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:

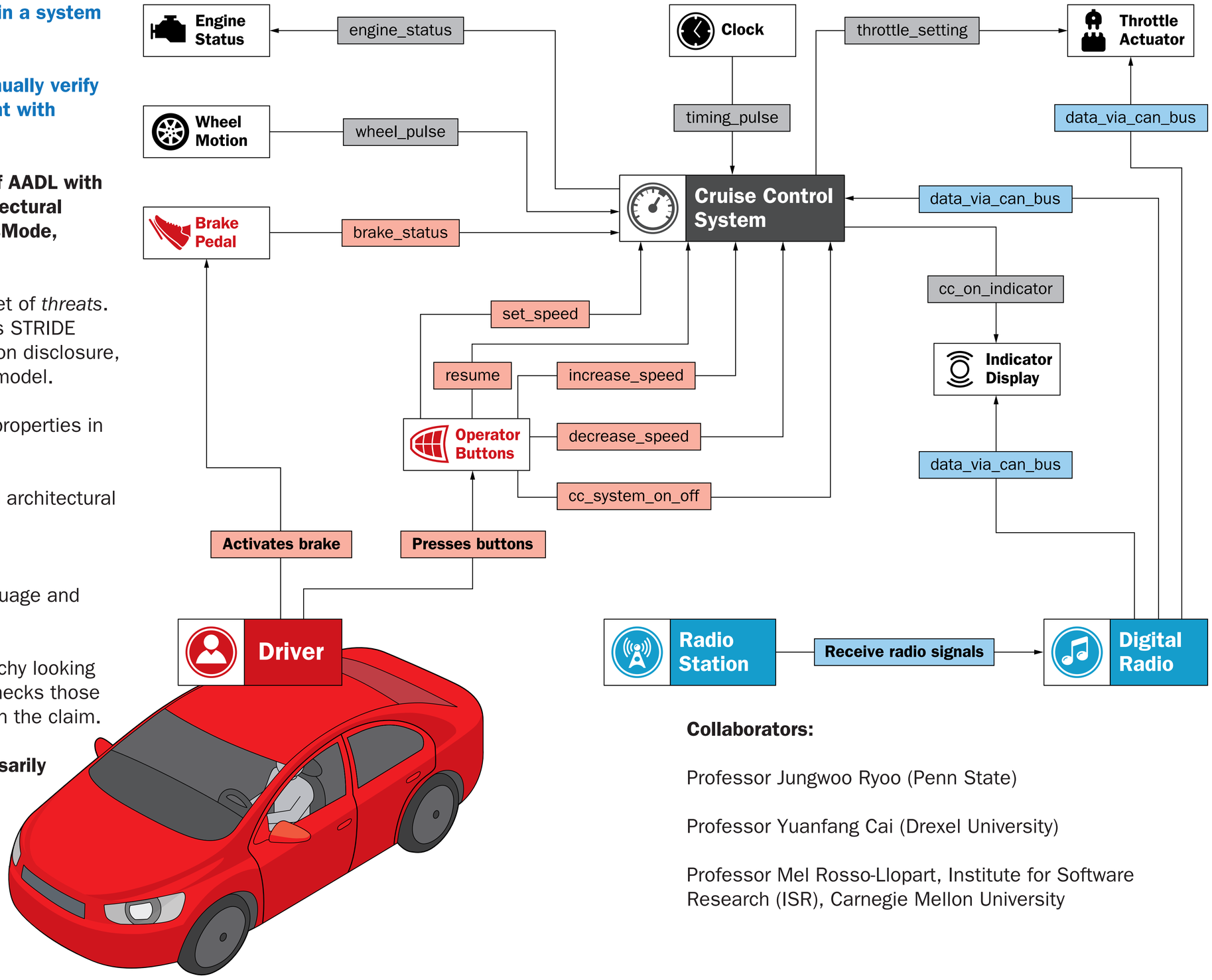
1. Specify the properties, and the associated architectural elements in AADL; and then
2. Analyze claims over those properties.

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.

DM-0002946