ModelPlex: Verified Runtime Validation of Verified CPS Models
From Model Checking to Checking Models

Stefan Mitsch  André Platzer

Computer Science Department, Carnegie Mellon University

Clarke Symposium, Sept. 20, 2014

For details, see ModelPlex paper at RV’14
Formal Verification in CPS Development

Real CPS

Verification Results

safe

Proof

Reachability Analysis ...

Stefan Mitsch, André Platzer—ModelPlex: Verified Runtime Validation of Verified CPS Models
Formal Verification in CPS Development

Real CPS

Model $\alpha^*$

Control $\alpha_{\text{ctrl}}$

$$v := v + 1$$

Plant $\alpha_{\text{plant}}$

$$x' = v$$

abstract

Proof

Reachability Analysis

Verification Results

safe

Challenge

Verification results about models only apply if CPS fits to the model

$\Rightarrow$ Verifiably correct runtime model validation
Formal Verification in CPS Development

Real CPS

Model

Challenge

Verification results about models
only apply if CPS fits to the model

Verifiably correct runtime model validation
ModelPlex ensures that verification results about models apply to CPS implementations.
ModelPlex ensures that verification results about models apply to CPS implementations.

Contributions

- Verification results transfer to CPS when validating model compliance for current run
- Compliance with model is characterizable in logic
- Compliance formula transformed by proof to executable monitor

model adequate? control safe? until next cycle?
ModelPlex at Runtime

“Simplex for Models”

Compliance Monitor Checks CPS for compliance with model at runtime
- Model Monitor: model adequate?
- Controller Monitor: control safe?
- Prediction Monitor: until next cycle?

Fallback Safe action, executed when monitor is not satisfied

Challenge What conditions do the monitors need to check to be safe?
When are two states linked through a run of model $\alpha$?

$\text{Model } \alpha \subseteq \text{Offline}(x^{-}, x^{+}) \in \rho(\alpha)$

Semantical: reachability relation of $\alpha$

Logic (dL): starting at $x = x^{-}$ exists a run of $\alpha$ to a state where $x = x^{+}$

$\text{Real arithmetic: check at runtime (efficient)}$

$\text{Theorem } (x = x^{-}) \rightarrow \langle \alpha(x) \rangle (x = x^{+})$

$\text{dL proof } \gg \text{dL proof}$

$F(x^{-}, x^{+})$
When are two states linked through a run of model $\alpha$?

- A prior state characterized by $x^-$
- Model $\alpha$
- A posterior state characterized by $x^+$

Semantical: $(x^-, x^+) \in \rho(\alpha)$

reachability relation of $\alpha$
When are two states linked through a run of model $\alpha$?

A prior state characterized by $x^-$

Model $\alpha$

A posterior state characterized by $x^+$

Offline

Semantical: $(x^-, x^+) \in \rho(\alpha)$

Logic ($d\mathcal{L}$): $(x = x^-) \rightarrow \langle \alpha(x) \rangle (x = x^+)$

Theorem

starting at $x = x^-$ exists a run of $\alpha$ to a state where $x = x^+$

Stefan Mitsch, André Platzer—ModelPlex:Verified Runtime Validation of Verified CPS Models
When are two states linked through a run of model $\alpha$?

- **a prior state characterized by $x^-$**
- **a posterior state characterized by $x^+$**

**Offline**

- **Semantical:** $(x^-, x^+) \in \rho(\alpha)$
  - $\iff$ Theorem
- **Logic ($\mathcal{DL}$):** $(x = x^-) \rightarrow \langle \alpha(x) \rangle (x = x^+)$
  - $\iff$ $\mathcal{DL}$ proof
- **Real arithmetic:** $F(x^-, x^+) \iff$ check at runtime (efficient)

Starting at $x = x^-$, there exists a run of $\alpha$ to a state where $x = x^+$. Check at runtime (efficient).
Monitor Characterization

When are two states linked through a run of model $\alpha$?

- A prior state characterized by $x^{-}$
- A posterior state characterized by $x^{+}$

**Offline**

- **Semantical:** $\left(x^{-}, x^{+}\right) \in \rho(\alpha)$
  - Theorem
- **Logic ($d\mathcal{L}$):** $\left(x = x^{-}\right) \rightarrow \langle \alpha(x) \rangle \left(x = x^{+}\right)$
  - $d\mathcal{L}$ proof
- **Real arithmetic:** $F \left(x^{-}, x^{+}\right)$
  - Check at runtime (efficient)

**Starting at $x = x^{-}$ exists a run of $\alpha$ to a state where $x = x^{+}$**
Proof calculus of $d\mathcal{L}$ executes models symbolically

Proof attempt

$(x = x^-) \rightarrow \langle \alpha \rangle (x = x^+)$
Proof calculus of $d\mathcal{L}$ executes models symbolically

Proof attempt:

$\text{prior state } x^{-} \rightarrow \text{climb} \rightarrow \text{posterior state } x^{+}$

$(x = x^{-}) \rightarrow \langle \text{climb} \cup \text{descend} \rangle (x = x^{+})$

$$\langle \text{climb} \rangle \phi \lor \langle \text{descend} \rangle \phi \quad \frac{\langle \cup \rangle \langle \text{climb} \cup \text{descend} \rangle \phi}$$
Proof calculus of $\mathcal{dL}$ executes models symbolically

Proof attempt:

$$(x = x^-) \rightarrow (\langle \text{climb} \cup \text{descend} \rangle (x = x^+))$$

$$\langle \text{climb} \rangle (x = x^+) \lor \langle \text{descend} \rangle (x = x^+)$$
Proof calculus of $d\mathcal{L}$ executes models symbolically

Proof attempt:

$$ (x = x^-) \rightarrow \langle \text{climb} \cup \text{descend} \rangle (x = x^+) $$

$$ \langle \text{climb} \rangle (x = x^+) $$

$$ \langle \text{descend} \rangle (x = x^+) $$

$$ F_1 (x^-, x^+) $$

$$ F_2 (x^-, x^+) $$
**Provably Correct Synthesis of Monitors**

- Proof calculus of $d\mathcal{L}$ executes models symbolically

![Diagram of state transition](attachment:image.png)

prior state $x^-$ $i - 1$ $\xrightarrow{\text{climb}}$ $i$ $\xleftarrow{\text{descend}}$ posterior state $x^+$

**proof attempt**

$$\begin{align*}
& (x = x^-) \rightarrow \langle \text{climb} \cup \text{descend} \rangle (x = x^+) \\
& \langle \text{climb} \rangle (x = x^+) \\
& \langle \text{descend} \rangle (x = x^+) \\
& F_1(x^-, x^+) \\
& F_2(x^-, x^+) \\
\end{align*}$$

Monitor: $F_1(x^-, x^+) \lor F_2(x^-, x^+)$

- The subgoals that cannot be proved express all the conditions on the relations of variables imposed by the model
Proof calculus of $dL$ executes models symbolically.

Model $\alpha$

Prior state $x^{-}$ $\rightarrow$ $i-1$ $\rightarrow$ $i$ $\rightarrow$ Posterior state $x^{+}$

Model Monitor

Immediate detection of model violation

$\rightsquigarrow$ Mitigates safety issues with safe fallback action

$F_{1}(x^{-}, x^{+})$ $\lor$ $F_{2}(x^{-}, x^{+})$

Monitor: $F_{1}(x^{-}, x^{+}) \lor F_{2}(x^{-}, x^{+})$

The subgoals that cannot be proved express all the conditions on the relations of variables imposed by the model.
ModelPlex ensures that proofs apply to real CPS

- Validate model compliance
- Characterize compliance with model in logic
- Prover transforms compliance formula to executable monitor
Thank You!
Evaluation

- Evaluated on hybrid system case studies
  - Water tank
  - Cruise control
  - Traffic control
  - Ground robots
  - Train control

- Model sizes: 5–16 variables
- Monitor sizes: 20–150 operations (larger if automated simplification to remove redundant checks is computationally infeasible)

**Theorem:** ModelPlex is decidable and monitor synthesis can be automated in important classes