

Extending AADL for Security Design Assurance of the Internet of Things

Important decisions that establish security in a system are made in the architecture.

Formal modeling provides a means to continually verify that design and code changes are consistent with security requirements.

We extended the core modeling concepts of AADL with security properties, to formally model architectural properties relevant to security (e.g., AccessMode, AccessGroup).

To drive the analysis we first established a set of threats. Our threat analysis was guided by Microsoft's STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privilege) model.

To reason about the satisfaction of security properties in a system architecture we need to:

1. Specify the properties, and the associated architectural elements in AADL; and then
2. Analyze claims over those properties.

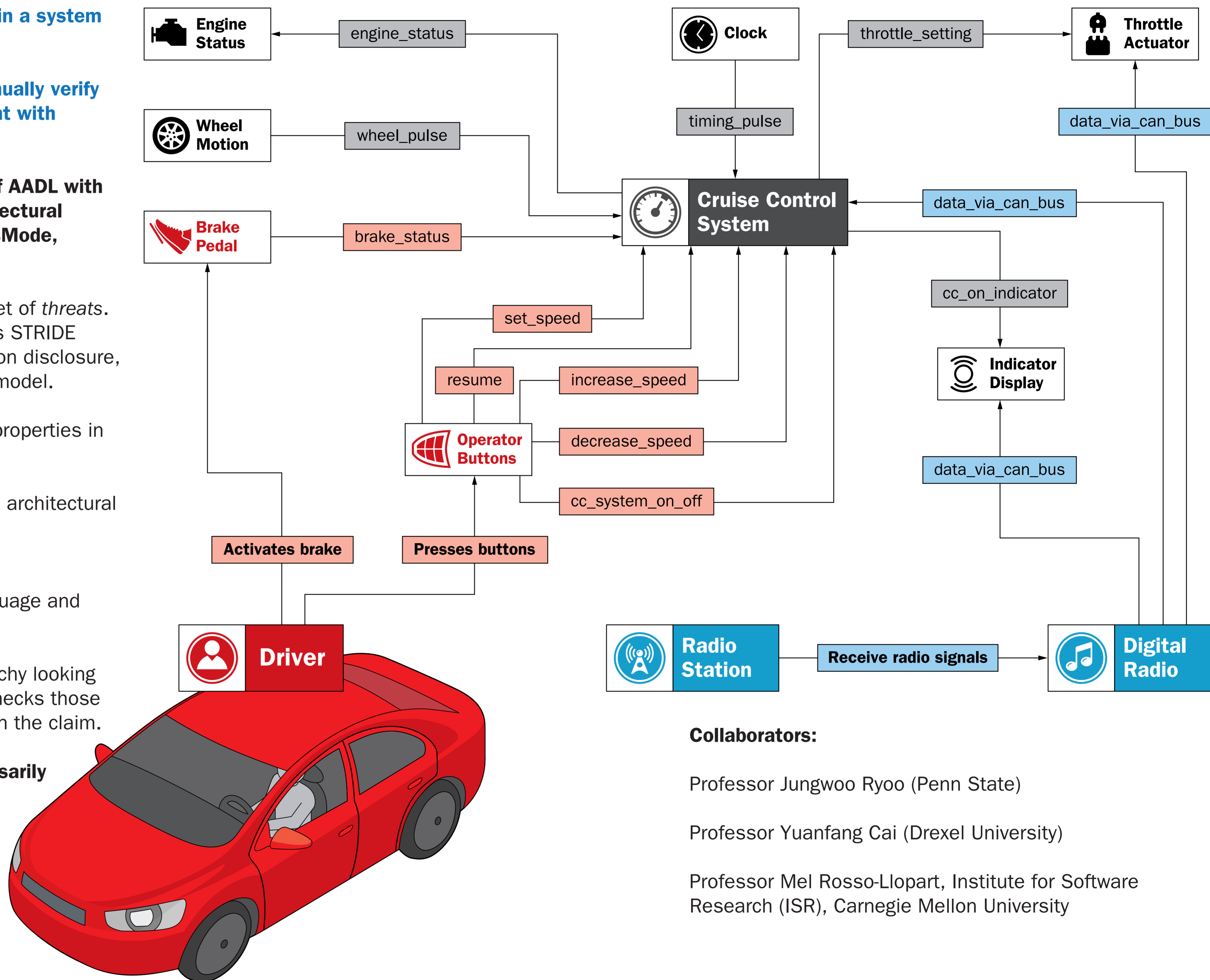
To analyze claims we used the Resolute language and model-checker.

Resolute walks the instantiated model hierarchy looking for components specified in its claims and checks those components according to the logic encoded in the claim.

Limitations: An AADL model does not necessarily support security validation. Is the system sufficiently secure for the planned usage?

Possibilities:

- An external weakness may enable an attacker to operate outside the model.
- The specifications may not provide the desired level of security assurance.



Collaborators:

Professor Jungwoo Ryoo (Penn State)

Professor Yuanfang Cai (Drexel University)

Professor Mel Rosso-Llopart, Institute for Software Research (ISR), Carnegie Mellon University

Contact: Rick Kazman and Carol Woody rkazman@sei.cmu.edu, cwoody@cert.org

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

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.

This material has been approved for public release and unlimited distribution except as restricted below.

Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use:* 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 external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

* These restrictions do not apply to U.S. government entities.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM-0002946