

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE 01 OCT 2014		2. REPORT TYPE N/A		3. DATES COVERED -	
4. TITLE AND SUBTITLE Contract-Based Integration of Cyber-Physical Analyses				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Ivan Ruchkin, Dionisio De Niz, Sagar Chaki, David Garlan				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213				8. PERFORMING ORGANIZATION REPORT NUMBER	
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 SAR	18. NUMBER OF PAGES 1	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

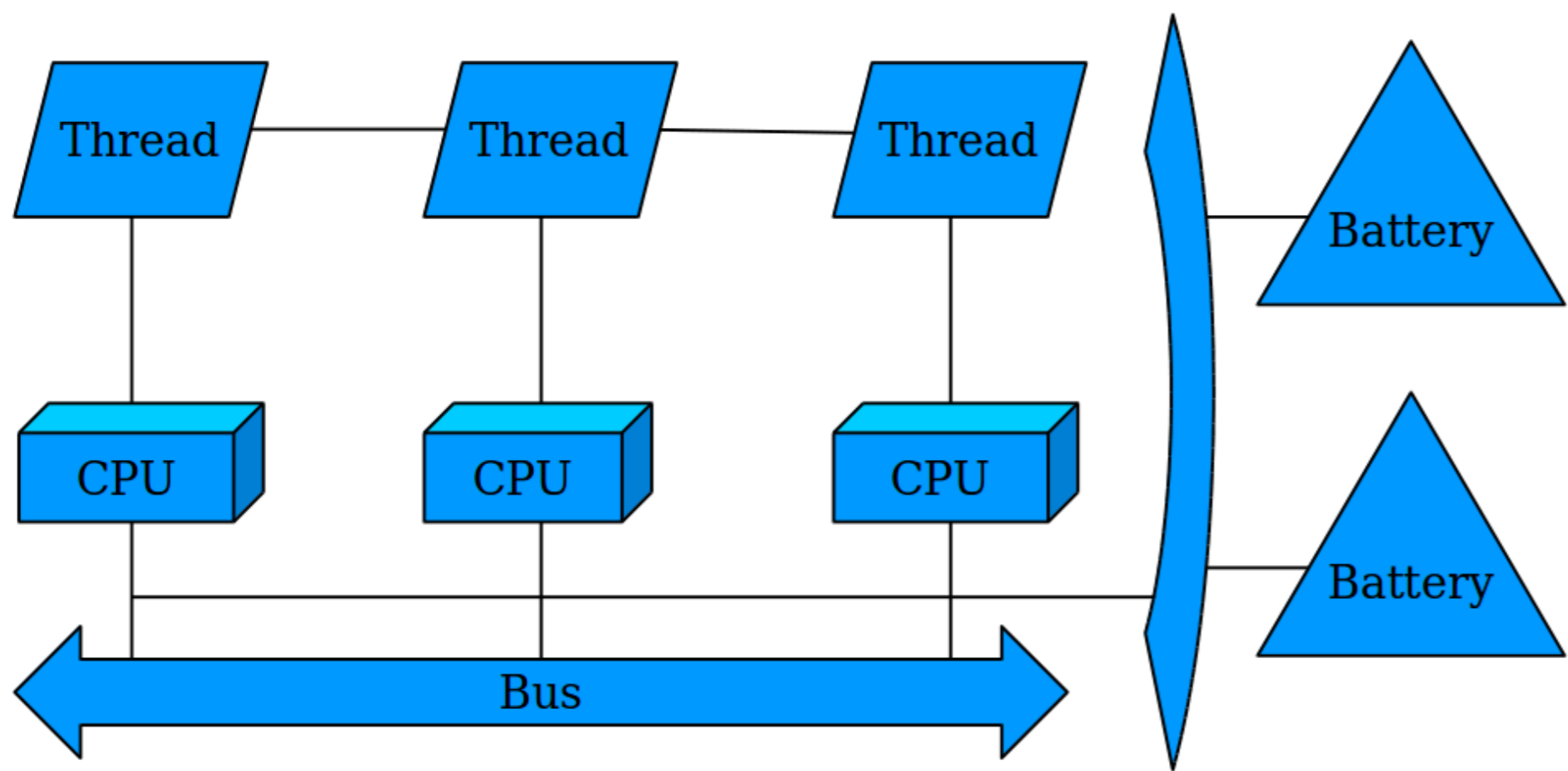
- er, the outputs of the former may be invalid.
- Specification of such implicit assumptions and detection of their violation is left to human designers, who are often unable to cope with complexity.
- Analysis integration problems discovered late in development lead to expensive changes to the system.

Hence the research question:

- How to specify analysis compositions and verify their correctness?

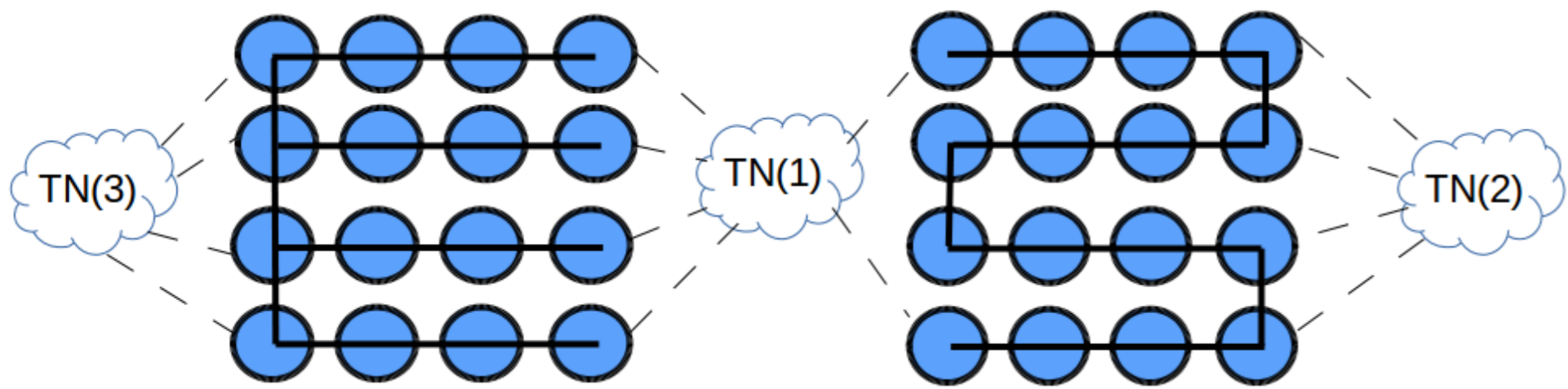
Example System

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.



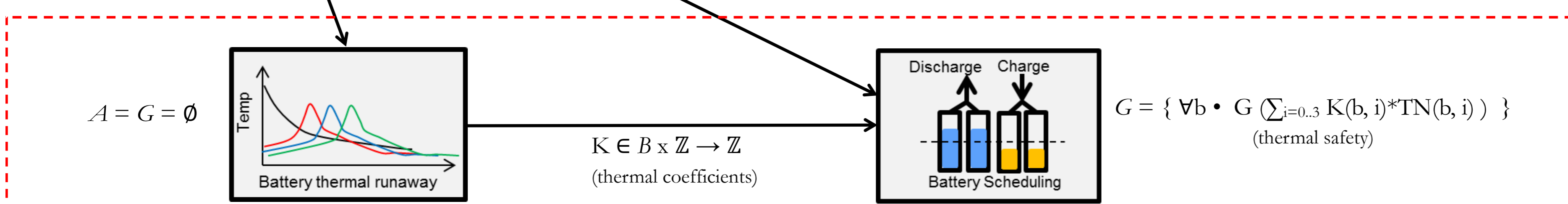
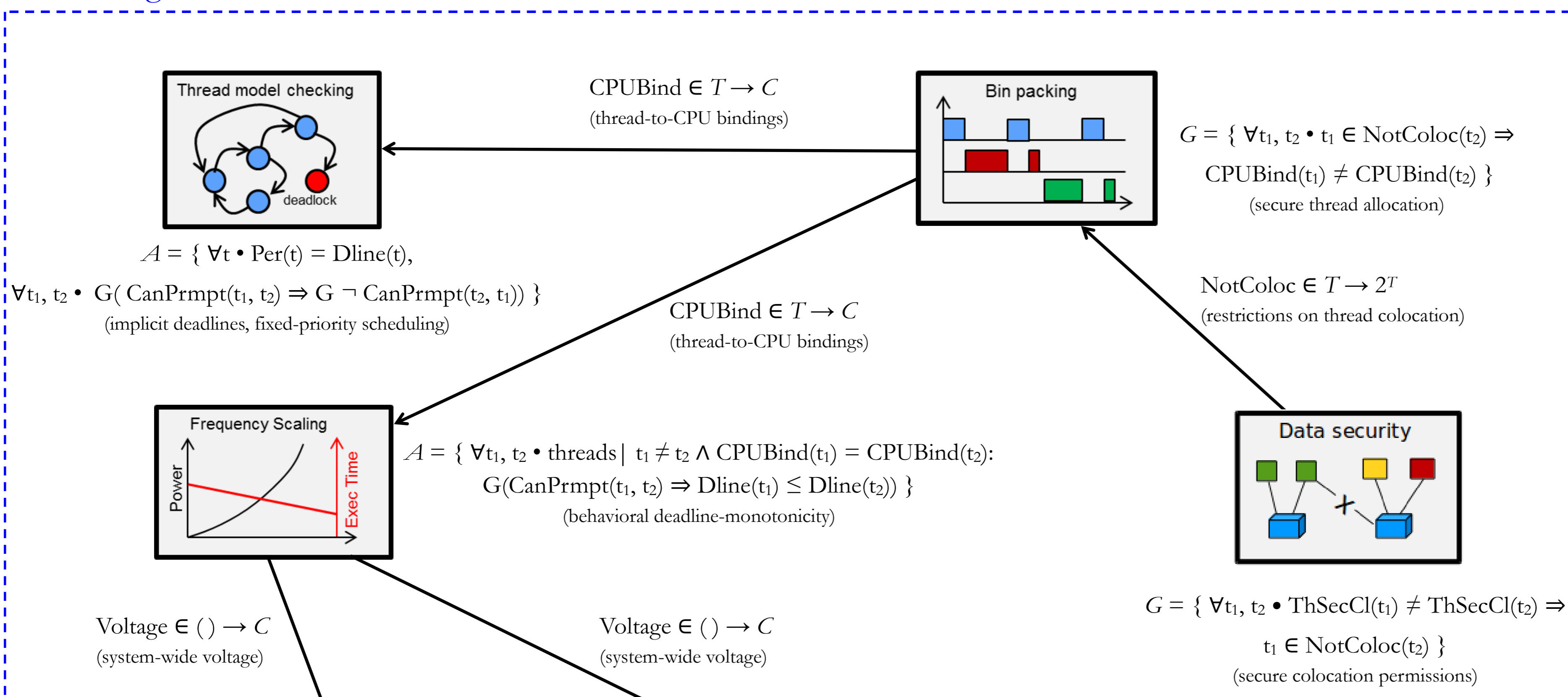
A battery has a matrix of cells, and each cell has a current level of charge. A battery scheduler determines parallel and sequential connections between groups of cells in order to satisfy voltage and current output requirements.

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 $\text{canPrmpt}(t_1, t_2)$ and number of cells in a battery b with i thermal neighbors $\text{TN}(b, i)$.
- \mathcal{T} — execution semantics of σ — a set of sequences of assignments to \mathcal{R} . We use Promela programs to implement the semantics.
- $\llbracket \cdot \rrbracket_{\sigma}$ — a domain interpretation of \mathcal{A} , \mathcal{S} , and \mathcal{T} . E.g., $\llbracket \text{SchedPol} \rrbracket_{\sigma} = \{\text{RMS, DMS, EDF}\}$.

Formally, an AADL architectural model \mathbf{m} is an interpretation $\llbracket \cdot \rrbracket_{\mathbf{m}}$ of \mathcal{A} , \mathcal{S} , and \mathcal{T} . E.g., $\llbracket T \rrbracket_{\mathbf{m}} = \{\text{SensorSample, Ctrl}_1, \text{Ctrl}_2\}$, $\llbracket \text{CPUBind} \rrbracket_{\mathbf{m}} = \{(\text{Ctrl}_1, \text{CPU}_1), (\text{Ctrl}_2, \text{CPU}_2), \dots\}$.

$\llbracket \cdot \rrbracket_{\sigma} \cup \llbracket \cdot \rrbracket_{\mathbf{m}}$ 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, \mathcal{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 $\mathcal{A} \subseteq \mathcal{F}_{\sigma}$ are logical statements that must be satisfied by every input model to the analysis: $\mathbf{m} \models \mathcal{A}$.
- Guarantees $G \subseteq \mathcal{F}_{\sigma}$ are logical statements that must be satisfied by every output model of the analysis: $\mathbf{m} \models G$.

Assumption and guarantee formulas have the following syntax:

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

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 $\langle C_1 \dots C_n \rangle$ of contract

if predecessors $\langle C_1 \dots C_n \rangle$ are not dependent on

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

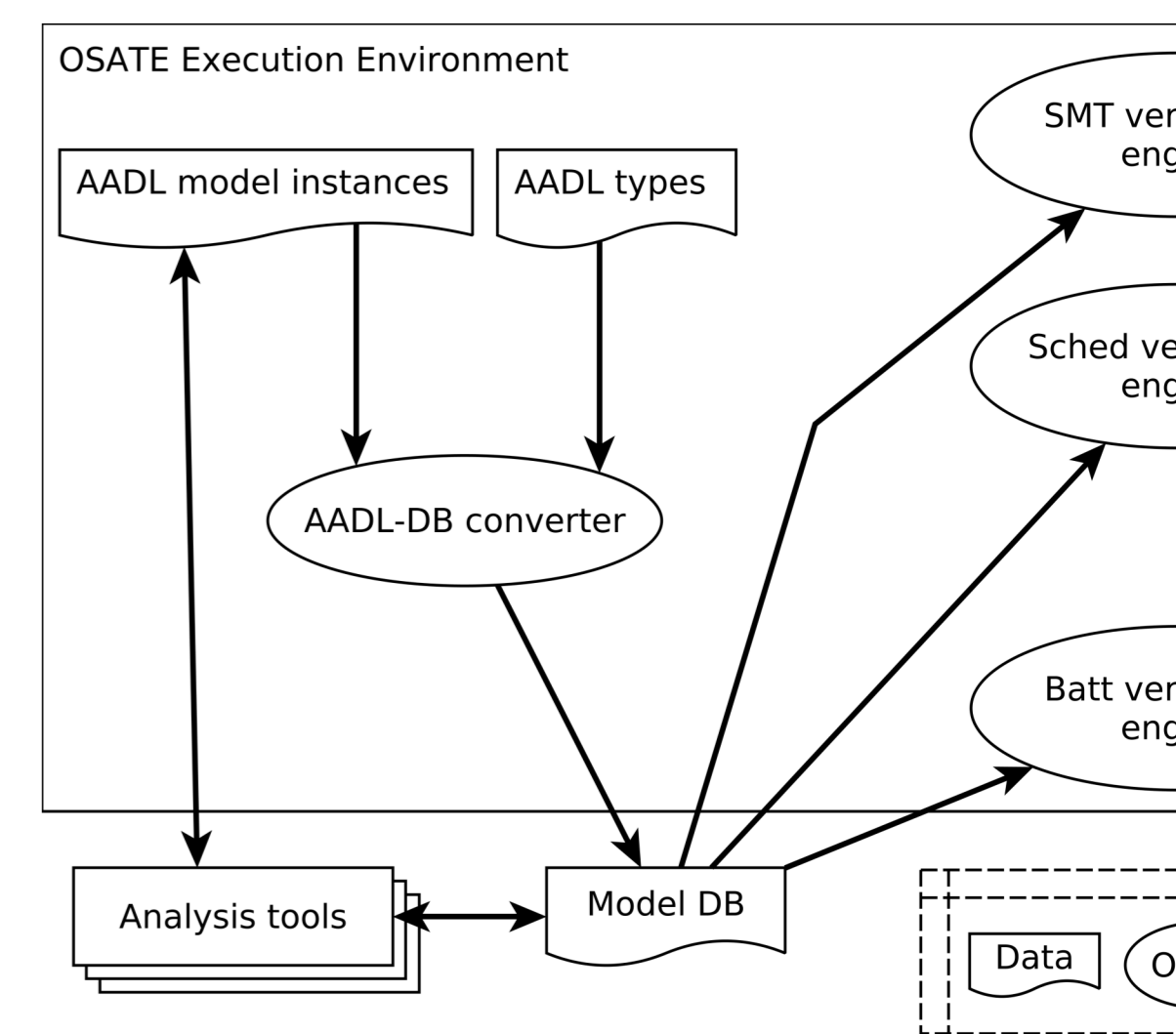
Consider a graph with vertices being contract dependencies. There being contract dependencies. There ing of contracts if and only if the g it is not cyclic, any topological orde

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} mentioned. 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 SMT find all valuations of $v_1 \dots v_j$ that satisfy valuation we call Spin on a Promela elements \mathcal{T} for \mathbf{m} in the domain of ψ . formed into an LTL property specified 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 detect errors and verify their absence in the example.

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

$\mathcal{T}_{\text{sched}}$:			$\mathcal{T}_{\text{batt}}$:
Threads	(R/D)MS time*	EDF time*	Cells
3	0.01	0.01	9
4	0.01	0.52	12
5	0.07	33.4	16
6	0.37	2290.0	20
7	2.18	memlim	25
8	12.4	memlim	
9	71.2	memlim	* All ti
10	421	memlim	
11	memlim	memlim	

Copyright 2014 ACM. This material is based upon work funded and supported by the Department of Defense under award number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute. This work was sponsored by the U.S. Army Research Office. ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSES OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY WILL NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM COPYRIGHT INFRINGEMENT. This material has been approved for public release and distribution. Carnegie Mellon is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.