Logic for Distributed Hybrid Systems

André Platzer

Carnegie Mellon University, Pittsburgh, PA
Outline

1 Motivation

2 Quantified Differential Dynamic Logic QdL
   - Design
   - Syntax

3 Verification of Distributed Hybrid Systems
   - Soundness and Completeness

4 Conclusions
Q: I want to verify my car

Challenge

Q: I want to verify my car

A: Hybrid systems

Continuous dynamics (differential equations)

Discrete dynamics (control decisions)
Q: I want to verify my car  
A: Hybrid systems

Challenge (Hybrid Systems)

- Continuous dynamics  
  (differential equations)
- Discrete dynamics  
  (control decisions)
Q: I want to verify my car
A: Hybrid systems
Q: But there’s a lot of cars!

Challenge (Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
Q: I want to verify a lot of cars

Challenge
Q: I want to verify a lot of cars  
A: Distributed systems

**Challenge (Distributed Systems)**

- Local computation  
  (finite state automaton)
- Remote communication  
  (network graph)
Q: I want to verify a lot of cars 
A: Distributed systems 
Q: But they move!

Challenge (Distributed Systems)

- Local computation (finite state automaton)
- Remote communication (network graph)
Q: I want to verify lots of moving cars

Challenge

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (remote communication)
- Dimensional dynamics (appearance)
Q: I want to verify lots of moving cars  
A: Distributed hybrid systems

**Challenge (Distributed Hybrid Systems)**

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (remote communication)
Q: I want to verify lots of moving cars  A: Distributed hybrid systems

Challenge (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (remote communication)
- Dimensional dynamics (appearance)
Q: I want to verify lots of moving cars
A: Distributed hybrid systems
Q: How?

Challenge (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (remote communication)
- Dimensional dynamics (appearance)
State of the Art:

**Shift** [DGV96] The Hybrid System Simulation Programming Language

**R-Charon** [KSPL06] Modeling Language for Reconfigurable Hybrid Systems

**Hybrid CSP** [CJR95] Semantics in Extended Duration Calculus

**Φ-calculus** [Rou04] Semantics in rich set theory

**HyPA** [CR05] Translate fragment into normal form.

**ACP^srt** [BM05] Modeling language proposal

**χ process algebra** [vBMR^+06] Simulation, translation of fragments to PHAVER, UPPAAL

**OBSHS** [MS06] Partial random simulation of objects
<table>
<thead>
<tr>
<th>Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shift [DGV96]</td>
<td>The Hybrid System Simulation Programming Language</td>
</tr>
<tr>
<td>R-Charon [KSPL06]</td>
<td>Modeling Language for Reconfigurable Hybrid Systems</td>
</tr>
<tr>
<td>Hybrid CSP [CJR95]</td>
<td>Semantics in Extended Duration Calculus</td>
</tr>
<tr>
<td>Φ-calculus [Rou04]</td>
<td>Semantics in rich set theory</td>
</tr>
<tr>
<td>Χ process algebra [vBMR⁺06]</td>
<td>Simulation, translation of fragments to PHAVER, UPPAAL</td>
</tr>
<tr>
<td>ACP&lt;sub&gt;hs&lt;/sub&gt; [BM05]</td>
<td>Modeling language proposal</td>
</tr>
<tr>
<td>OBSHS [MS06]</td>
<td>Partial random simulation of objects</td>
</tr>
</tbody>
</table>
State of the Art: Modeling and Simulation

No formal verification of distributed hybrid systems

- **Shift** [DGV96]  The Hybrid System Simulation Programming Language
- **R-Charon** [KSPL06]  Modeling Language for Reconfigurable Hybrid Systems
- **Hybrid CSP** [CJR95]  Semantics in Extended Duration Calculus
- **Φ-calculus** [Rou04]  Semantics in rich set theory
- **HyPA** [CR05]  Translate fragment into normal form.
- **ACP_{srt}** [BM05]  Modeling language proposal
- **χ**  Process algebra [vBMR+06]  Simulation, translation of fragments to PHAVER, UPPAAL
- **OBSHS** [MS06]  Partial random simulation of objects
Contributions

1. System model and semantics for distributed hybrid systems: QHP
2. Specification and verification logic: QdL
3. Compositional verification for QdL
4. First verification approach for distributed hybrid systems
   Sound and complete relative to differential equations
6. Verify collision freedom in a (simple) distributed car control system, where new cars may appear dynamically on the road
7. Logical foundation for analysis of distributed hybrid systems
8. Fundamental extension: first-order $x(i)$ versus primitive $x$
1 Motivation

2 Quantified Differential Dynamic Logic $\mathcal{QdL}$
   - Design
   - Syntax

3 Verification of Distributed Hybrid Systems
   - Soundness and Completeness

4 Conclusions
Outline (Conceptual Approach)

1. Motivation

2. Quantified Differential Dynamic Logic $QdL$
   - Design
   - Syntax

3. Verification of Distributed Hybrid Systems
   - Soundness and Completeness

4. Conclusions
Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (remote communication)
Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  \[ x'' = a \]
- Discrete dynamics (control decisions)
- Structural dynamics (remote communication)
Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  \[ x'' = a \]

- Discrete dynamics (control decisions)
  \[ a := \text{if } .. \text{then } A \text{ else } -b \]

- Structural dynamics (remote communication)
Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  \[ x'' = a \]
- Discrete dynamics (control decisions)
  \[ a := \text{if .. then } A \text{ else } -b \]
- Structural dynamics (remote communication)
Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  \[ x'' = a \]

- Discrete dynamics (control decisions)
  \[ a := \text{if .. then } A \text{ else } -b \]

- Structural dynamics (remote communication)
Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  \[ x(i)'' = a(i) \]
- Discrete dynamics (control decisions)
  \[ a(i) := \text{if .. then } A \text{ else } -b \]
- Structural dynamics (remote communication)
Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  \[ \forall i \ x(i)'' = a(i) \]
- Discrete dynamics (control decisions)
  \[ \forall i \ a(i) := \text{if .. then } A \text{ else } -b \]
- Structural dynamics (remote communication)
Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- **Continuous dynamics** (differential equations)
  \[ \forall i \quad x(i)'' = a(i) \]
- **Discrete dynamics** (control decisions)
  \[ \forall i \quad a(i) := \text{if } \ldots \text{ then } A \text{ else } -b \]
- **Structural dynamics** (remote communication)
  \[ \ell(i) := \text{carInFrontOf}(i) \]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

André Platzer (CMU)

Logic for Distributed Hybrid Systems

CMACS 7 / 10
Q: How to model distributed hybrid systems
A: Quantified Hybrid Programs

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  \( \forall i \, x(i)'' = a(i) \)
- Discrete dynamics (control decisions)
  \( \forall i \, a(i) := \text{if} \ldots \text{then } A \text{ else } -b \)
- Structural dynamics (remote communication)
  \( \ell(i) := \text{carInFrontOf}(i) \)
- Dimensional dynamics (appearance)
Q: How to model distributed hybrid systems

A: Quantified Hybrid Programs

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  \[ \forall i \ x(i)'' = a(i) \]

- Discrete dynamics (control decisions)
  \[ \forall i \ a(i) := \text{if .. then } A \text{ else } -b \]

- Structural dynamics (remote communication)
  \[ \ell(i) := \text{carInFrontOf}(i) \]

- Dimensional dynamics (appearance)
  \[ n := \text{new Car} \]
Definition (Quantified hybrid program $\alpha$)

$$\forall i: C \ x(s)' = \theta$$  (quantified ODE)

$$\forall i: C \ x(s) := \theta$$  (quantified assignment)

?$\chi$  (conditional execution)

$\alpha; \beta$  (seq. composition)

$\alpha \cup \beta$  (nondet. choice)

$\alpha^*$  (nondet. repetition)

\{ jump & test \}

\{ Kleene algebra \}
### Definition (Quantified hybrid program $\alpha$)

- $\forall i : C \ x(s)' = \theta$ (quantified ODE)
- $\forall i : C \ x(s) := \theta$ (quantified assignment)
- $?\chi$ (conditional execution)
- $\alpha; \beta$ (seq. composition)
- $\alpha \cup \beta$ (nondet. choice)
- $\alpha^*$ (nondet. repetition)

### Kleene algebra

- $\alpha \cup \beta$ (nondet. choice)
- $\alpha^*$ (nondet. repetition)

### Jump & Test

- $\forall i : C \ x(s)' = \theta$
- $\forall i : C \ x(s) := \theta$
- $?\chi$
- $\alpha; \beta$
- $\alpha \cup \beta$
- $\alpha^*$

### DCCS

$$DCCS \equiv (\text{ctrl}; \text{drive})^*$$

- $\text{ctrl} \equiv \forall i : C \ a(i) := \text{if } \forall j : C \ \text{far}(i, j) \ \text{then } A \ \text{else } -b$
- $\text{drive} \equiv \forall i : C \ x(i)'' = a(i)$
Definition (Quantified hybrid program $\alpha$)

\[
\begin{align*}
\forall i : C \ x(s)' &= \theta \quad \text{(quantified ODE)} \\
\forall i : C \ x(s) &:= \theta \quad \text{(quantified assignment)} \\
?\chi &\quad \text{(conditional execution)} \\
\alpha;\beta &\quad \text{(seq. composition)} \\
\alpha \cup \beta &\quad \text{(nondet. choice)} \\
\alpha^* &\quad \text{(nondet. repetition)} \\
\end{align*}
\]

\{ jump \& test \}

\{ Kleene algebra \}

\[DCCS \equiv (appear; ctrl; drive)^*\]

appear $\equiv n := \text{new } C; \ ?(\forall j : C \ \text{far}(j, n))$

ctrl $\equiv \forall i : C \ a(i) := \text{if } \forall j : C \ \text{far}(i, j) \ \text{then } A \ \text{else } -b$

drive $\equiv \forall i : C \ x(i)'' = a(i)$
Definition (Quantified hybrid program $\alpha$)

$$\forall i : C \; x(s)' = \theta$$  (quantified ODE)

$$\forall i : C \; x(s) := \theta$$  (quantified assignment)

$$?x$$  (conditional execution)

$$\alpha; \beta$$  (seq. composition)

$$\alpha \cup \beta$$  (nondet. choice)

$$\alpha^*$$  (nondet. repetition)

\[ \{ \text{jump \\& test} \} \quad \{ \text{Kleene algebra} \} \]

\[
DCCS \equiv (appear; ctrl; drive)^* \\
appear \equiv n := \text{new } C; \; ?(\forall j : C \; \text{far}(j, n)) \\
ctrl \equiv \forall i : C \; a(i) := \text{if } \forall j : C \; \text{far}(i, j) \text{ then } A \text{ else } \neg b \\
drive \equiv \forall i : C \; x(i)'' = a(i) \\
\text{new } C \text{ is definable!}
\]
Quantified Differential Dynamic Logic \( \mathcal{QdL} \): Syntax

**Definition (\( \mathcal{QdL} \) Formula \( \phi \))**

\[ \neg, \land, \lor, \to, \forall x, \exists x, =, \leq, +, \cdot \quad (\mathbb{R}\text{-first-order part}) \\
[\alpha]\phi, \langle \alpha \rangle \phi \quad (\text{dynamic part}) \]

\[
\forall i, j : C \text{ far}(i, j) \rightarrow [(\text{appear}; ctrl; drive)^*] \quad \forall i \neq j : C \ x(i) \neq x(j)
\]

\[
\text{far}(i, j) \equiv i \neq j \rightarrow x(i) < x(j) \land v(i) \leq v(j) \land a(i) \leq a(j) \\
\land \ x(i) > x(j) \land v(i) \geq v(j) \land a(i) \geq a(j) \ldots
\]
Outline (Verification Approach)

1. Motivation

2. Quantified Differential Dynamic Logic QdŁ
   - Design
   - Syntax

3. Verification of Distributed Hybrid Systems
   - Soundness and Completeness

4. Conclusions
Theorem (Relative Completeness)

$\text{QdL}$ verification sound & complete axiomatisation of distributed hybrid systems relative to quantified differential equations.

Proof 16p.
Theorem (Relative Completeness)

QdŁ verification sound & complete axiomatisation of distributed hybrid systems relative to quantified differential equations.

Corollary (Proof-theoretical Alignment)

proving distributed hybrid systems = proving dynamical systems!
Soundness and Completeness

Theorem (Relative Completeness)

\[ \text{QdL verification sound & complete axiomatisation of distributed hybrid systems relative to quantified differential equations.} \]

Proof 16p.

Corollary (Proof-theoretical Alignment)

proving distributed hybrid systems = proving dynamical systems!

Corollary (Yes, we can!)

distributed hybrid systems can be verified by recursive decomposition
1 Motivation

2 Quantified Differential Dynamic Logic QdŁ
   • Design
   • Syntax

3 Verification of Distributed Hybrid Systems
   • Soundness and Completeness

4 Conclusions
Distributed hybrid systems everywhere
System model and semantics
Logic for distributed hybrid systems
Compositional verification
First verification approach
Sound & complete / diff. eqn.
Simple distributed car control verified
Conclusions

quantified differential dynamic logic

\[ \text{QdL} = \text{FOL} + \text{DL} + \text{QHP} \]

- Distributed hybrid systems everywhere
- System model and semantics
- Logic for distributed hybrid systems
- Compositional verification
- First verification approach
- Sound & complete / diff. eqn.
- Simple distributed car control verified
Jan A. Bergstra and C. A. Middelburg.  
Process algebra for hybrid systems.  

A formal description of hybrid systems.  

Pieter J. L. Cuijpers and Michel A. Reniers.  
Hybrid process algebra.  

Akash Deshpande, Aleks Göllü, and Pravin Varaiya.  
SHIFT: A formalism and a programming language for dynamic networks of hybrid automata.  
João P. Hespanha and Ashish Tiwari, editors.

Fabian Kratz, Oleg Sokolsky, George J. Pappas, and Insup Lee.
R-Charon, a modeling language for reconfigurable hybrid systems.
In Hespanha and Tiwari [HT06], pages 392–406.

José Meseguer and Raman Sharykin.
Specification and analysis of distributed object-based stochastic hybrid systems.
In Hespanha and Tiwari [HT06], pages 460–475.

William C. Rounds.
A spatial logic for the hybrid $\pi$-calculus.

D. A. van Beek, Ka L. Man, Michel A. Reniers, J. E. Rooda, and Ramon R. H. Schifflers.
Syntax and consistent equation semantics of hybrid Chi.