HD-Sec: Holistic Design of Secure Systems on Capability Hardware

https://hd-sec.github.io

DSbD
Digital Security by Design

Robert Thorburn, Asieh Salehi, Dana Dghaym, Michael Butler, Thai Son Hoang, Leonardo Aniello, Vladimiro Sassone
University of Southampton, UK {robert.thorburn, a.salehi-fathabadi, d.dghaym, m.j.butler, t.s.hoang, l.aniello, vsassone}@soton.ac.uk

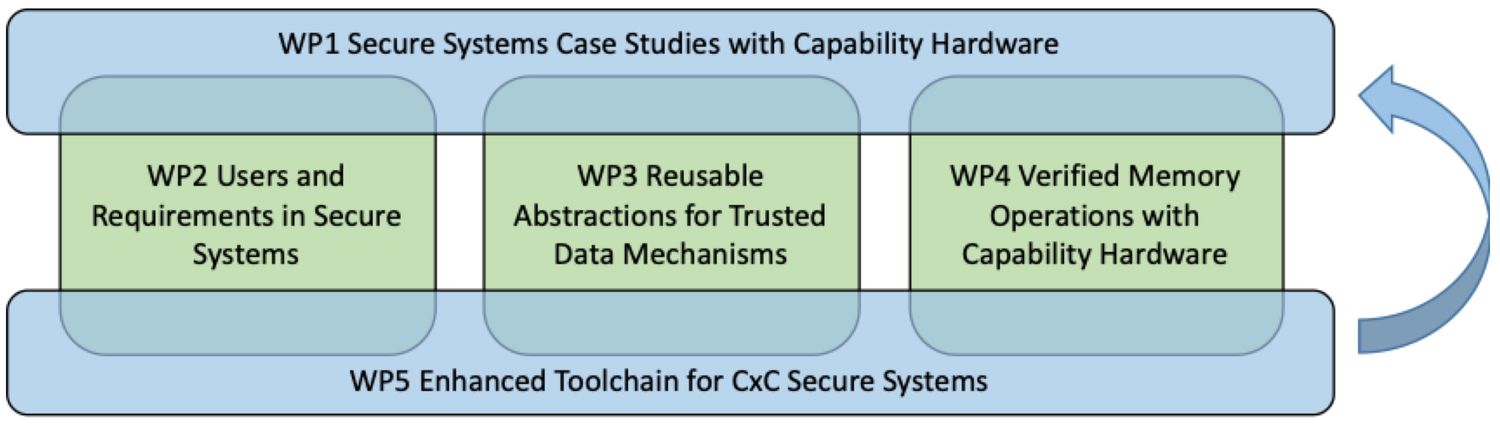
University of Southampton

UKRI Engineering and Physical Sciences Research Council

Overview

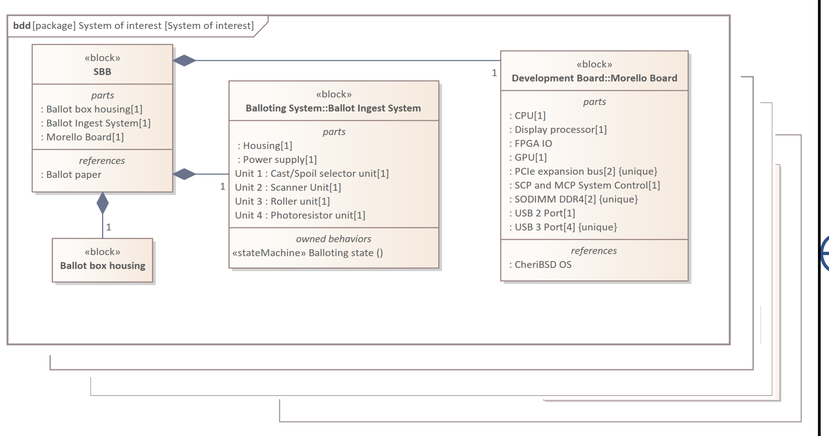
Transformation of security-critical software development

- From an expensive iterative test-and-fix approach
  - To a correctness-by-construction (CxC) approach
- The design of software from requirements to implementation:
  - Formal modelling, Reusable formal abstractions
  - Verification
  - Model transformation
  - CxC tools and running on capability hardware



Workflow of HD-Sec methodology: *Integration of **SysML, STRIDE, Formal Validation and Verification** and **Code Generation** to support Morello*

SysML System Model



Threat reporting

RIS

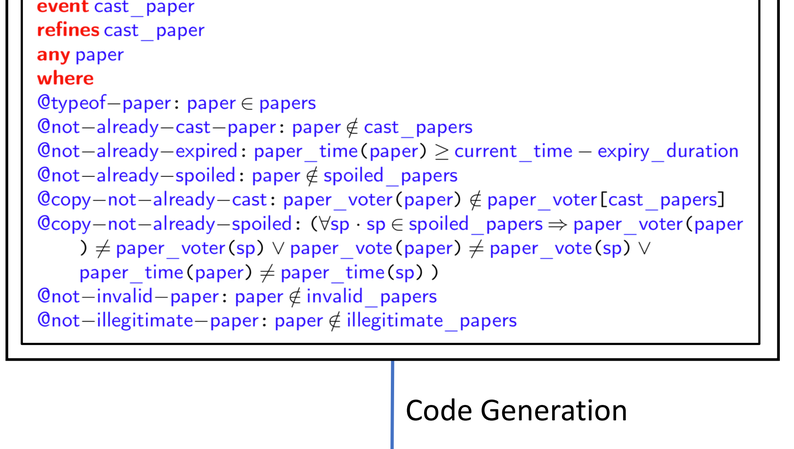
Event-B Formal Model (Validation & Verification)

```
event cast_paper
refines cast_paper
any paper
where
@typeof−paper: paper ∈ papers
@not−already−cast−paper: paper ∉ cast_papers
@not−already−expired: paper_time(paper) ≥ current_time − expiry_duration
@not−already−spoiled: paper ∉ spoiled_papers
@copy−not−already−cast: paper_voter(paper) ∉ paper_voter[cast_papers]
@copy−not−already−spoiled: (∀sp · sp ∈ spoiled_papers ⇒ paper_voter(paper
    ) ≠ paper_voter(sp) ∨ paper_vote(paper) ≠ paper_vote(sp) ∨
    paper_time(paper) ≠ paper_time(sp) )
@not−invalid−paper: paper ∉ invalid_papers
@not−illegitimate−paper: paper ∉ illegitimate_papers
```

Code Generation

Executable Code

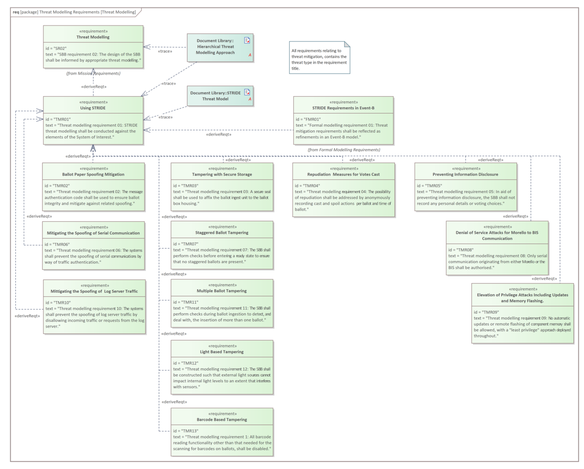
```
procedure cast(paper : in barcode) with
  Global => (Proof_In => (spoiled_arr, curr_time, spoil_count),
  In_Out => (cast_arr, cast_count)),
   Pre => ( Invariants(spoil_count,cast_arr)
         and then not already_cast(paper)
         and then not already_spoiled(paper)
         and then valid_time (paper)
         and then validate_barcode(paper)
         and then cast_count in 0 .. Max_Votes−1),
   Post => (Invariants(spoil_count,cast_arr)
        and then already_cast(paper)
        −− and then cast_arr(cast_ count) = paper
        and then cast_count = cast_count' old + 1);
```



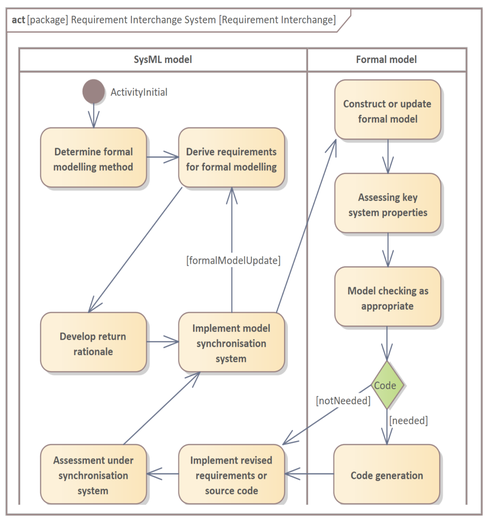
STRIDE Threat Modeling

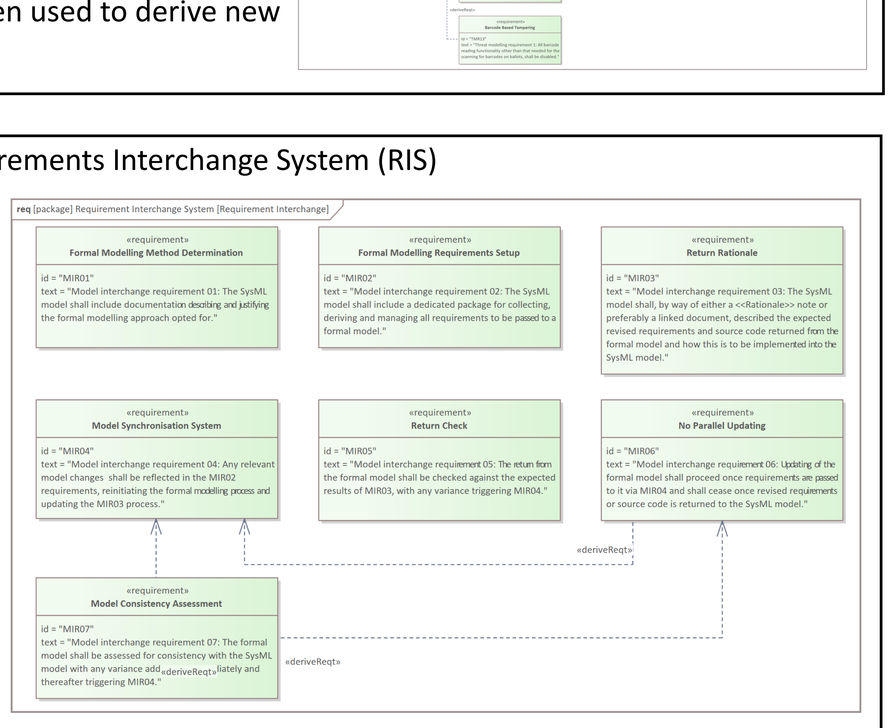
Conducting STRIDE threat modelling

- SysML requirements, derived from mission needs statement, establish and direct the need for threat modelling
- The SysML model exposes set model elements (system of interest) for threat modelling
- The SysML model, by way of a document repository, records the outcomes of the threat modelling
- Threat modelling outcomes are then used to derive new requirements



Requirements Interchange System (RIS)





Integration workflow

- Once STRIDE threat modelling is conducted and documented
  - Newly derived requirements are integrated into system model
  - Requirements are filtered via RIS and passed to Event-B model as appropriate
  - Verification and validation conducted in Event-B model
  - Feedback from Event-B to SysML model
- Iterative update process between SysML and Event-B models
- Structured via RIS, including expected outcomes and bounds of interchange
  - Which requirements are passed and which are not
- Ada Spark code generated from verified Event-B formal model
- Generated code executed on capability hardware

Current work

- Testing and expanding RIS as a standalone model library
- Investigating the effects of temporal memory safety on our codebase
  - A smaller codebase does not equal assurance
  - An environment under active development makes security a moving target
  - Potential actions of attackers -> future penetration testing
- Releasing code generation tool -> Spark Ada
- Preparing for code generation in not just C but Cheri C
- Testbed running CheriBSD main and dev branches on Morello
  - Includes a FreeBSD reference machine
  - Speeds up trouble shooting and allows for future comparative analysis
- Automated uptime and login reporting for Morello test bed in Teams

References

[1] Galois and Free & Fair. The BESSPIN Voting System (2019).
[2] Praxis: Tokeneer. https://www.adacore.com/tokeneer (2022).
[3] D. Dghaym, T.S. Hoang, M. Butler, R. Hu, L. Aniello, V. Sassone (2021) Verifying System-level Security of a Smart Ballot Box. In ABZ 2021- 8th International Conference on rigorous State Based Methods.
[4] E. Poorhadi, E. Troubitsyna, G. Dán (2022) Analysing the Impact of Security Attacks on Safety Using SysML and Event-B. In IMBSA 2022 - International Symposium on Model-Based Safety and Assessment.
[5] R.H. Thorburn, A. Salehi-Fathabadi, D. Dghaym, T.S. Hoang, M. Butler, L. Aniello, V. Sassone (2022) A Light-weight Approach to the Concurrent Use and Integration of SysML and Formal Methods in Systems Design. In ACM/IEEE 25th International Conference on Model Driven Engineering Languages and Systems (MODELS '22 Companion)