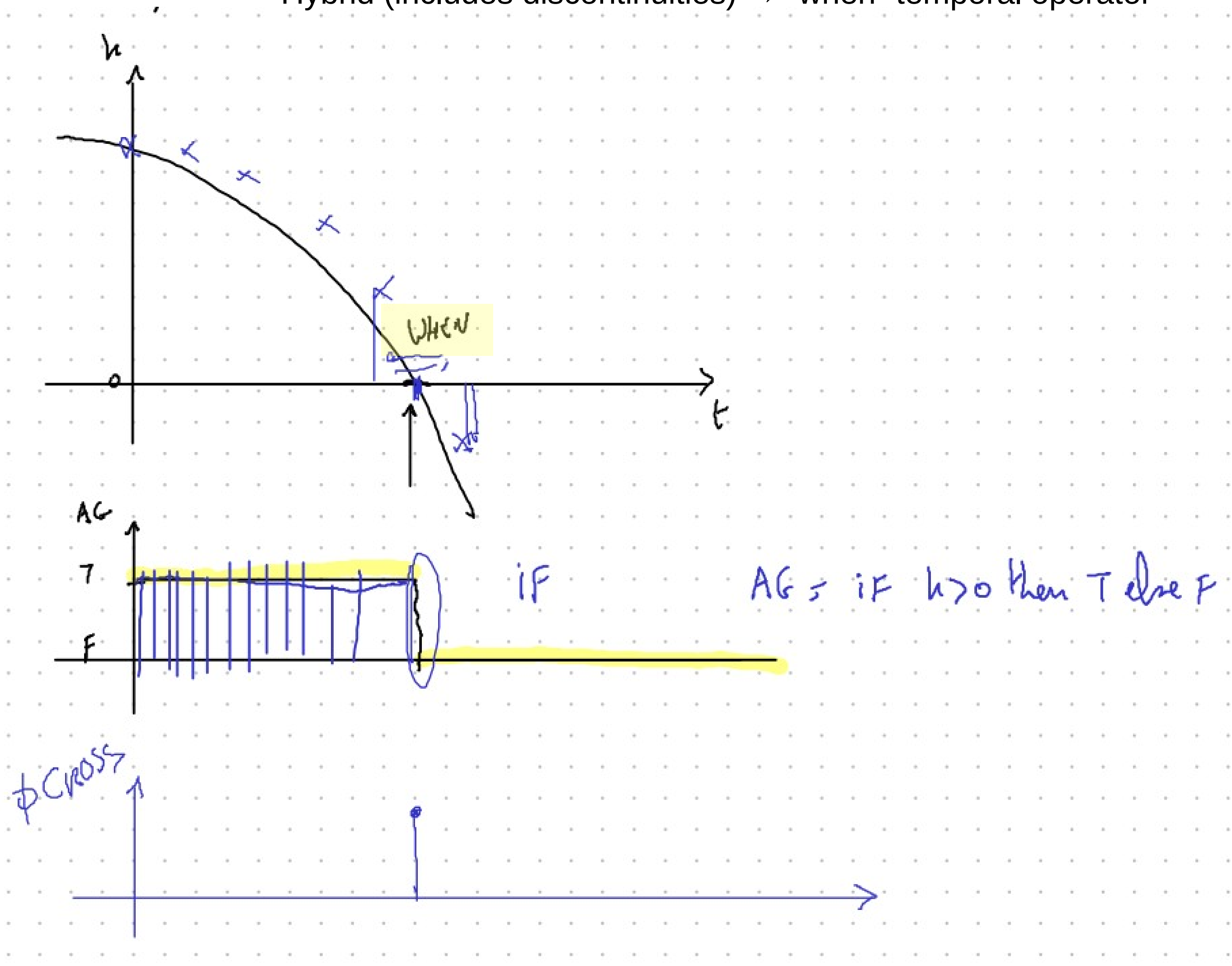
"low (frequency) pass filter"





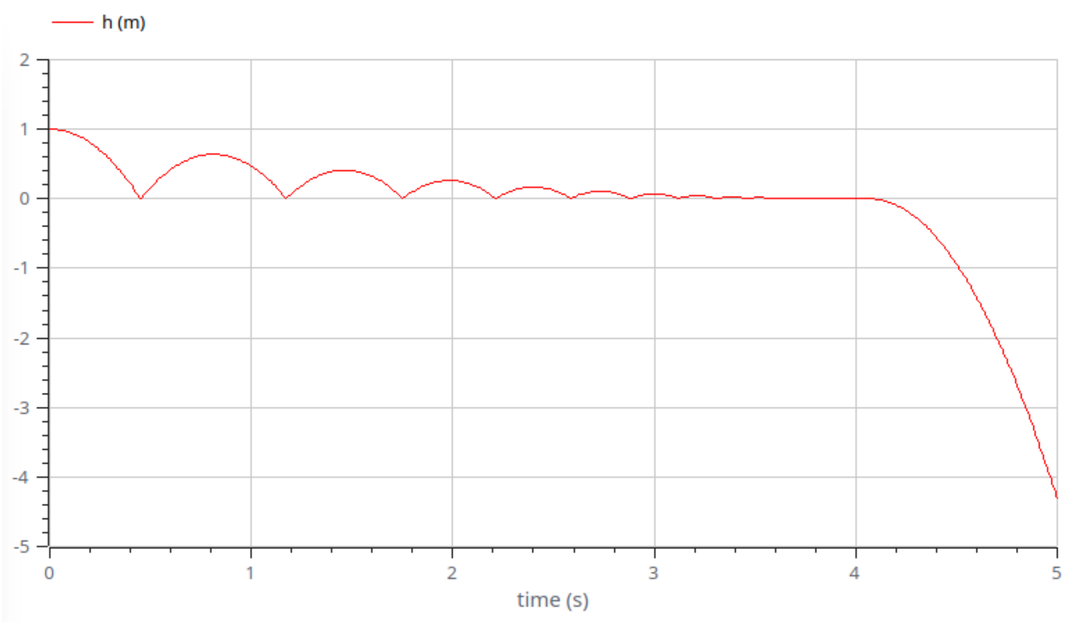


Hybrid (includes discontinuities) → "when" temporal operator



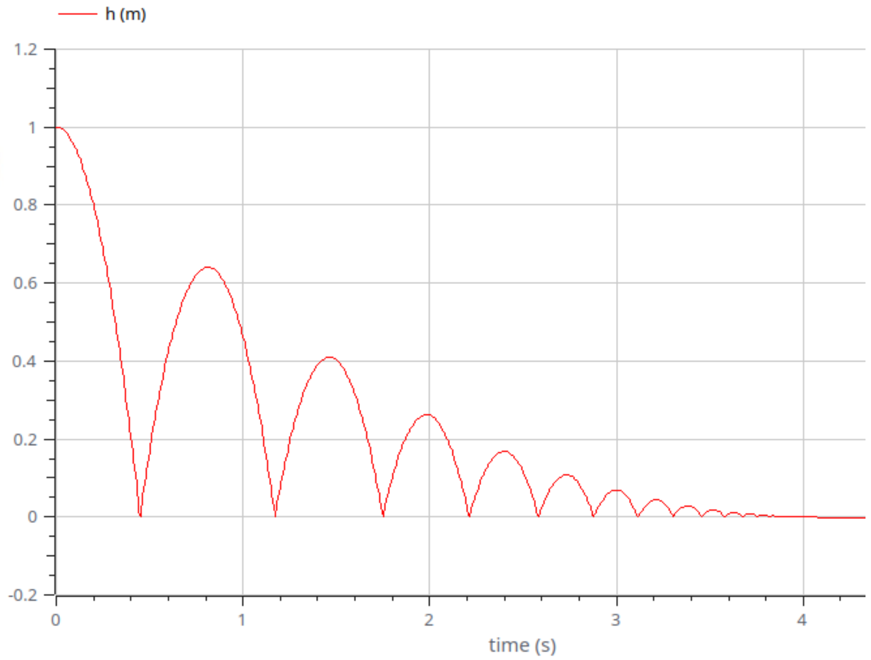
Hybrid (includes discontinuities) -- unstable

```
model unstable_bouncing_ball "The 'classic' bouncing ball model"
  type Height=Real(unit="m");
  type Velocity=Real(unit="m/s");
  parameter Real e=0.8 "Coefficient of restitution";
  parameter Height h0=1.0 "Initial height";
  Height h "Height";
  Velocity v(start=0.0, fixed=true) "Velocity";
initial equation
  h = h0;
equation
  v = der(h);
  der(v) = -9.81;
  when h<0 then
    reinit(v, -e*pre(v));
  end when;
end unstable_bouncing_ball;
```

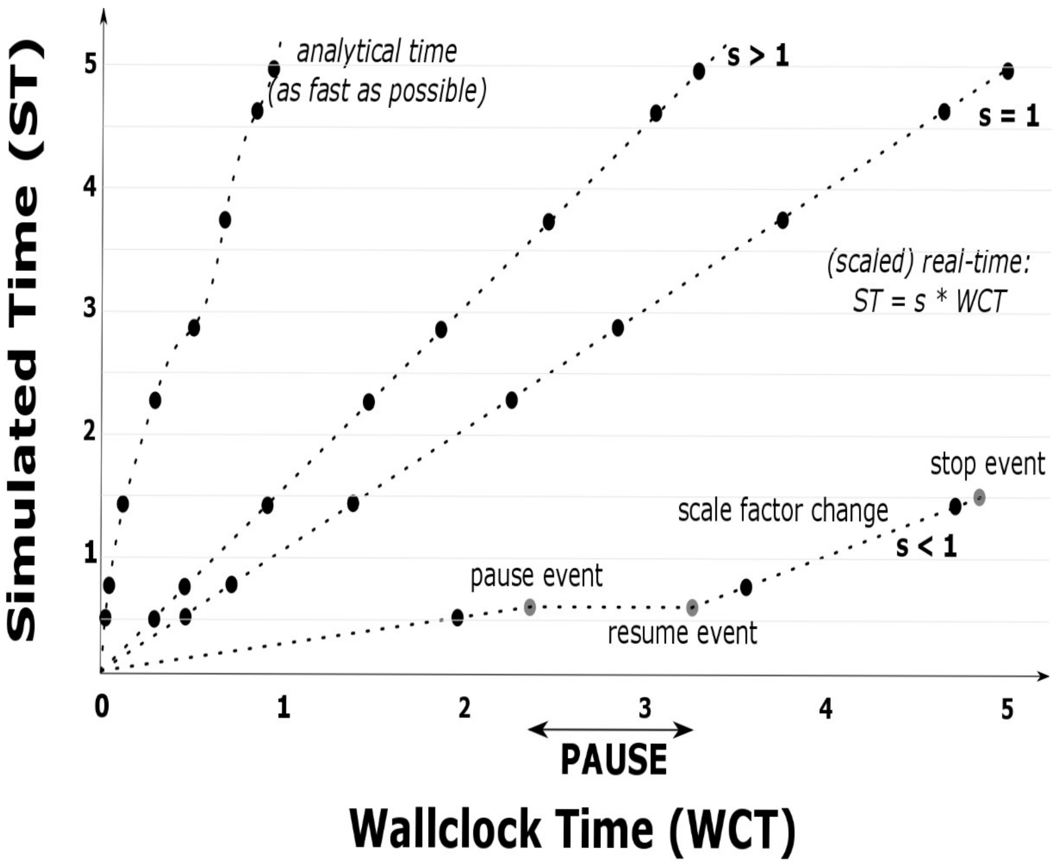


Hybrid (includes discontinuities) -- stable

```
model StableBouncingBall
  "The 'classic' bouncing ball model with numerical tolerances"
  type Height=Real(unit="m");
  type Velocity=Real(unit="m/s");
  parameter Real e=0.8 "Coefficient of restitution";
  parameter Height h0=1.0 "Initial height";
  constant Height eps=1e-3 "Small height";
  Boolean done "Flag when to turn off gravity";
  Height h "Height";
  Velocity v(start=0.0, fixed=true) "Velocity";
initial equation
  h = h0;
  done = false;
equation
  v = der(h);
  der(v) = if done then 0 else -9.81;
  when {h<0, h<-eps} then
    done = h<-eps;
    reinit(v, if h<-eps then 0 else -e*pre(v));
  end when;
end StableBouncingBall;
```



(scaled) real-time simulation (execution)

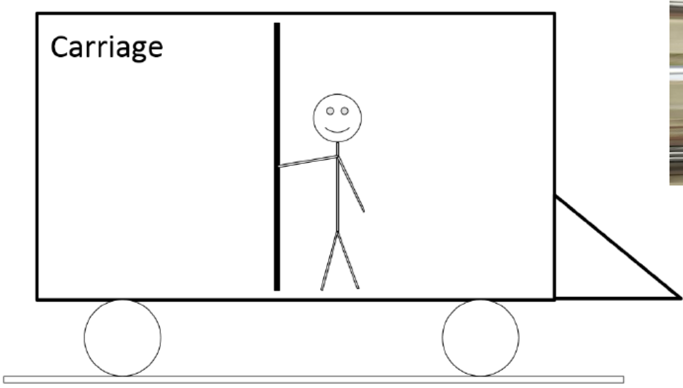


Controller Design and Tuning

Control System

- A *control system* (or "controller") is a system whose purpose is to command, direct, or regulate itself, or another system.
- The System under Control is often called a "plant" (as in "chemical production plant").
- There are open-loop and closed-loop control systems.
  - Closed-loop control system: e.g., human picking an object
    - Eyes are *sensors*.
    - Hands are *actuators*.
    - Brain is the *controller* that estimates the distance between hand and object based on sensor input.
      It determines/computes an appropriate *control action* that satisfies requirements and implements it through the *actuators*.
  - Open-loop control system: e.g., blindfolded picking
    - Only the current state and a model of the plant are used. The output of the system under control is not observed.
- Our example (closed loop): velocity control in rail car

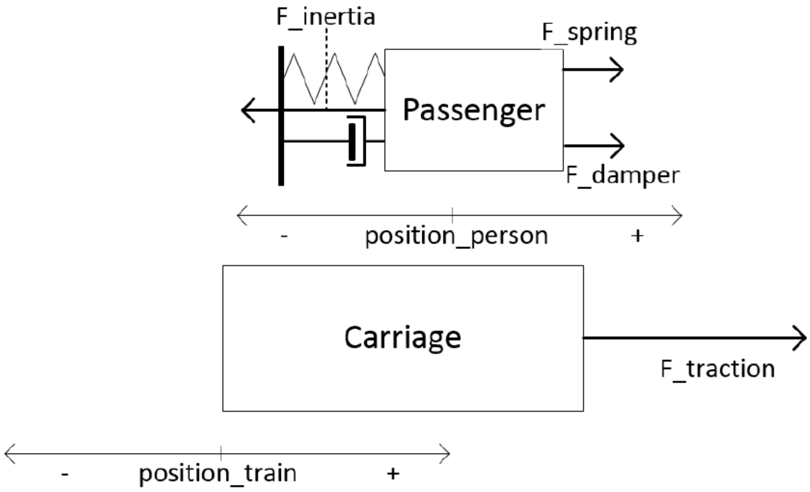
Rail Car Case



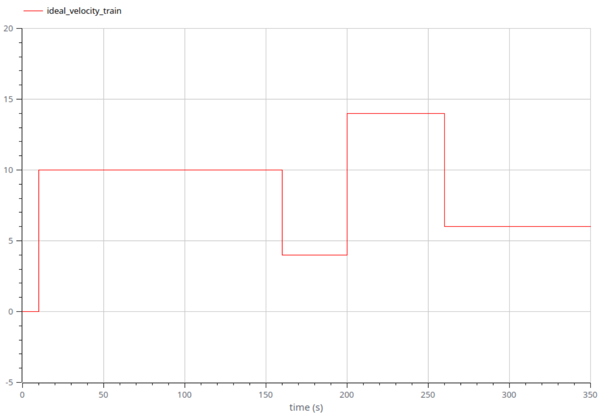
- Build the controller for a driverless rail car.
- The controller determines the acceleration of the train, in an attempt to match (i.e., deviate as little as possible from) a predefined profile of desired velocities.

  The desired (piecewise constant) velocity profile is known beforehand by a central coordinator (and is encoded in a file).
- Passengers should not fall (*i.e.,* accelerate too much).
- Other requirements such as minimizing total energy consumption could be added.

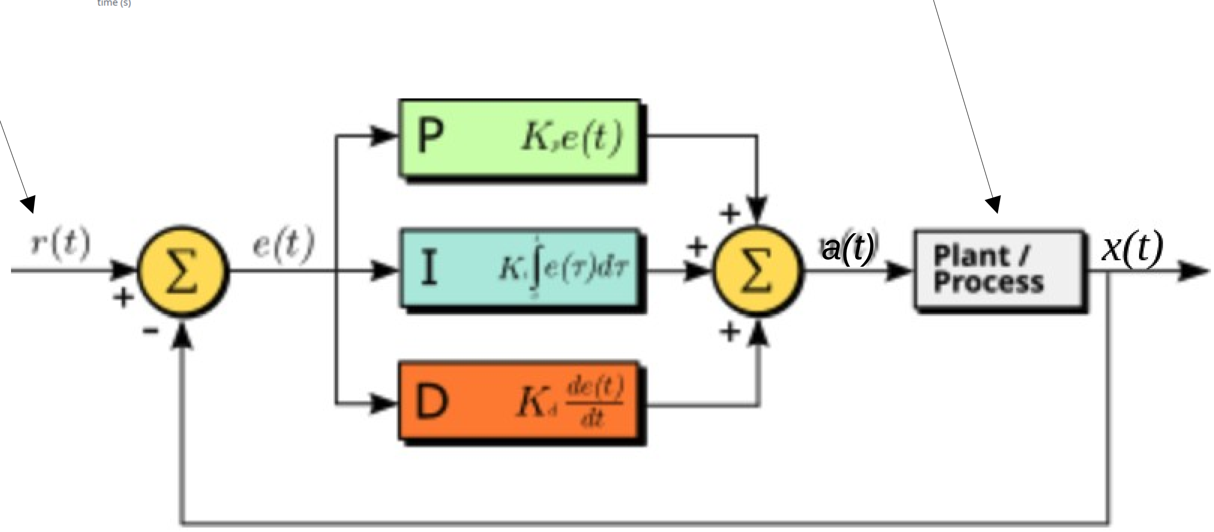
Abstracting Train-and-Passenger ("Plant" model)

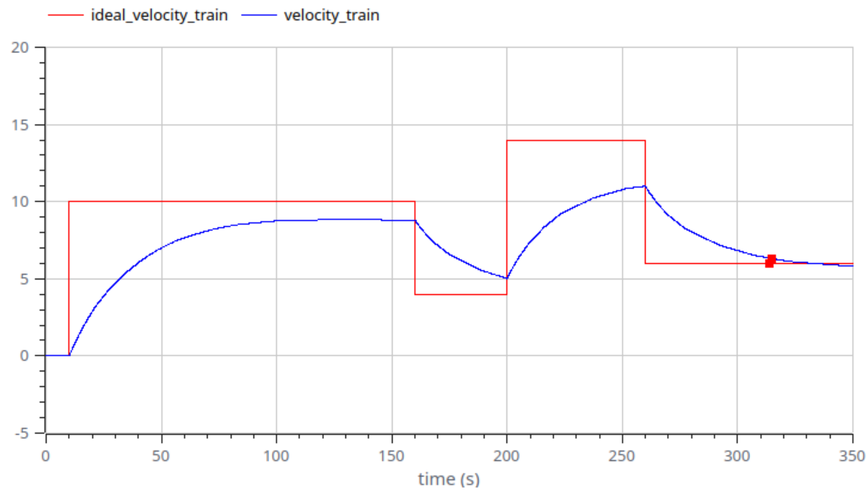


$$
\begin{cases}
m_{passger} * a_{passger} & = k(-x_{passger}) + c(-v_{passger}) - m_{passger} * a_{train} \\
F_{traction} & = (m_{train} + m_{passger}) * a_{train} \\
a_{passger} & = \frac{dv_{passger}}{dt} \\
v_{passger} & = \frac{dx_{passger}}{dt} \\
a_{train} & = \frac{dv_{train}}{dt} \\
v_{train} & = \frac{dx_{train}}{dt}
\end{cases}
$$

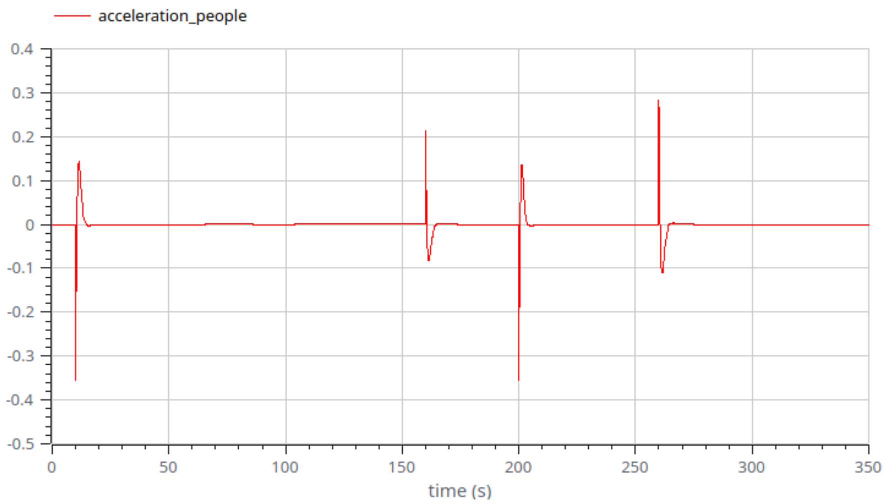


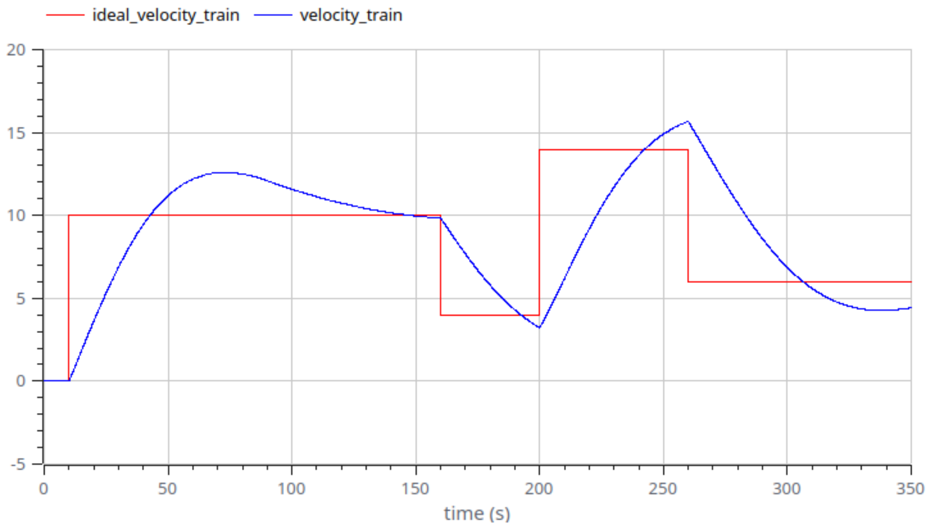
$$\begin{cases} m_{passger} * a_{passger} & = k(-x_{passger}) + c(-v_{passger}) - m_{passger} * a_{train} \\ F_{traction} & = (m_{train} + m_{passger}) * a_{train} \\ a_{passger} & = \frac{dv_{passger}}{dt} \\ v_{passger} & = \frac{dx_{passger}}{dt} \\ a_{train} & = \frac{dv_{train}}{dt} \\ v_{train} & = \frac{dx_{train}}{dt} \end{cases}$$



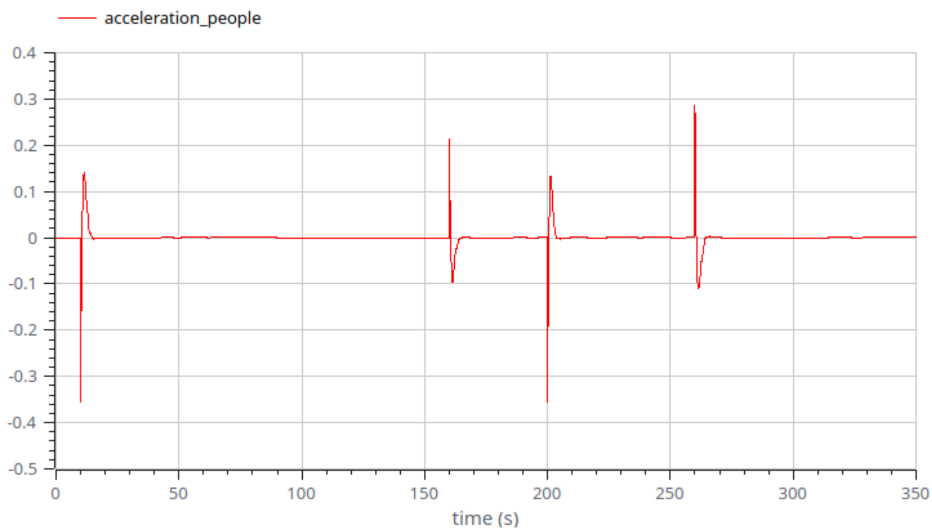


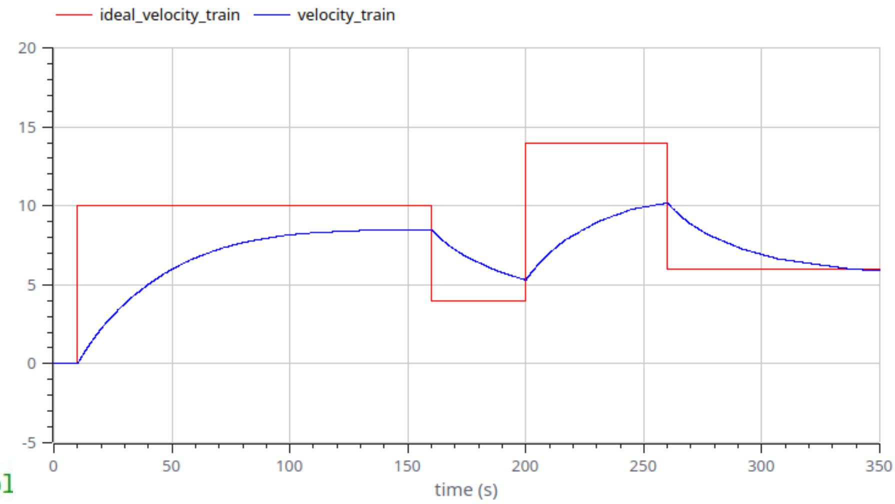

```
  // a Proportional (P) controller
parameter Real control_error_proportion = 200;
parameter Real control_int_error_proportion = 0;
parameter Real control_der_error_proportion = 0;
```



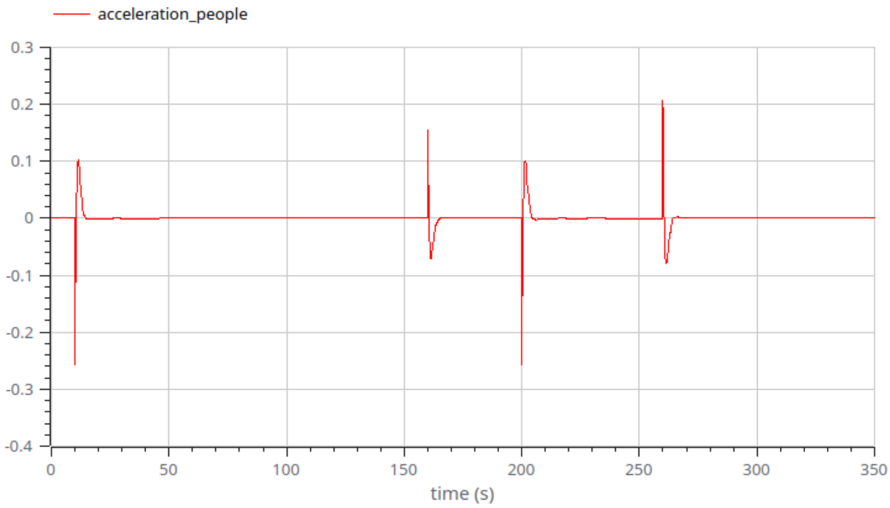



```
  // a Proportional and Integral (PI) controller
parameter Real control_error_proportion = 200;
parameter Real control_int_error_proportion = 10;
parameter Real control_der_error_proportion = 0;
```

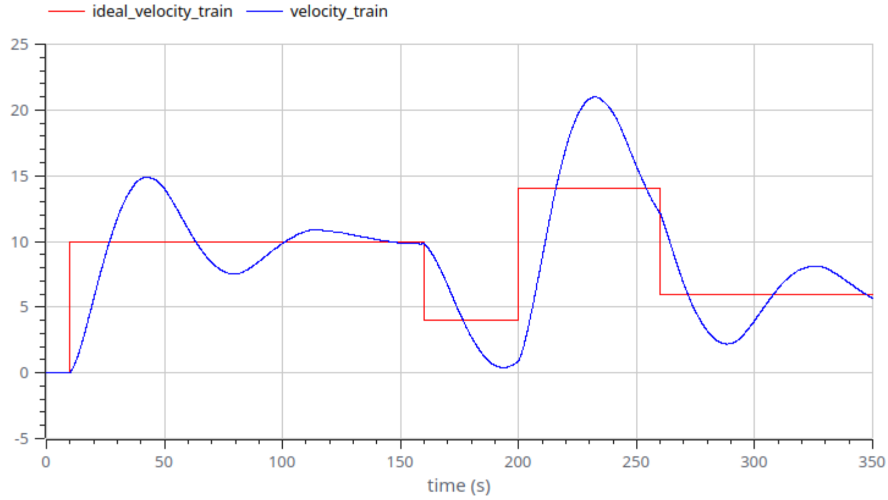




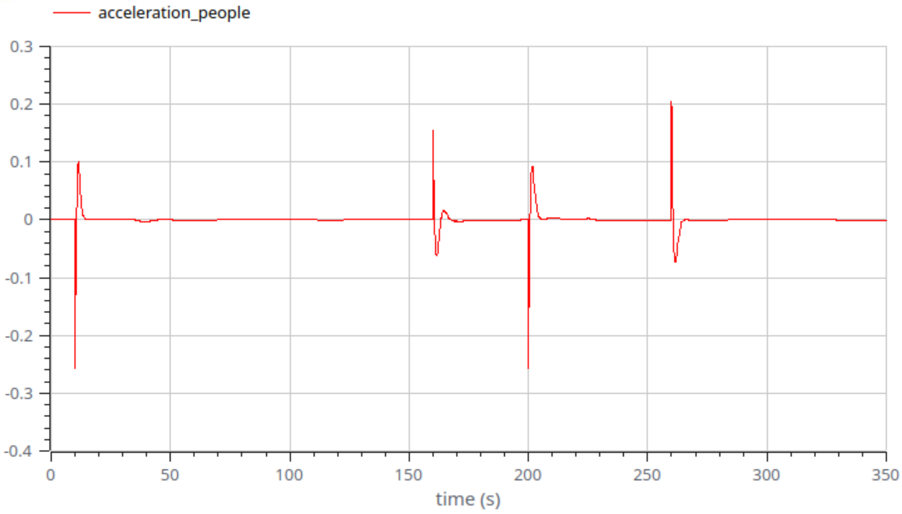

```
  // a Proportional and Derivative (PD) control
parameter Real control_error_proportion = 150;
parameter Real control_int_error_proportion = 0;
parameter Real control_der_error_proportion = 200;
```



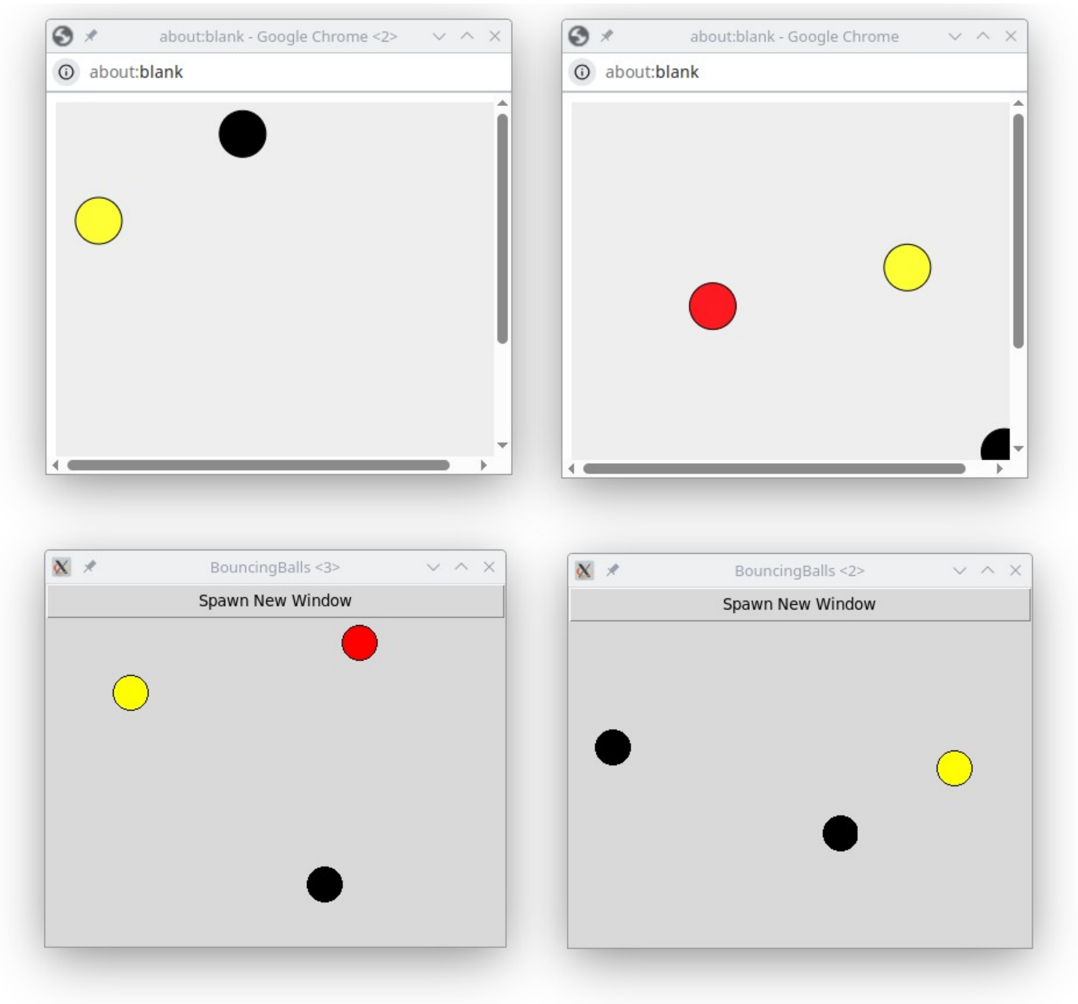
PID controller
constrained multi-criteria **optimization**




```
  // a PID controller
parameter Real control_error_proportion = 150;
parameter Real control_int_error_proportion = 50;
parameter Real control_der_error_proportion = 200;
```



Dynamic Structure



SS-CPSIoT

6th Annual Summer School on
Cyber Physical Systems and Internet of Things

Developing Reactive Systems using Statecharts

for Modelling, Simulation and Synthesis of Software-Intensive Systems

Hans Vangheluwe, Rakshit Mittal, Joeri Exelmans and Simon Van Mierlo

13 June 2025 – Budva, Montenegro

Universiteit Antwerpen
Ansymo Antwerp Systems & Software Modelling University of Antwerp
Nexor Cyber-Physical Systems University of Antwerp
FLANDERS MAKE MANUFACTURING INNOVATION NETWORK
MPM4CPS
cost

Introduction

Modelling Reactive Systems

- Interaction with the environment: reactive to *events*
- Autonomous behaviour: *timeouts* + *spontaneous transitions*
- System behaviour: *modes* (*hierarchical*) + *concurrent* units
- Use programming language + threads and timeouts (OS)?

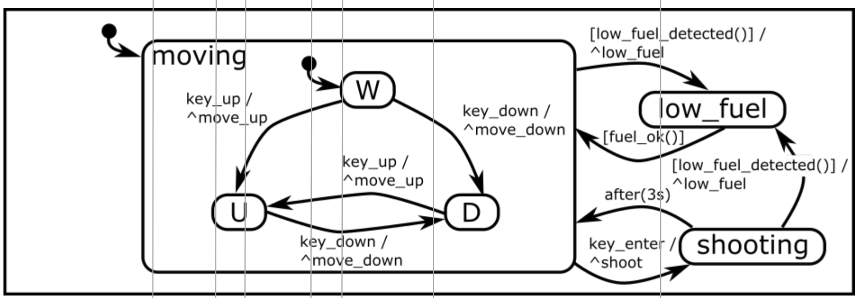
Programming language (and OS) is too low-level
-> most appropriate formalism: “what” vs. “how”

“Nontrivial software written with threads, semaphores, and mutexes are incomprehensible to humans”

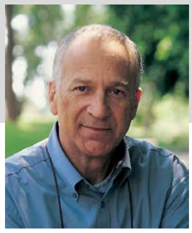
E. A. Lee, "The problem with threads," in *Computer*, vol. 39, no. 5, pp. 33-42, May 2006.

Discrete-Event Abstraction

behavioural
model



Statecharts History



- Introduced by David Harel in 1987
- Notation based on higraphs = hypergraphs + Euler diagrams + unordered Cartesian product
- Extends deterministic finite state automata with:
  - Depth (Hierarchy)
  - Orthogonality
  - Broadcast Communication
  - Time
  - History
  - Syntactic sugar, such as enter/exit actions

David Harel, Statecharts: a visual formalism for complex systems, Science of Computer Programming, Volume 8, Issue 3, 1987, pages 231-274

Statecharts History

- Incorporated in UML: State Machines (1995)
- xUML for semantics of UML subset (2002)
- W3 Recommendation: State Chart XML (SCXML) (2015)

  https://www.w3.org/TR/scxml/
- Standard: Precise Semantics for State Machines (2019)

  https://www.omg.org/spec/PSSM/

Statechart (Variants) Tools

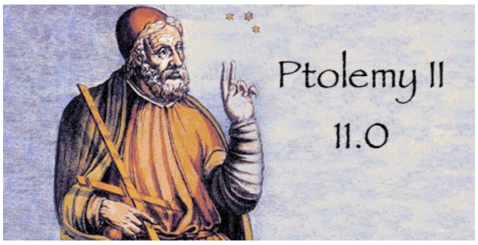
STATEMATE: A Working Environment for the Development of Complex Reactive Systems

Rational software

https://www.ibm.com/us-en/marketplace/systems-design-rhapsody

Matlab Simulink Stateflow

https://www.mathworks.com/products/stateflow.html



https://ptolemy.berkeley.edu/ptolemyII/ptII11.0/index.htm

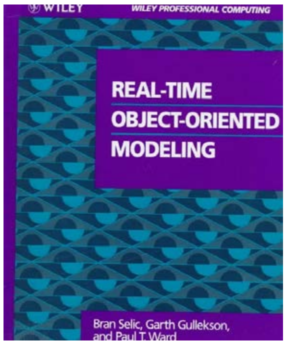
YAKINDU STATECHART TOOLS

itemis CREATE

https://www.itemis.com/en/products/itemis-create/

PAPYRUS REALTIME

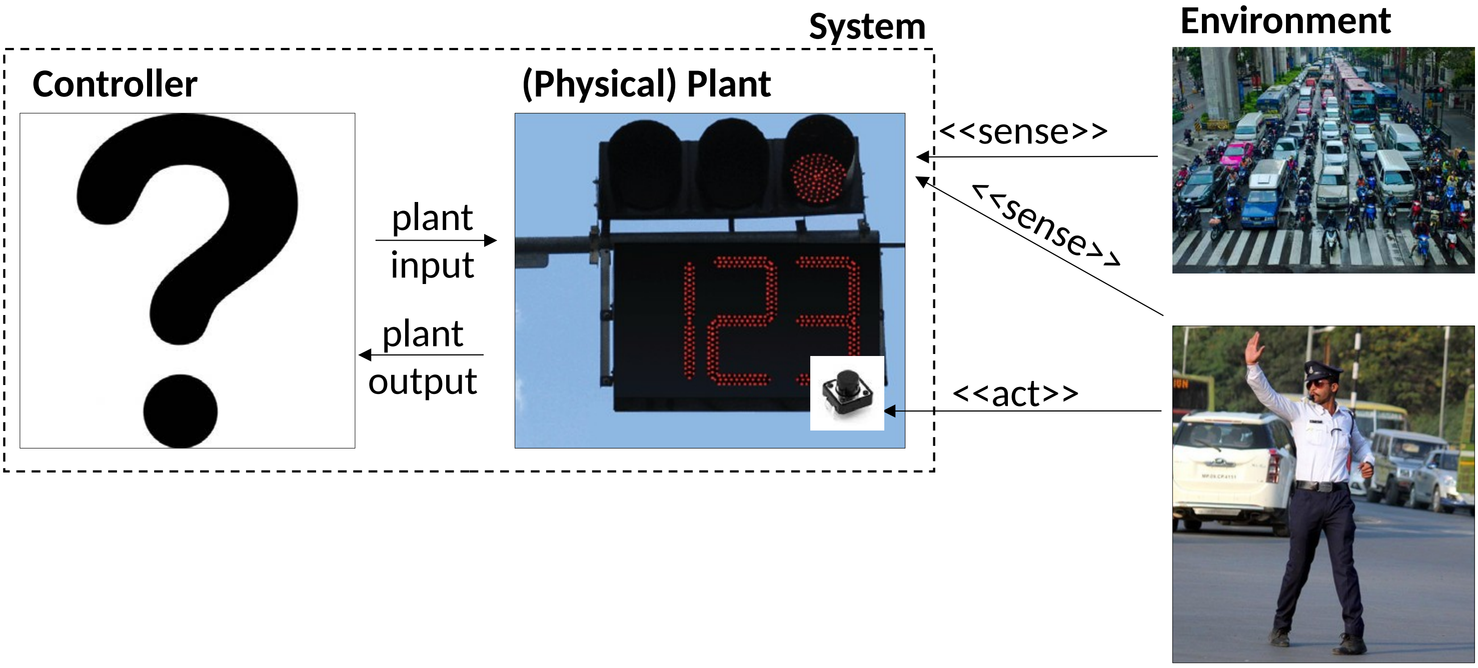
https://www.eclipse.org/papyrus-rt/



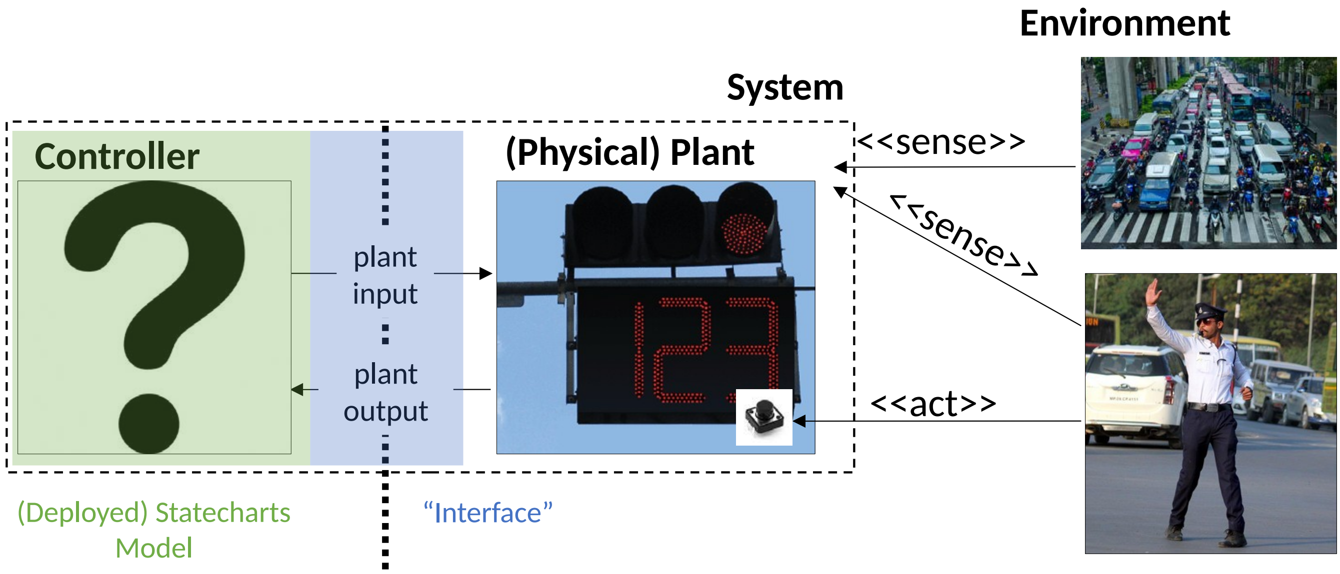
etrice.

https://www.eclipse.org/etrice/

Running Example



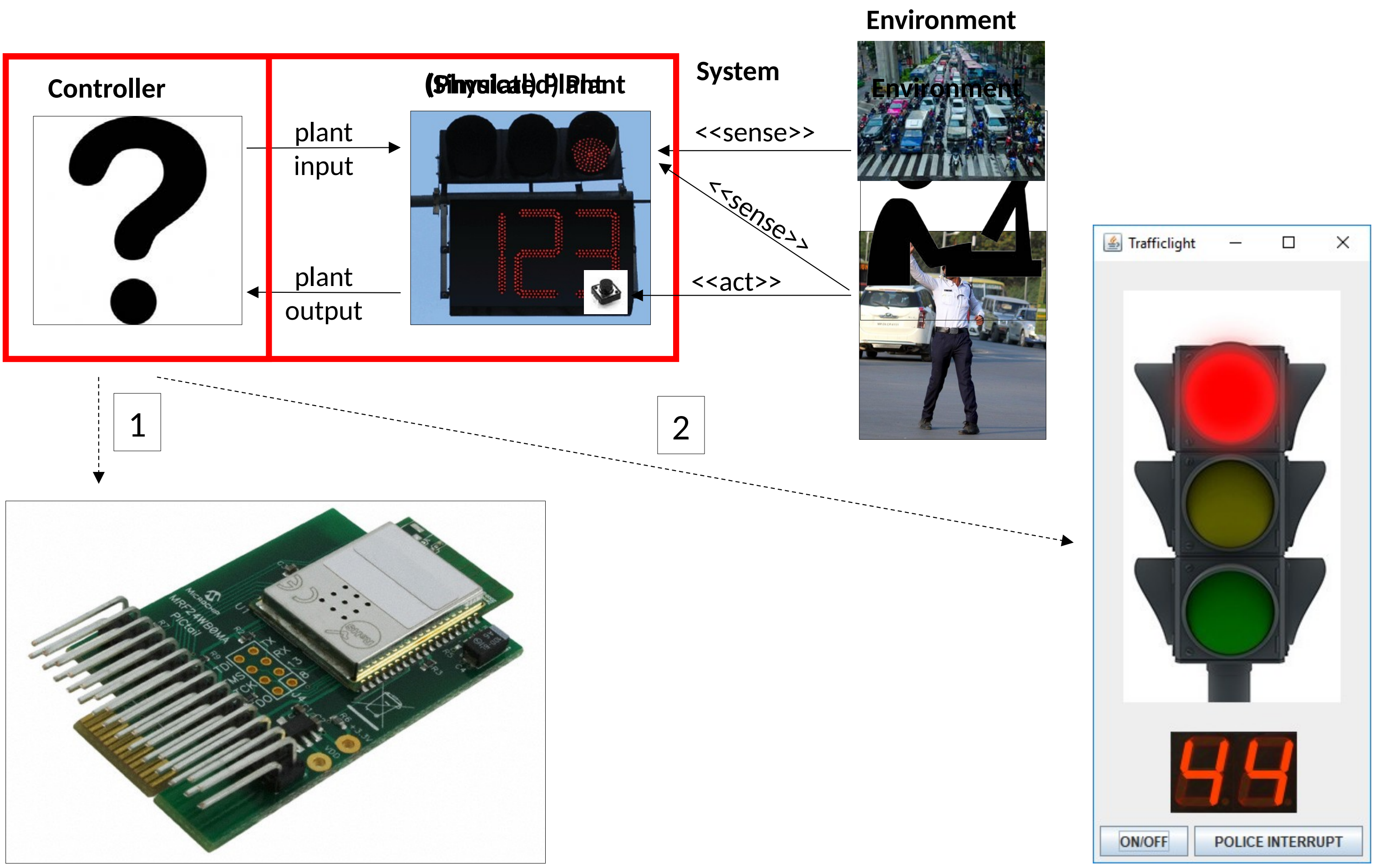
What are we developing?



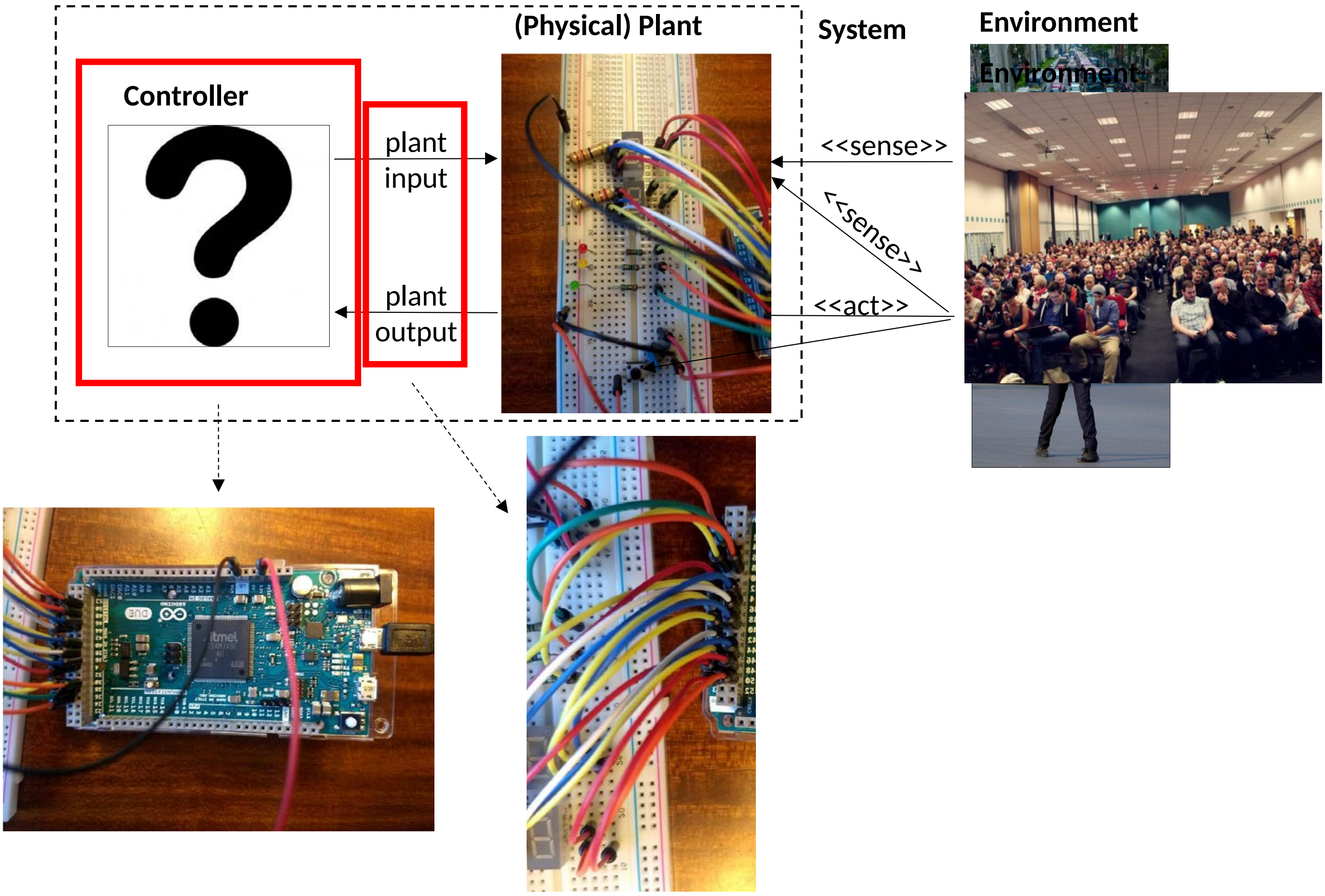
- Autonomous (timed) behaviour
- Interrupt logic
- Orthogonal (traffic light/timer) behaviour

- Turn on/off traffic lights (red/green/yellow)
- Display counter value (three-digit)
- Change counter colour (red/green)
- Sense button presses

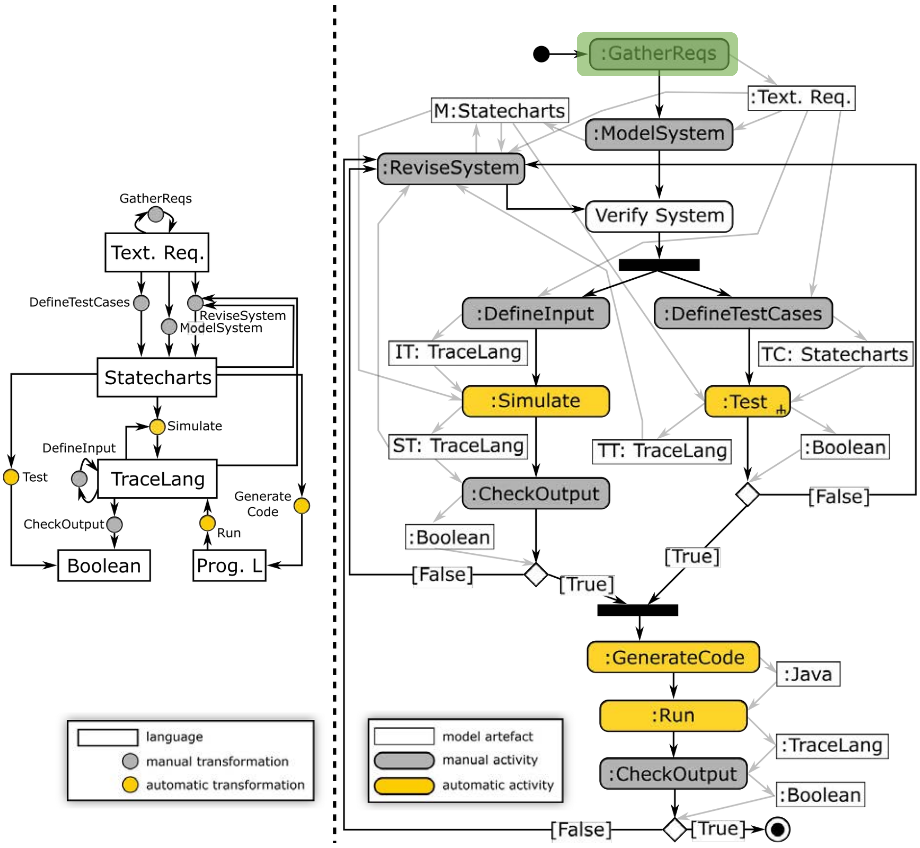
Deployment (Simulation)



Deployment (Hardware)



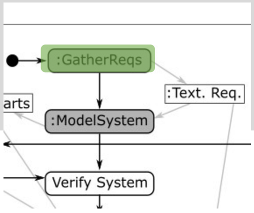
Workflow



Hans Vangheluwe and Ghislain C. Vansteenkiste. A multi-paradigm modeling and simulation methodology: Formalisms and languages. In European Simulation Symposium (ESS), pages 168-172. Society for Computer Simulation International (SCS), October 1996. Genoa, Italy.

Levi Lúcio, Sadaf Mustafiz, Joachim Denil, Hans Vangheluwe, Maris Jukss. FTG+PM: An Integrated Framework for Investigating Model Transformation Chains. System Design Languages Forum (SDL) 2013, Montreal, Quebec. LNCS Volume 7916, pp 182-202, 2013.

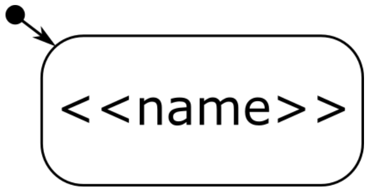
Requirements



- R1: three differently coloured lights: red, green, yellow
- R2: at most one light is on at any point in time
- R3: at system start-up, the red light is on
- R4: cycles through red on, green on, and yellow on
- R5: red is on for 60s, green is on for 55s, yellow is on for 5s
- R6: time periods of different phases are configurable.
- R7: police can interrupt autonomous operation
  - Result = blinking yellow light (on -> 1s, off -> 1s)
- R8: police can resume an interrupted traffic light
  - Result = light which was on at time of interrupt is turned on again
- R9: traffic light can be switched on and off and restores its state
- R10: a timer displays the remaining time while the light is red or green; this timer decreases and displays its value every second. The colour of the timer reflects the colour of the traffic light.

The Statecharts Language

States



being **in** a state

= state <<*name*>> is **active**

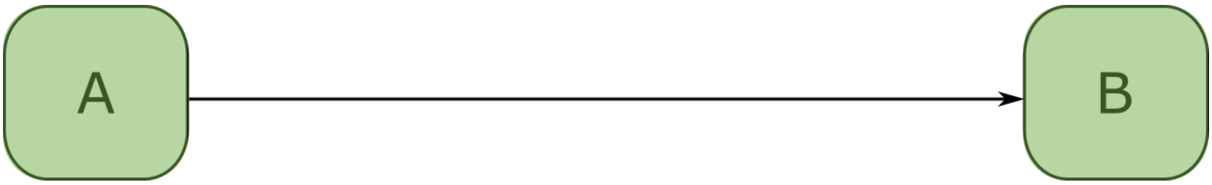
= the system is in **configuration** <<*name*>>

initial state

exactly one per model

“entry point”

Transitions

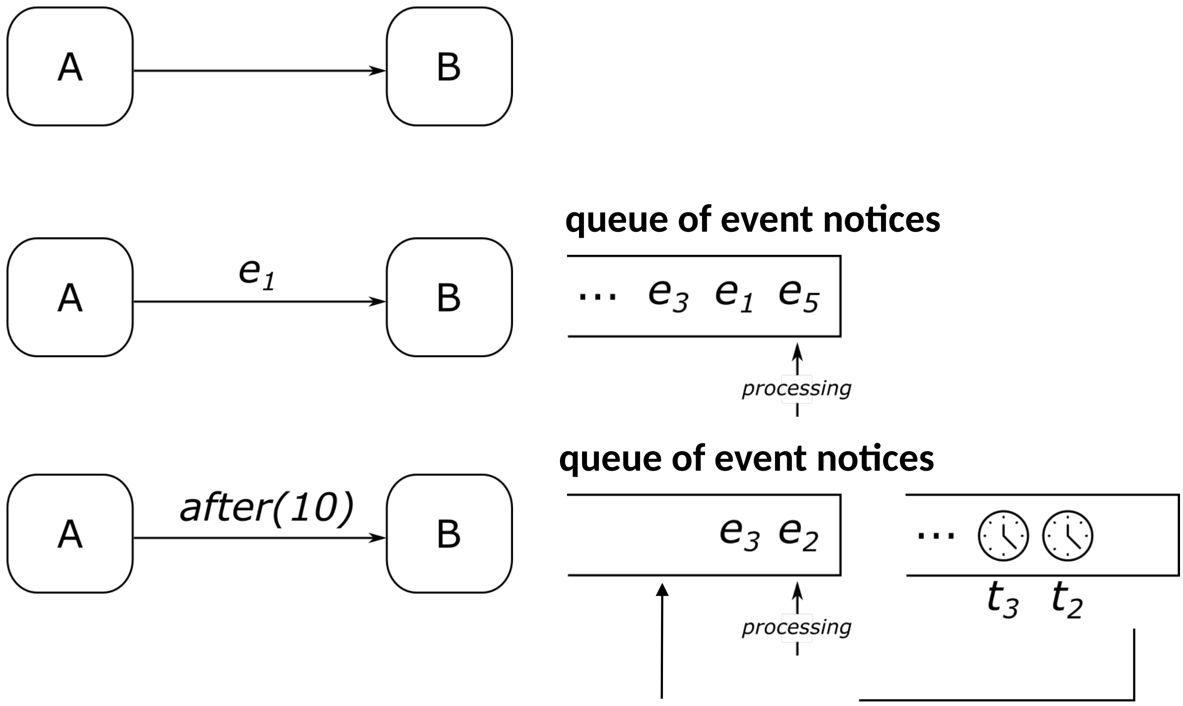


- Model the **dynamics** of the system:
  - *when*
    - the system is **in state A** and
    - the **event** is **received**
  - *then*
    1. **output_action** is evaluated and
    2. the new **active state** becomes B

Transitions: Events

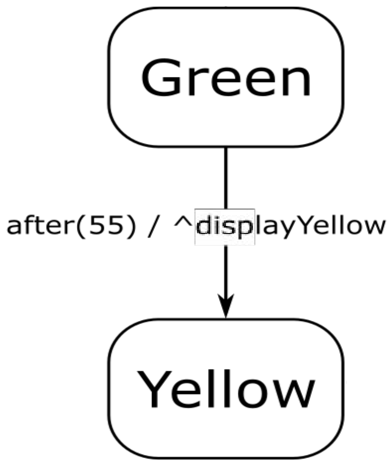
event(in_params) / output_action(out_params)

- Spontaneous
- Input Event
- After Event



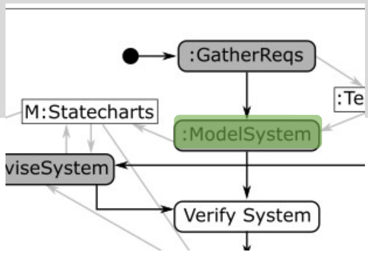
Transitions: Raising Output Events

event(in_params) / output_action(out_params)



Syntax for output action:
^*output_event*
means "raise the event *output_event* (to the environment)"

Exercise 1 - Requirements



Your model here.

- R1: three differently coloured lights: red (R), green (G), yellow (Y)
- R2: at most one light is on at any point in time
- R3: at system start-up, the red light is on
- R4: cycles through red on, green on, and yellow on
- R5: red is on for 60s, green is on for 55s, yellow is on for 5s

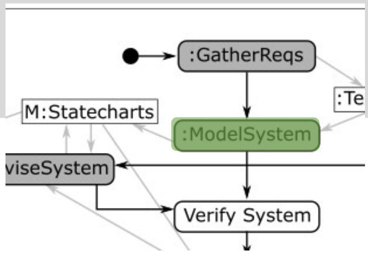
(Simulated) Plant

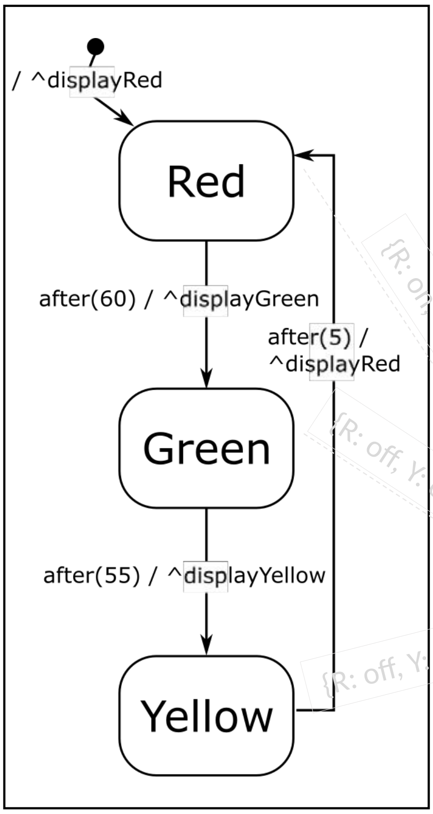
Environment

<<control>>

<<observe>>

Exercise 1 - Solution





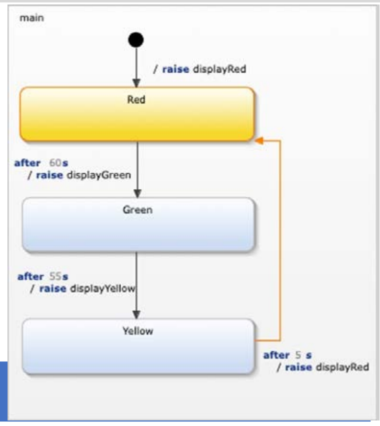
- R1: three differently coloured **lights**: red (R), green (G), yellow (Y)
- R2: at most one light is on at any point in time
- R3: at system start-up, the **red** light is on
- R4: cycles through red on, green on, and yellow on
- R5: **red** is on for 60s, **green** is on for 55s, **yellow** is on for 5s

(Simulated) Plant

<<observe>>

Environment

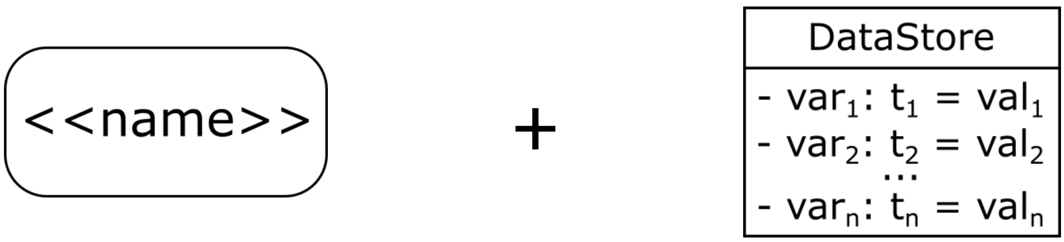
Exercise 1 - Solution



requirement	modelling approach
R1: three differently coloured **lights**: red (R), green (G), yellow (Y)	For each colour a **state** is defined. Transitions that lead to a state raise the proper out event which interacts with the plant.
R2: at most one light is on at any point in time	The states are all contained in a single region and thus a exclusive to each other ("or" states).
R3: at system start-up, the red light is on	The entry node points to state Red and the entry transition raises the event displayRed.
R4: cycles through red on, green on, and yellow on	The transitions define this cycle.
R5: red is on for 60s, green is on for 55s, yellow is on for 5s	Time events are specified on the transitions.


Data Store

Full System State

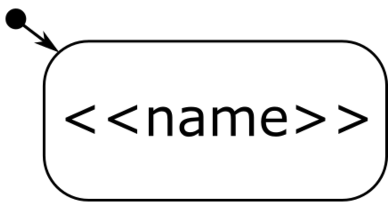


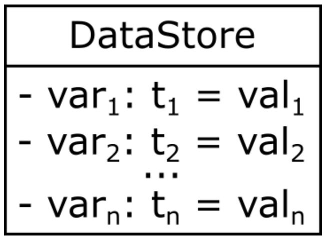
being **in** a state = state <<*name*>> is **active** = the system is in **configuration** <<*name*>>	data store **snapshot** = variable values

=

full system state

Full System State: Initialization






```
int main() {

}
```

initial state

exactly one per model

"entry point"

provide **default value** for each variable

"initial snapshot"

Compare:

C++ initialization

implicit state

(program counter)

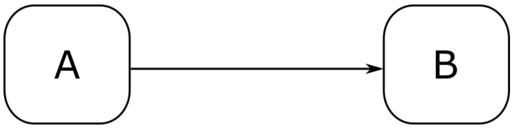
\+ **data store**

Transitions: Guards

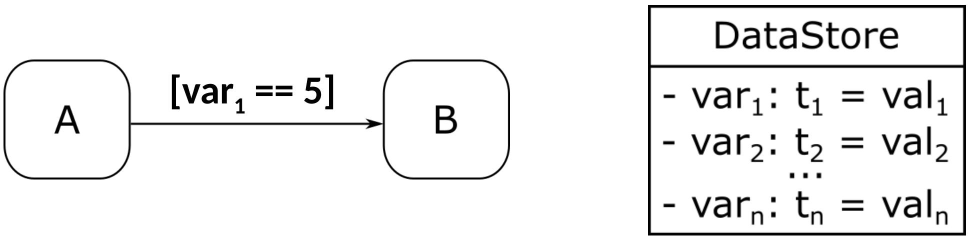
event(in_params) [guard] / output_action(out_params)

Modelled by "guard expression" (evaluates to Boolean) in some appropriate language

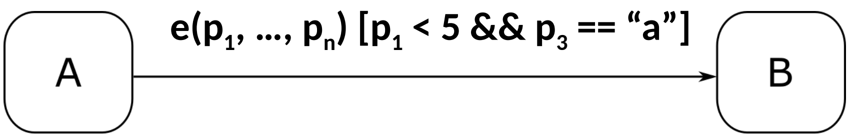
- Spontaneous [True]



- Data Store Variable Value



- Parameter Value

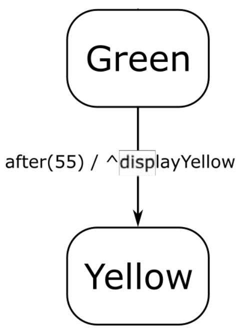


Transitions: Output Actions

event(params) [guard] / output_action(params)

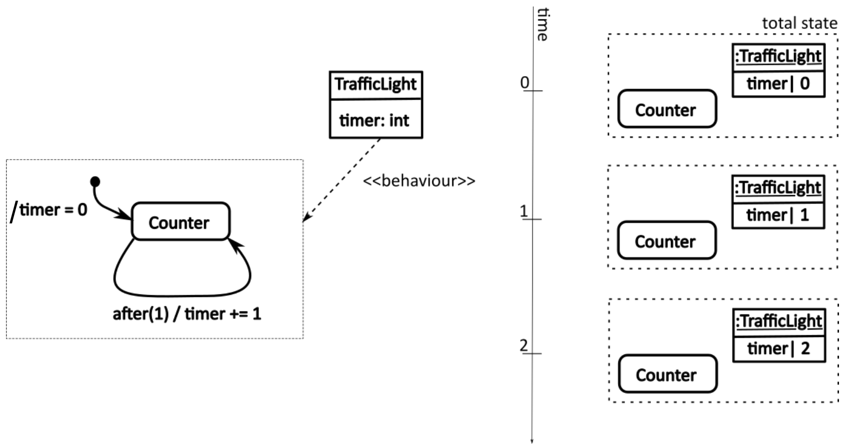
Output Event

^*output_event*(p_1, p_2, ..., p_n)

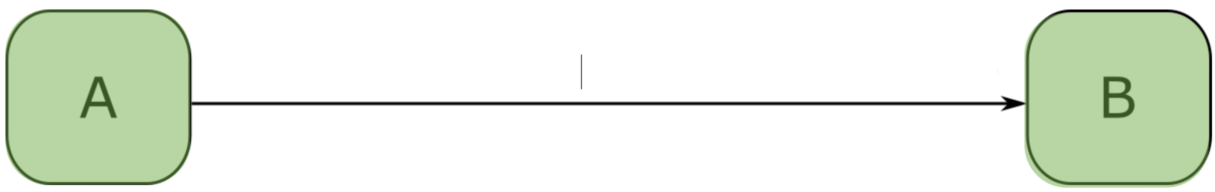


Assignment (to the non-modal part of the state)

- by action code in some appropriate language



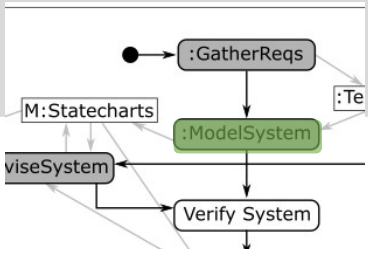
Transitions



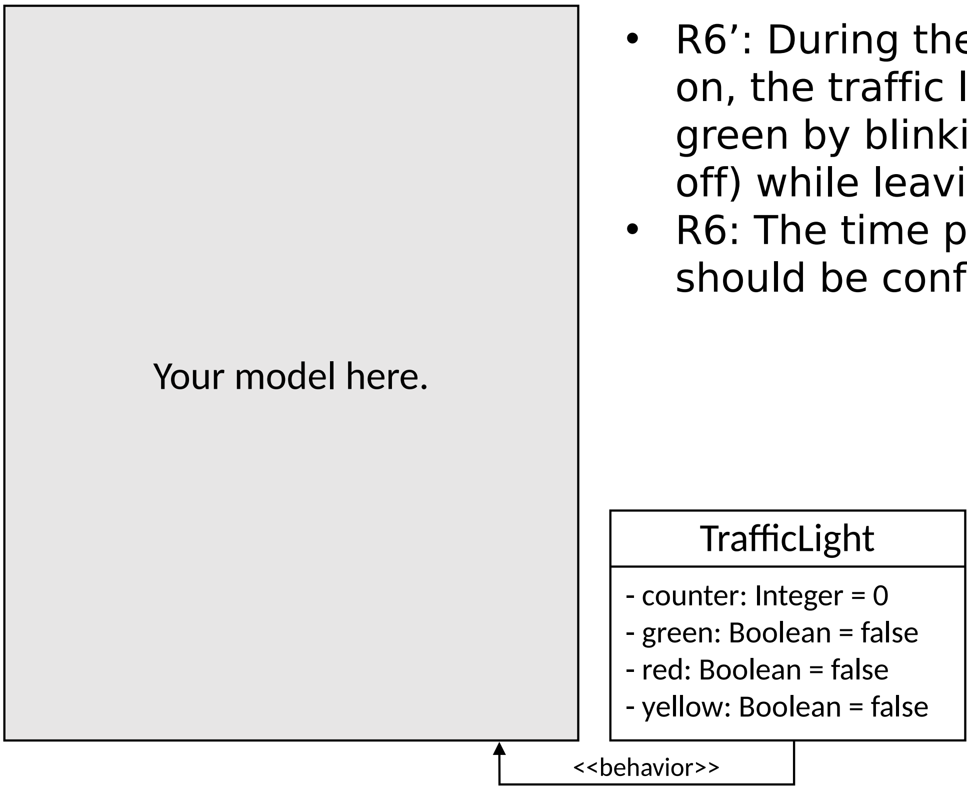
- Model the **dynamics** of the system:
  - *when*
    - the system is **in state A** and
    - **event is received** and
    - **guard** evaluates to **True**
  - *then*
    1. **output_action** is evaluated and
    2. the new **active state** becomes B


Exercise 2
Add data stores

Exercise 2 - Requirements

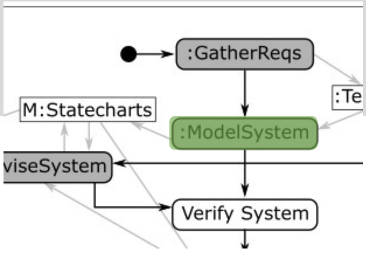


- R6': During the last 6 seconds of red being on, the traffic light announces to go to green by blinking its yellow light (1s on, 1s off) while leaving its red light on.
- R6: The time period of the different phases should be configurable.

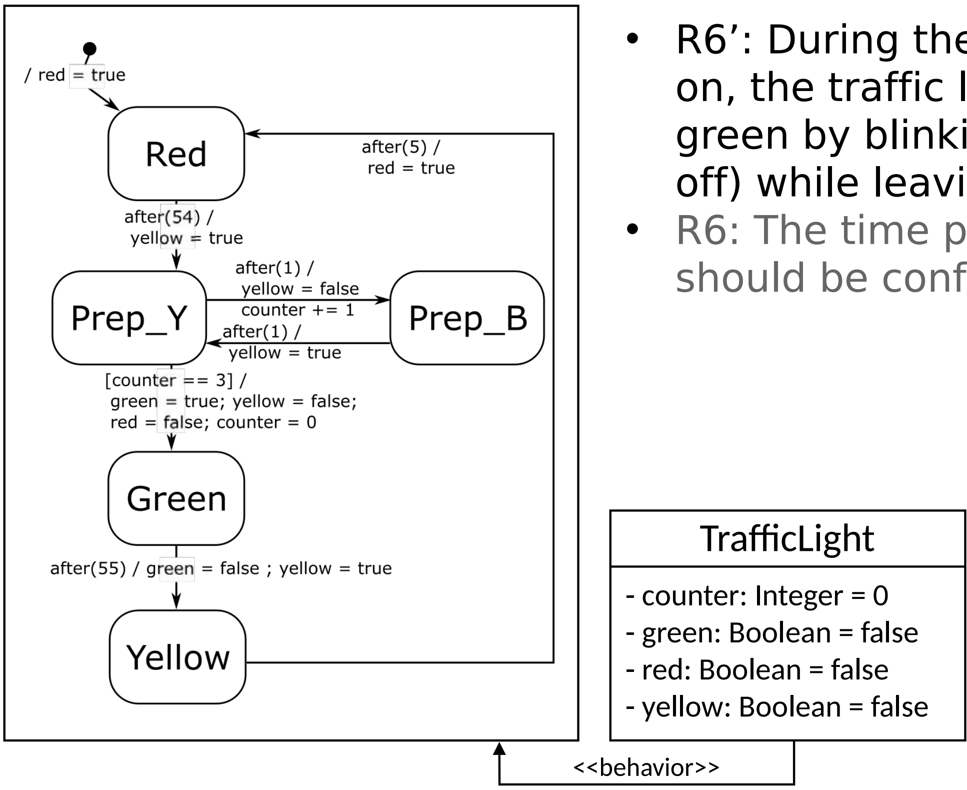


Make sure that:
- **the values of the variables reflect which lights are on/off**
- **you use at least one conditional transition**

Exercise 2: Solution



- R6': During the last 6 seconds of red being on, the traffic light announces to go to green by blinking its yellow light (1s on, 1s off) while leaving its red light on.
- R6: The time period of the different phases should be configurable.



Statechart Execution

8

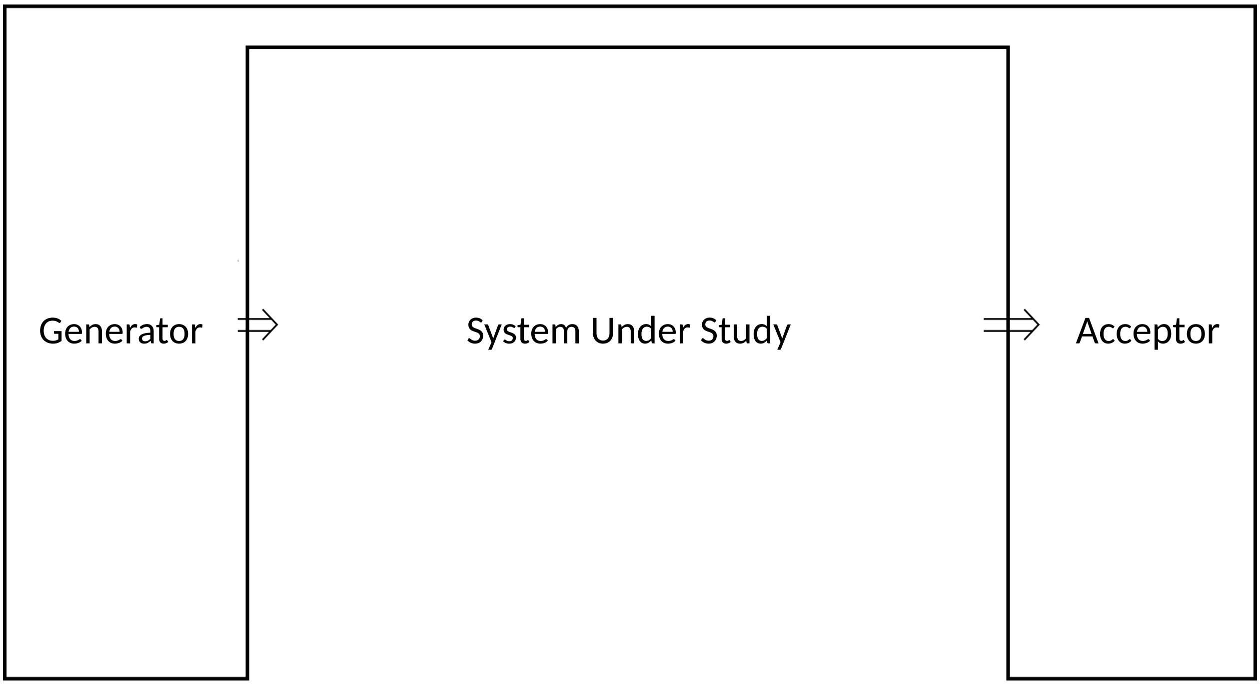
Run-To-Completion Step

- A Run-To-Completion (RTC) step is an *atomic execution step* of a state machine.
- It transitions the state machine from a *valid state configuration* into the next *valid state configuration*.
- RTC steps are executed one after the other - they must *not interleave*.
- New incoming events *cannot interrupt* the processing of the current event and must be stored in an *event queue*

Testing Statecharts

3

Testing Statecharts

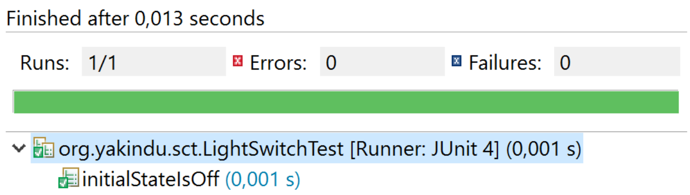


Zeigler BP. Theory of modelling and simulation. New York: Wiley-Interscience, 1976.

Mamadou K. Traoré, Alexandre Muzy, Capturing the dual relationship between simulation models and their context, Simulation Modelling Practice and Theory, Volume 14, Issue 2, February 2006, Pages 126-142.

SCTUnit (beta)

- X-unit testing framework for YAKINDU Statechart Tools
- Test-driven development of Statechart models
- Test generation for various platforms
- Executable in YAKINDU Statechart Tools
- Virtual Time




```
testclass LightSwitchTest for statechart Light_Switch{
    @Test
    operation initialStateIsOff(){
        enter
        assert active(Light_Switch.main_region.Off)
    }
}
```

Expressions

```
// entering / exiting the statechart
enter, exit
// raising an event
raise event : value

// proceeding time or cycles
proceed 2 cycle
proceed 200 ms

// asserting an expression, expression must evaluate to boolean
assert expression
// is a state active
active(someStatechart.someRegion.someState)
```

Mocking Statements

SCTUnit allows to

- mock operations defined in the statechart model
- verify that an operation was called with certain values

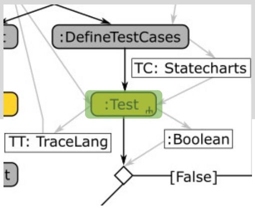
```
// mocking the return value of an operation
mock mockOperation returns (20)
mock mockOperation(5) returns (30)

// verifying the call of an operation
assert called verifyOperation
assert called verifyOperation with (5, 10)
```

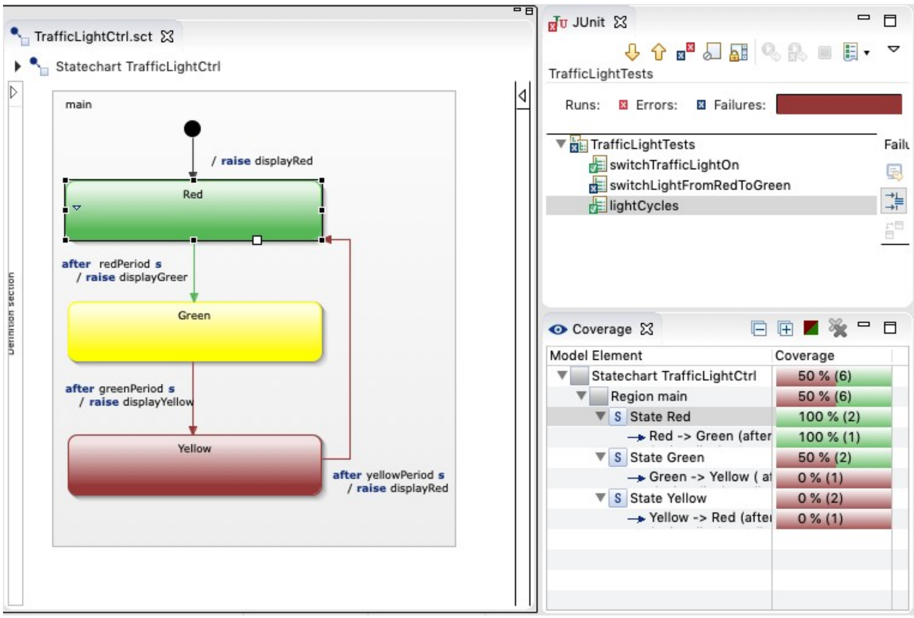
Exercise 3

Testing Models

3

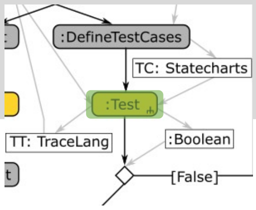
Exercise 3 – Unit testing Statecharts





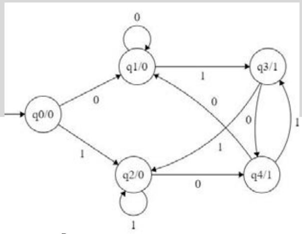
- Create a test that checks the following requirements:
  - R3: at system start-up, the red light is on
  - R4: cycles through red on, green on, and yellow on
  - R5: red is on for 60s, green is on for 55s, yellow is on for 5s

Exercise 3 – Solution

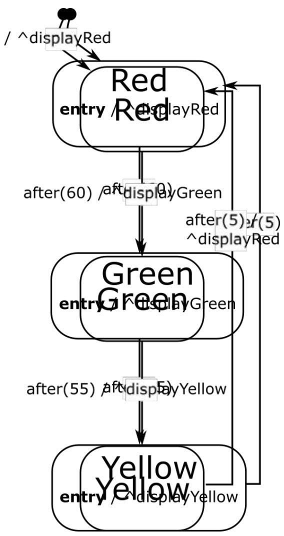


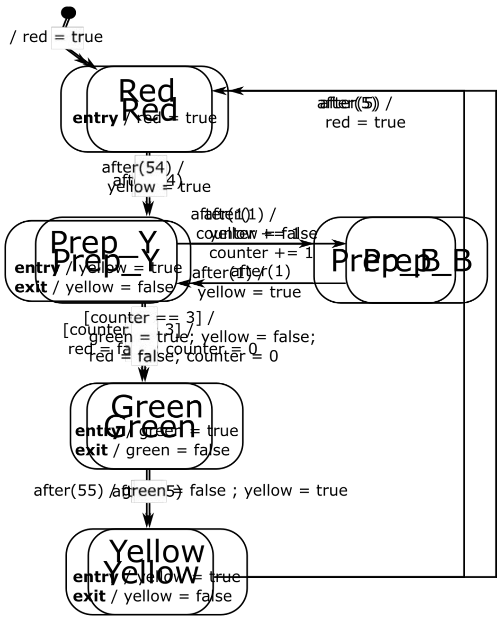

```
package trafficlight.test

testclass TrafficLightTests for statechart TrafficLightCtrl {

    @Test operation switchTrafficLightOn () {

        // given the traffic light is inactive
        assert !is_active
        // when
        enter
        // then traffic light is off which means no color was switched on
        assert displayRed
        assert !displayGreen
        assert !displayYellow
    }


    @Test operation switchLightFromRedToGreen () {

        // given
        switchTrafficLightOn
        // when
        proceed 60s
        // then
        assert displayGreen
    }


    @Test operation switchLightFromGreenToYellow () {

        // given
        switchLightFromRedToGreen
        // when
        proceed 55s
        // then
        assert displayYellow
    }


    @Test operation switchLightFromYellowToRed () {

        // given
        switchLightFromGreenToYellow
        // when
        proceed 5s
        // then
        assert displayRed
    }

    @Test operation lightCycles () {

        // given
        switchLightFromYellowToRed

        var i : integer = 10

        while (i > 0) {
            i=i-1

            //when
            proceed 60 s
            // then
            assert displayGreen

            //when
            proceed 55 s
            // then
            assert displayYellow

            //when
            proceed 5 s
            // then
            assert displayRed
        }
    }
}
```

Entry/Exit Actions

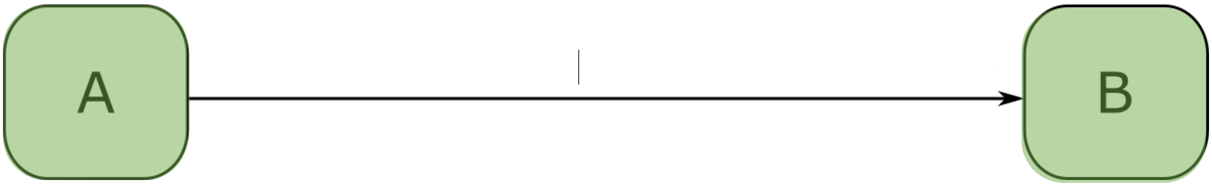


- A state can have entry and exit actions.
- An *entry action* is executed whenever a state is entered (made active).
- An *exit action* is executed whenever a state is exited (made *inactive*).
- Same expressiveness as *transition actions* (*i.e.,* syntactic sugar).





Transitions

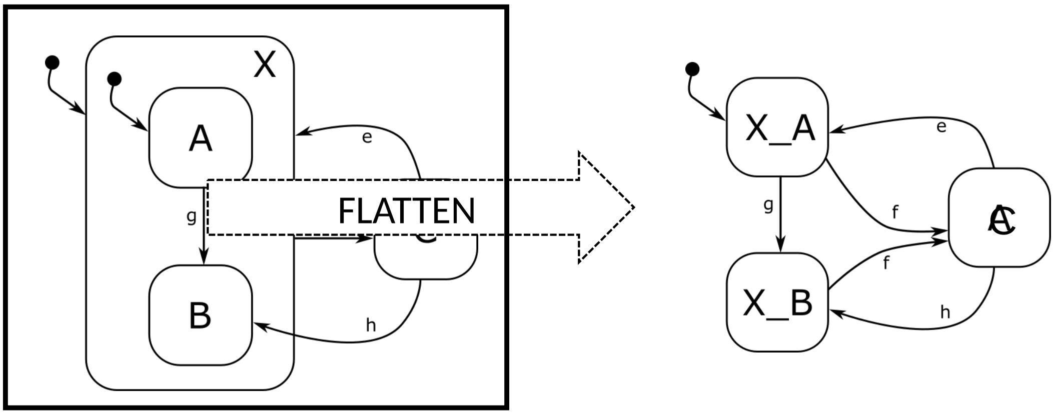


- Model the **dynamics** of the system:
  - *when*
    - the system is **in state A** and
    - **event is received** and
    - **guard** evaluates to **true**
  - *then*
    1. the **exit actions** of state A are evaluated and
    2. **output_action** is evaluated and
    3. the **enter actions** of state B are evaluated and
    4. the new **active state** becomes B

Hierarchy

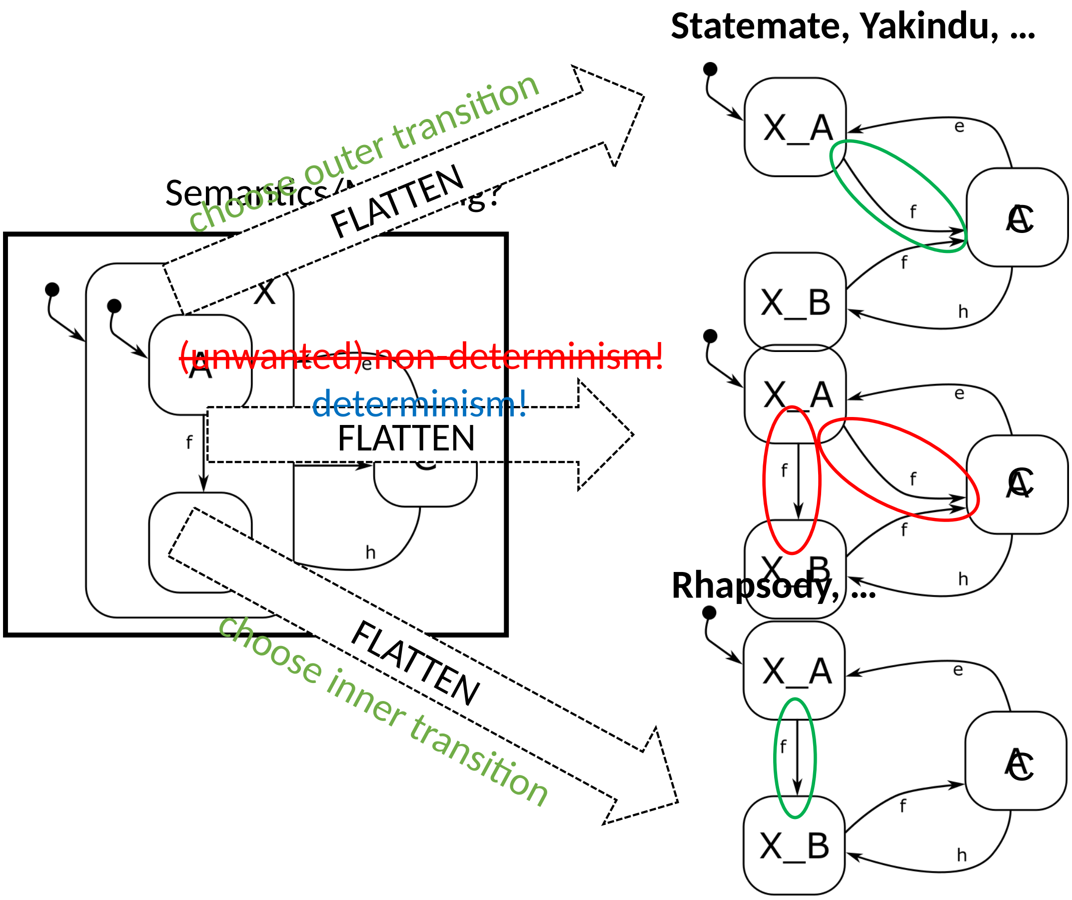
- Statechart states can be hierarchically (de-)**composed**
- Each hierarchical state has exactly one **initial/default state**
- An active hierarchical state has **exactly one active child** (down to leaf/atomic state)

Semantics/Meaning?



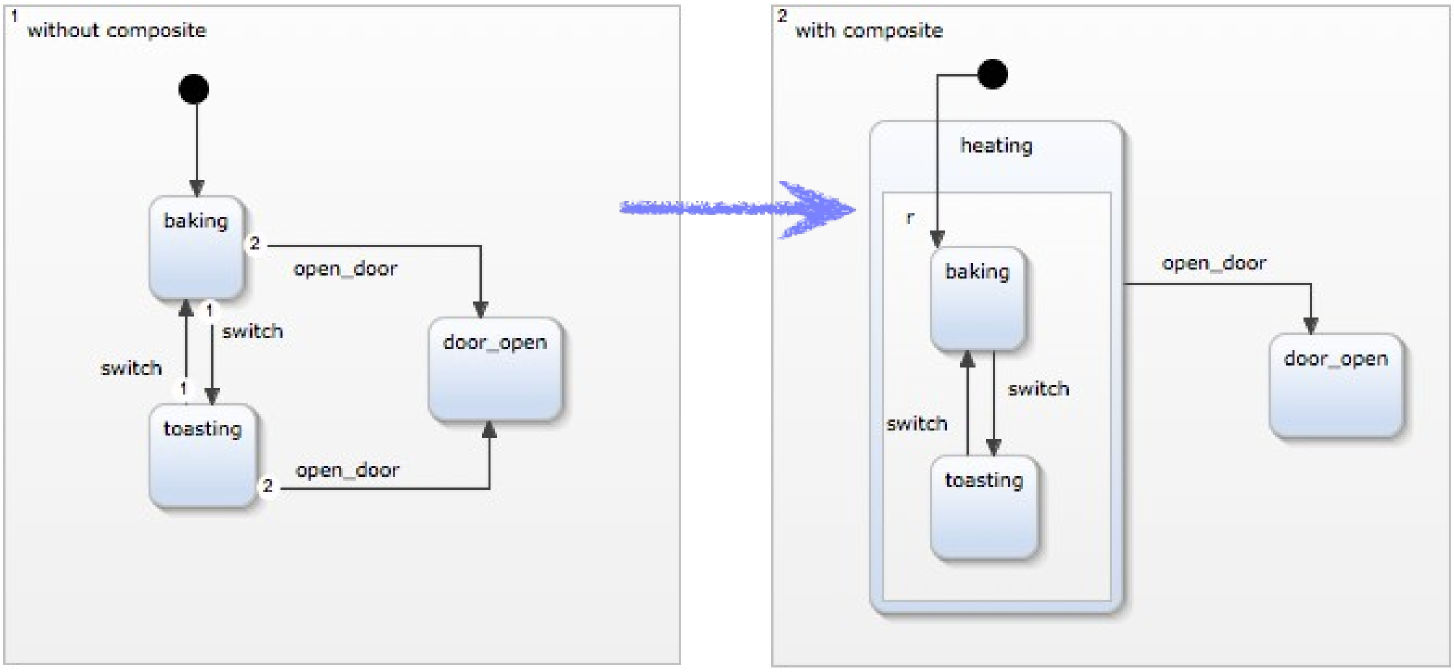

Hierarchy

Hiearchy: Modified Example

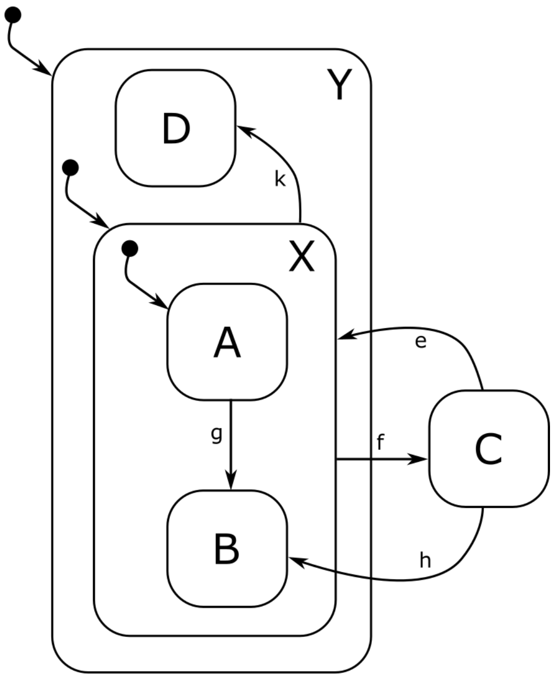


Composite States

- Hierarchical states are an ideal mechanism for hiding complexity
- Parent states can implement common behaviour for their substates
- Hierachical event processing **reduces the number of transitions**
- Refactoring support: group states into a composite state

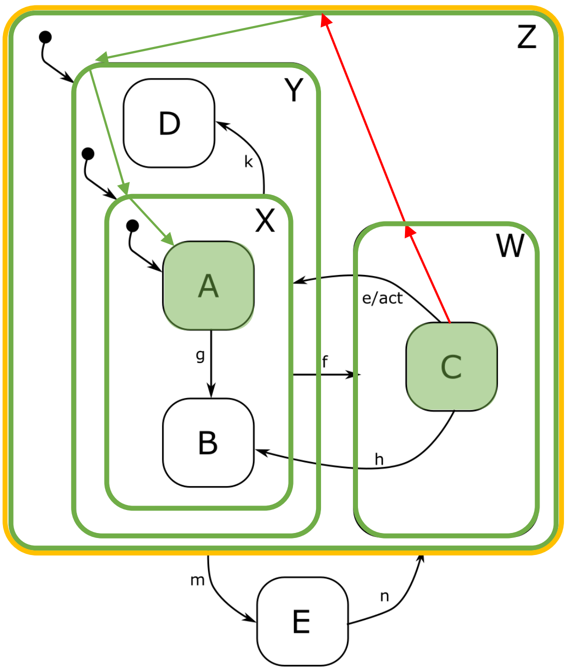


Hierarchy: Initialization



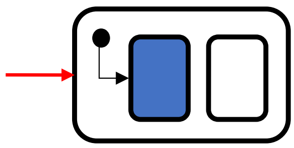
- Concept of **effective target state**
  - Recursive: the effective target state of a composite state is its initial state
- Effective target state of initial transition is *Y/X/A*
- Initialization:
  1. Enter Y, execute enter action
  2. Enter X, execute enter action
  3. Enter A, execute enter action

Hierarchy: Transitions



- Assume *Z/W/C* is active and *e* is processed.
- Semantics:
  1. Find LCA, collect states to leave
  2. Leave states up the hierarchy
  3. Execute action *act*
  4. Find effective target state set, enter states down the hierarchy

Effective target states:



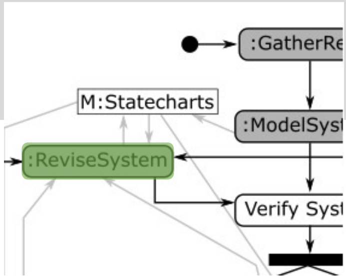
RECURSIVE!

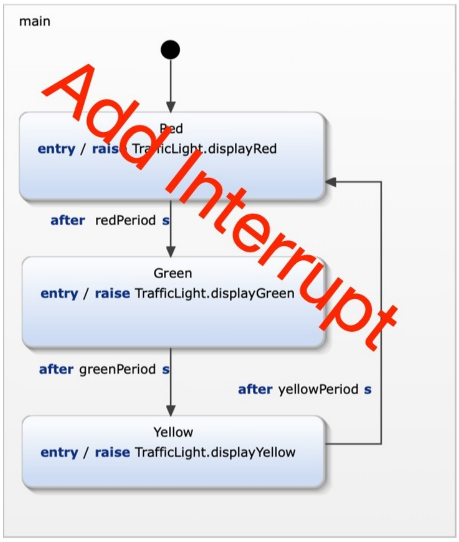
Exercise 5

Model an interruptible traffic light

6

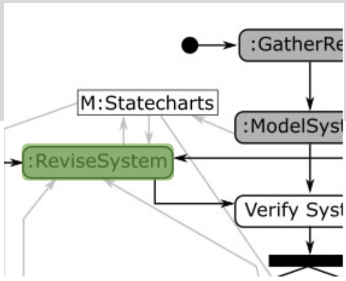
Exercise 5 - Requirements



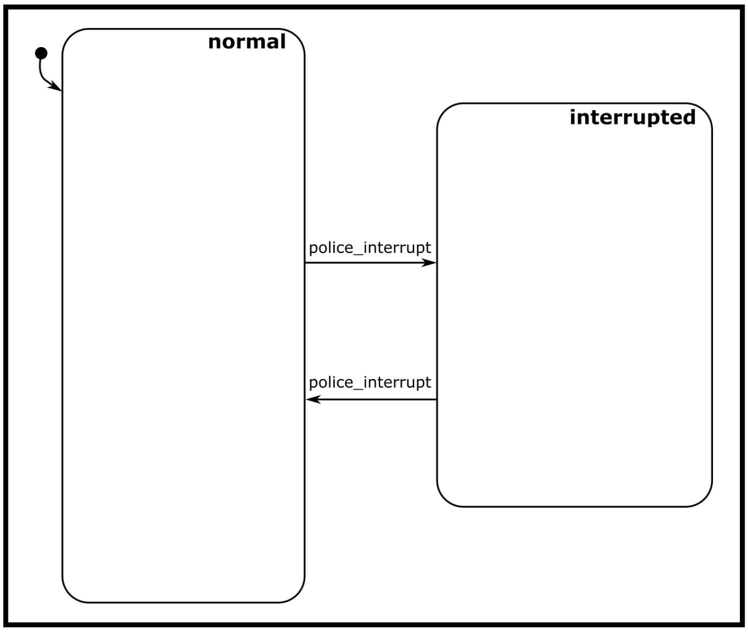


- R7a: police can interrupt autonomous operation .
- R7b: autonomous operation can be interrupted during any phase of constant red, yellow and green lights.
- R7c: in interrupted mode the yellow light blinks with a constant frequency of 1 Hz (on 0.5s, off 0.5s).
- R8a: police can resume to regular autonomous operation.
- R8b: when regular operation is resumed, the traffic light restarts with red (R) light on.

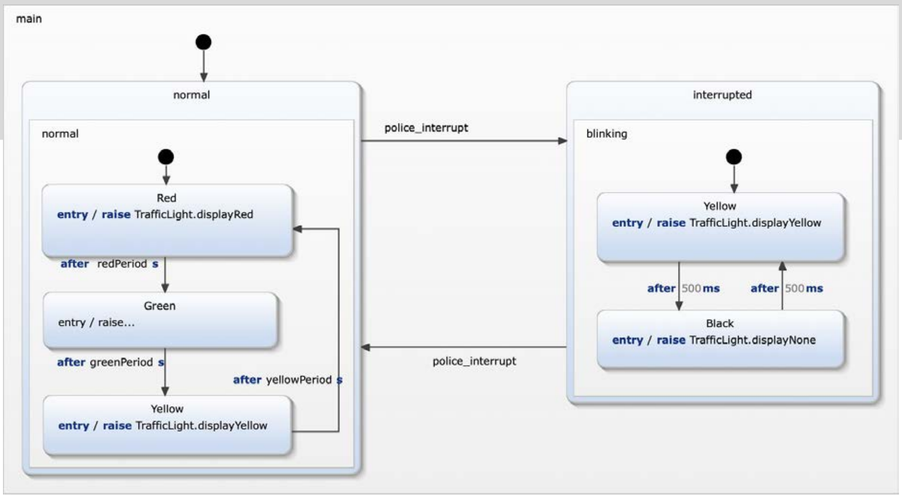
Exercise 5: Solution



- R7a: police can interrupt autonomous operation .
- R7b: autonomous operation can be interrupted during any phase of constant red, yellow and green lights.
- R7c: in interrupted mode the yellow light blinks with a constant frequency of 1 Hz (on 0.5s, off 0.5s).
- R8a: police can resume to regular autonomous operation.
- R8b: when regular operation is resumed, the traffic light restarts with red (R) light on.



Exercise 5 - Solution

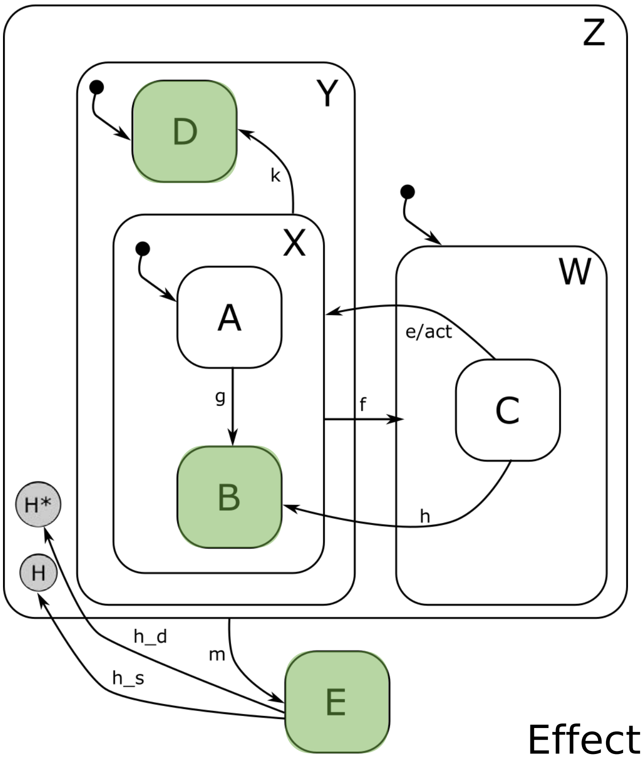


requirement	modelling approach
R6: police can interrupt autonomous operation.	An new incoming event police_interrupt triggers a transition to a new state interrupted.
R6a: autonomous operation can be interrupted during any phase of constant red, yellow and green lights.	The states Red, Green, and Yellow are grouped within a new composite state normal. This state is the source state of the transition to state interrupted and thus also applies to all substates.
R7: in interruptetd mode the yellow light blinks with a constant frequency of 1 Hz. (on 0.5s, off 0.5s).	State interrupted is a composite state with two substates Yellow and Black. These switch the yellow light on and off. Timed transitions between these states ensure correct timing for blinking.
R8: police can resume to regular autonomous operation.	A transition triggered by police_interrupt leads from state interrupted to state normal.
R8a: when regular operation is resumed the traffic light restarts with red (R) light on.	When activating state normal its substate Red is activated by default.


History

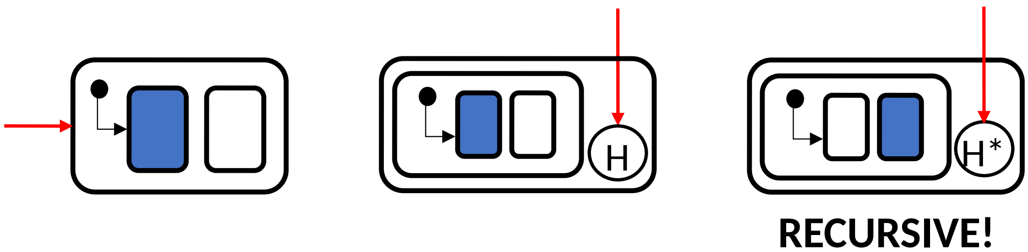
History: pseudo-states

H *shallow* history H* *deep* history



- Assume *Z/Y/X/B* is active, and *m* is processed
  - Effective target state: *E*
- If *h_s* is processed
  - Effective target state: *Z/Y/D*
- If *h_d* is processed
  - Effective target state: *Z/Y/X/B*

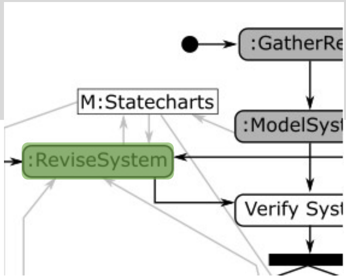
Effective target states:



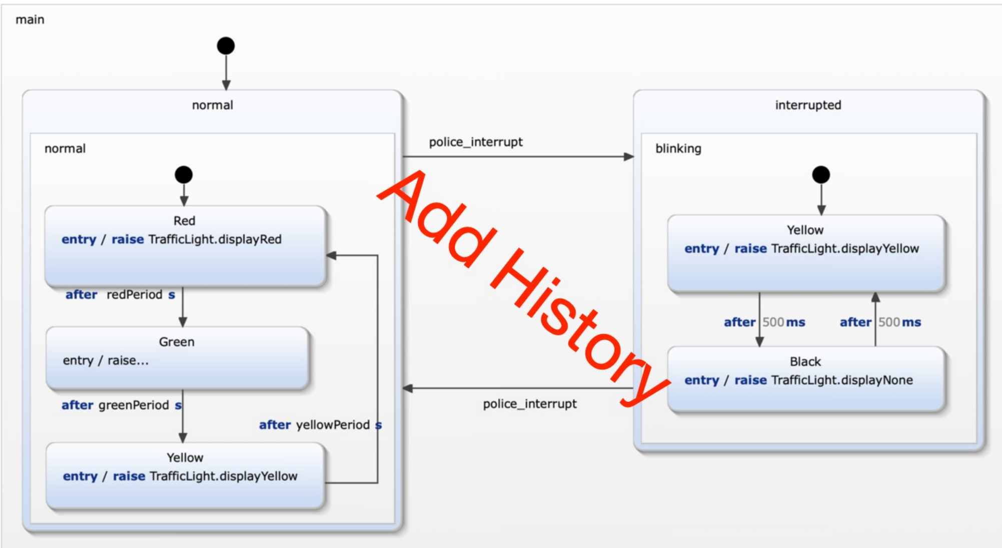
Exercise 6

Model an interruptible traffic light that restores its state

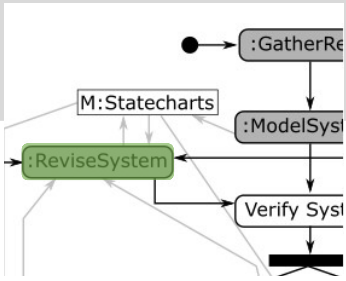
Exercise 6: Requirements



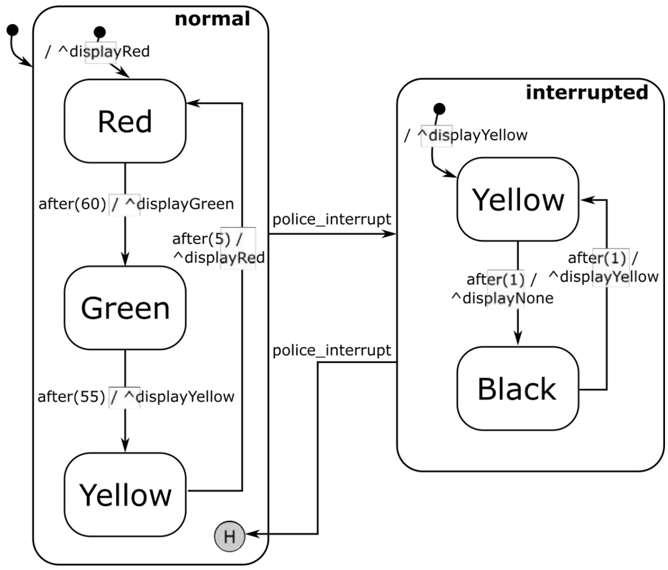
- R8b: when regular operation is resumed the traffic light restarts with the last active light color red (R), green (G), or yellow (Y) on.



Exercise 6: Solution



- R8b: when regular operation is resumed the traffic light restarts with the last active light color red (R), green (G), or yellow (Y) on.

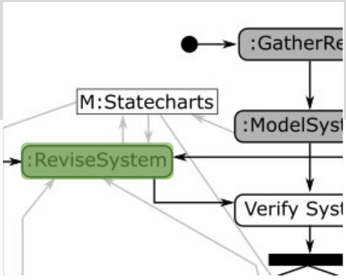


Exercise 7

Model an interruptible traffic light that restores its state and can be switched on/off

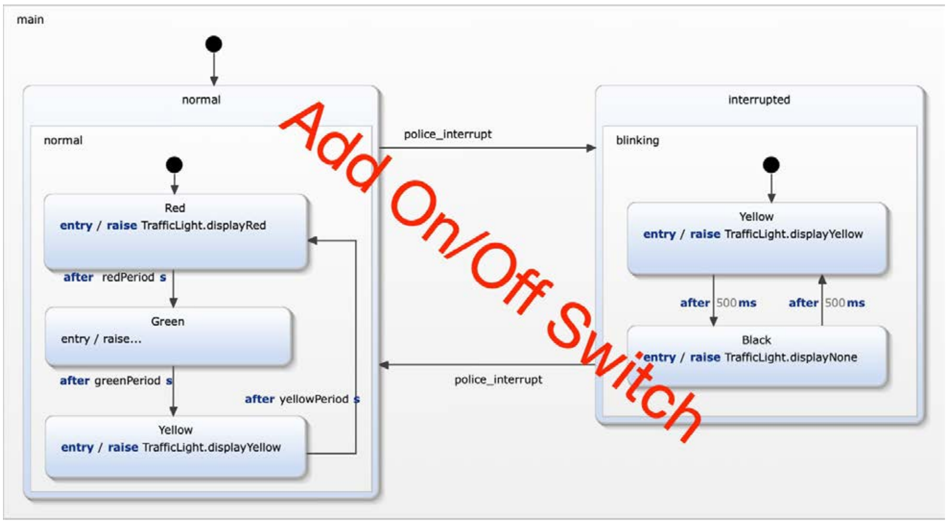
5

Exercise 7: Requirements

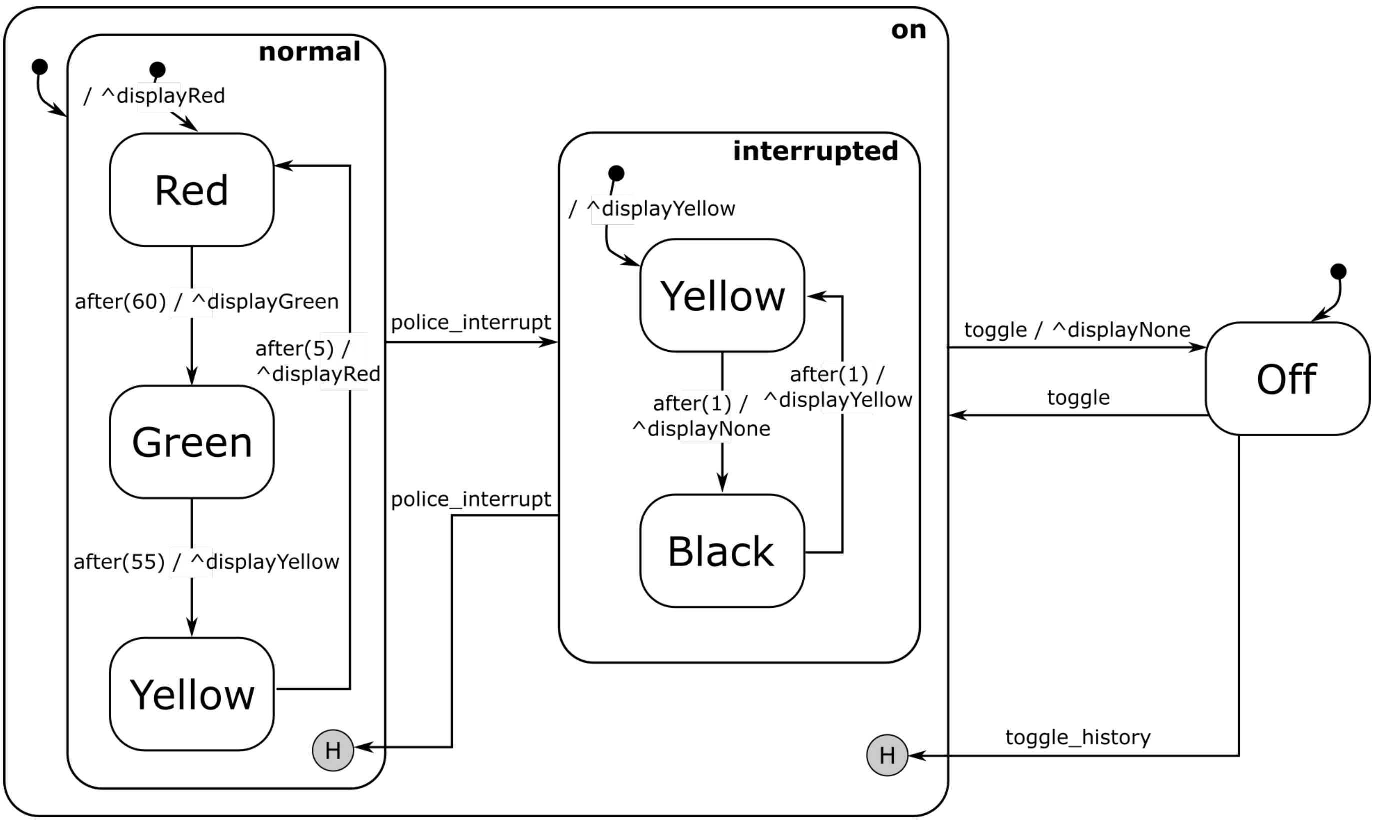


Add another level of hierarchy that supports switching on and off the entire traffic light. Go into detail with shallow and deep histories.

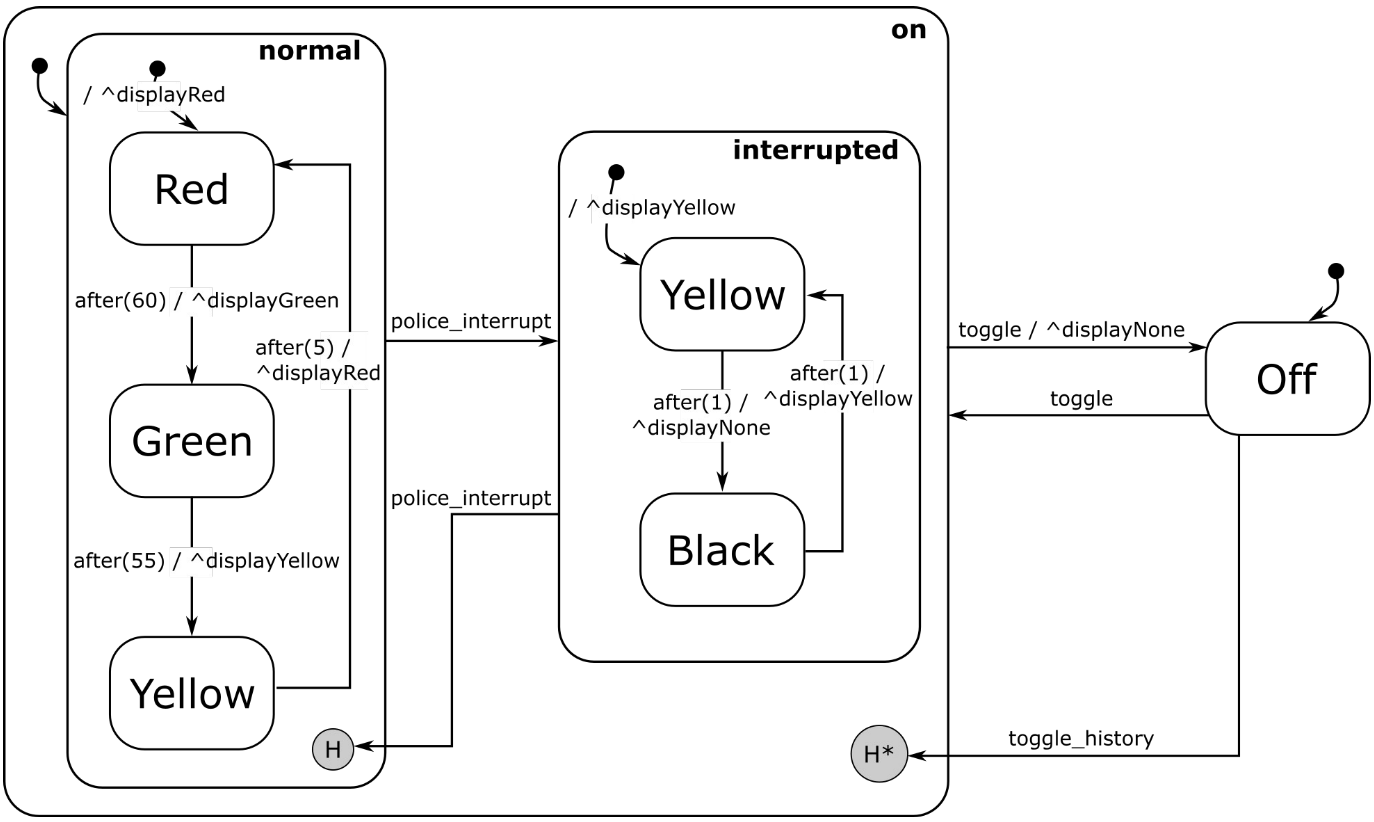
- R9: The traffic light can be switched on and off.
- R9a: The traffic light is initially off.
- R9b: If the traffic light is off none of its lights (R/G/Y) are on.
- R9c: After switching off and on again the traffic light must switch on the light that was on before the switching off.



Exercise 7: Solution



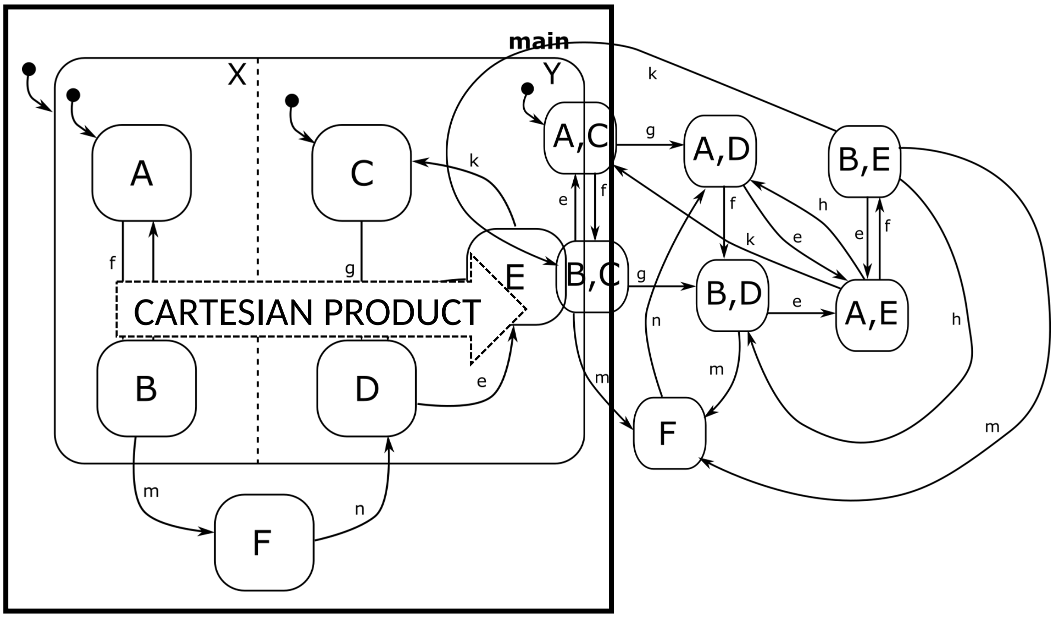
Exercise 7: Alternative Solution



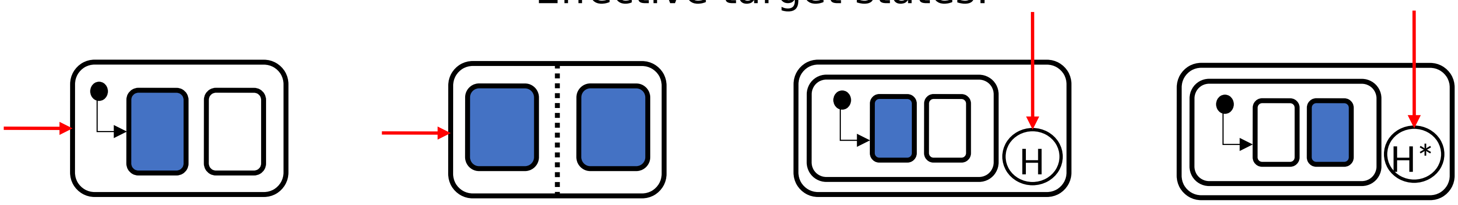

Orthogonality

Orthogonal Components/Regions: "and" states

Semantics/Meaning?

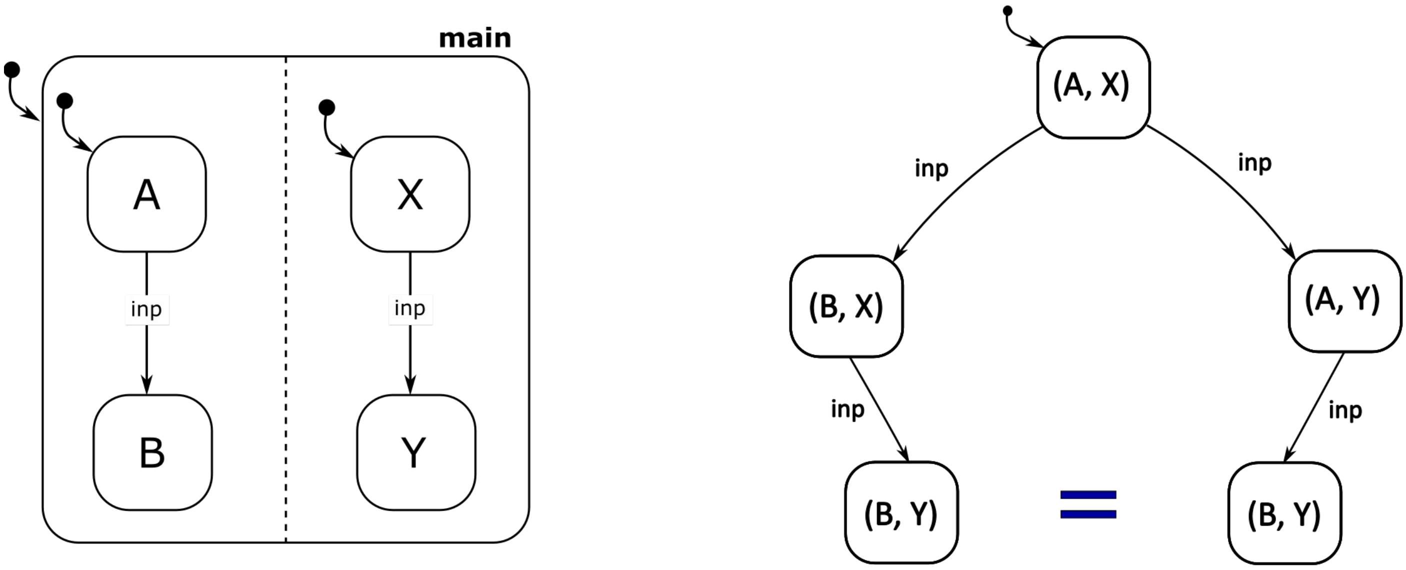


Effective target states:

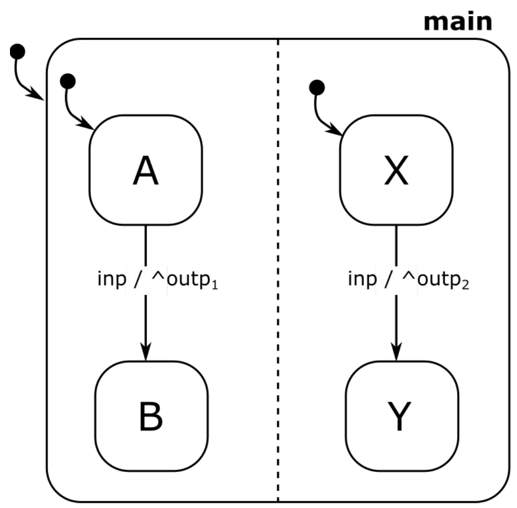


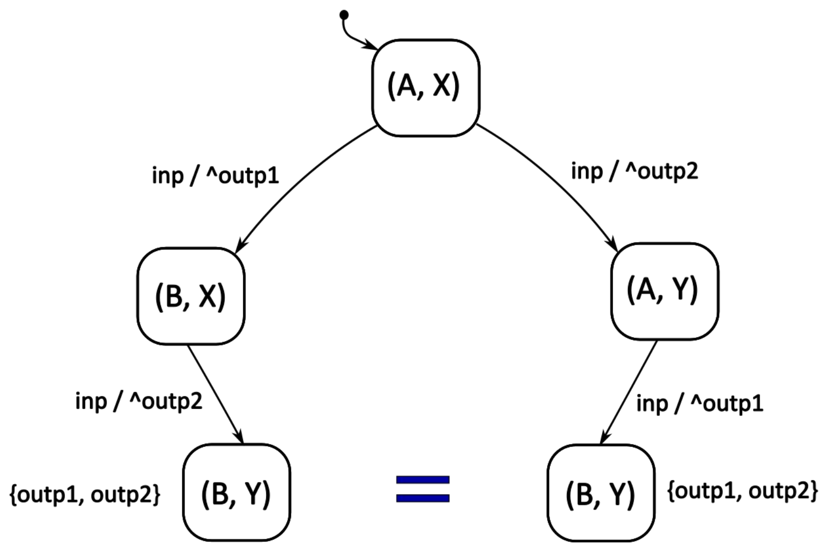
RECURSIVE!

Parallel (In)Dependence

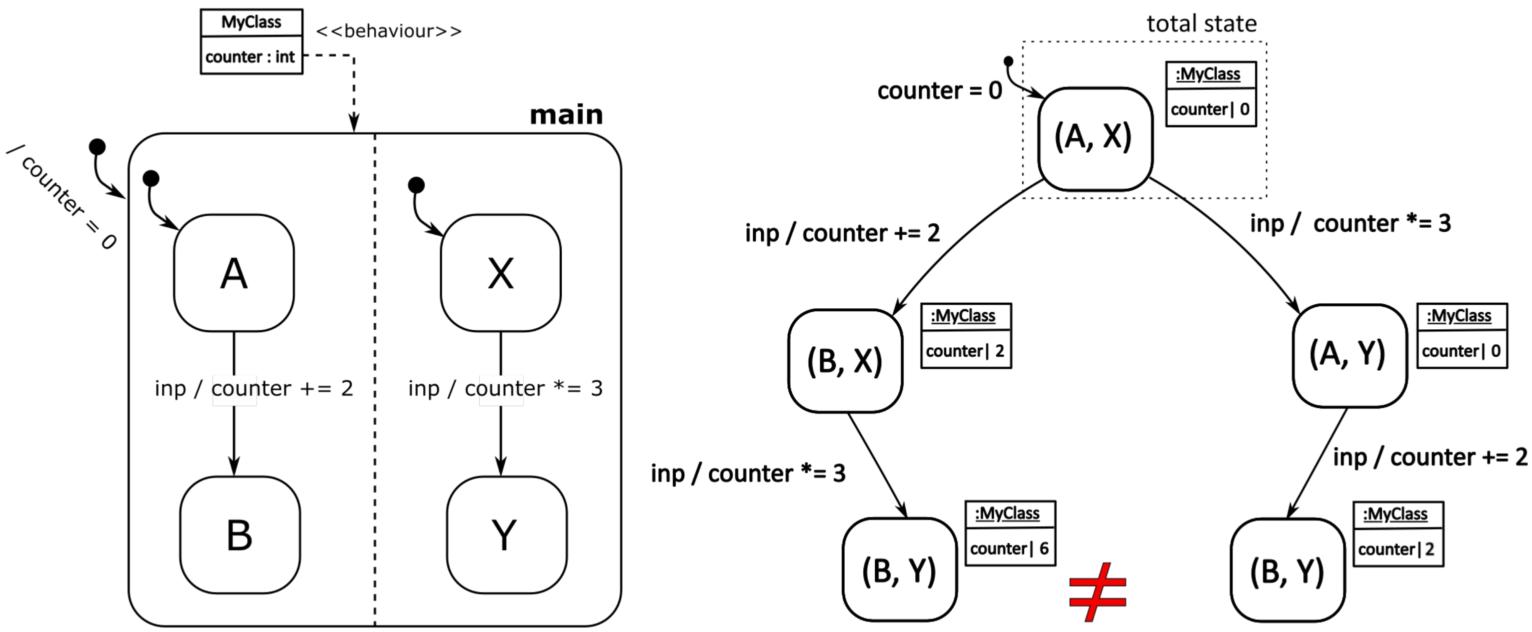


Parallel (In)Dependence

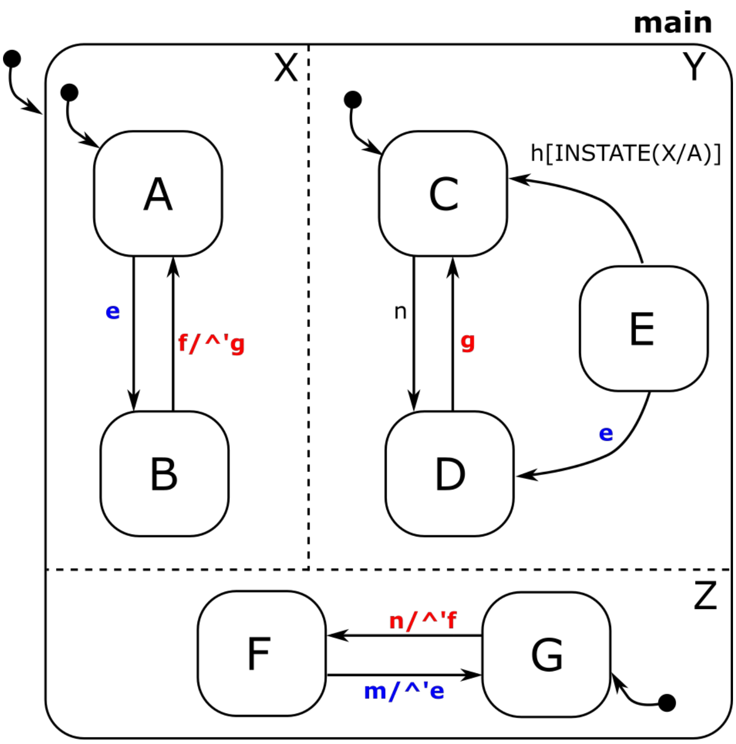




Parallel (In)Dependence



Orthogonality: Communication

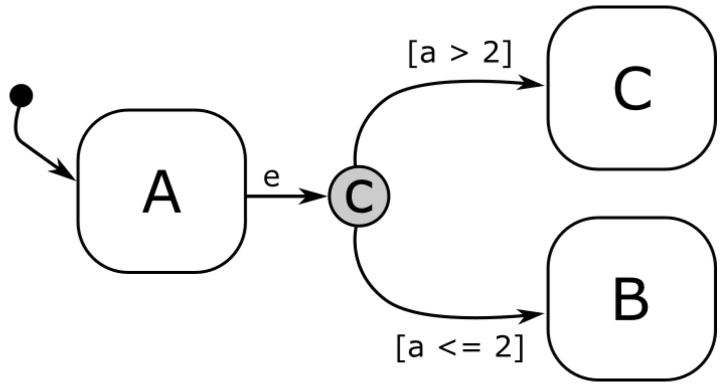


Input Segment: **nmnn**

Orthogonal Components can communicate:

- raising/broadcasting local events:
  ^’<<event name>>
- state reference is a transition guard:
  INSTATE(<<state location>>)

Conditional Transitions

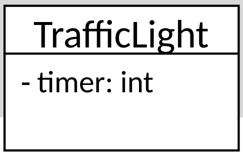


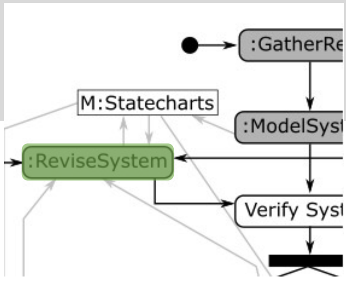
- getEffectiveTargetStates(): select one *True*-branch
- Conditions should not overlap to avoid non-determinism
  - in Yakindu, priority makes deterministic
  - “else” branch is required
- Equivalent (in this case) to two transitions:
  - A – *e[a > 2]* -> C
  - A – *e[a <= 2]* -> B

Exercise 8

Add a timer to the traffic light

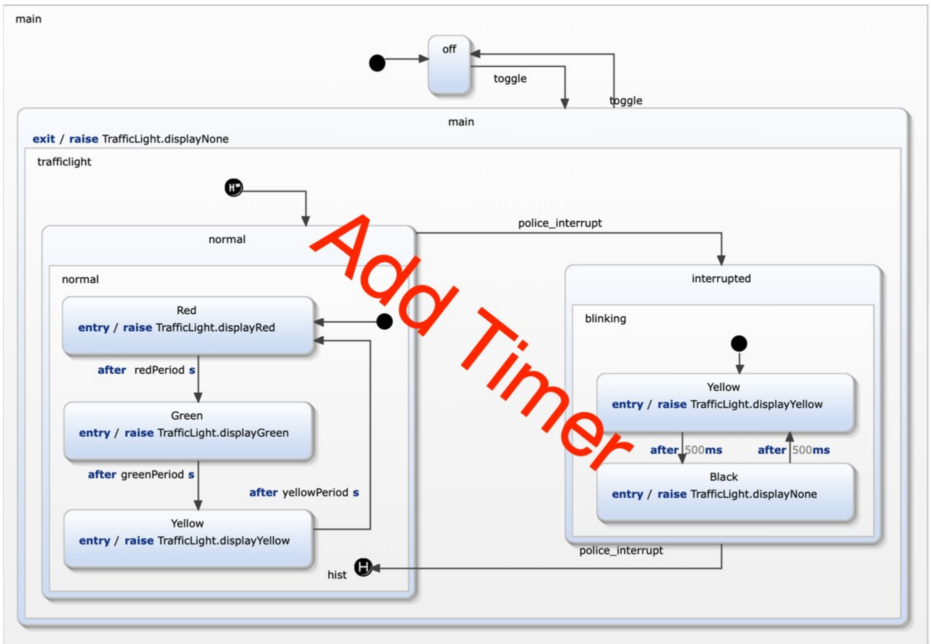
Exercise 8: Requirements



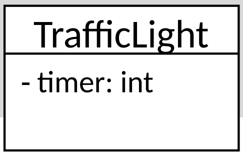


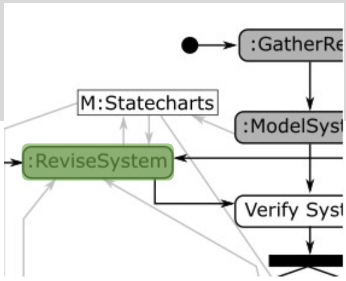
In this exercise a timer must be modelled.
It introduces the use of orthogonal regions.

- R10a: A timer displays the remaining time while the light is red or green.
- R10b: This timer decreases and displays its value every second.
- R10c: The colour of the timer reflects the colour of the traffic light.

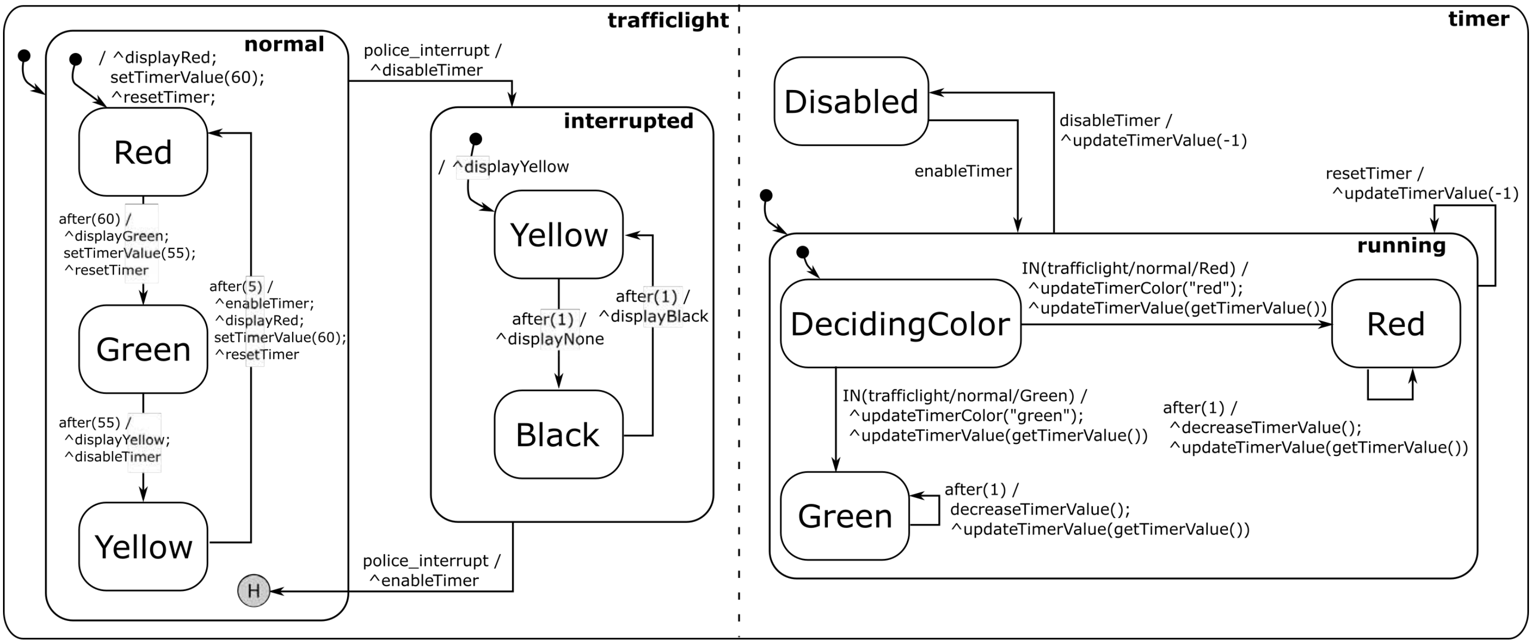


Exercise 8: Solution



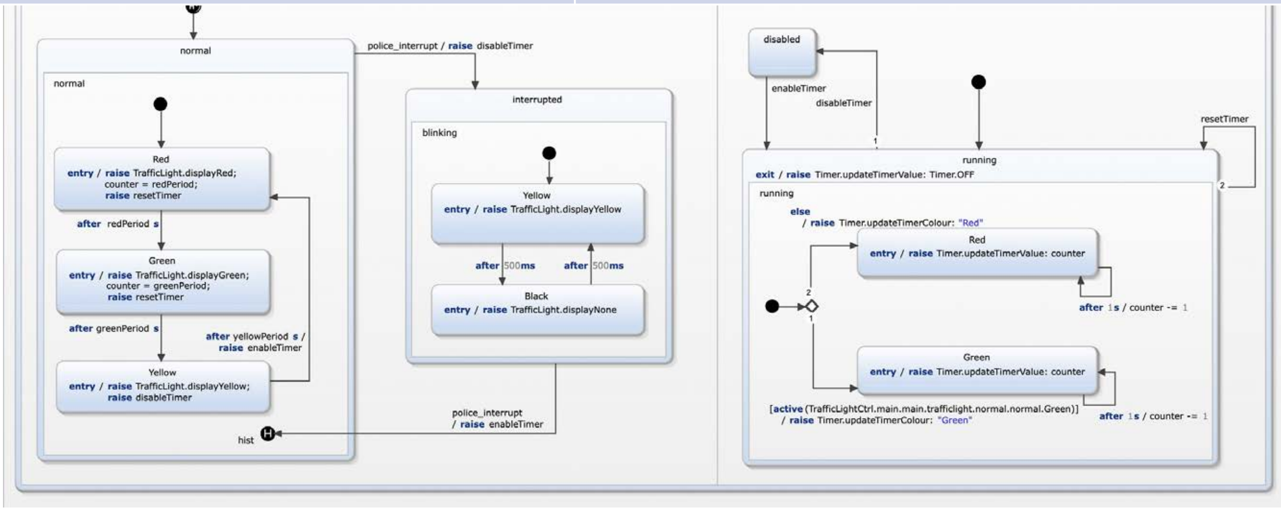


- R10a: A timer displays the remaining time while the light is red or green.
- R10b: This timer decreases and displays its value every second.
- R10c: The colour of the timer reflects the colour of the traffic light.



Solution 8

requirement	**modelling approach**
R10: a timer displays the remaining time while the light is red or green	The timer is defined in a second region within state on (main in the Yakindu model).
R10a: This timer decreases and displays its value every second.	An internal variable for the counter is introduced. When switching the traffic light phase, the counter value is set to how long the light has been in that phase. Additionally, the local events resetTimer, enableTimer, and disableTimer are used to synchronize traffic light phase switches with the timer.
R10b: The colour of the timer reflects the colour of the traffic light.	When the timer is enabled it checks the active traffic light phase using the active() function.



Yakindu syntax

Yakindu:

- **raise** e == ^e
- strict alternation between "or" and "and" states →

  TrafficLightCtrl.**main.main**.trafficlight.**normal.normal**.Green
- **active**() == INSTATE() == IN()

Code Generation

02

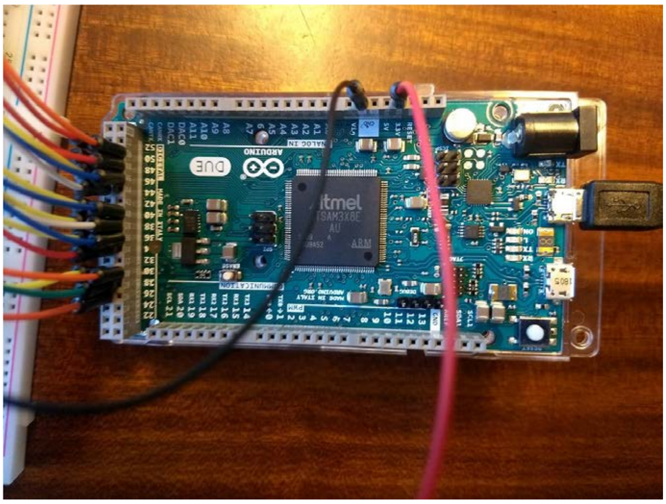
Code Generation

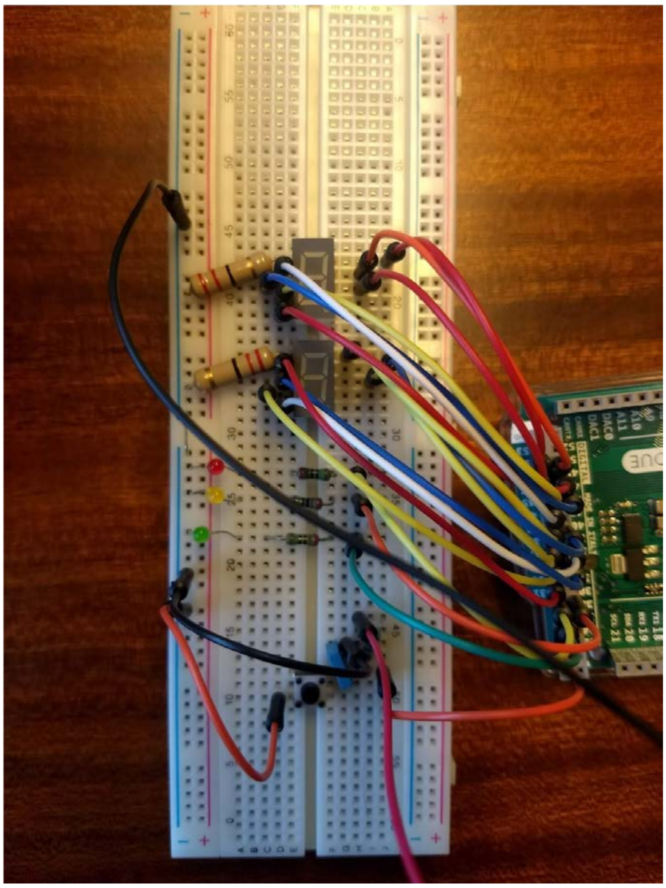
- Code generators for C, C++, Java, Python, Swift, Typescript, SCXML
- Plain-code approach by default
- Very efficient code
- Easy integration of custom generators

Deployed Application (Scaled Real-Time)

Deploying onto Hardware

Interface:
- pinMode(*pin_nr*, *mode)*
- digitalWrite(*pin_nr*, *{0, 1}*)
- digitalRead(*pin_nr*): {0, 1}





Deploying onto Hardware

Runner

```
#define CYCLE_PERIOD (10)
static unsigned long cycle_count = 0L;
static unsigned long last_cycle_time = 0L;

void loop() {
  unsigned lon
  read_pushbut
  if ( cycle_c                                             _PERIOD) ) {
    sc_timer_s                                             e_time);
    synchroniz
    trafficLig
    last_cycle
    cycle_coun
  }
}
```

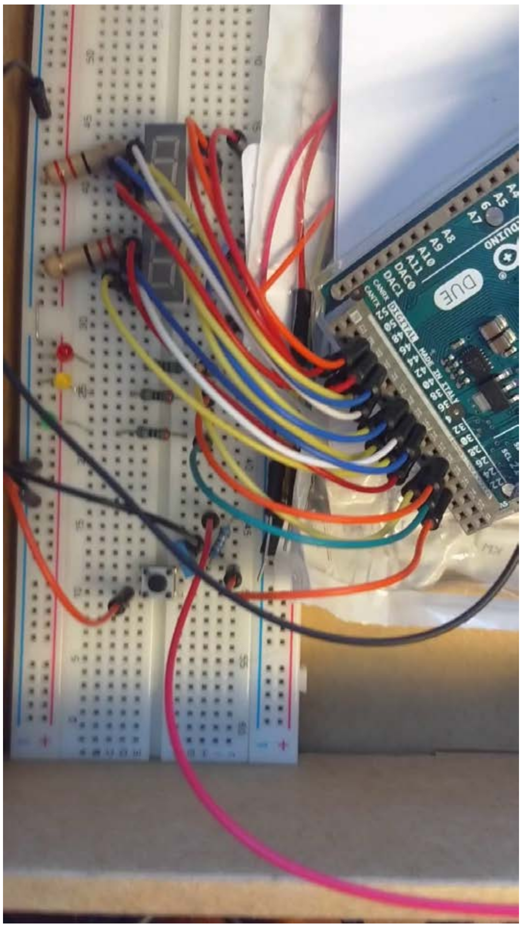
Generator

```
GeneratorModel for yakindu::c {

    statechart TrafficLightCtrl {

        feature Outlet {
            targetProject = "traffic_light_arduino"
            targetFolder = "src-gen"
            libraryTargetFolder = "src-gen"
        }

        feature FunctionInlining {
            inlineReactions = true
            inlineEntryActions = true
            inlineExitActions = true
            inlineEnterSequences = true
            inlineExitSequences = true
            inlineChoices = true
            inlineEnterRegion = true
            inlineExitRegion = true
            inlineEntries = true
        }
    }
}
```

Button Co

```
void read_pushb
  int pin_value
  if (pin_value
    button->las
  }
  if ((millis()                                              ay) {
    if (pin_val
      button->s
      button->c
    }
  }
  button->debounce_state = pin_value;
}
```

Deployed Application



TELECOM Paris

IP PARIS

Summer School on CPS & IoT 2025

Modeling, Analyzing and Synthesizing Embedded Systems with AADL using RAMSES

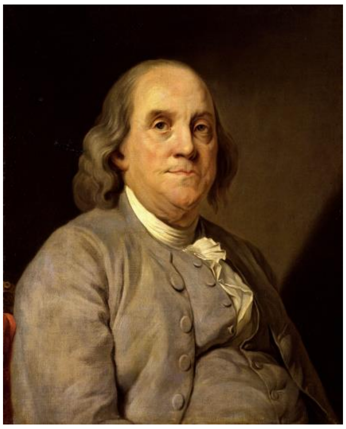
Dominique Blouin, Associate Professor
Télécom Paris, Institut Polytechnique de Paris
dominique.blouin@telecom-paris.fr

Content

- Introduction to AADL
- Modeling Software Applications
- Modeling Execution Platforms
- Organization of Declarations
- Introduction to OSATE and AADL Inspector
- Timing Analysis with AADL Inspector
- Model Refinement and Code Synthesis with RAMSES

The Importance of Architectures

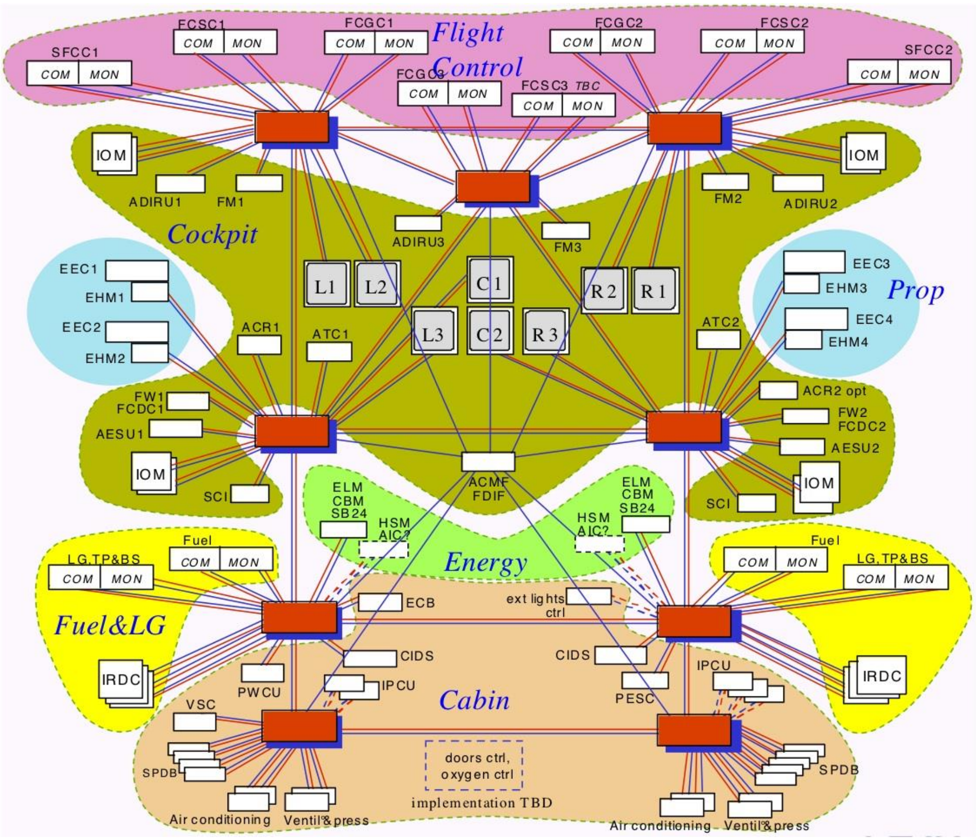
- If you fail to plan, you are planning to fail!



Painting by Duplessis.
Source: Wikipedia

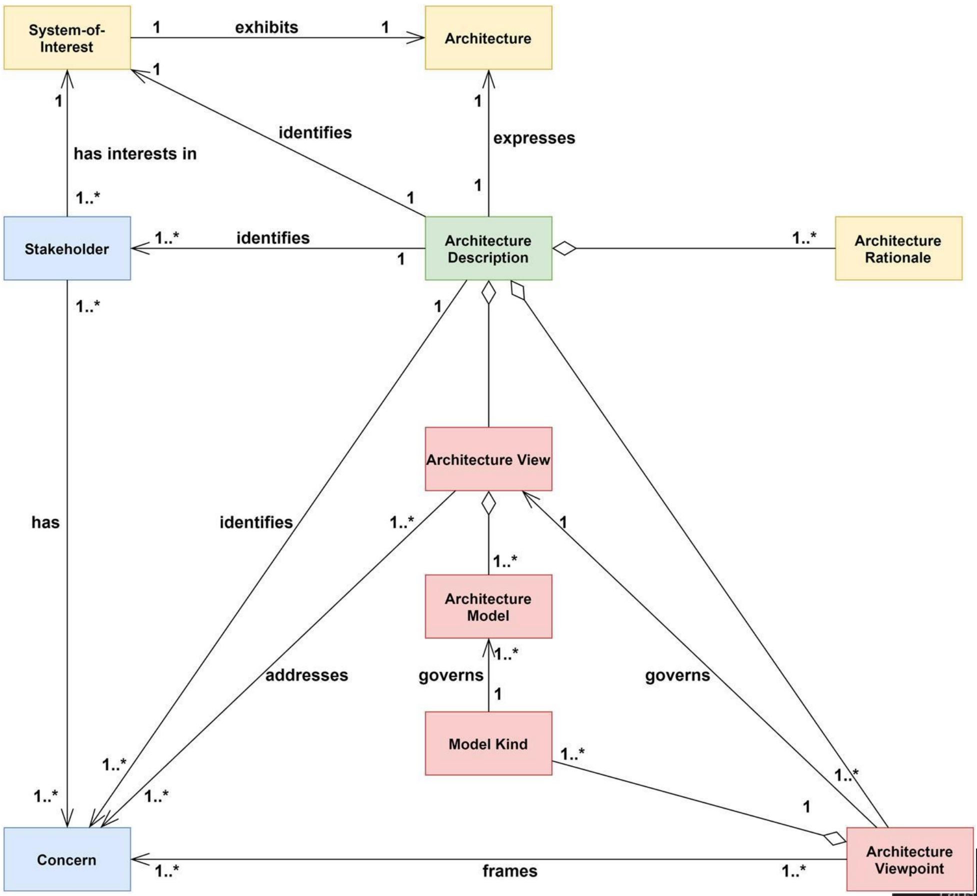
- Architectures are not only useful for buildings, but for complex (software) systems too!

Example of a Complex Avionics Architecture



Architecture Description Languages (ADL)

- **ISO/IEC/IEEE 42010** *Systems and software engineering— Architecture description*
- Define an ADL as "any form of expression for use in architecture descriptions".
- Specifies minimum **requirements** on ADLs.



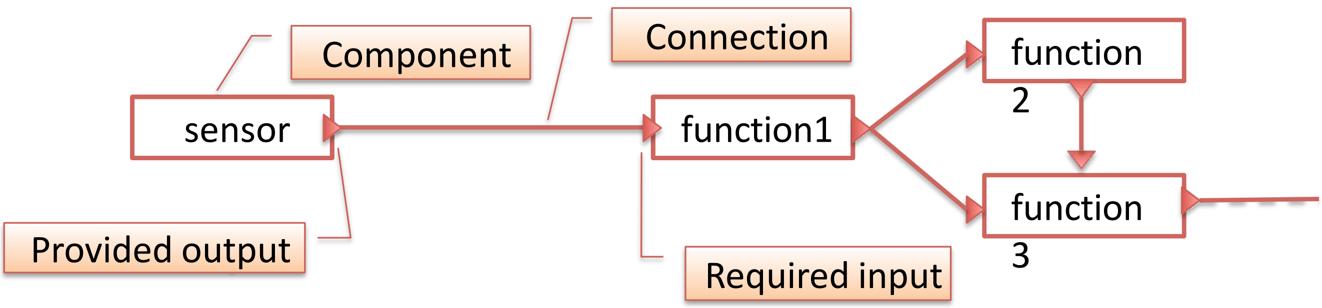
Architecture Description Languages (ADL)

- Languages used to express the **structural** and **behavioral** aspects of systems.
  - Including properties, elements, components, and their relationships.
- Enable stakeholders to communicate and collaborate in the design and management of systems.
- A plethora of ADLs exist.
- Some are **standardized** (e.g., UML, UML MARTE, AUTOSAR, AADL).
  - Provide a common understanding of the notation.
  - But the cost is **slow evolutions** through a committee…

Components-Based ADLs

- Architecture models represent the **organization** of a **computer system** as a set of **components** and their **interactions**.

- Main artefacts: Boxes and arrows
  - Components: Main elements of the design
  - Interfaces: What components offer and what they need
  - Connections: Satisfy components needs



- Then drawing becomes programming… or at least designing…
  - Nothing new conceptually…

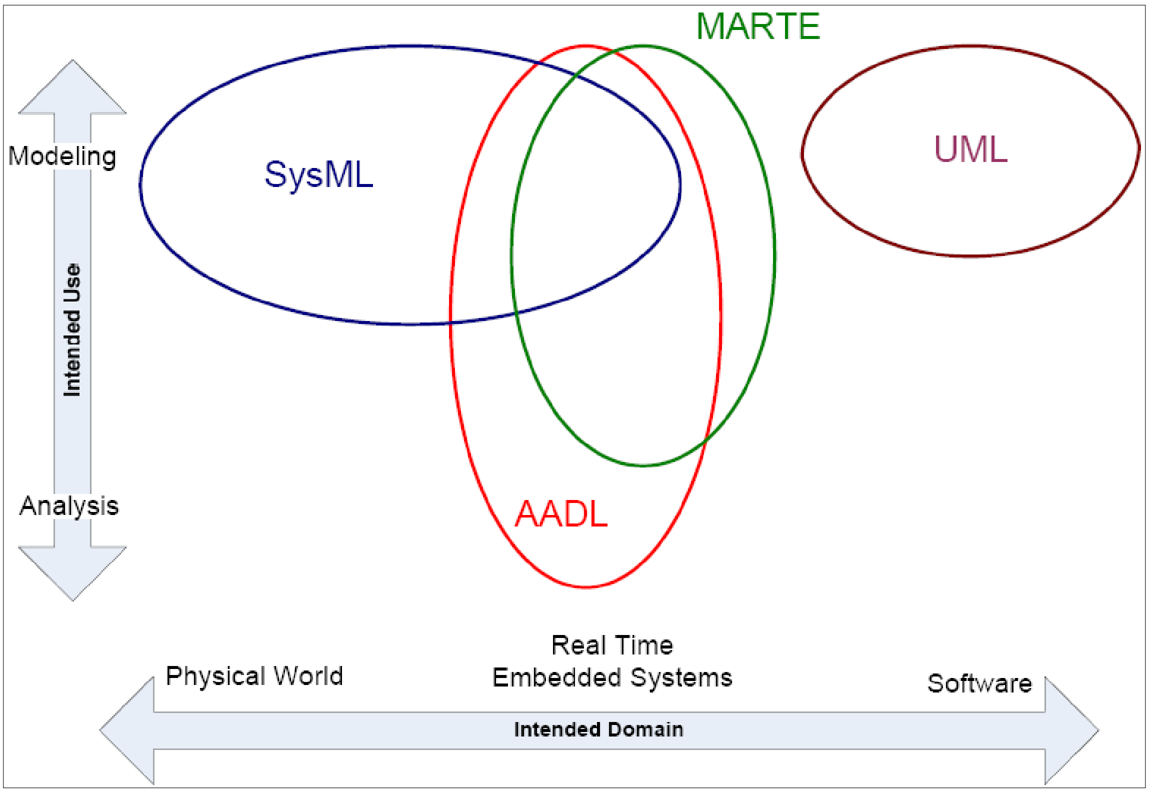
- What about the **semantics**?

AADL: Architecture Analysis & Design Language

- An **ADL** for real-time embedded systems.
  - Component-based (components, interfaces and connections).
  - Defines properties for real-time and embedded systems analysis.
    - Scheduling policy, compute execution time, latency…
    - Software components to hardware components allocation.
  - SAE Standard AS5506
    - https://www.sae.org/standards/content/as5506d/

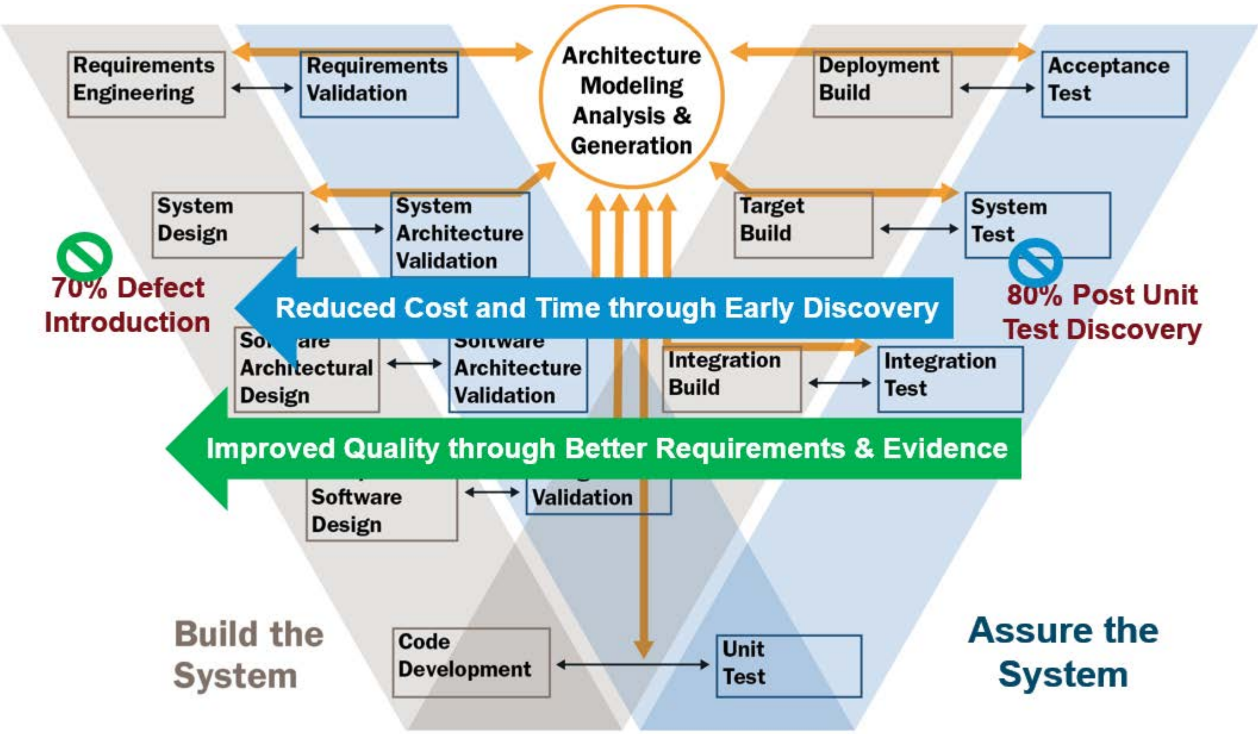
- **Objective**: Support the design of such systems.
  - Standardized semantics (formulated with natural language).
  - Textual and graphical syntax (blended syntax).
  - Strongly typed (components category, composition rules, …).
  - Extensible (property definition language and annexes).

Comparison with other Architecture Description Languages (ADL)



Source: Steven P. Miller, AADL Standards Winter Meeting, 2011

Architecture-Centric Virtual Integration Process (ACVIP)



Source: J. McGregor, P. Gluch and P. Feiler, "Analysis and Design of Safety-critical, Cyber-physical Systems", 2017.

General Characteristics of AADL

- **Components** are the main modeling entities.

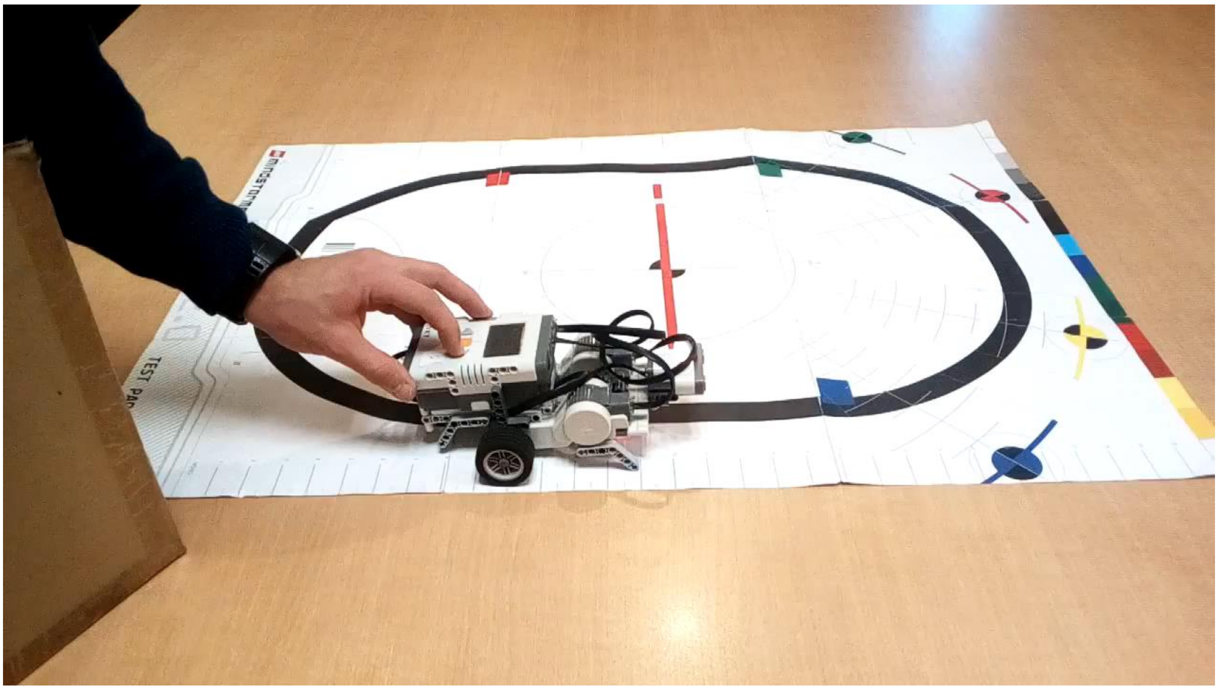
- The standard defines **categories** of components (keywords of the language), e.g.:
  - Composite: System, Abstract
  - Software: Process, Thread, Data, Subprogram…
  - Hardware: Processor, Memory, Bus, Device…

- Components definition is divided into **types**, **implementations**, and **subcomponents:**
  - Type: Defines how the component is viewed from outside (e.g., interaction interfaces)
  - Implementation: Defines the internal structure of the component (e.g., subcomponents)
  - Subcomponents: Instances of components, starting from a root system implementation.

- Components are structured into **sections** identified by keywords of the language (e.g., **features**, **properties**, **subcomponents**).
  - Components can be declared in any order.
  - The language is case insensitive.

Running Example (for the lecture only): Line-Follower Robot with Obstacle Detection

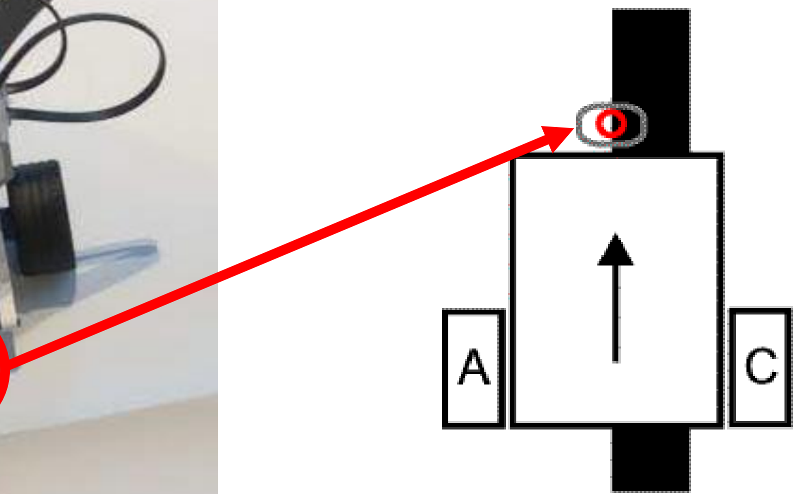
- Purpose: **Follow a line** to carry an object from point A to point B.
  - **Stop** when there is an obstacle (e.g., another robot), and **restart** when the obstacle is no longer there.



How it works…

- **Sensors**: Light and sonar sensors.
- **Actuators**: Two wheels motors.
- **Brick** (includes the execution platform)

- **Control software**
  - PID controller…



AADL System Level Component Categories

- Two categories: **System** and **Abstract**.

System | **Abstract**

- Different purposes:
  - Represent from a very abstract viewpoint the main constituent of a system, its interfaces and connections.
  - System:
    - Aggregates by composition subcomponents describing the execution platform and subcomponents describing the software architecture.
    - Define the main operational modes of the system
  - Abstract:
    - Define structure and interaction without knowing yet the nature of the component. E.g., system functions…

- **What category should we use for the overall robot system?**

Content

- **Introduction to AADL**
- **Modeling Software Applications**
- **Modeling Execution Platforms**
- **Organization of Declarations**
- **Introduction to OSATE and AADL Inspector**
- **Timing Analysis with AADL Inspector**
- **Model Refinement and Code Synthesis with RAMSES**

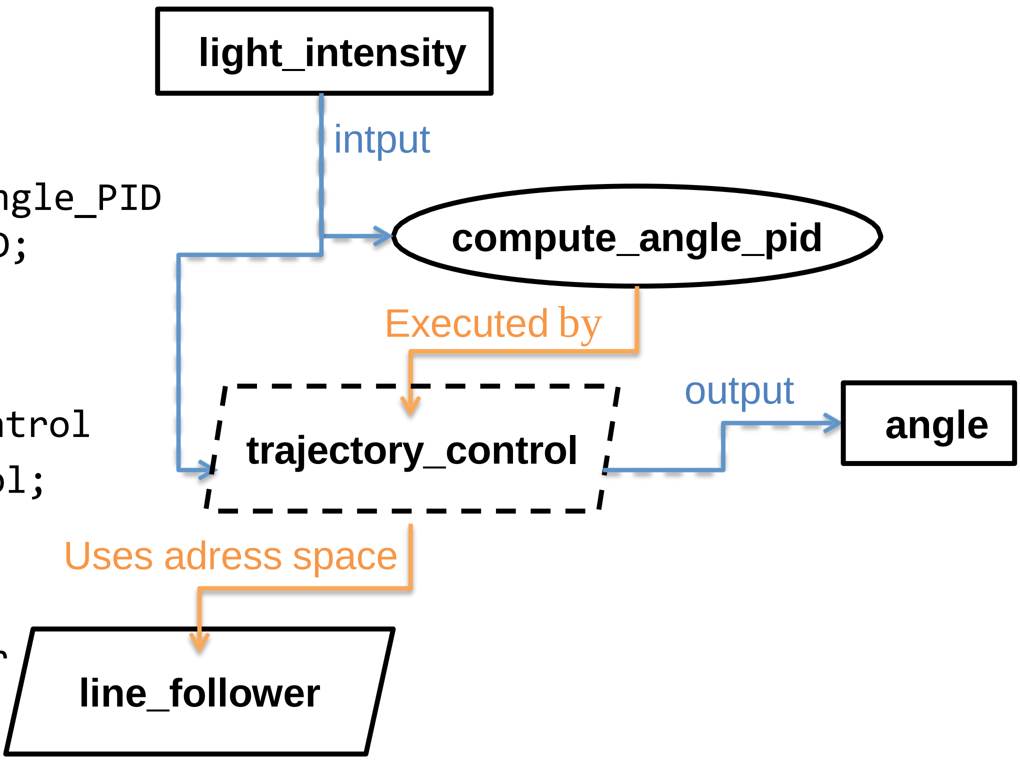
Software Component Categories

- AADL component categories for **software**:
  - Data: Information that can be exchanged among software components.
  - Subprogram: Sequentially executable software, like functions in C programming language.
  - Thread: Task (schedulable unit) executing a sequence of subprograms.
  - Process: Memory address space (software program) allocated for the execution of its thread subcomponents.
  - Etc.

- These categories focus on **operating system** and **programming** components.

Example of Software Components Types

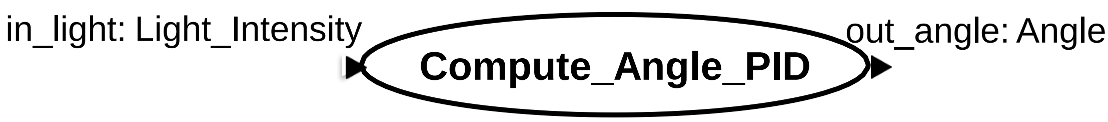
```
data Light_Intensity
end Light_Intensity;

subprogram Compute_Angle_PID
end Compute_Angle_PID;

thread Trajectory_Control
end Trajectory_Control;

process Line_Follower
end Line_Follower;
```



How to represent these interactions and allocations in AADL?

First, define component interfaces Features

- Parameters:
  - **in**, **out**, or **inout**
  - Usable for subprograms only.



- Requires or provides data access:
  - Usable for subprograms and threads.

- Ports:
  - **in**, **out**, or **inout**
  - **data**, **event** or **event data**
  - Usable for threads, processes and systems.

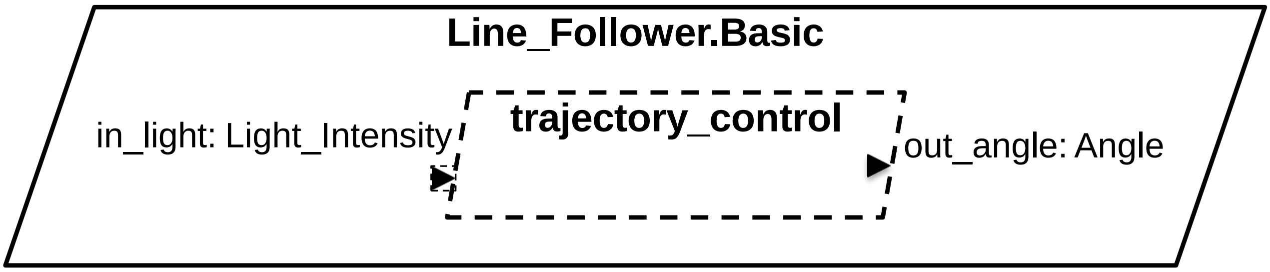
Semantics of Software Components Features

- **Data Port** versus **Event Data Port**:
  - Data Port: Single value shared among components (no queueing).
  - Event or Event Data Port: Multiple values **queued**.

- **Data Port** versus **Data Access**:
  - Data Access allows access to the data at anytime during the execution of a task / subprogram.
  - Data Port defines the following semantics:
    - Data becomes **available** on an **input port** when the thread **starts its execution**. Data not used during the previous execution of the thread is lost. Data is **not updated during the execution** of the task.
    - Data produced on an **output port** is sent to the recipient port at the **end** of the producer task execution.

Composing Components

- Create thread subcomponent(s) in a process implementation:
  - Graphical syntax:



  - Textual syntax:

```
process implementation Line_Follower.Basic
    subcomponents
        trajectory_control: thread Trajectory_Control.Basic;
end Line_Follower.Basic;
```

Defining Subprogram Calls in Thread Implementations

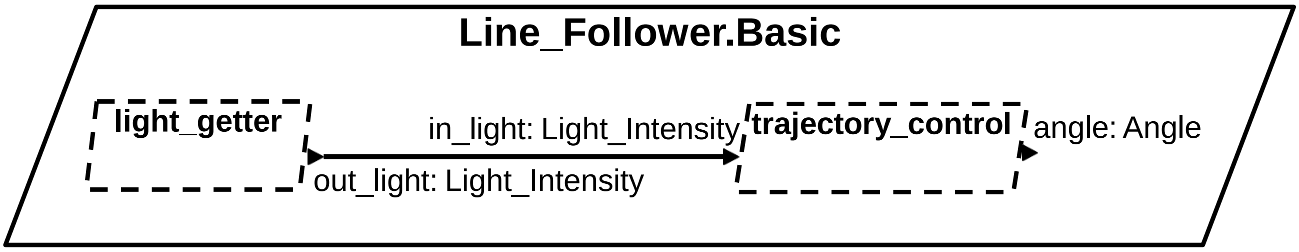
- Subprogram calls in threads:

```
thread implementation Trajectory_Control.Basic
    calls
        call_sequences {
            compute_angle: subprogram Compute_Angle;
            compute_power: subprogram Compute_Power;
        };
end Trajectory_Control.Basic;
```

Sequence of subprogram calls

Connecting Components

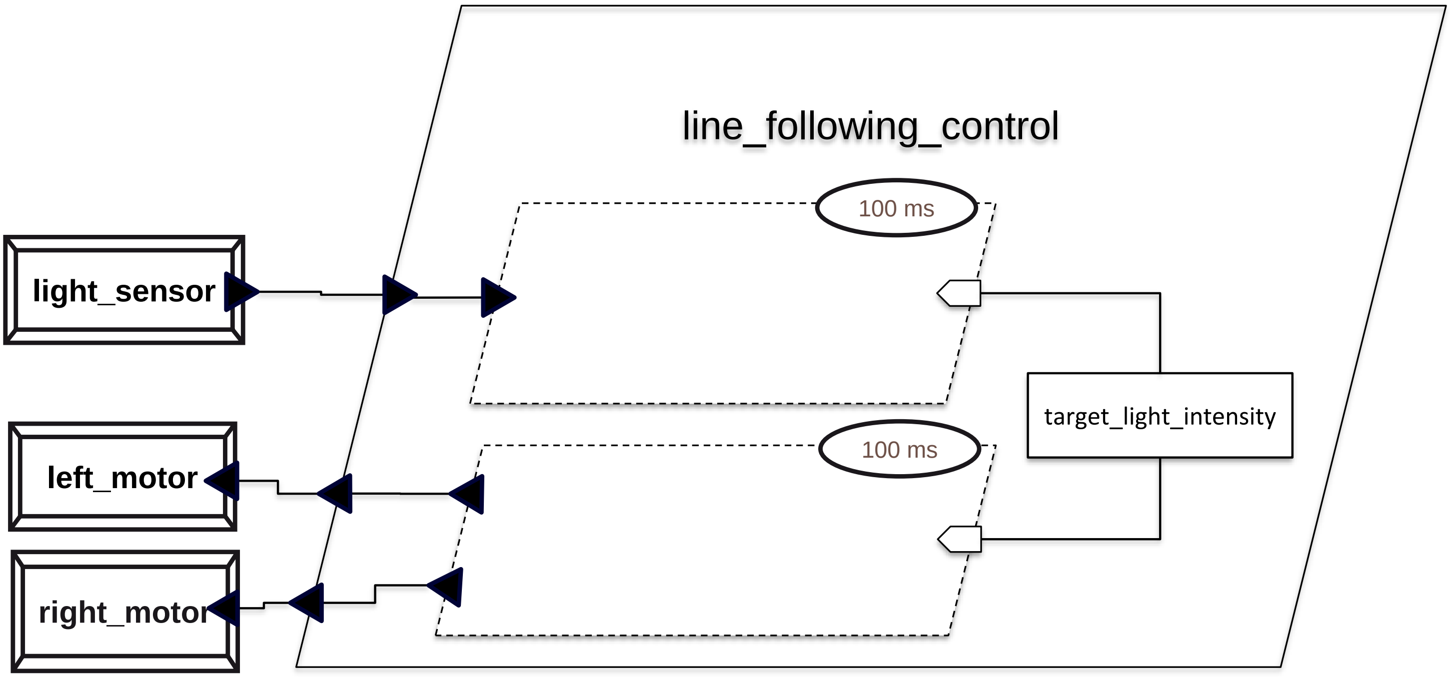
- Components features are connected **hierarchically**:
  - Thread subcomponents located inside a process.
- Graphical syntax:



- Textual syntax:

```
process implementation Line_Follower.Basic
    subcomponents
        trajectory_control: thread Trajectory_Control.Basic;
        light_getter: thread Light_getter.Basic
    connections
        light_intensity_con: port light_getter.out_light -> trajectory_control.in_light;
end Line_Follower.Basic;
```

Sensors and Actuators as Software Device Drivers

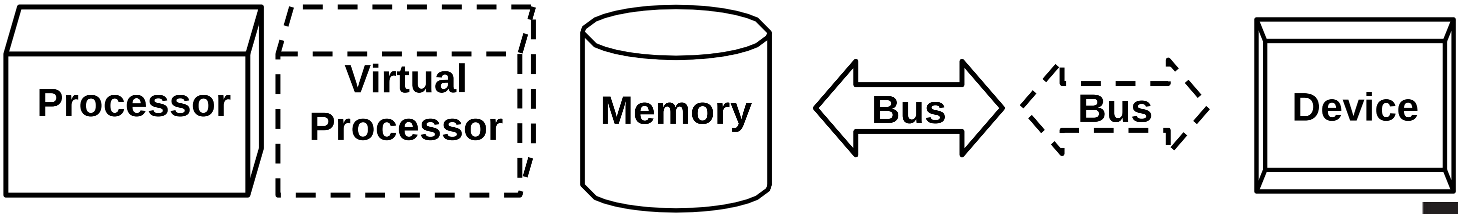


Content

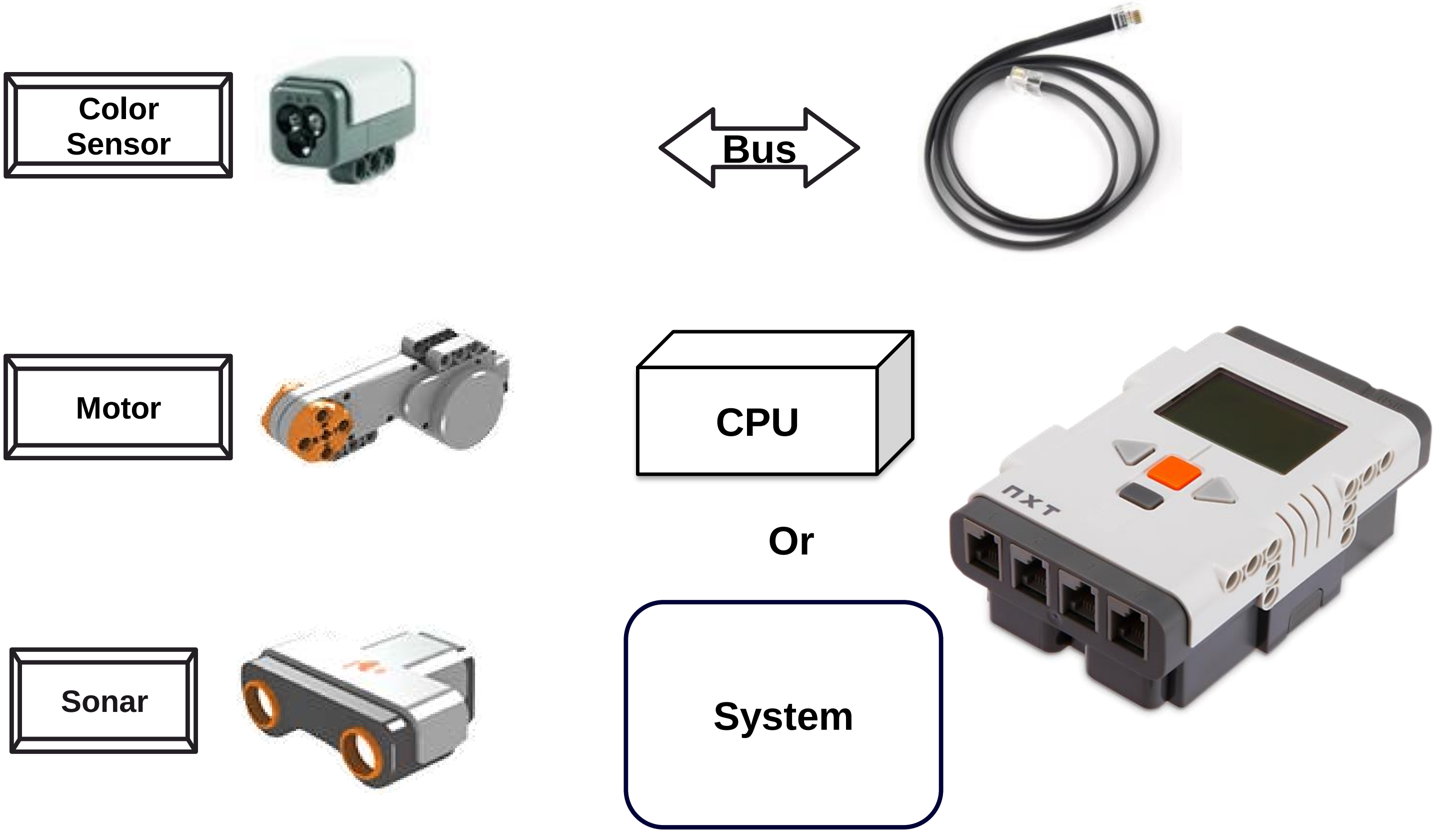
- Introduction to AADL
- Modeling Software Applications
- Modeling Execution Platforms
- Organization of Declarations
- Introduction to OSATE and AADL Inspector
- Timing Analysis with AADL Inspector
- Model Refinement and Code Synthesis with RAMSES

Execution Platform Component Categories

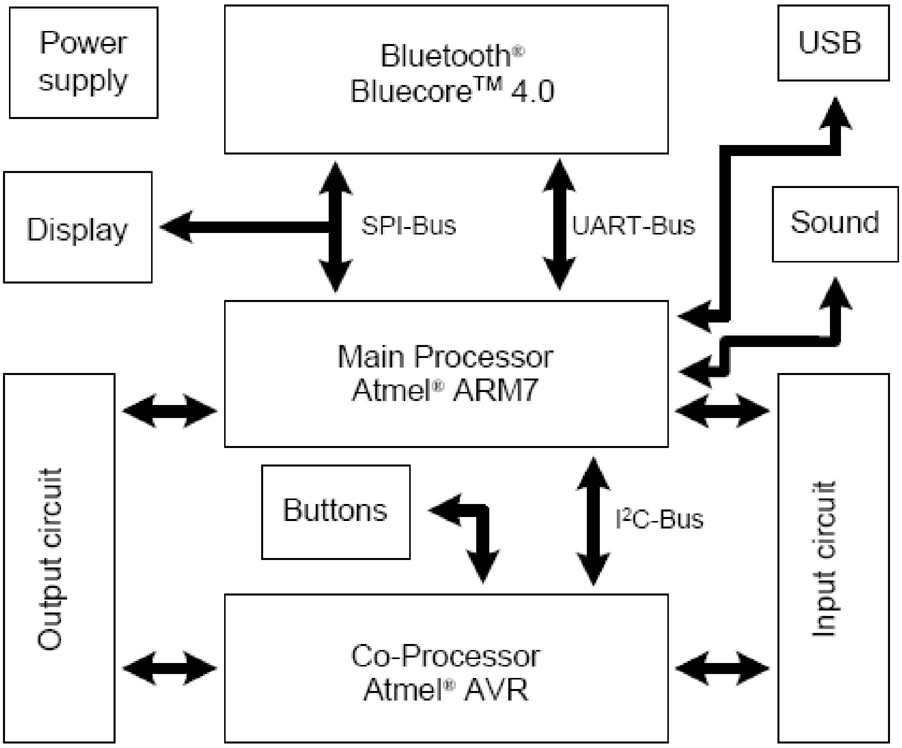
- Components categories dedicated to the **execution platform** specification:
  - **Processor**: Hardware computation unit + tasks scheduling capabilities
    - **Virtual Processor**: Processor logical partition.
  - **Memory**: Storage component (may be RAM, hard disk / flash drive, cache, etc.).
  - **Bus**: Physical communication link (cable) and protocol.
    - **Virtual Bus**: Network.
  - **Device**: Interface with the physical environment of the system (sensors/ actuators).
  - Etc.



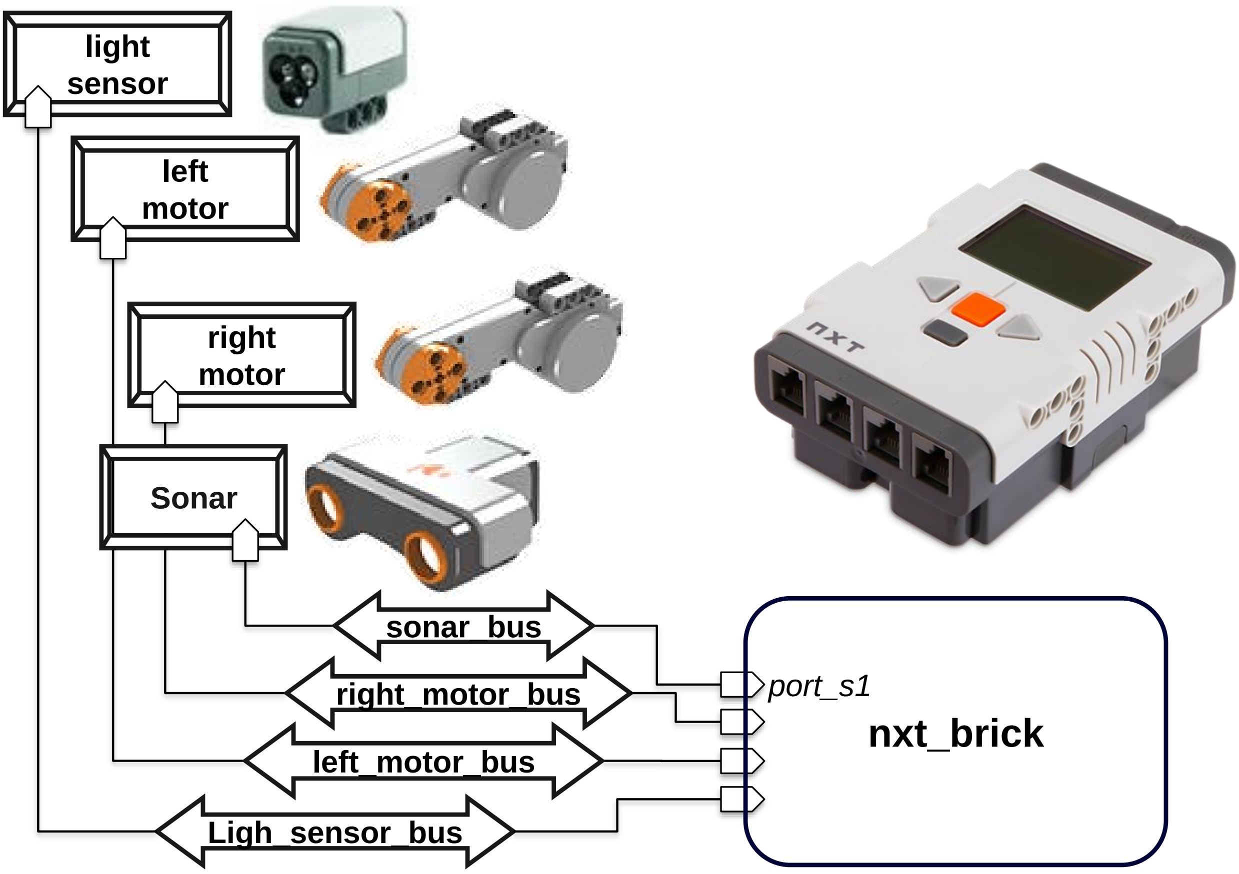
Example Execution Platform Components



Lego Robot Brick System



Lego Robot Hardware Platform



Lego Robot Hardware Platform in Textual Notation

- Define a **bus** for sensors and actuators wires.
- Add a **requires bus access** feature to the device.

```
bus Device_Bus
end Device_Bus;

device Light_Sensor
    features bus_access: requires bus access Device_Bus;
end Light_Sensor;
```

- **Connect** components to bus via bus access connections.

```
system implementation Robot_Hardware.Basic
    subcomponents
        light_sensor: device Light_Sensor;
        light_sensor_bus: bus Device_Bus;
        ...
        exec_platform: system Robot_Platform.Basic;
    connections
        light_sensor_con: bus access light_sensor_bus -> light_sensor.bus_access;
        platform_light_con: bus access light_sensor_bus -> exec_platform.light_sensor;
        ...
```

Content

- Introduction to AADL
- Modeling Software Applications
- Modeling Execution Platforms
- Organization of Declarations
- Introduction to OSATE and AADL Inspector
- Timing Analysis with AADL Inspector
- Model Refinement and Code Synthesis with RAMSES

Components Composition Rules

- A component implementation may declare **subcomponents**.
- An AADL model is therefore a **tree of components** starting by a root component, usually of **system** category.
- **Legality rules** define what categories of subcomponents a component can contain.
  - Same case for which features a component type can own.
- Look at **OSATE help** for more info.
  - OSATE documentation includes nearly all the SAE standard content.

Components Composition Rules

Category	Allowed Subcomponent Categories
abstract	data, subprogram, subprogram group, thread, thread group, process, processor, virtual processor, memory, bus, virtual bus, device, system, abstract
data	data, subprogram, abstract
subprogram	data, subprogram, data
subprogram group	subprogram, subprogram group, data, abstract
thread	data, subprogram, subprogram group, abstract
thread group	data, subprogram, subprogram group, thread, thread group, abstract
process	data, subprogram, subprogram group, thread, thread group, abstract
processor	memory, bus, virtual processor, virtual bus, abstract
virtual processor	virtual processor, virtual bus, abstract
memory	Memory, bus, abstract
bus	virtual bus, abstract
virtual bus	virtual bus, abstract
device	bus, virtual bus, data, abstract
system	data, subprogram, subprogram group, process, processor, virtual processor, memory, bus, virtual bus, device, system, abstract

Packages

- Like in programming languages such as Java, AADL includes a **package** notion to contain **component declarations**.

- Packages contain a **public** section and optionally a **private** section.
  - Component declarations are contained in these sections.

- Example:

```
package robot_deployment
public
    with robot_platform, robot_software;

    system Robot_Deployed
    end Robot_Deployed;

    system implementation Robot_Deployed.Basic
        subcomponents
            light_sensor_driver: device robot_software::Light_Sensor;
    ...
end robot_deployment;
```

Content

- **Introduction to AADL**
- **Modeling Software Applications**
- **Modeling Execution Platforms**
- **Organization of Declarations**
- **Introduction to OSATE and AADL Inspector**
- **Timing Analysis with AADL Inspector**
- **Model Refinement and Code Synthesis with RAMSES**

OSATE: Open-Source AADL Tool Environment

- Developed at the Software Engineering Institute (SEI) of the Carnegie Mellon University (CMU).
- **Synchronized** textual and graphical editors.
- Eclipse-based: Eclipse Modeling Framework (EMF);
  - Ecore meta-meta-model.
  - Xtext to define textual languages
  - Etc.
- Actively maintained.

This is the OSATE Open Source AADL Tool Environment.

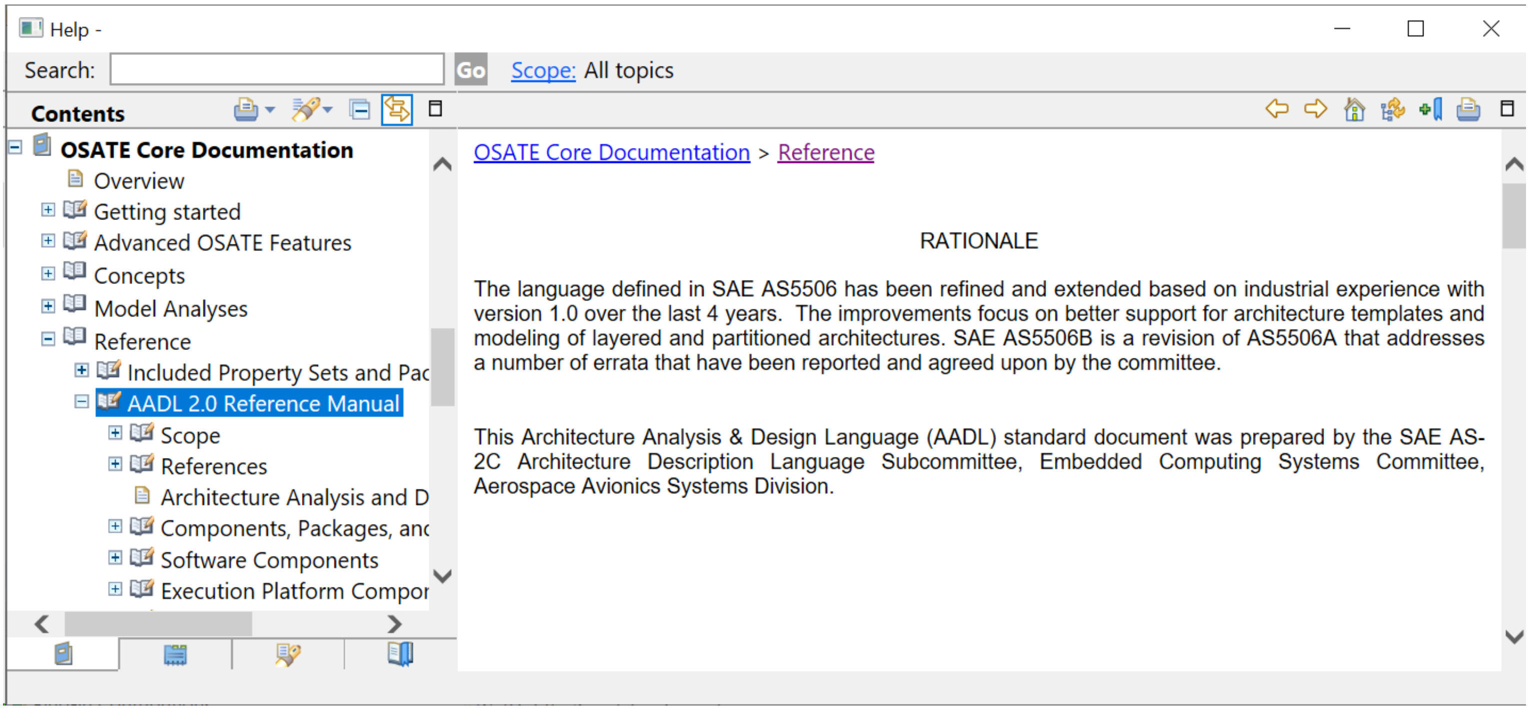
Version: 2.11.0.vfinal -- Build id: 2022-06-16

Copyright (c) 2004-2022 Carnegie Mellon University.
All Rights Reserved.

This offering is based on technology from the Eclipse Project.
Visit http://osate.org and http://www.eclipse.org

OSATE Documentation

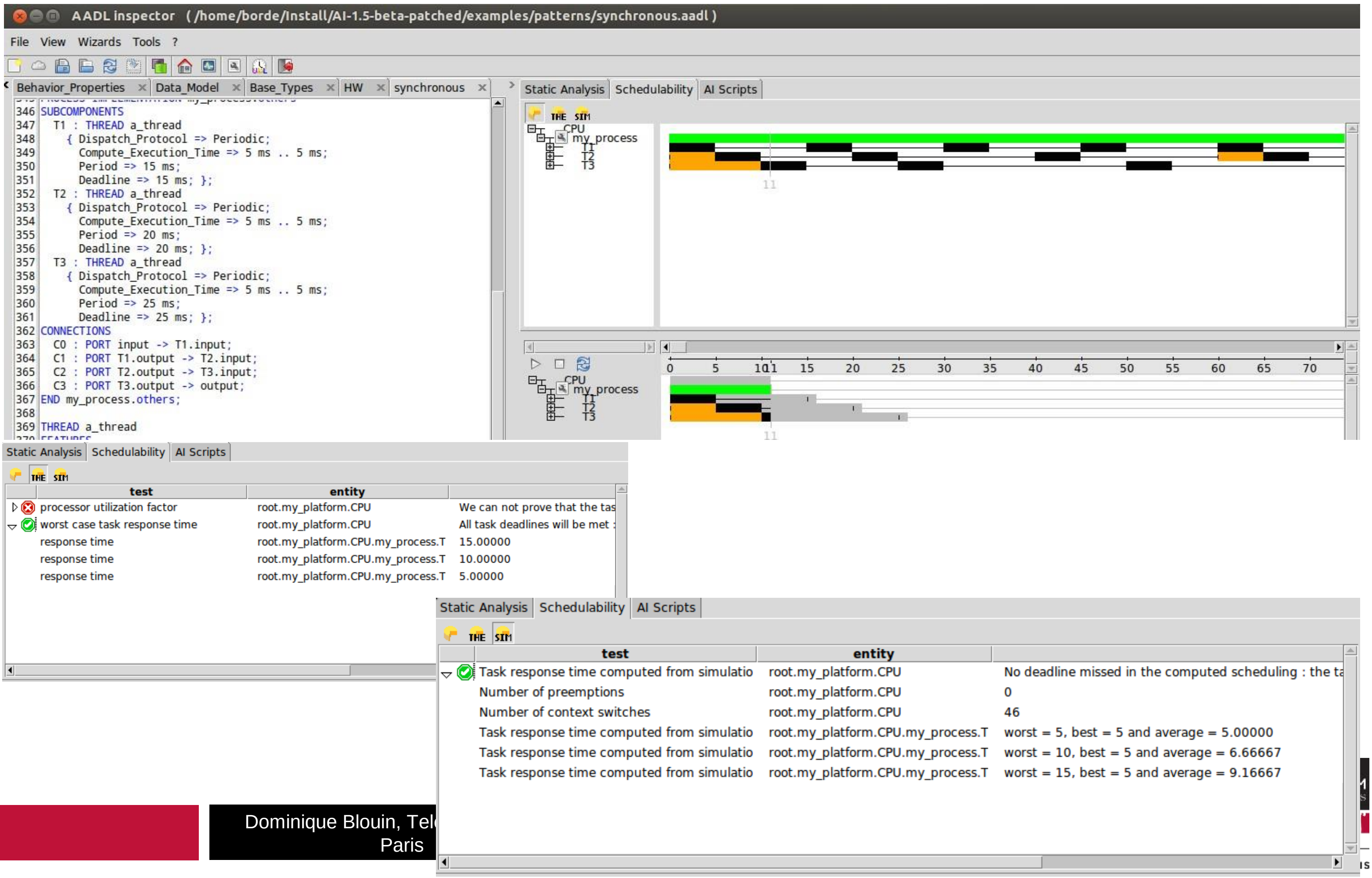
- Click menu *Help>>Help Contents.*
- Navigate to the *OSATE Core Documentation* branch.



Another Analysis Tool

- AADL Inspector: Standalone (not in Eclipse IDE) analysis tool for AADL.
- Developed by Ellidiss Technologies in France
  - Active member of AADL standards committee from the beginning.
  - Provided a **free evaluation license** for this course.
- Integrates other tools from academia such as Cheddar.
  - Scheduling analyzer and simulator.
- Take as input same AADL textual files (.aadl) as for OSATE.

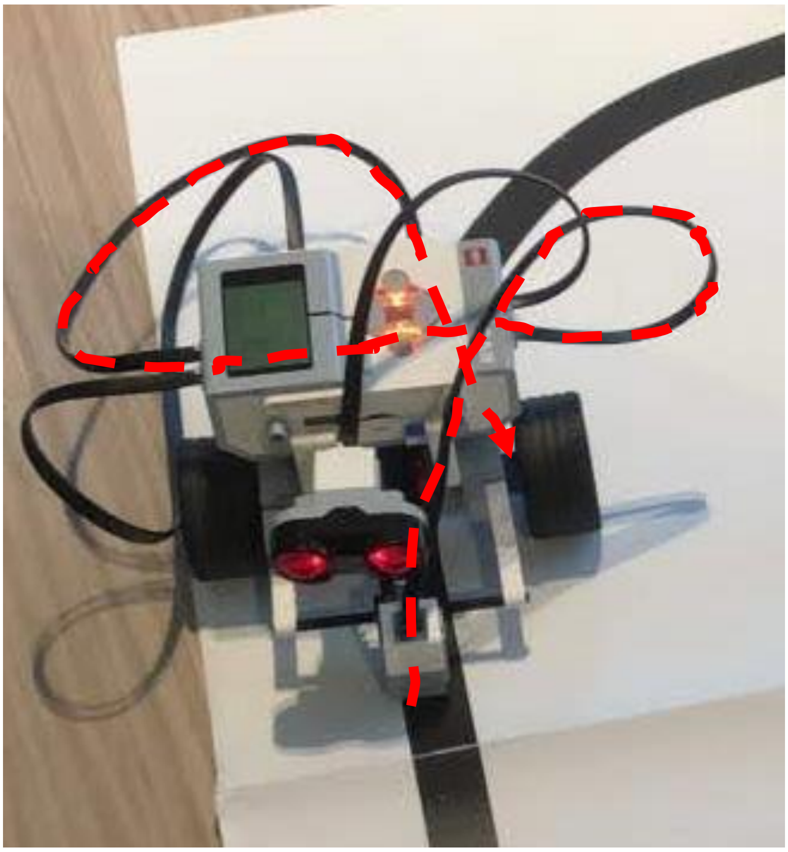
AADL Inspector User Interface Views



Content

- Introduction to AADL
- Modeling Software Applications
- Modeling Execution Platforms
- Organization of Declarations
- Introduction to OSATE and AADL Inspector
- Timing Analysis with AADL Inspector
- Model Refinement and Code Synthesis with RAMSES

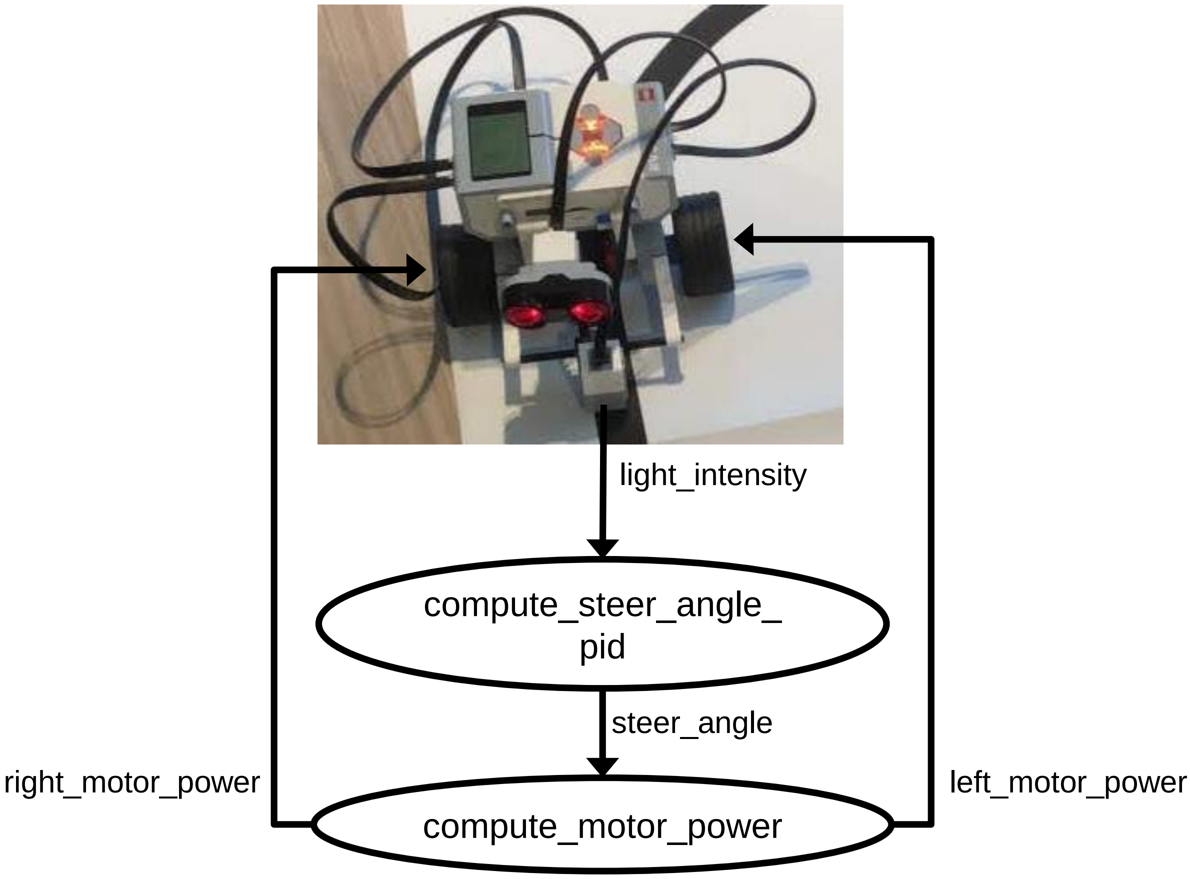
Importance of End-to-End Latency



- The **maximum allowed latency** on steering the robot upon a change of light intensity may depend on several parameters:
  - How fast the robot needs to go:
    - The faster the robot goes, the lower the latency will need to be.
  - Minimum curvature radius of the path to follow:
    - The smaller the radius is, the lover the latency will need to be.

- Latency is a **primary concern** in designing real-time systems.

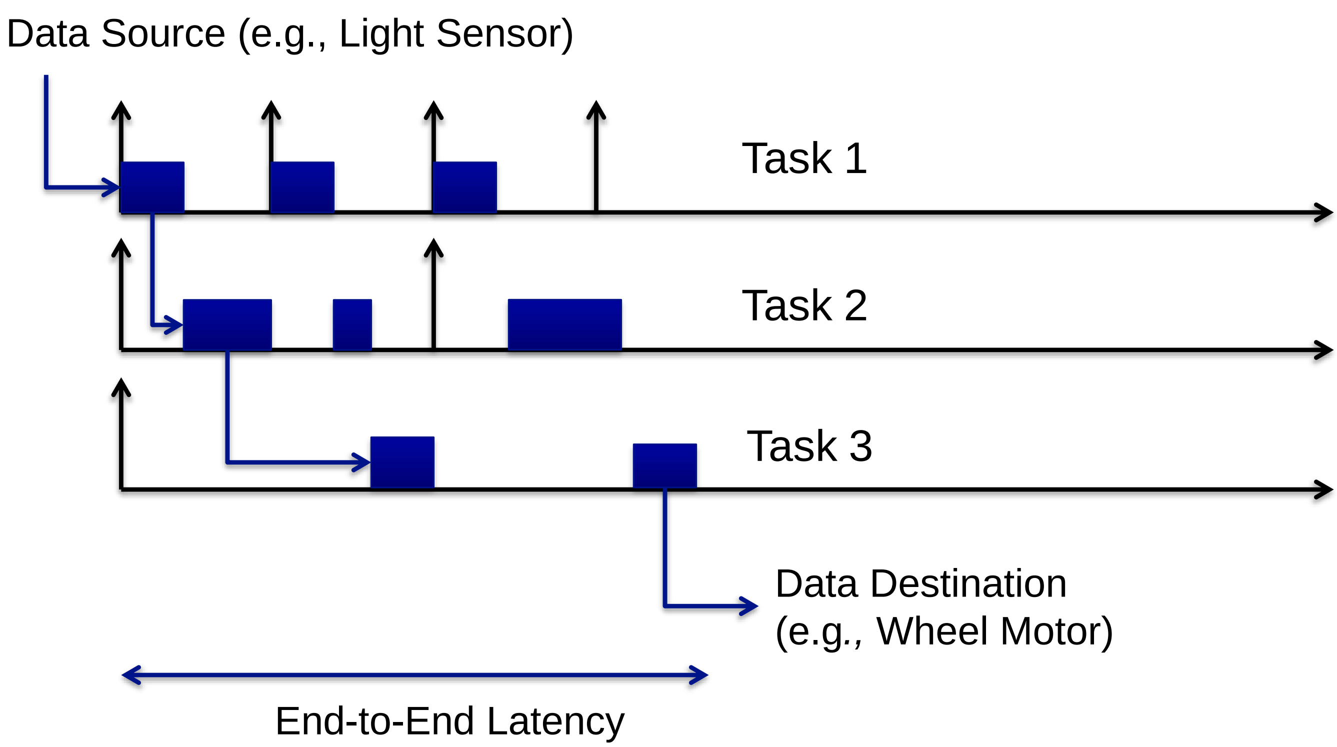
Importance of Response Time



Other Important Parameters

- **Sampling frequency**: How often do we need to execute the control loop function?
  - E.g., 50Hz (every 20 ms).
- What happens if the **computation time** is greater than the **period**?
  - E.g. 40 ms.
- What will be the impact on the system dynamics?
  - Any performance issues?
- What will be the impacts for users?
  - Any safety issues?
- Besides correct system functions,
  - Data must be available **in time**.
- **Real-time ≠ fast computing**.

Latency Example



Latency Contributors

- The tasks **response time** contributes to latency and jitter.
- What are other contributors?
- **Communications** between tasks:
  - Running on the **same** processor.
  - Or running on **different** processors.
    - Remote communication between tasks.
- Therefore, we need to add to our model information on **communications between components**…

AADL Properties

- A property allows associating a value of a given type to any model element in AADL (any component type, implementation, feature, etc.).
- The AADL standard defines a set of properties for common analyses of embedded systems.
- It is also possible to define user-specific properties declared in **property sets**.
- Properties is a sub-language of the core AADL.
- Properties can define constraints on which kinds of component they can be applied to (applicability).

Properties Syntax

- A property has a **type**. It can be:
  - Boolean: `aadlboolean`
  - Integer: `aadlinteger`
  - Real: `aadlreal`
  - String: `aadlstring`
  - Enumeration: `enumeration`
  - Component classifier: `classifier` (component, connection, etc.)
  - Reference to a model element: `reference` (component…)
  - A range of values: `range` …
  - A list of values: `list of` …
  - A quantity unit: `unit`

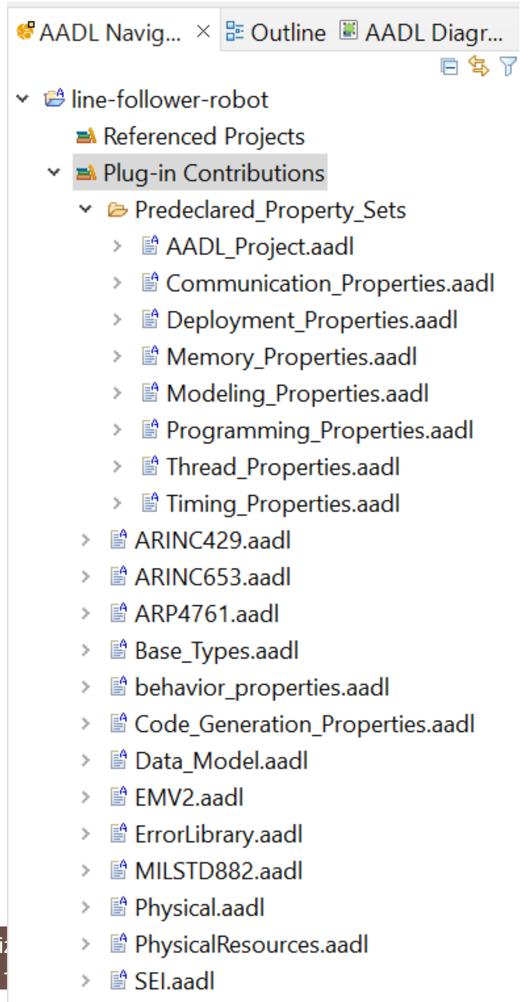
- It is possible to:
  - Define property types **reusing** existing types.
  - Associate a **default value** to a property.
  - Define the **applicability** of a property to component **categories** but also to specific component **types** or **implementations** (classifiers).

Standard Predefined Properties

- The AADL standard defines properties to be used for common analyses related to communication, memory, threading and timing.
  - Those are described in the standard documentation (available under OSATE help).

- Standard properties can be viewed within OSATE for any AADL project under the **Plug-in Contributions** library folder as shown in the screenshot.

- Other visible property sets are also part of the standard but are provided by annexes of AADL such as the Error Model Annex (EMV2).



Dominique Blouin, Telecom Paris, IP Paris

Modeling, Analyzing and Synthesiz with AADL using RAMSES -

AADL Properties that Influence Latency

- Connection **Timing**:

```
Timing: enumeration (sampled, immediate, delayed) => sampled applies to (port connection);
```

- **Deadline** or **Compute_Execution_Time**
  - Which one is used depends on the connection **timing**.

- **Transmission_Time** to be set on buses.

```
Transmission_Time: record (Fixed: Time_Range; PerByte: Time_Range;) applies to (bus, system, device,
processor, memory, virtual bus, virtual processor);
```

- **Latency** used to specify previously known latencies to various model elements or requirement on an end-to-end flow (see later).

```
Latency: Time_Range applies to (flow, connection, virtual bus, bus, processor, virtual processor, device,
system, feature, memory);
```

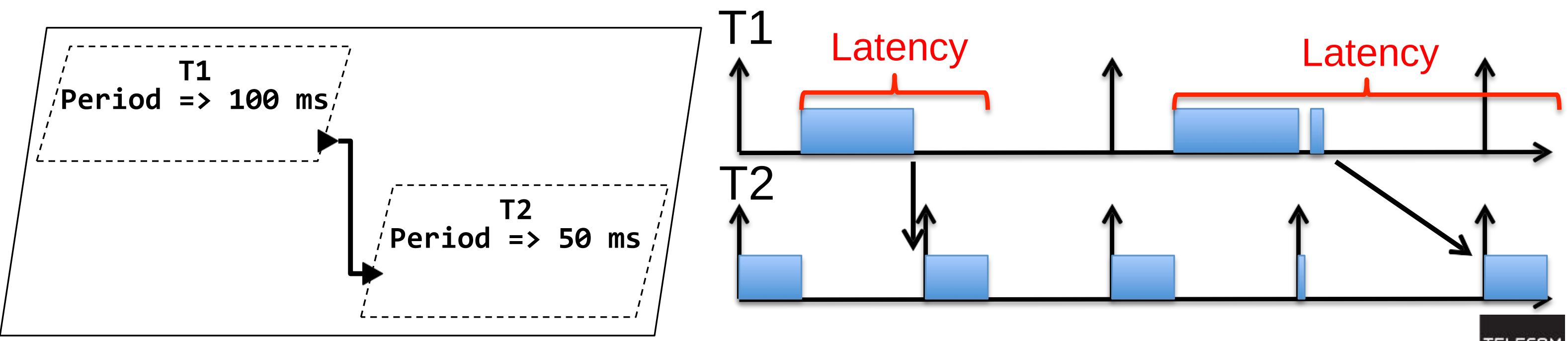
Examples

```
system implementation Exec_Platform.Basic
    subcomponents
        cpu_1: processor Arm.V7 {Scheduling_Protocol => (ARINC653);};
    properties
        Scheduling_Protocol => (RMS) applies to cpu_1;
end Exec_Platform.Basic;

processor Arm
    properties
        Scheduling_Protocol => (Round_Robin_Protocol);
end Arm;

processor implementation Arm.V7
    properties
      Scheduling_Protocol => (POSIX_1003_HIGHEST_PRIORITY_FIRST_PROTOCOL);
end Arm.V7;
```

What will be the scheduling protocol?

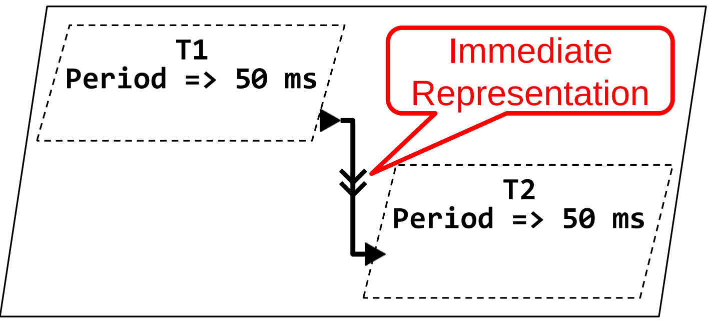
Data Port Connection Timing Property

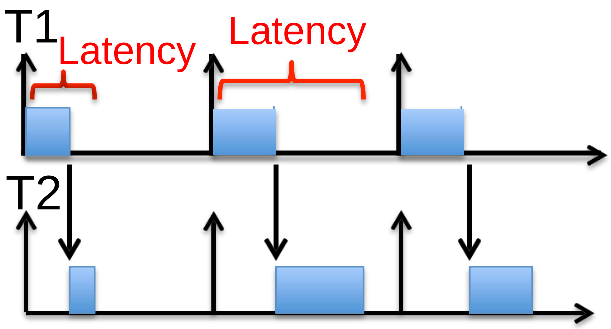
- **Sampled**:
  - Like a shared variable.
  - Thread T1 writes its data at the **end** of its execution and thread T2 reads the data at the **beginning** of its execution.
  - Advantage: Simplicity.
  - Disadvantage: **Non-deterministic**.



Data Port Connection Timing Property

- **Immediate**: The recipient only starts when the output port of the connected thread has been updated.
  - Advantages:
    - Deterministic.
    - Reduces latency.
  - Disadvantages: Imposes constraints on the scheduler and the thread model:
    - No task dependency cycle.
    - The execution of T1 must precede T2 so the period of T2 should be >= the period of T1.



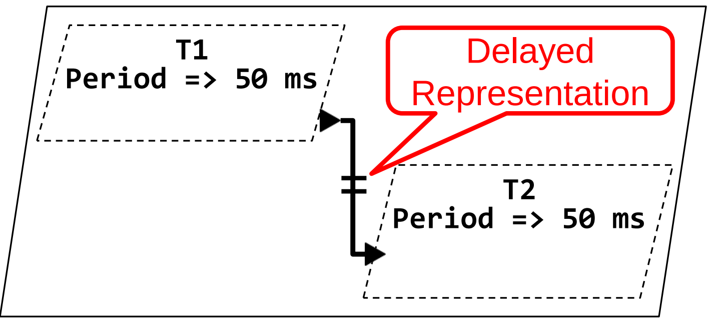


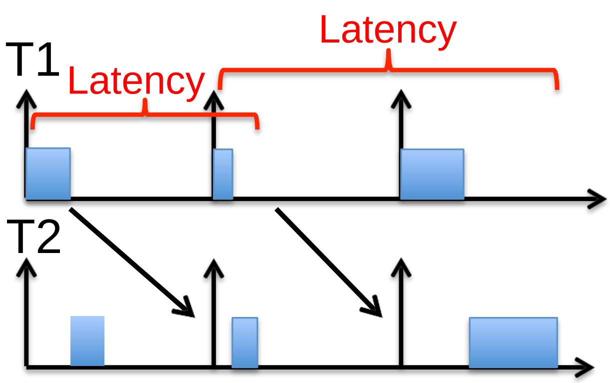
Data Port Connection Timing Property

- **Delayed**: The output port is updated at the **deadline** of its thread.

- The data is processed:
  - At the earliest after a period of T1 and a short execution of T2.
  - At the latest after a period of T1 and a long execution of T2.

- Advantage:
  - Deterministic and reduces jitter.
- Disadvantage:
  - Increases latency.





Deployment Properties

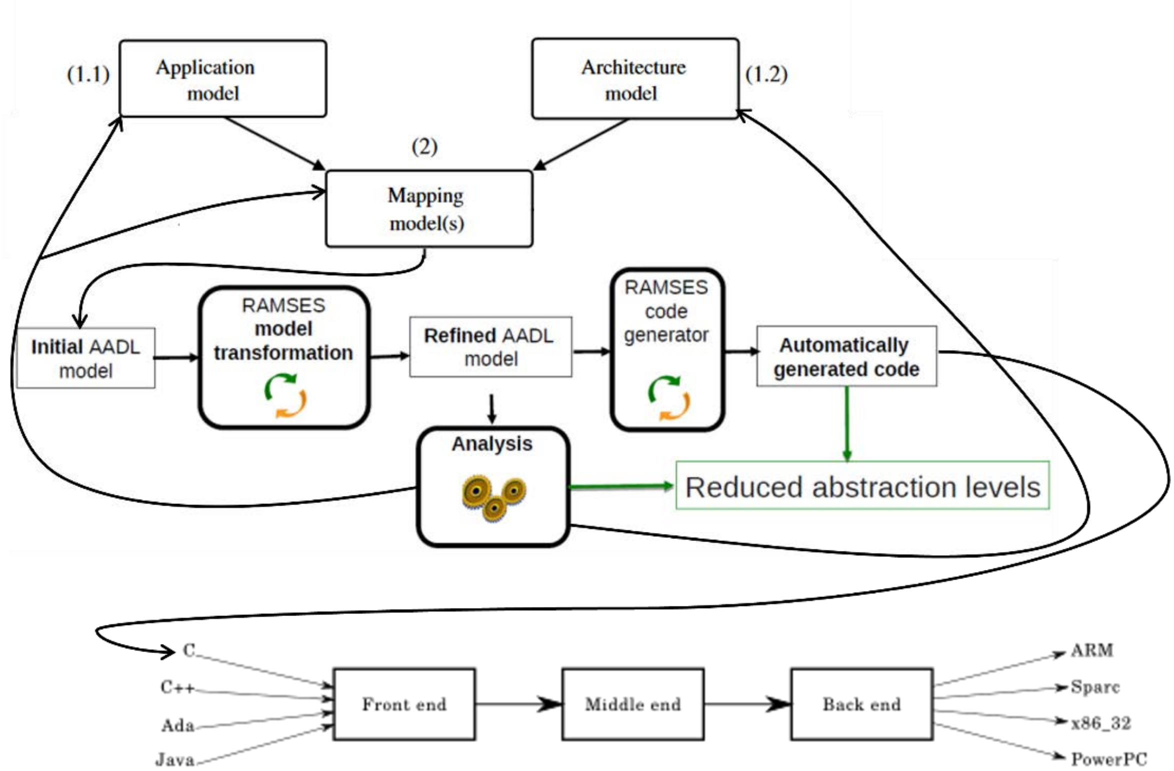
```
system implementation Synchronous.Others
    subcomponents
        my_platform: processor CPU;
        my_process: process My_Process.Basic;
    properties
        Actual_Processor_Binding => (reference(my_platform)) applies to my_process;

end Synchronous.Others;

-- Binding to nested subcomponents
system implementation Line_Follower_Robot.Basic
    subcomponents
        soft_app: system robot_software::Line_Follower_Application.Basic;
        hard_platform: system robot_platform::Robot_Hardware.Basic;
    properties

        Actual_Processor_Binding =>
(reference(hard_platform.exec_platform.basic_processor)) applies to
soft_app.line_following_controler;

        Actual_Memory_Binding => (reference(hard_platform.exec_platform.ram)) applies
to soft_app.line_following_controler;

end Line_Follower_Robot.Basic;
```

Content

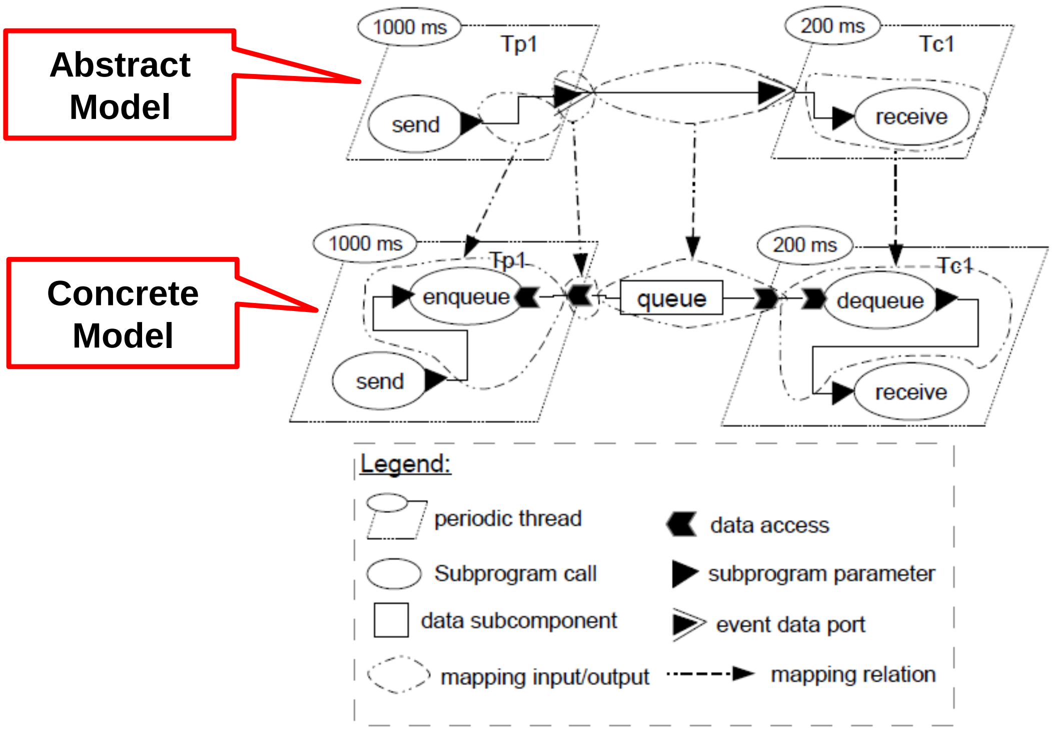
- **Introduction to AADL**
- **Modeling Software Applications**
- **Modeling Execution Platforms**
- **Organization of Declarations**
- **Introduction to OSATE and AADL Inspector**
- **Timing Analysis with AADL Inspector**
- **Model Refinement and Code Synthesis with RAMSES**

Model Refinement and Synthesis (Code Generation)

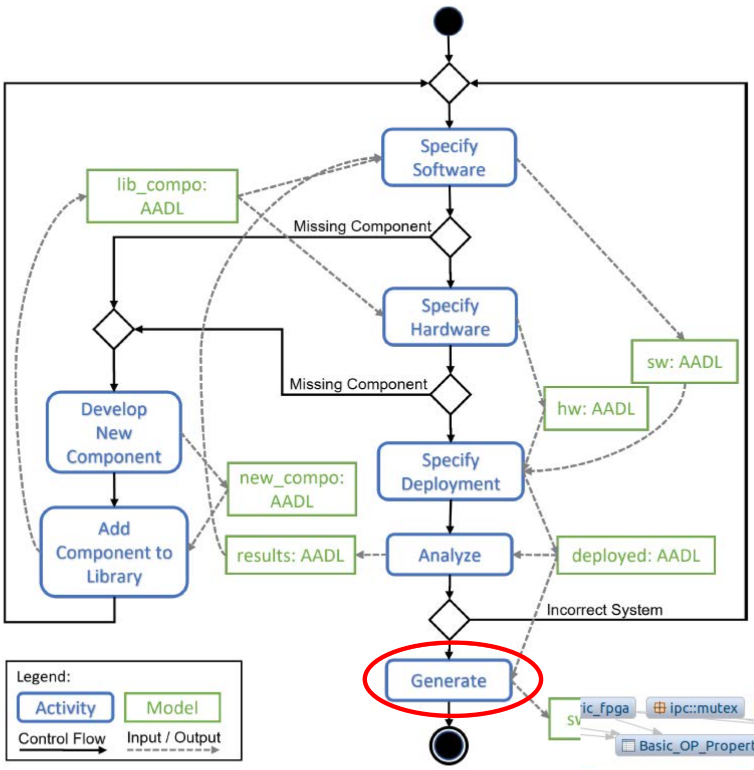
- **RAMSES**: Refinement of AADL Models for the Synthesis of Embedded Systems
  - https://mem4csd.telecom-paristech.fr/blog/index.php/ramses/

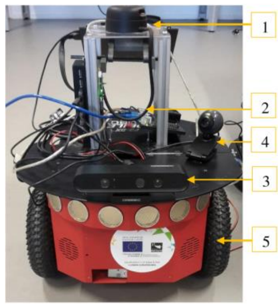


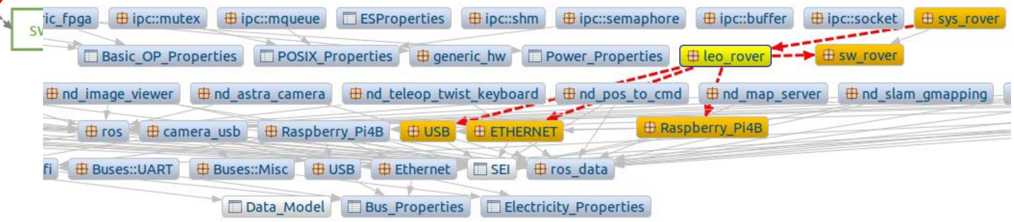
Example of a RAMSES Refinement Rule



Ongoing Project: RAMSES-ROS Extension for Complex Robotics Systems



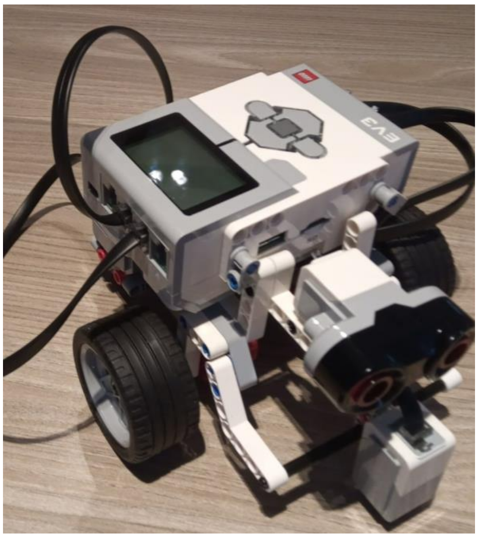




- AADL component library

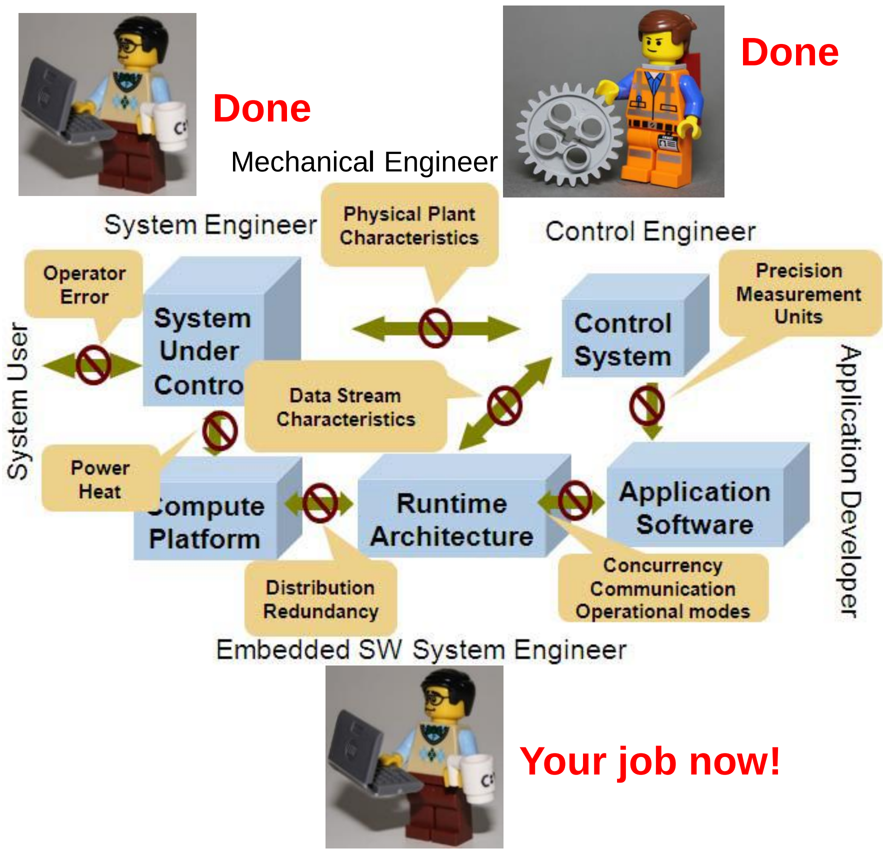
Hands-On Exercise: Model and Synthesize a Cruise-Control System

- Objectives:
  - Learn how to model a simple embedded system in AADL.
  - Learn how to analyze the model and modify the design to ensure the program execution is not jeopardized due to performance limitations.
  - Synthesize the C code of the system from this model.



- Follow the instructions on our website:
  - https://mem4csd.telecom-paristech.fr/

Engineering Domains



Making Teams

- 10 robots ➔ How many students per team?
- **Linux** required for the exercise.
  - GCC must be installed.
- Define the teams according to the available robots and Linux computers.

CPS+IoT+AI for Biomedical Applications with emphasis to CreativeIntegration principle

Prof. dr Radovan Stojanović*, Jovan Djurković**

*University of Montenegro

** MANT and MECOnet

stox@ucg.ac.me

Ver. 1.1

Content

- Introduction
- Architecture
- Challenges
- CI principle with examples
- Conclusions

Introduction

- **What is Biomedical Engineering (BME)?**
  - Biomedical Engineering (BME) is a multidisciplinary discipline that applies principles of engineering, technology, and science to medical and biological systems. Its goal is the development of devices, equipment, software systems and technologies that improve the health, diagnosis and treatment of patients and other living beings. BME combines knowledge from engineering, biology, medicine, physics, chemistry, ICT in order to create innovative products and solutions for purposes of health care.
  - BME includes many areas, ranging from development of medical devices as pacemakers, defibrillators, MRI scanners, ultrasound devices, ventilators and other diagnostic devices through prosthetics, orthotics, biomaterials, therapy, medical informatics etc.

Introduction

- Medical instruments and devices, big fan.

https://www.indiamart.com/proddetail/pathology-lab-equipments-27622230388.html

https://www.nairaland.com/6649159/medical-surgical-laboratory-equipment

CPSIoT Summer School, Budva, June 2025

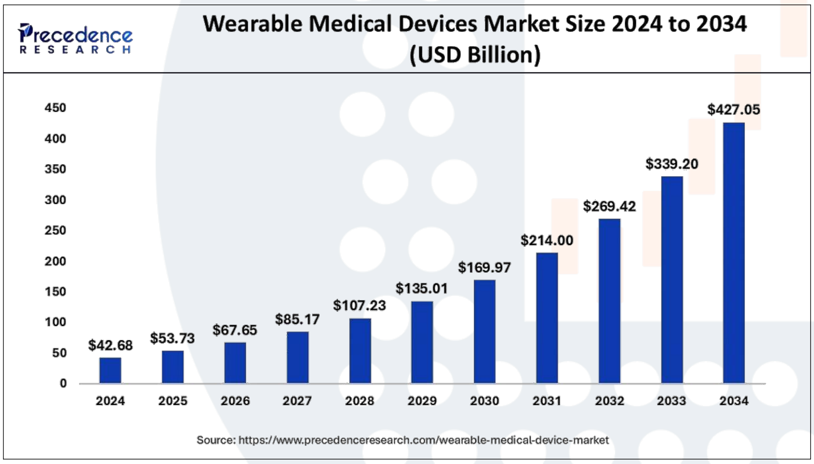
Inroduction

- We focus on "medical wearables", "wearable medical devices", or “wearables”, further **MW**, placed/fixed on the patient's body.
- Most often, they monitor patients vital parameters as: temperature, pulse, heart rate, blood pressure, breathing rhythm, blood glucose content, physical activity, etc...
- We will mainly address non-invasive wearables for monitoring physiological vital signs.

https://www.researchgate.net/publication/338316751_The_growing_role_of_Internet_of_Things_in_healthcare_wearables

Introduction

- The global market of MW is very dynamic and is estimated to grow at an upward rate for the next 10 years, from USD 42.68 billion in 2024 to USD 427.05 billion in 2034 with an annual growth rate of 25.9%.



https://www.precedenceresearch.com/

https://www.healthworkscollective.com/credit-suisse-says-wearable-tech-next-big-thing/

Introduction

- Today MW are Artificial Intelligence Medical Internet of things (AIMIoT). Evolution Embedded (ES) -> Cyber Physical Systems (CPS) -> Medical Internet of Things (MIoT)-> AIMIot.
- Example Glucose Monitoring.

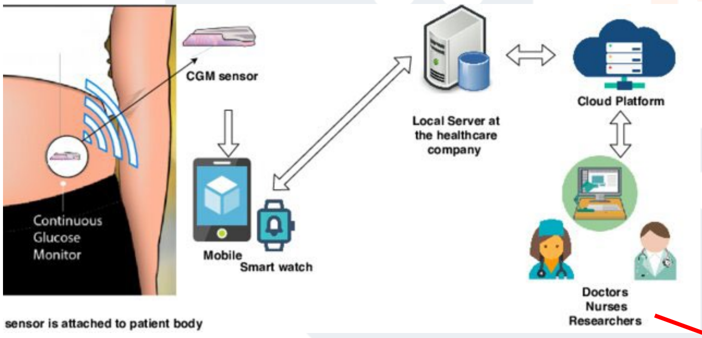
ES

- **Microcontroller**
- **Memory**
- **I/O interfaces,**
- **Software (firmware)**

CPS

- **Sensors**
- **CPU**
- **software**
- **networking**
- **actuators**

https://www.diabeloop.com/products



https://www.researchgate.net/publication/336733965_Designing_Security_and_Privacy_Requirements_in_Internet_of_Things_A_Survey/figures?lo=1

MIoT

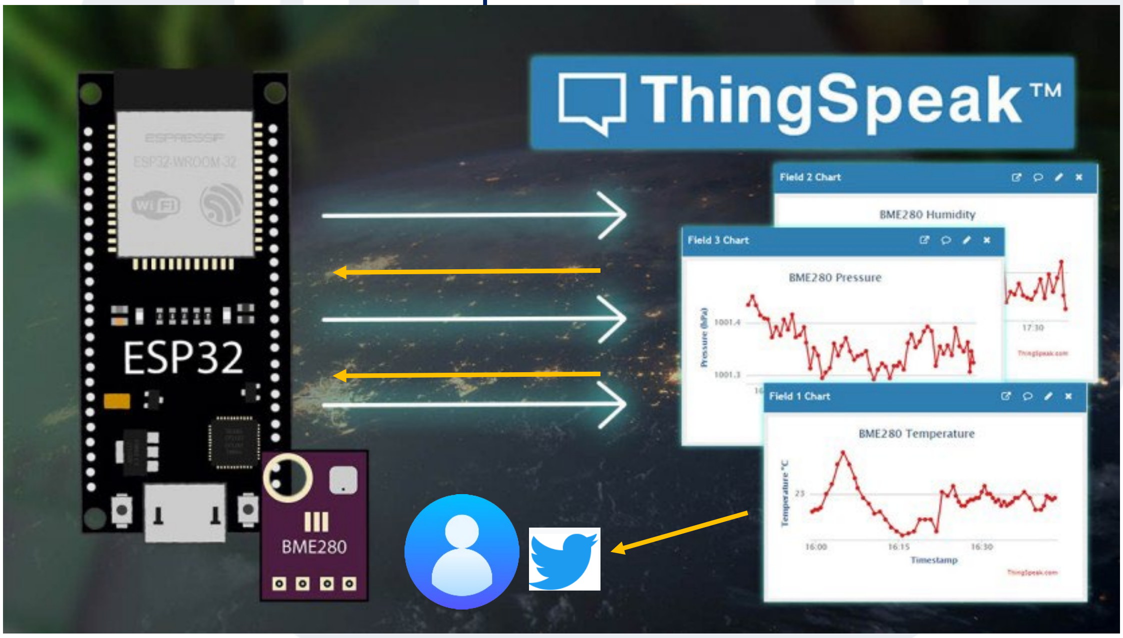
- **CPS with Internet**

+AI **AIMIoT**

CPSIoT Summer School, Budva, June 2025

Introduction

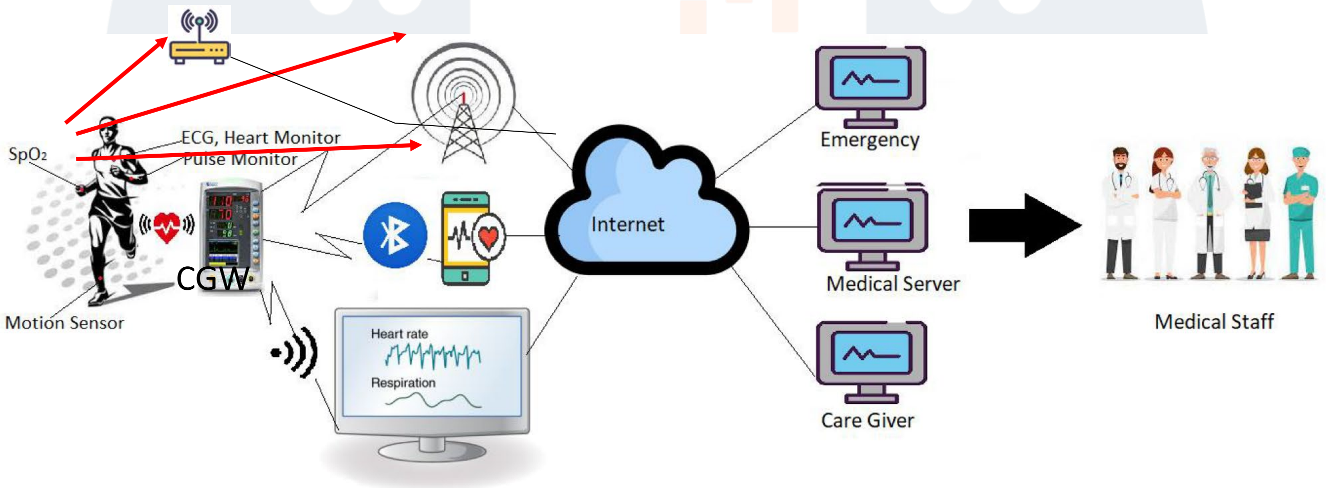
- Today, there is no boundary between ES, CPS, MIoT and AIMIoT systems. For example, ESP32+BME280+ThingSpeak can be considered e.g. IoT system, in some cases AIMIoT with a minimum number of components.



https://randomnerdtutorials.com/esp32-thingspeak-publish-arduino/

Introduction

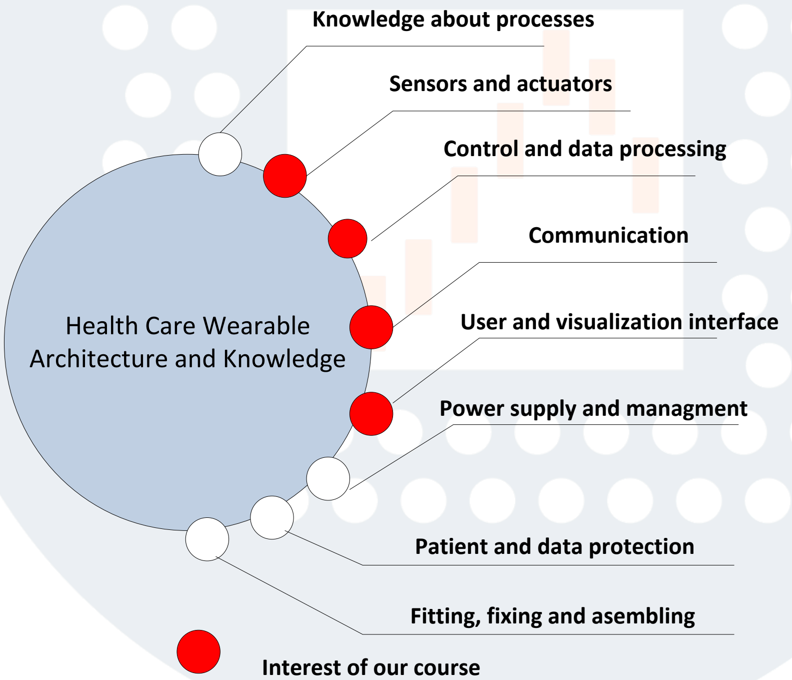
- A typical MIoT architecture in a broad sense.
  - A body-worn version of the MIoT client, which communicates with the medical staff in different ways.



https://www.mdpi.com/1424-8220/22/20/7722

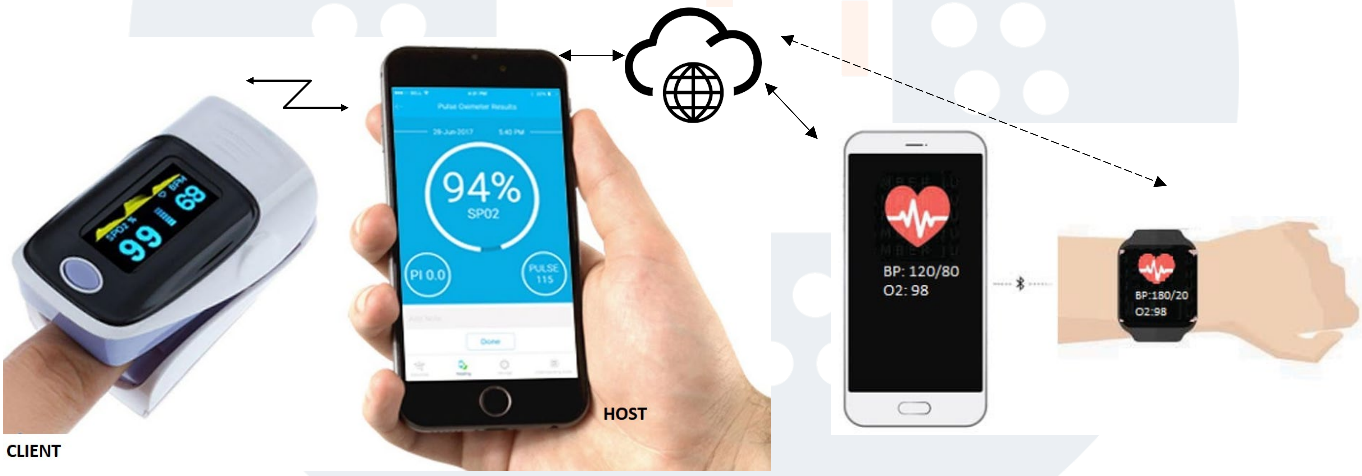
MW Architecture

- MWs require a synthesis of diverse knowledge and techniques, where each of them are story for themself.



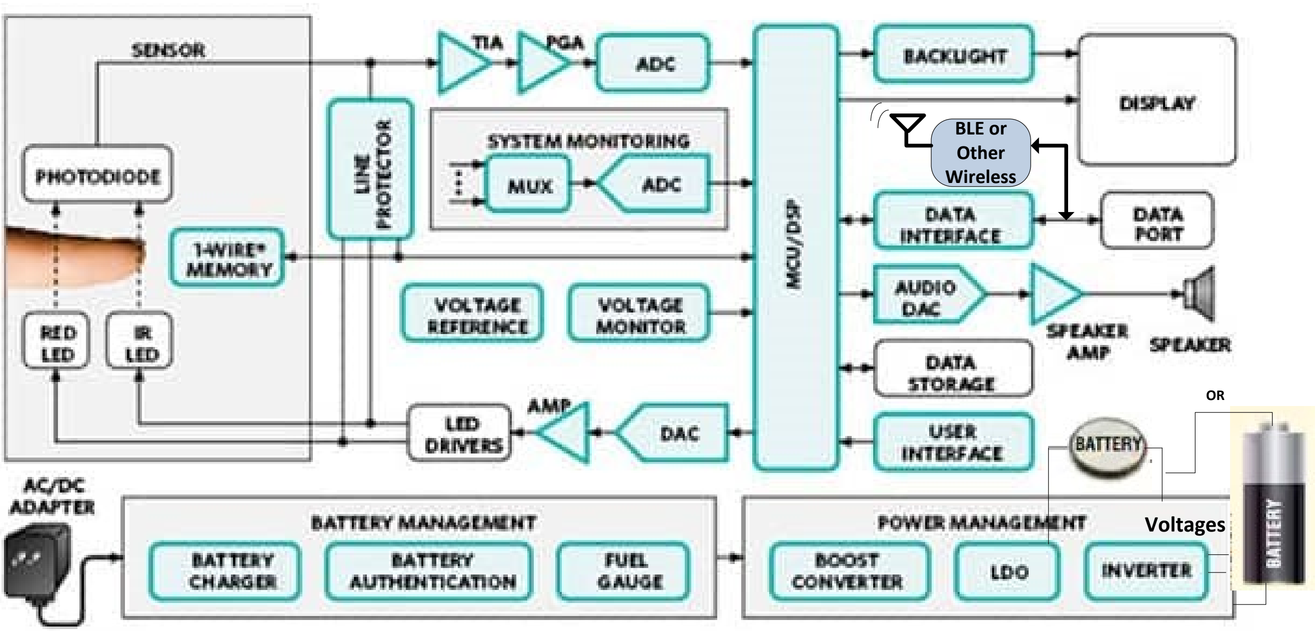
MW Architecture

- Generally seen, it is a client-host approach. Gadget (phone) can be all in 1, collector+gateway+analyser+...+, practically a host. A smart watch is an example of a very powerful device, "all in one."



MW Architecture

- Client architecture in a broader sense. Each its module is a story for itself.

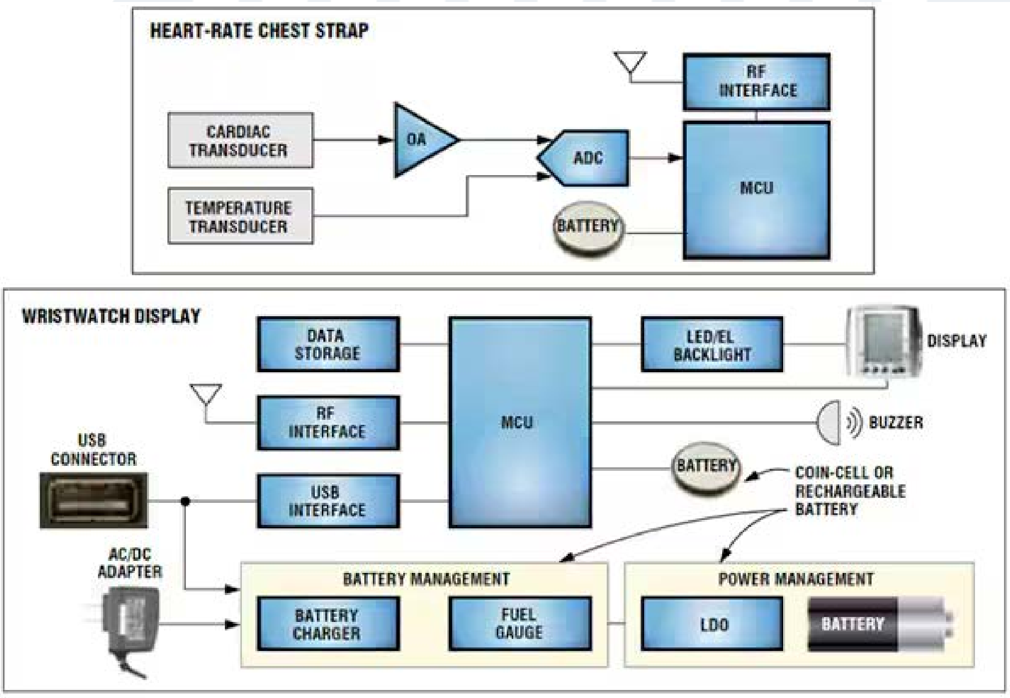


HeCaWe Architecture. From Maxim electronics with author's modifications.
https://www.digikey.com/en/articles/reducing-wearable-health-fitness-device-design-time

CPSIoT Summer School, Budva, June 2025

MW Architecture

- Simplified architecture



HeCaWe Architecture in different configurations in term of complexity. Different complexity options. From Maxim electronics with author's modification
https://www.digikey.com/en/articles/reducing-wearable-health-fitness-device-design-time

MW Design challenges

- Applies to both client and host, including hardware and software aspects.
  - Client challenge.
    - Battery: choice (according to the need for powering the processor and the rest of the circuit), price of the battery, rechargeable or not, capacity, budget, charging method, number of charging, possibility of charging in the field, dimensions, flexibility of change, self-discharge, impact on the environment, redundant or not, switching time, etc..etc....
  - Battery charger:
    - Design according to battery characteristics, operating voltage, LDO or DC-DC, fast, slow, solar, generator (self-charging, e.g. from movement, etc.).
  - Inverters for working circuit voltages:
    - Passive, active, DC-DC, inductor or capacitor-based DC-DC converters, linear, production of reference voltage, whether the processor itself can produce these voltages, charge pumps, negative voltage, stand by current...

MW design challenges

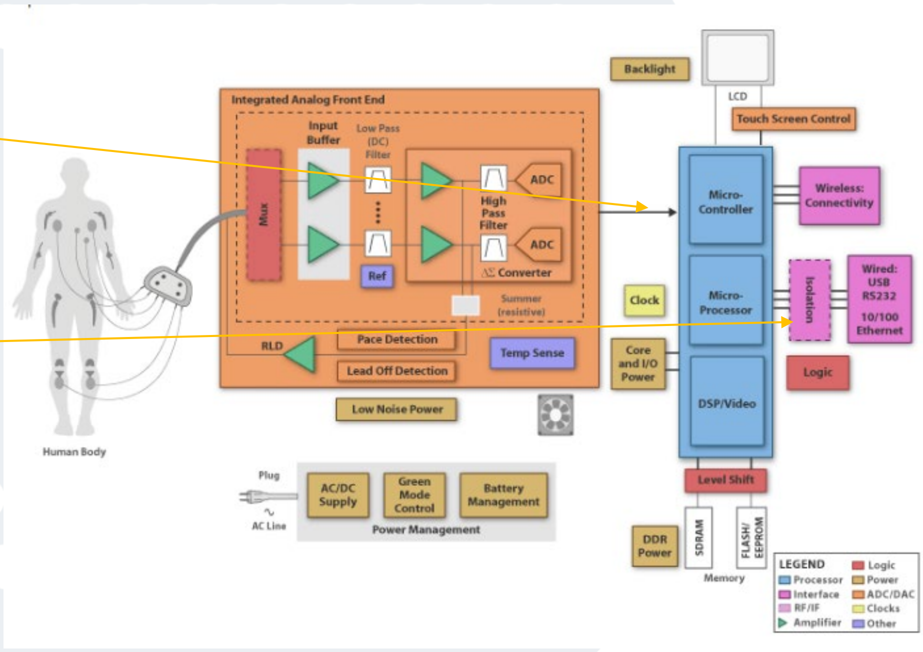
- **Analog Front End (AFE):**
  - Dedicated in the form of existing chipsets, designed by us on the basis of off-the-shelf components. Low suspicious, low consumption, single sided, accurate. If possible, use the existing analog components in the microcontroller. Some MCs have up to 4 OPs, comparators, references, etc. Use MC pins to control AFE operation, even for AGC, use PWM technique or D-A converters of MC. Input output protection according to standards, PGA, current voltage signal inputs etc..etc...
- **Drivers:**
  - Usually in optical and pulse-based measurements. Big driver power from a small battery.
- **CPU:**
  - Low consumption, fast, DSP, NPU, GPU, I-O interfaces, timers, A-D, D-A, comparator, RTOS, communication interfaces, price, housing, number of pins, visualization (display) etc...etc...Any processor can be useful...

MW design challenges

- **I/O user interface:**
  - Display, graphic, alphanumeric, resolution, consumption, manipulation, keyboard, touch etc...
- **Memorija:**
  - Programska, radna, SD card, flash, brzina, format itd
- **Communication:**
  - The most important component for CPS and IoT. Dominant wireless protocols: NFC. BLE, BT classic, Zigbee, 802.15.6 WBAN, WiFi, LoRa, ANT, ANT+, Cellularr. Dominant wire protocols (W1, I2C, SPI, UART)

MW design challenges

- **Communication**
- Between CPU and perphery (W1, I2C, SPI, UART, PWM, FREQ, CUSTOM)
- Between CPU and HOSTA (WIRE, WIRELESS)
- Between HOST and HER (electronics health record)

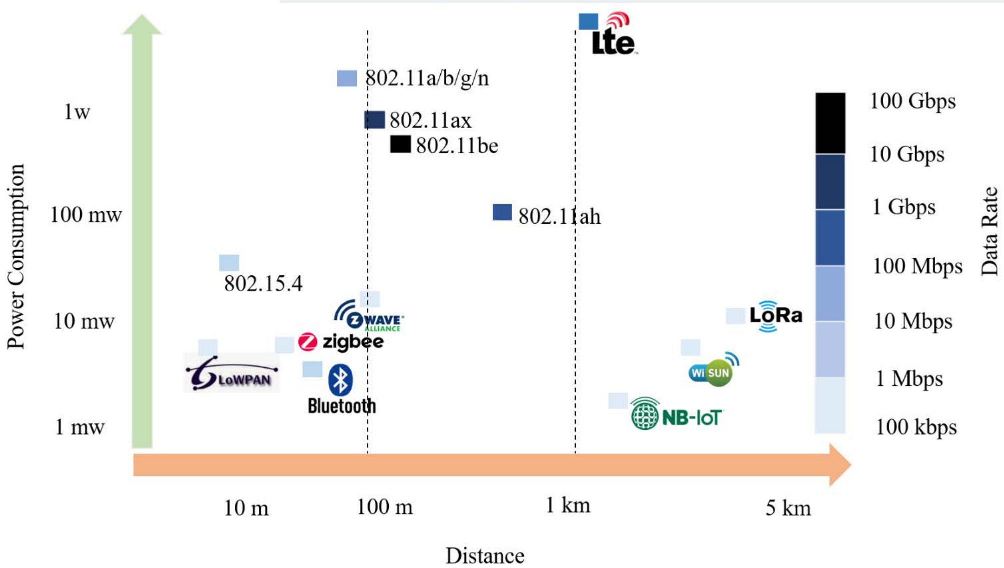


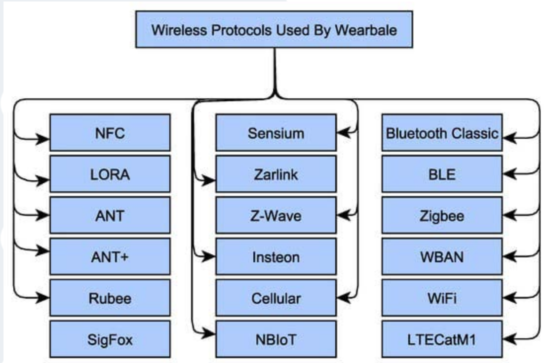
https://www.eeweb.com/dual-24-bit-ecg-front-end-ic/

CPSIoT Summer School, Budva, June 2025

MW design challenges

- **Wireless communication**





from: https://www.mdpi.com/1424-8220/22/20/7722

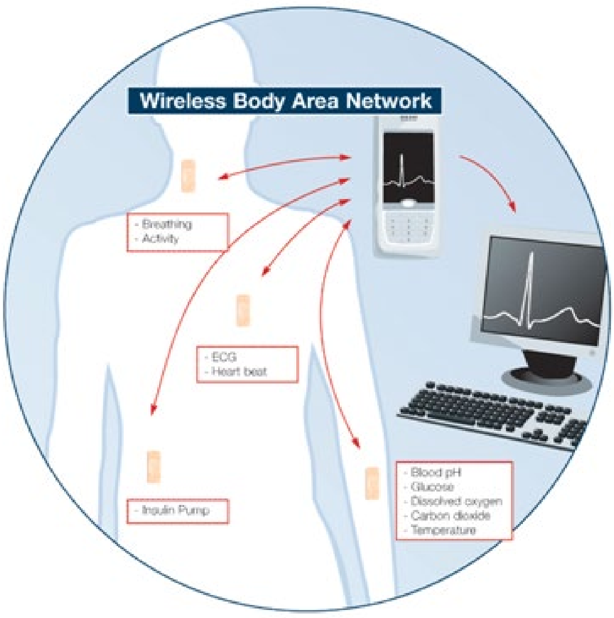
r

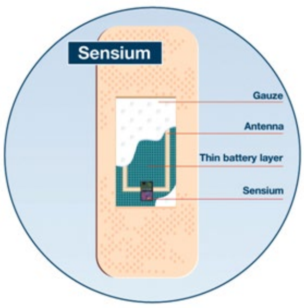
From: https://www.mdpi.com/1996-1073/16/8/3465

CPSIoT Summer School, Budva, June 2025

MW design challenges

- **Ideal wearable**
  - Miniature (patch), noise and movement resistant, easy to mount, battery powered, low consumption, communication. Example….





	Bluetooth	NORDIC	ANSEM	CHIPCON	AMIS	TOUMAZ
Frequency Band	2.4GHz	2.4 GHz	900 MHz	2.4 GHz	400 MHz	**900 MHz**
Data Rate	1Mbps	1 Mbps	115 kbps	250 kbps	16 kbps	**160 kbps**
Modulation	GMSK	GFSK	2-FSK	BPSK	ASK	**GMSK**
TX Current	25mA	19 mA	14 mA	17 mA	25 mA	**3 mA**
RX Current	37mA	25 mA	19 mA	20 mA	7.5 mA	**3 mA**
Voltage supply	2.7-5.4V	1.9 -3.6V	2.7V	2.1 -3.6V	2.4-3.6V	**1-1.5V**
Package size	6.3x8mm	5x5mm	6x6mm	7x7mm	5x5mm	**5x5mm**
RX Power	100mW	47 mW	51 mW	42 mW	18 mW	**3mW**

http://www.sensium-healthcare.com

CPSIoT Summer School, Budva, June 2025

CreativeIntegration (CI) or DIY4.0 principle in BME

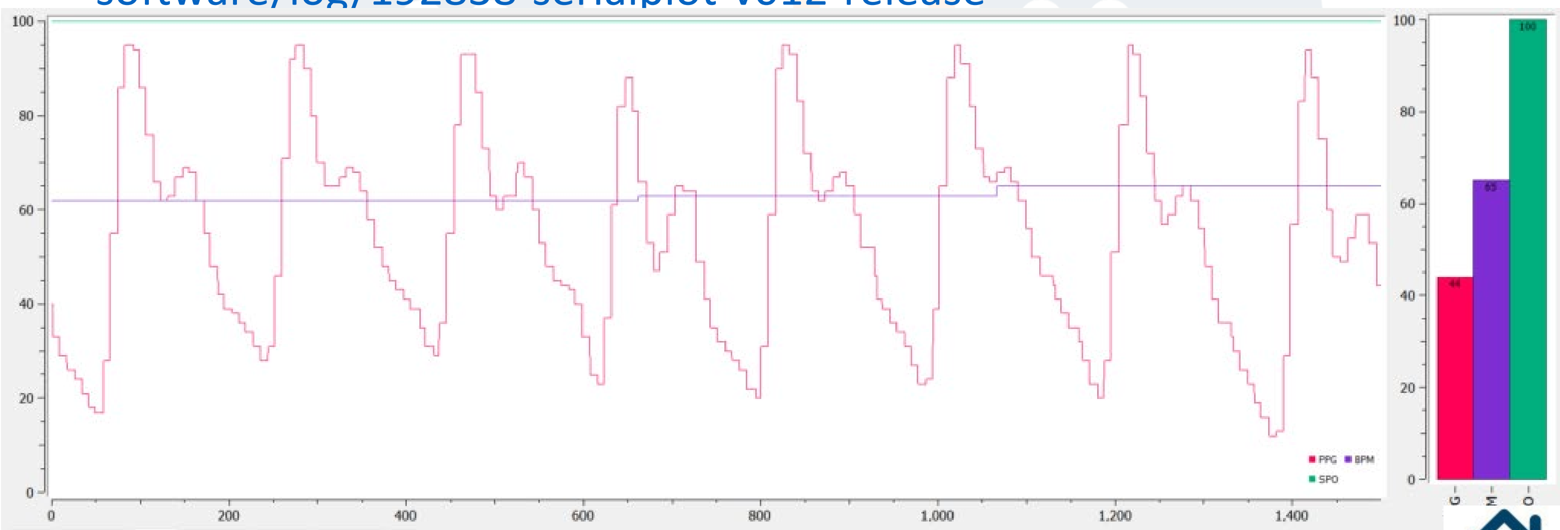
- MECOnet design approach in BME and in overall
  - #CreativeIntegration (**#CI**), or #DIY4.0
  - "Where DIY meets design“
  - “DIY isn’t dead, it’s evolving”. DIY culture is not outdated, it's just changing with the times. It's smarter, digital, open-source, and connected. You're no longer just using duct tape and hot glue, you’re integrating:
    - Analog/digital/RF front ends boards,
    - Microcontrollers
    - 3D printing
    - Open (or low cost) hardware/software (**OHW/OSW**)
    - Cloud APIs
    - AI tools
  - By CI principle, even most sophisticated and accurace medical instruments can be designed.

CI in BME…

- **Example #1.** A commercial (low-cost) oximeter connected to BLE<->USB gateway (DIY done by ESP32), which is then connected to any gadget, including PC for further signal processing, based on OSW applications.

#CI in BME

- **Example #1.** We process and analyze PPG (PLETH) signal, sent by gateway, by using commercial software, standard graphics plotters/emulators, MATLAB, LabVIEW, even free ones.
- An example of using a standard serial plotter emulator. https://hackaday.io/project/5334-serialplot-realtime-plotting-software/log/192838-serialplot-v012-release

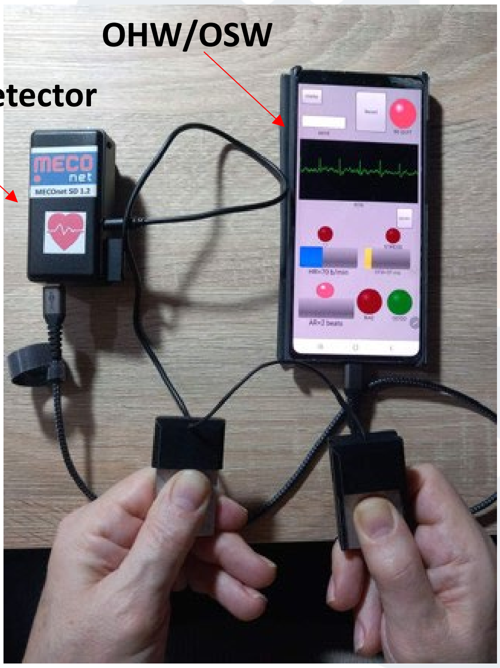


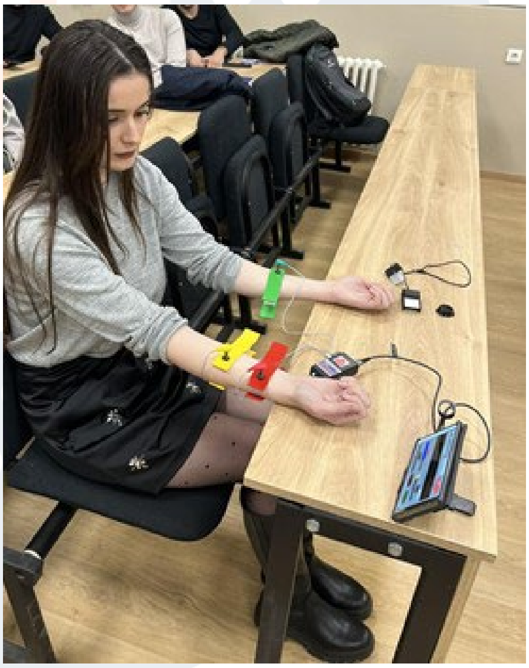
#CI in BME…

- **Example #2. Stress detector**. Monitor of human stress and arrythmias. DIY Stress detector is based on ArduinoNano +AD8234 ECG Amp. GUI implemented on the platform https://roboremo.app/ .

DIY Stress Detector

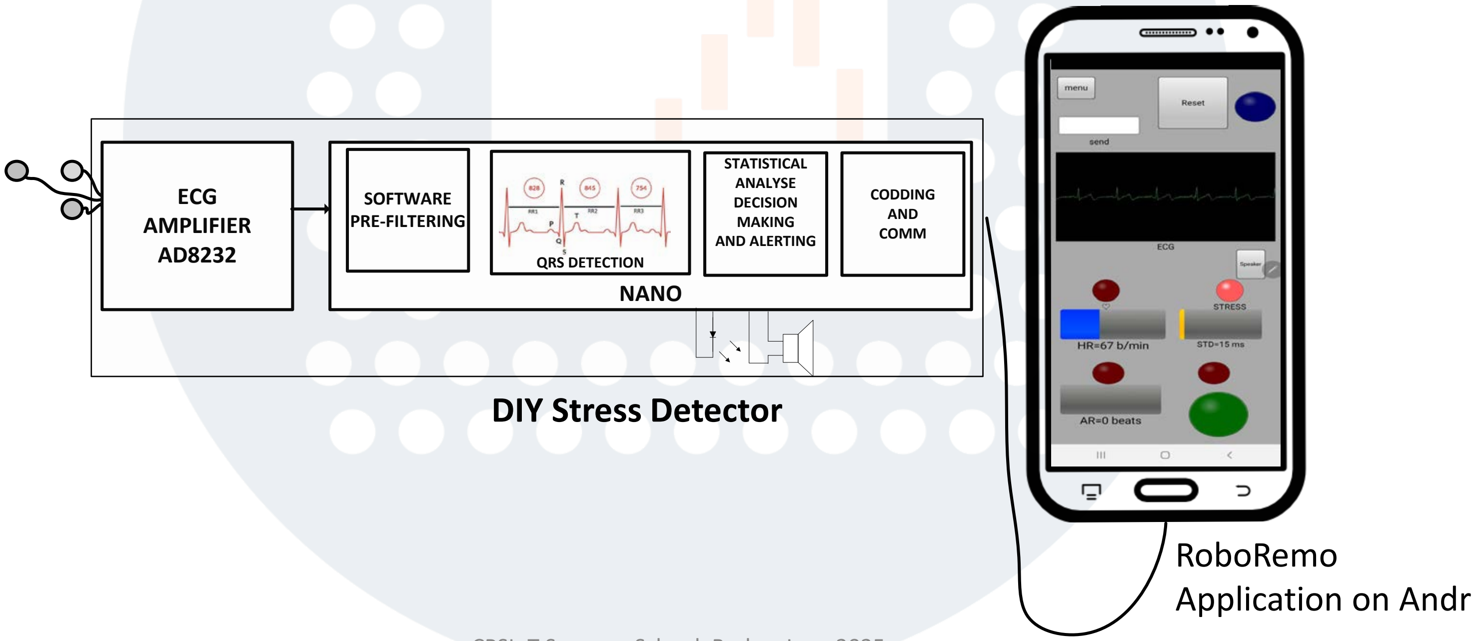
OHW/OSW





CI in BME...

- **Stress detector can work as** stand-alone device or connected to a gadget (android) application that is more comfortable way.



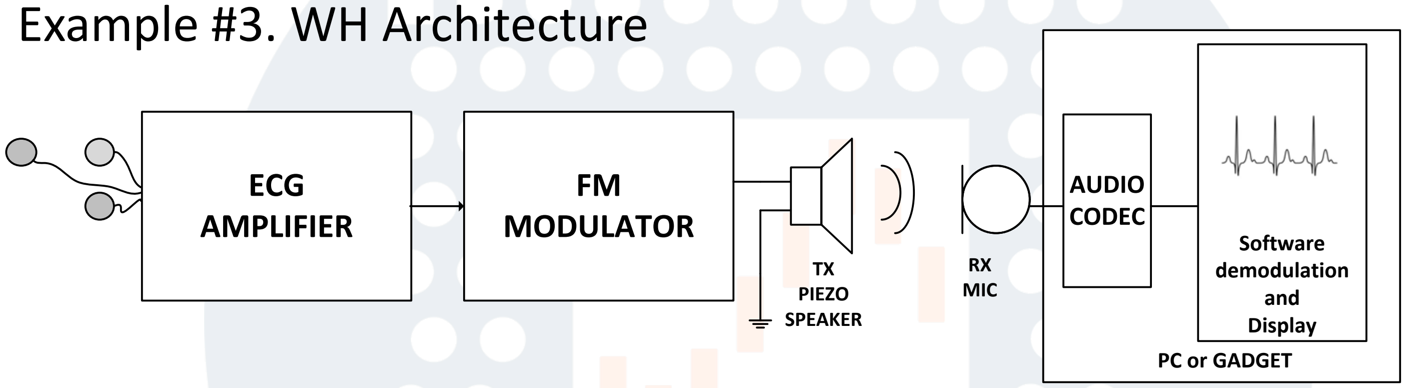
#CI in BME…

- **Example #3. "Whispering heart (WH)".** Medical data over sound. We send ECG signal in range 16KHz-18KHz. The signal is simultaneously received and displayed on several devices: OHW/OSW #1, standard android which runs Spectroid APK (by Carl Reinke), OHW/OSW #2, Android + custom made JavaScript+HTML+CSS APK, OHW/OSW #3, PC + custom made JavaScript+HTML+CSS APK. Zero setting and platform independent approach

CPSIoT Summer School, Budva, June 2025

CI in BME…

- Example #3. WH Architecture



- FM signal modulation in the near ultrasonic range. Then, the signal is observed in the time, frequency, and time-frequency domains using host software implemented in MATLAB or JavaScript
- Different ways of demodulation including Hilbert transform and discriminator techniques.

CI in BME…

- **Example #4. CardiaTalker. The system that integrate ChatGPT in monitoring cardiac illnesses**
  - Cardiovascular diseases (CVDs) remain the leading global cause of death.
  - Electrocardiography (ECG), remain most common used non-invasive to assess cardiac functions in term of rhythm based disorders as arrhythmia, bradycardia, tachycardia, pre-infraction and pre-strokes states.
  - Cardio wearables are transforming personal and preventive heart health care, and the trend is accelerating fast. There are plenty of such devices and holters, loop recorders, hand held monitors and watches are commonly used.

https://www.ccjm.org/content/86/7/483

CI in BME…

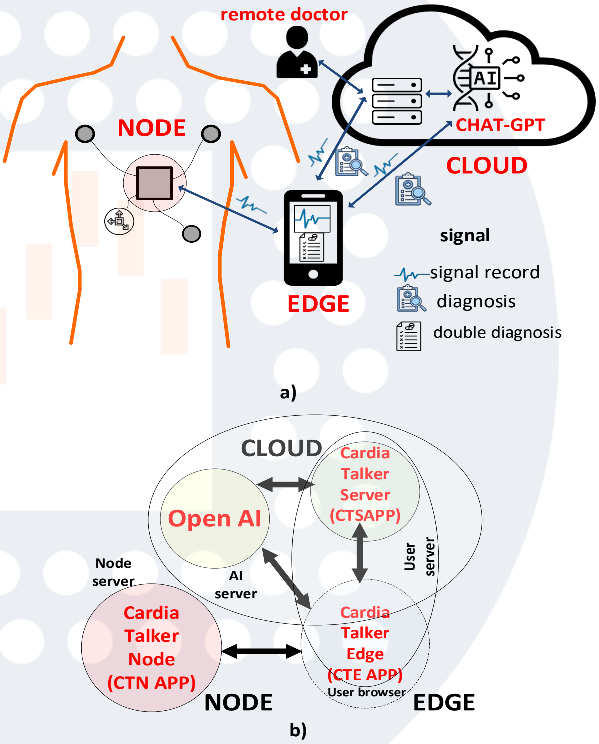
- Commercial Holter or Loop recorder devices. Device is logging data, which are later off-line read and analyzed in PC based application and can be sent to remote doctor

Source: https://getwellue.com/products/12-lead-holter-monitor

CPSIoT Summer School, Budva, June 2025
MECO 2025, Djurkovi et all

CI in BME...

- **MECOnet CardiaTalker (CaTa).** Feasible AI supported system for cardia disorders monitoring
- With current state of the art in IoT and AI can be easy implemented.
- MECOnet developed CardiaTalker concept

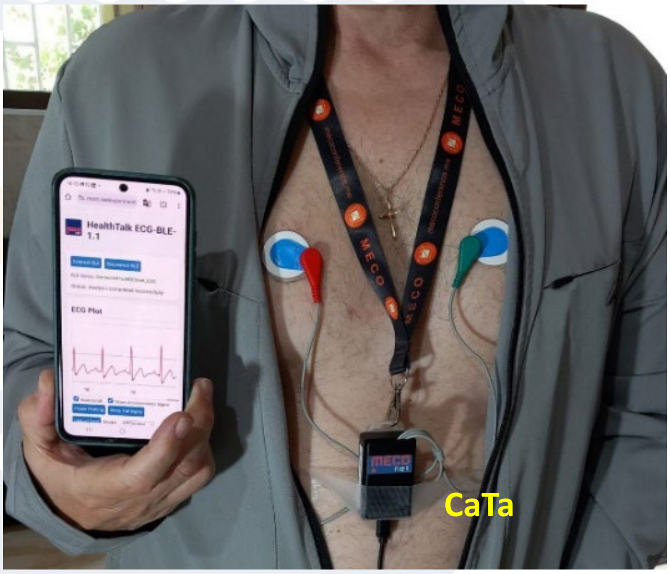


The system architecture of CardiaTalker, a) overall system components, b) application layer architecture.

CPSIoT Summer School, Budva, June 2025
MECO 2025, Djurkovi et all

CI in BME…

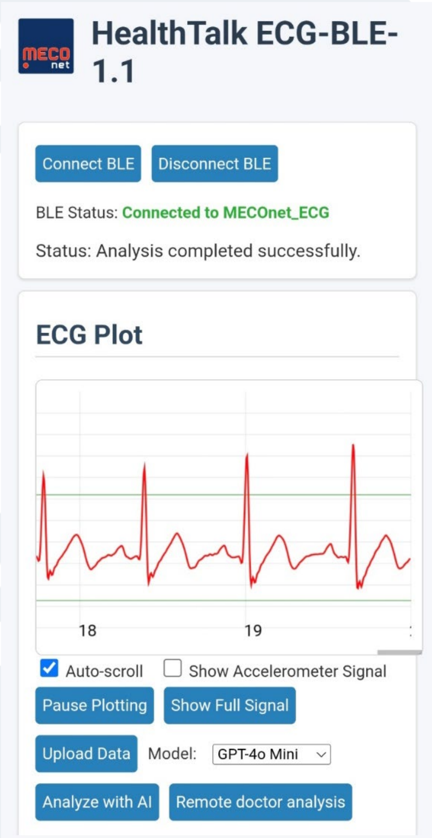
- **CaTa** is attached to the patient's body, records the ECG signal and communicates with the mobile phone application and further to the cloud (**OHW/OSW**).
- **CaTa** Provide double diagnosis from ChatGPT and remote doctor.
- **CaTa** has hardware and 3 layers software, firmware, Edge, Cloud



CPSIoT Summer School, Budva, June 2025
MECO 2025, Djurkovi et all

CI in BME…

Parameter	Specification / Result
Microcontroller Unit (MCU)	ESP32 (dual-core, Wi-Fi + BLE, low-power operation)
ECG Module	AD8232 (analog front-end for biopotential measurement)
Accelerometer (ACC)	Integrated DXL335 3-axis accelerometer
Power Source	5000 mAh Li-ion power bank
Battery Life	>60 hours of continuous operation
ADC Resolution / Sampling Rate	12-bit resolution, 200 Hz sampling frequency
Average Power Consumption	~75 mA
Edge Communication Protocol	BLE (Bluetooth Low Energy)
Cloud Communication Protocol	Wi-Fi (HTTP POST to REST API)
Edge Communication Range	~10 meters indoor with >98% packet reception rate
Cloud Upload Time (30s ECG)	~2 seconds from CET App to CST App
OpenAI Response Time (60s ECG, gpt-4o-mini model)	~40 seconds (includes upload, processing, and return)
OpenAI Response Time (60s ECG, gpt-4-turbo)	~19 seconds (includes upload, processing, and return)



CPSIoT Summer School, Budva, June 2025
MECO 2025, Djurkovi et all

CI in BME…

AI Interpretation Quality

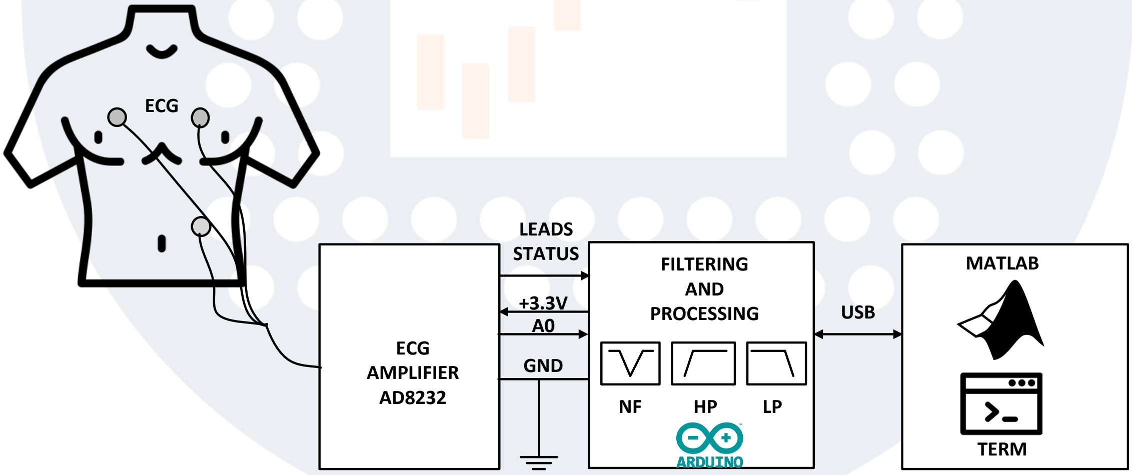
- ChatGPT consistently provided coherent interpretations for well-formatted ECG prompts, identifying conditions like "normal rhythm" or "possible irregularity"
- Still it is not reliable in determining amplitude and time characteristics of typical segments of ECG signal as P-amp[mV], PR-dur[s], R-amp[mV], QT-dur[s], Q-amp[mV], ST-dur[s], T-amp[mV], QRS-dur[mV], S-amp[mV], RR-dur-av[s]. It does not even dare to determine the HR from the data available, for known selection frequency.
- Evaluations with ST segment elevation should be taken with a reserve, because it still does not have a mathematical model for the global evaluation of the ECG signal, not to mention the evaluation in the vicinity of the point.
- Assessment of arrhythmias can be done at the level of assessment of rhythm regularity, without going into the type of arrhythmias. In some testing it can not detect tachycardia between 120-150 bps as well as brachicardia of 30 bps. Irregular rhythm on basis of ectopic beat can be detected.
- Although slower gpt-4o-mini shown better accuracy than gpt-4-turbo.
- It is very important to well define input ECG record, recommended time and mVs stamps as well as better description of the input parameters.

CI in BME…

- **MedWearables from MEConet,**
- https://meconet.me/smarthealth/
- Syntrofos – A headset like wearable device to track COVID-19 symptoms
- Wireless and hands free vital signs monitors
- Portable monitors, stress and arrhythmia analyzers
- "Whispering heart", proximity contactless, browser-based ECG analyzer
- Wireless pulse oximeter monitor
- Fall detection using beacon and node MCU
- True measurement of blood pressure
- HRV analysis and arrhythmia detection
- https://github.com/orgs/MEConet-Code/repositories

CI in BME…

- **Exercise. Acquisition, Filtering, Communication and Visualization of ECG signal, docs and files**
- **How to design the acquisition, software filtering, serial sending and visualization of ECG signal by using AD8232 ECG module, ARDUINO NANO and serial plotter or MATLAB GUI?**. Notch, HP and LP filtering are implemented in ARDUINO NANO.
- Observe the signals in Terminal-Plotter?
- Observe and process signal in the MATLAB? Implement other signsl processing in Matlab.



CI in BME...

- The effect of the software filtering by cascade Notch (50Hz) -> LP(25Hz) -> HP(0.5Hz), implemented in ATMega328 and observed in free version of the serial oscilloscope, https://x-io.co.uk/serial-oscilloscope/

Conclusions

- **Biomedical engineering combined with ICT, CPS, IoT and AI is the technology of the future.**
- **Embedded, cyber-physical and networked systems, wearables, are intensively used in medicine and home health care.**
- **Their market is growing every day and they are considered one of the technologies of the future.**
- **Their design is the result of a wide range of knowledge in analog, digital and telecommunication electronics, as well as programming.**
- **Designing can be simplified and made cheaper, while maintaining good system performance, by integrating existing hardware-software modules and techniques and modules designed by designers (where necessary) as well as open source technologies.**
- **#CreativeIntegration (#CI) or #DIY4.0 principle in BME is introduced that shows Do-It-Yourself (DIY) culture is not outdated, it's just changing with the times. It's smarter, digital, open-source, and connected.**
- **Several examples of #CI designs are given for educational and professional purposes.**
- **The mobile phone and any gadgets are proven to be a very powerful tool in wearables, as it can perform a large number of tasks in low cost.**
- **Spot-Edge-Cloud is promising combination.**

References

1. J. Đurković, R. Stojanović, A. Škraba, M. Vukmirović, B. Babić and V. Miranović, "Cardiac Rhythm Monitoring with ChatGPT Integration," 2025 14th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2025, pp. 1-4, doi: 10.1109/MECO66322.2025.11049165.

2. R. Stojanović, J. Đurković, M. Vukmirović, B. Babić and V. Miranović, "Medical Data Over Sound - HealthTalk Concept," 2025 14th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2025, pp. 1-5, doi: 10.1109/MECO66322.2025.11049300.

3. R. Stojanović, J. Djurković and A. Škraba, "ECG and PPG Signals Monitoring by Using Web Audio API," 2024 13th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2024, pp. 1-6, doi: 10.1109/MECO62516.2024.10577786.

4. R. Stojanović, J. Djurković, B. Babić, V. N. Ivanović, B. Lutovac and M. Stork, "A Toolset for Blood Pressure Visualization and Measurement in Time, Frequency and Time-Frequency Domains," 2024 13th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2024, pp. 1-6, doi: 10.1109/MECO62516.2024.10577773.

5. Djurkovic, J., Stojanović, R., & Cico, B. (2023). An Experimental Platform for Fall Detection Using Beacon, Node MCU and MATLAB. WiPiEC Journal - Works in Progress in Embedded Computing Journal, 9(2). Retrieved from https://wipiec.digitalheritage.me/index.php/wipiecjournal/article/view/46

6. Jóźwiak, L., & Stojanovic, R. (2024). Proceedings of the 5th Summer School on Cyber-Physical Systems and Internet-of-Things, Vol. V, 2024 (1.1) [Computer software]. 5th Summer School on Cyber-Physical Systems and Internet-of-Things, Budva, Montenegro. Zenodo. https://doi.org/10.5281/zenodo.12608092

7. Lech Jóźwiak, Radovan Stojanovic, & Christos Antonopoulos. (2023). Proceedings of the 4th Summer School on Cyber-Physical Systems and Internet-of-Things, Vol. IV, 2023 (1.0) [Computer software]. 4th Summer School on Cyber-Physical Systems and Internet-of-Things (SS-CPSIoT2023), Budva, Montenegro. Zenodo. https://doi.org/10.5281/zenodo.8113313

8. https://meconet.me/smarthealth/

CPSIoT Summer School, Budva, June 2025

Thank you, questions, comments?

CPSIoT Summer School, Budva, June 2025

CPS&IoT'2025 Summer School on Cyber-Physical Systems and Internet-of-Things

Budva, Montenegro, June 10-14, 2025

Schedule

Day 1, Tuesday 10June:
09:00:10:00 Wolfgang Ecker, Infineon and TU Munich, DE
Keynote: **How to Circumvent a RISCV Dead End**
10:00-10:30 Event Chairs and Special Guests
Title: Opening Ceremony of the CPS&IoT'2025 Summer School, and MECO'2025 and CPS&IoT'2025 Conferences
10.30-11.00 Break
11.00-13.00 Lech Jóźwiak, TU/e, **NL**
Title: Green CPS and IoT for Green World
13.00-14.00 Lunch Break
14.00-15.30 Said Hamdioui, TU Delft, **NL**
Title: The Future of Energy-Efficient Computing: Beyond CMOS and Von Neumann
15.30-17.00 Natalie Simson & Wolfgang Ecker, **DE**
Title: RISCV vs. ARM
17.00-17.30 Break
17.30-19.00 Natalie Simson & Wolfgang Ecker, **DE**
Title: RISCV vs. ARM

Day 2, Wednesday 11 June:
09.30-10.30 Reiner John, **AVL**, **AT**
Title: **Keynote**: **AI+ — Self-Awareness as the Key to Resilience, Efficiency, and ROI**
10.30-11.00 Break
11.00-12.30 Reiner John, AVL, **AT**
Title: Applications of Quantum Sensing and AI in Real-Time Edge Computing Systems (lecture/hands-on)
12.30-14.00 Lunch Break
14.00-17.00 Nabil Abdennadher, Univ. of Applied Sciences, West. Switzerland, **CH**
Title: Towards a decentralized Federated Learning based edge-to-edge Continuum Computing Framework
17.00-17.30 Break
17.30-19.00 Iryna DE ALBUQUERQUE SILVA, CEA, **FR**
Title: Aidge: Embedded AI Open Source Platform
21.00 Gala Dinner

Day 3, Thursday 12 June:
09.00-10.30 Samir OUCHANI, CESI**, FR**
Title: Designing Trustworthy Smart Systems: Bridging Artificial Intelligence and Formal Methods
10.30-11.00 Break
11.00-12.30 Zakaria Chihani, CEA, **FR**
Title: Trustworthy AI: Methods & Tools
12.30-13.30 Lech Jóźwiak, TU/e, **NL**
Title: Quality-driven Design of Cyber-Physical Systems
13.30-14.30 Lunch Break
14.30-16.30 Hans Vangheluwe and Joachim Denil, University of Antwerp, **BE**
Title: General systems engineering concepts (for CPS) and General modelling and simulation concepts
16.30-17.30 Joachim Denil, University of Antwerp, **BE**
Title: Requirements Modelling for Testing and Run-time Monitoring
17.30-18.00 Break
18.00-19.00 Rakshit Mittal and Hans Vangheluwe, University of Antwerp, **BE**
Title: Modelling and simulation of physical systems using OpenModelica (lecture/hands-on)

Day 4, Friday 13 June:
09.00-10.00 Rakshit Mittal and Hans Vangheluwe, University of Antwerp, **BE**
Title: Calibration and control of physical systems using OpenModelica (lecture/hands-on)
10.00-12.00 Joeri Exelmans and Hans Vangheluwe, University of Antwerp, **BE**
Title: Modelling, simulation, and synthesis of discrete-event systems using Statecharts (lecture/hands-on)
12.00-13.00 Lunch Break
13.00-16.30 Anish Bhobe, Dominique Blouin, and Yara Hallak, Telecom Paris, **FR**
Title: Modelling, verification, and synthesis of embedded cyber-physical systems using AADL and RAMSES (lecture/hands-on)
16.30-17.00 Break
17.00-18.30 Radovan Stojanovic, University of Montenegro and MECOnet, **ME**
Title: Design of performance and energy efficient nodes for smart systems
18.30-19.00 Closing of the CPS&IoT'2025 Summer School

+ Free participation in sessions of the CPS&IoT'2025 Conference and MECO'2025 Conference
Summer School participants are expected to come with their own laptops. Internet access will be guaranteed.

Day 5, Saturday 14 June: Excursion possible (excursion fee is not included in the summer school fee)

Summer School on CPS&IoT 2025 – Attendees

#	Students	Country	Affiliation
1	Aleksandar Krivokapić	Serbia	University of Novi Sad
2	Hichame Ben Youssef	France	Telecom Paris
3	Barbara da Silva Oliveira	France	INRIA
4	Koushik Chakraborty	USA	Utah State University
5	Sanghamitra Roy	USA	Utah State University
6	Dražen Jurišić	Croatia	University of Zagreb
7	Marina Bulat	Serbia	University of Novi Sad
8	Martin Shameti	Albania	Epoka University
9	Klei Qorri	Albania	Epoka University
10	Kristina Tomović	Montenegro	DeNe Deutsche Netzbau
11	Nebojša Ivanović	Montenegro	University of Montenegro
#	**Lecturers**	**Country**	**Affiliation**
1	Lech Jóźwiak	Netherlands	TU/e
2	Radovan Stojanović	Montenegro	University of Montenegro
3	Said Hamdioui	Netherlands	TU Delft
4	Natalie Simson	Germany	Infineon Technologies AG and Technical University of Munich
5	Wolfgang Ecker	Germany	Infineon Technologies AG and Technical University of Munich
6	Reiner John	Austria	AVL
7	Florian Lorber	Austria	AVL
8	Selim Solmaz	Austria	AVL
9	Nabil Abdennadher	Switzerland	Univ. of Applied Sciences, West. Switzerland
10	Iryna de Albuquerque Silva	France	CEA
11	Samir Ouchani	France	CESI
12	Zakaria Chihani	France	CEA
13	Hans Vangheluwe	Belgium	University of Antwerp
14	Joachim Denil	Belgium	University of Antwerp
15	Rakshit Mittal	Belgium	University of Antwerp
16	Joeri Exelmans	Belgium	University of Antwerp
17	Simon Van Mierlo	Belgium	University of Antwerp
18	Anish Bhobe	France	Telecom Paris
19	Dominique Blouin	France	Telecom Paris
20	Yara Hallak	France	Telecom Paris
21	Jovan Djurković	Montenegro	MECOnet

Certificate of Attendance

THIS ACKNOWLEDGES THAT

Marko Markovic

MONTENEGRO

MONTENEGRIN ASSOCIATION FOR NEW TECHNOLOGIES – MANT

has successfully attended and completed:

The 6th Summer School on

Cyber Physical Systems and Internet of Things (SS-CPSIoT'2025)

(3 ECTS)

in Budva, Montenegro, June 10-14 2025

On behalf of the organizers:

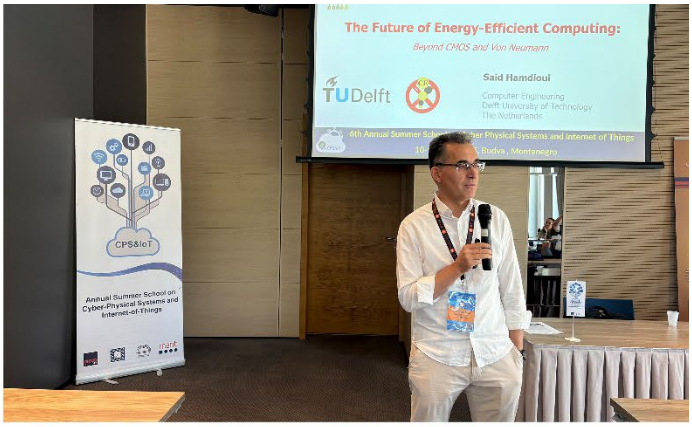
Prof. dr. Lech Jozwiak

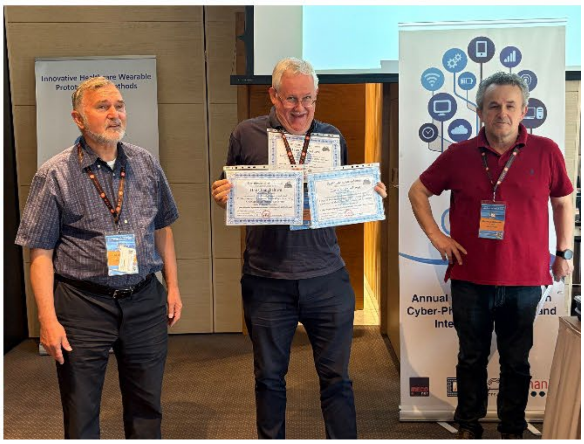
Prof. dr. Radovan Stojanović

Prof. dr. Betim Cicc

Author Index

SS-CPS&IoT 2025 Gallery





CPS&IoT'2025 6th Summer School on Cyber-Physical Systems and Internet-of-Things

Budva, Montenegro, June 10-14, 2025

Citation:

Author/s, "Title of contribution-presentation", in Proceedings of the 6th Summer School on Cyber-Physical Systems and Internet-of Things, Editors: Lech Jozwiak and Radovan Stojanovic, Authors: Anish Bhobe, Dominique Blouin, Florian Lorber, Hans Vangheluwe, Iryna de Albuquerque Silva, Joachim Denil, Joeri Exelmans, Jovan Djurković, Lech Jóźwiak, Nabil Abdennadher, Natalie Simson, Radovan Stojanović, Rakshit Mittal, Reiner John, Said Hamdioui, Selim Solmaz, Simon Van Mierlo, Wolfgang Ecker, Yara Hallak and Zakaria Chihani, Vol. VI, June 2025, pp. xx-yy, DOI: https://doi.org/10.5281/zenodo.16102824.

Technical editors:

Prof. dr Budimir Lutovac, University of Montenegro
Jovan Djurković, MECOnet

Publishers:

MECOnet Institute (Mediterranean Excellence in Computing and Ontology),
https://meconet.me

Montenegrin Association for New Technologies – MANT, https://mant.me

Place/Year: Podgorica, Montenegro, 2025