



# #18: Increasingly-Autonomous CPS: Taming Emergent Behaviors from an Architectural Perspective

**Jérôme Hugues** [jjhugues@sei.cmu.edu](mailto:jjhugues@sei.cmu.edu)

**Daniela Cancila** [Daniela.CANCILA@cea.fr](mailto:Daniela.CANCILA@cea.fr)

Software Engineering Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213

Copyright 2022 Carnegie Mellon University and CEA LIST.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

DM22-0628

# The IA/AI fresh breeze ...

General enthusiasm on **Increasingly-Autonomous** CPS [1] to

- Improve system efficiency (decrease # operators), system capability (automate high-level tasks), faster than human, ..

**Increasingly-Autonomous** Systems embeds advanced “intelligent” capabilities, from basic control to advanced **AI**

“Intelligence”



Autonomous  
Models, RL, ML



Automated  
Rule-Based, stochastic



Automatic  
(PID, Hysteresis)

[1] E. E. Alves, B. Devesh, B. Hall, K. Driscoll, A. Murugesan, and J. Rushby. Considerations in Assuring Safety of Increasingly Autonomous Systems. Technical Report NASA/CR-2018-220080, NF1676L-30426, NASA AIR TRANSPORTATION AND SAFETY, 2018.

Notation **Increasingly-Autonomous** → IA [1]

**Artificial Intelligence** → AI

Adapted from USD (R&E) Autonomy  
roadmap 2019 Webinar p2

# The IA/AI fresh breeze .. coming from an iceberg ..

General enthusiasm on **Increasingly-Autonomous** CPS [1] to

- Improve system efficiency (decrease # operators), system capability (automate high-level tasks), faster than human, ..

**Increasingly-Autonomous** Systems embeds advanced “intelligent” capabilities, from basic control to advanced **AI**

But fast pace of actions, poorly-defined safety mechanisms makes it impossible for a human to mitigate issues

⇒ Distrust in system, longer V&V, or capability not deployed

⇒ May jeopardize capabilities of future projects

## How to demonstrate that a system is safe?

[1] E. E. Alves, B. Devesh, B. Hall, K. Driscoll, A. Murugesan, and J. Rushby. Considerations in Assuring Safety of Increasingly Autonomous Systems. Technical Report NASA/CR-2018-220080, NF1676L-30426, NASA AIR TRANSPORTATION AND SAFETY, 2018.

Notation **Increasingly-Autonomous** → IA [1]

**Artificial Intelligence** → AI



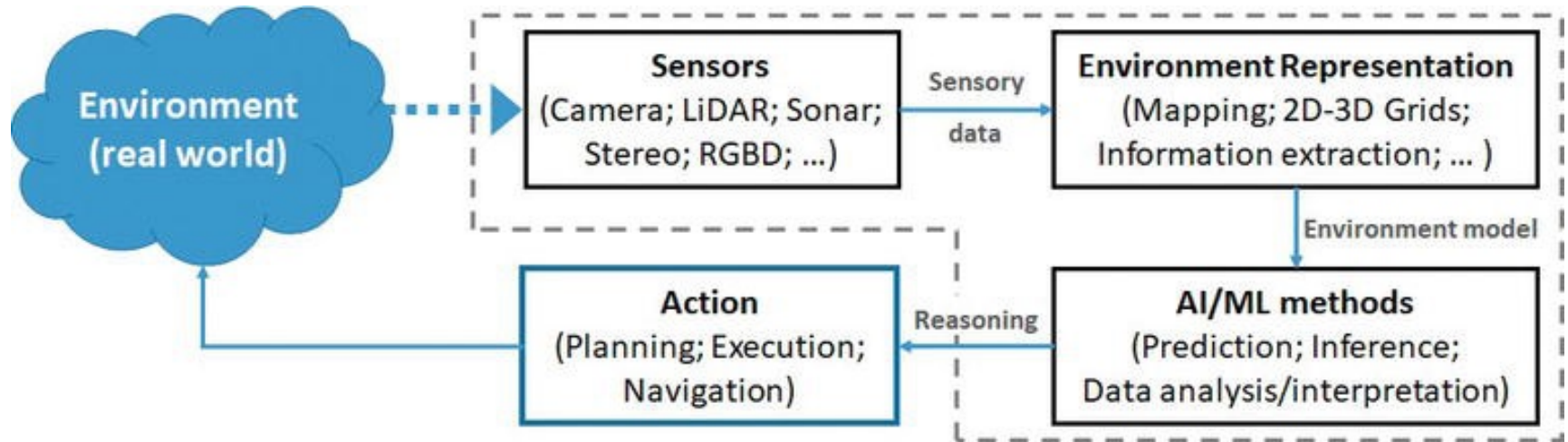
Image: CC BY-SA 3.0, Uwe Kils & Wiska Bodo, [File:Iceberg.jpg - Wikimedia Commons](https://commons.wikimedia.org/wiki/File:Iceberg.jpg), 2005.

# Architecture, Faults & IA-CPS

Notation

Increasingly-Autonomous → IA

Artificial Intelligence → AI



Adapted from DOI: 10.5772/intechopen.79742

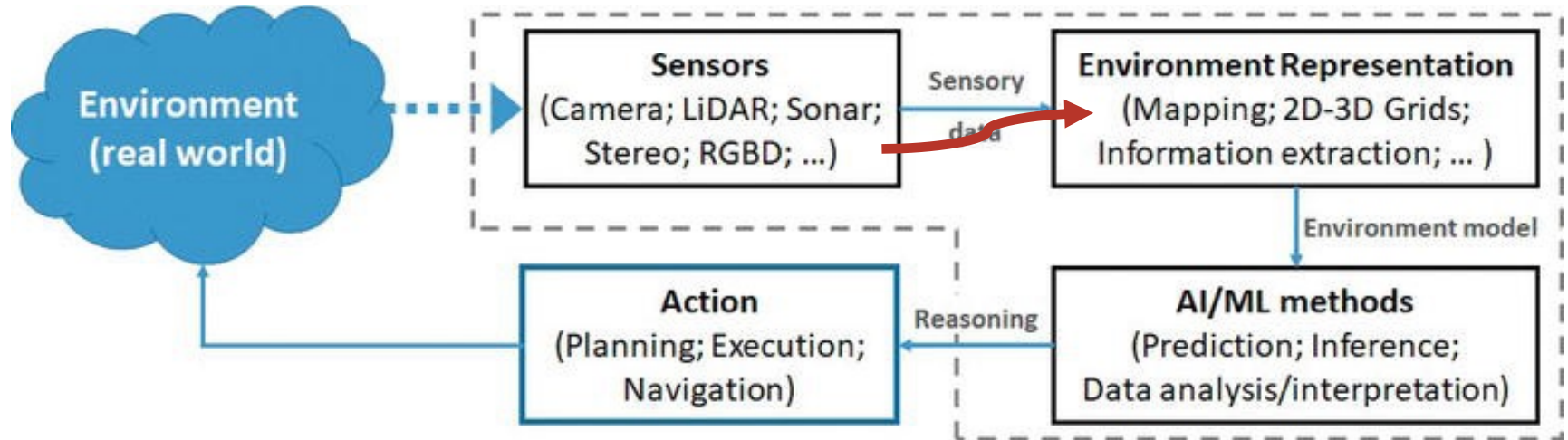
# Architecture, Faults & IA-CPS

Notation

Increasingly-Autonomous → IA

Artificial Intelligence → AI

Timing? Value?



Adapted from DOI: 10.5772/intechopen.79742

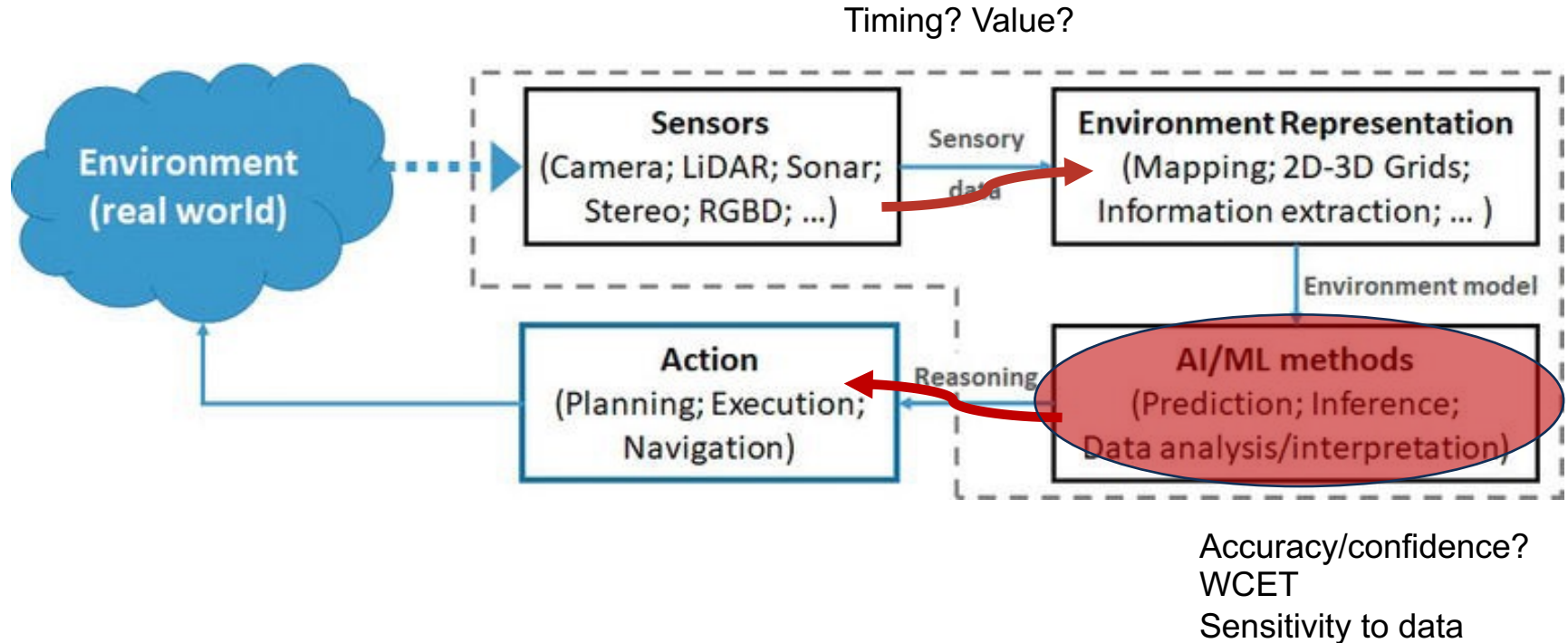


# Architecture, Faults & IA-CPS

Notation

Increasingly-Autonomous → IA

Artificial Intelligence → AI



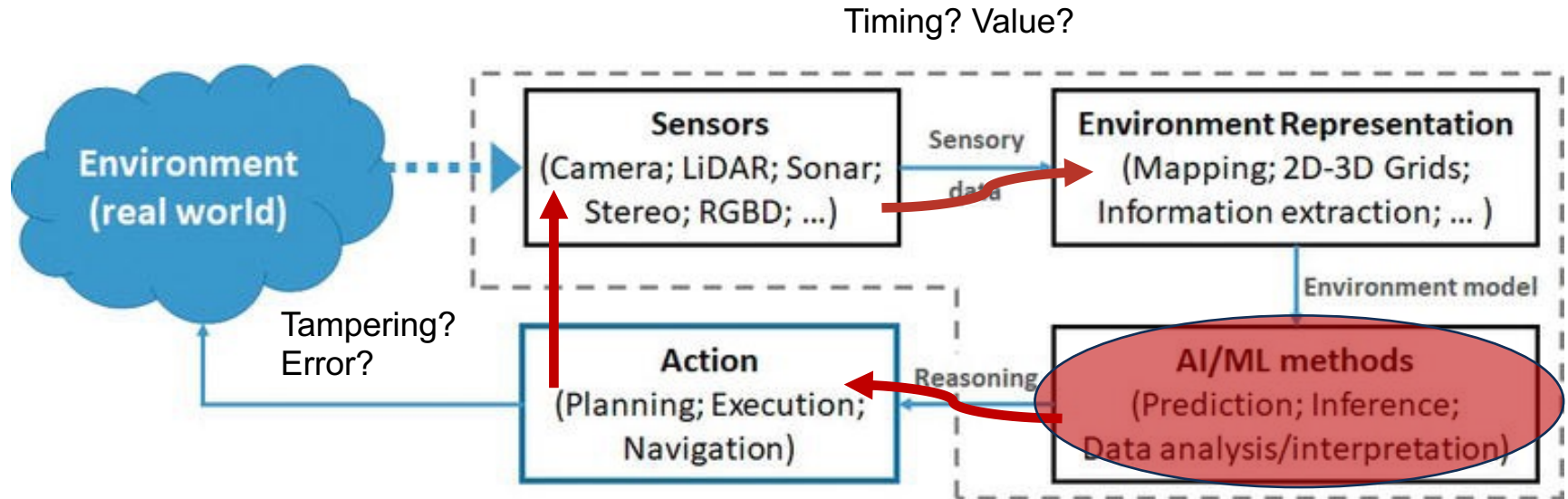
Adapted from DOI: 10.5772/intechopen.79742

# Architecture, Faults & IA-CPS

Notation

Increasingly-Autonomous → IA

Artificial Intelligence → AI



**Claim #1:** the IA/CPS boundary gives rise to emergent behaviors

Accuracy/confidence?  
WCET  
Sensitivity to data

Adapted from DOI: 10.5772/intechopen.79742



# Example of emergent behaviors and complexity

## Notation

Increasingly-Autonomous → IA for short

Artificial Intelligence → AI

Emergent behavior characterized in CPS and AI,

- what about on Increasingly-Autonomous/CPS boundary?

Bad resource management: CPS → AI

- Timing violations, jitter, non determinism may violate assumptions made by IA controllers
- Memory, bandwidth, battery scarcity may impede some functions

Bad resource management: AI → CPS

- Conversely: variability in resource utilization by IA functions negatively impacts system

Cybersecurity and Cyber-Physical Security: AI ↔ CPS

- RL-based systems can help detecting attacks or faults ; yet they introduce another layer of incidental complexity in system design

# From Safety Assessment to Resilience Assurance

Notation

Increasingly-Autonomous → IA for short

Artificial Intelligence → AI

Previous examples illustrate the impossibility to perform a safety assessment

Difficulty to

- describe the requirements of AI-based functions  
Except through examples ...
- the worst-case budget used by the function
- test all conditions, etc.

Safety evaluation becomes impossible, we claim one must concentrate on defining resilient architecture and strategies to assure system resilience by leveraging

- Patterns for resilience systems, e.g. power plant
- Model-Based Architecture Description Languages + V&V capabilities
- System-theoretic approaches to elicit hazardous scenarios and their mitigation
- + built-in fail-safe strategies whenever possible