Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
01 OCT 2014	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Contract-Based Integration of Cyber-Physical Analyses	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
Ivan Ruchkin, Dionisio De Niz, Sagar Chaki, David Garlan	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**SAR**	**1**	

er, the outputs of the former may be invalid.

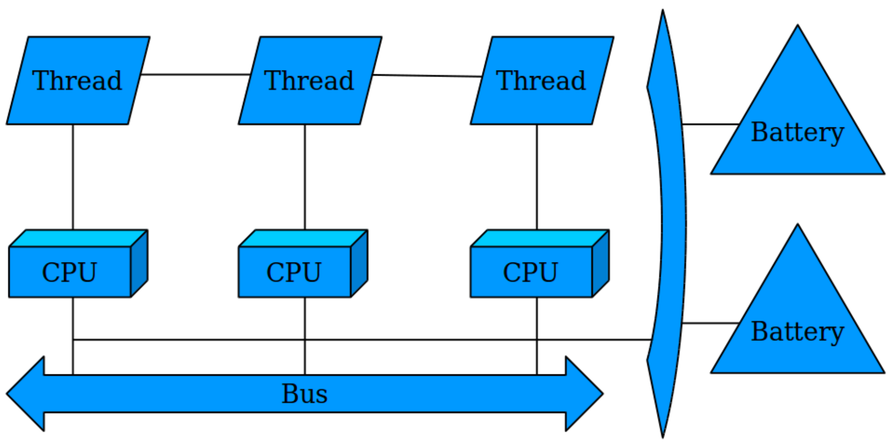
- Specification of such implicit assumptions and detection of their violation is left to human designers, who are often unable to cope with complexity.
- Analysis integration problems discovered late in development lead to expensive changes to the system.

Hence the research question:

- How to specify analysis compositions and verify their correctness?

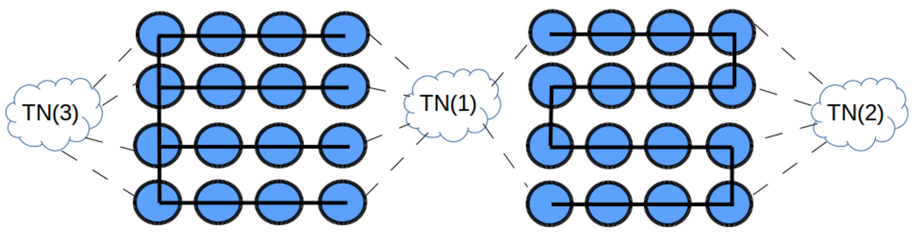
Example System

Consider an autonomous aircraft as an example CPS. It operates data with different classes of security, from normal to top secret (ThSecCl). Periodic threads (T) execute on several processors (C). The aircraft is powered with multi-cell reconfigurable batteries (B). The system's architecture shown below is specified in AADL.



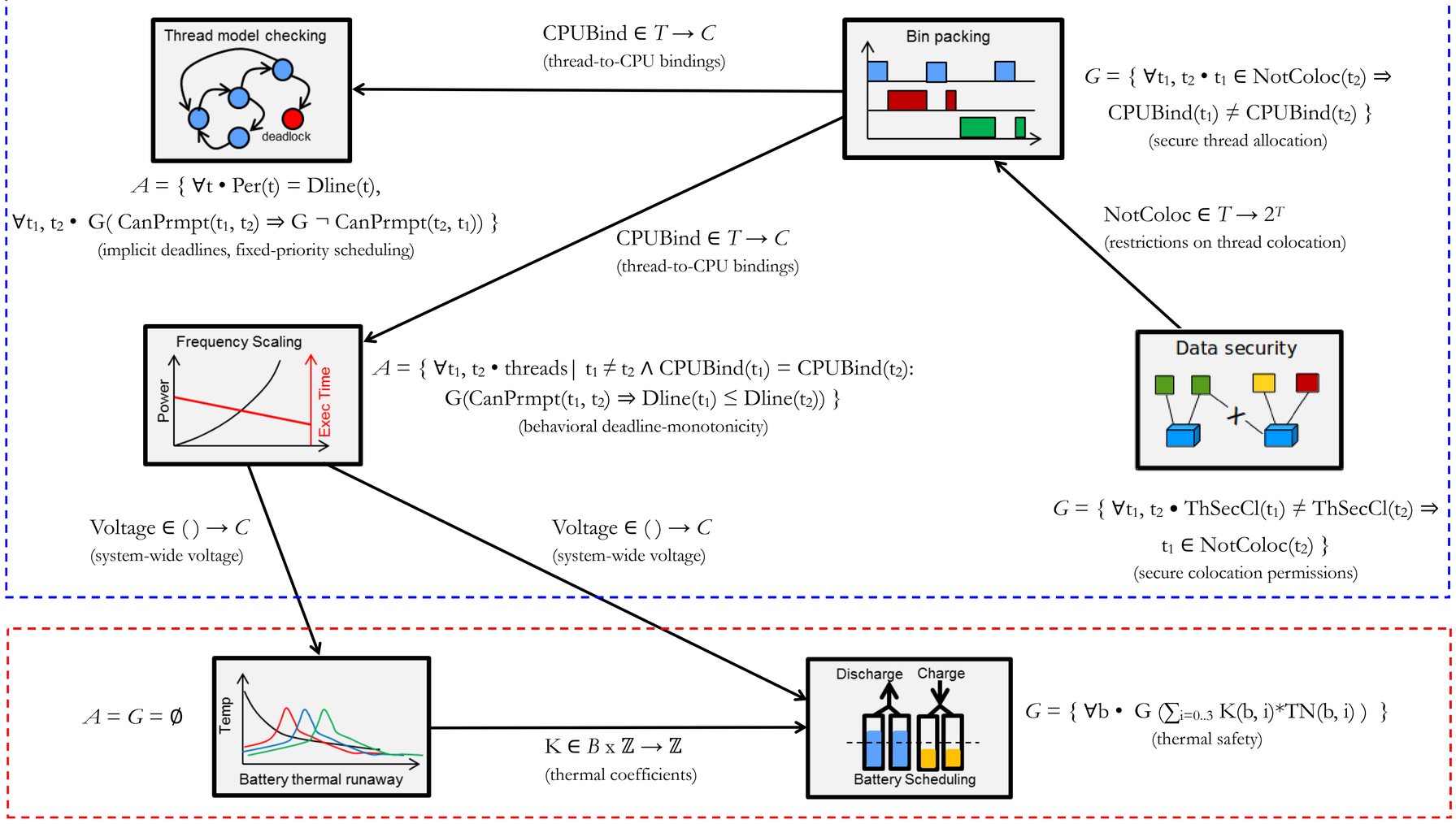
A battery has a matrix of cells, and each cell has a current level of charge. A battery scheduler determines parallel and sequential connections between groups of cells in order to satisfy voltage and current output requirements.

Thermally, each cell exchanges heat with its neighboring cells (*thermal neighbors,* TN) through an electrical connector, affecting the risk of a *thermal runaway*.



Example Analyses

Scheduling verification domain σ_{sched}



Battery verification domain σ_{batt}

policies SchedPol.

- $\mathcal{S}$ — a set of static functions that encode design-time properties. E.g., thread period Per, thread-to-CPU binding CPUBind, and system-wide Voltage.
- $\mathcal{R}$ — a set of runtime functions that encode dynamic properties. E.g., preemption relation canPrmpt(t_1, t_2) and number of cells in a battery b with i thermal neighbors TN(b, i).
- $\mathcal{T}$ — execution semantics of σ — a set of sequences of assignments to $\mathcal{R}$. We use Promela programs to implement the semantics.
- $[\![]\!]_\sigma$ — a domain interpretation of $\mathcal{A}$, $\mathcal{S}$, and $\mathcal{T}$. E.g., $[\![\text{SchedPol}]\!]_\sigma$ = {RMS, DMS, EDF}.

Formally, an AADL architectural model $\mathbf{m}$ is an interpretation $[\![]\!]_m$ of $\mathcal{A}$, $\mathcal{S}$, and $\mathcal{T}$. E.g., $[\![T]\!]_m$ = { SensorSample, $Ctrl_1$, $Ctrl_2$ }, $[\![\textbf{CPUBind}]\!]_m$ = { ($Ctrl_1$, CPU_1), ($Ctrl_2$, CPU_2), …) }.

$[\![]\!]_\sigma \cup [\![]\!]_\mathbf{m}$ form a full interpretation of $\mathcal{A}$, $\mathcal{S}$, $\mathcal{R}$, and $\mathcal{T}$.

Analysis Contracts

Each analysis is assigned a *contract* — a tuple (I, O, A, G).

- Inputs $I \subseteq \mathcal{A} \cup \mathcal{S}$ declare elements that the analysis reads.
- Outputs $O \subseteq \mathcal{A} \cup \mathcal{S}$ declare elements that the analysis writes.
- Assumptions $A \subseteq \mathcal{F}_\sigma$ are logical statements that must be satisfied by every input model to the analysis: $\mathbf{m} \vDash A$.
- Guarantees $G \subseteq \mathcal{F}_\sigma$ are logical statements that must be satisfied by every output model of the analysis: $\mathbf{m} \vDash G$.

Assumption and guarantee formulas have the following syntax:

$$\mathcal{F}_\sigma ::= \forall\, v_1 ... v_j \bullet \varphi \mid \exists\, v_1 ... v_j \bullet \varphi \mid$$
$$\forall\, v_1 ... v_j \bullet \varphi : \psi \mid \exists\, v_1 ... v_j \bullet \varphi : \psi,$$

where φ is a predicate logic formula over $\mathcal{A} \cup \mathcal{S}$, ψ is an LTL formula over $\mathcal{A} \cup \mathcal{S} \cup \mathcal{R}$.

Am ordering $\langle C_1 ... C_n \rangle$ of contract
if predecessors are not dependent o

$$\forall i \in [1, n] \bullet \forall j \in [1, i) \bullet C_j$$

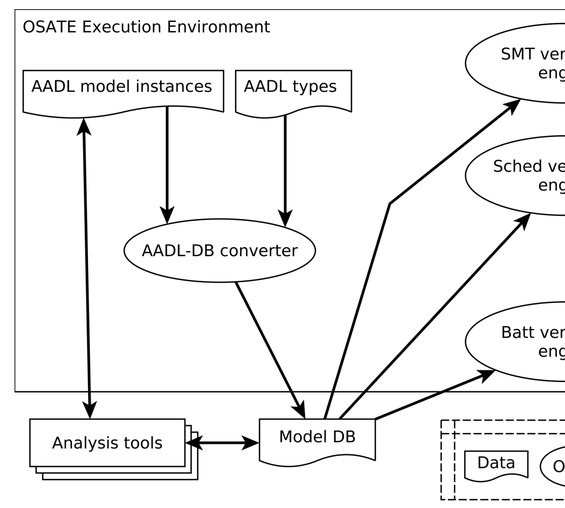
Consider a graph with vertices bein
being contract dependencies. There
ing of contracts if and only if the g
it is not cyclic, any topological orde

Contract Verification

The goal of contract verification is

For purely first-order formulas that
decide satisfiability via SMT solving
generated based on $\mathcal{A}$ and $\mathcal{S}$ menti
SMT solver is invoked on $\neg\, \varphi$ (or
tification). A universally (existential
is satisfied if and only if UNSAT (S

For formulas combining predicate
formula ψ, we first generate an SM
find all valuations of $v_1 ... v_j$ that satis
valuation we call Spin on a Promela
ments $\mathcal{T}$ for $\mathbf{m}$ in the domain of ψ.
formed into an LTL property speci
universally (existentially) quantified
and only if the LTL property holds
valuations. The architecture of our
shown below:



Experimental Results

Effectiveness: we have been able to de
tion errors and verify their absence
the example.

Scalability: the results of scalability e
implementations of $\mathcal{T}$ are shown in

$\mathcal{T}_{sched}$:

Threads	(R/D)MS time*	EDF time*
3	0.01	0.01
4	0.01	0.52
5	0.07	33.4
6	0.37	2290.0
7	2.18	memlim
8	12.4	memlim
9	71.2	memlim
10	421	memlim
11	memlim	memlim

$\mathcal{T}_{ba}$

Cells
9
12
16
20
25

* All ti

Carnegie Mellon University

institute for SOFTWARE RESEARCH