Verification and Validation in Cyber-Physical Systems: Research Challenges and a Way Forward

Xi (James) Zheng
Christine Julien
The Center for Advanced Research in Software Engineering
The University of Texas at Austin

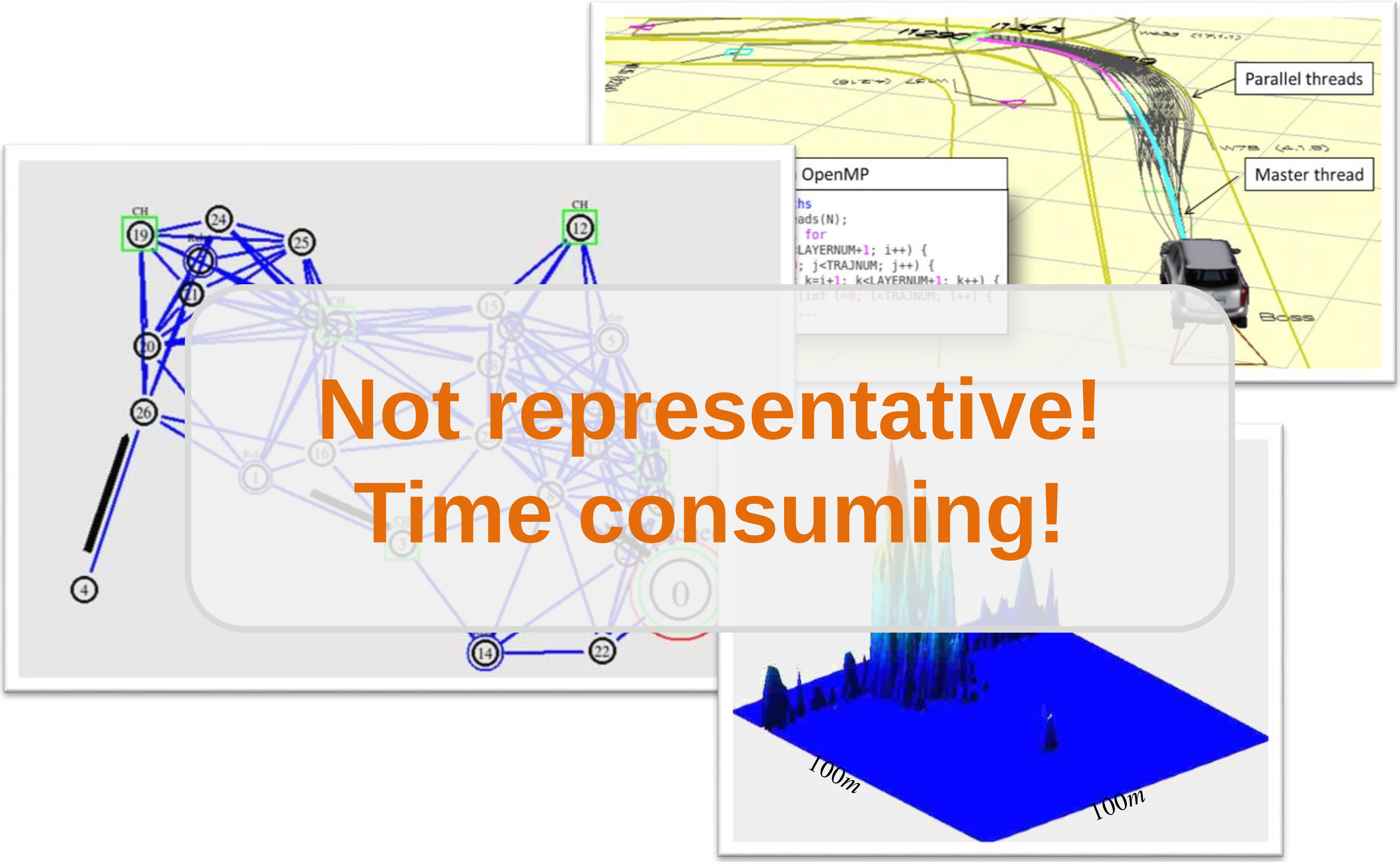
Does this look familiar?

Too complicated!

Not reproducible!

Dangerous!

Or is this more your cup of tea?



OpenMP

```
hs
ads(N);
 for
LAYERNUM+1; i++) {
; j<TRAJNUM; j++) {
 k=i+1; k<LAYERNUM+1; k++) {
(int l=0; l<TRAJNUM; l++) {
...
```

Not representative!

Time consuming!

Problem Statement

Opportunity:

Empirical evidence of the use and effectiveness of verification and validation strategies in CPS is largely anecdotal

Gap:

It is not clear what is truly demanded by modern CPS with respect to tools and techniques for verification and validation

Challenges: Real world scale, dynamics, safety, repeatability

This work starts with an **empirical study** of the state of the art and state of the practice of **verification and validation** of **cyber-physical systems**. It uses this study to motivate essential **research directions** for CPS V&V.

Strongly Held Belief 1

CPS developers are generally unfamiliar with traditional software verification and validation methodologies

- CPS developers are often **domain experts**, not software engineering experts
- Many often have a very different view of the software engineering process than we traditionally do

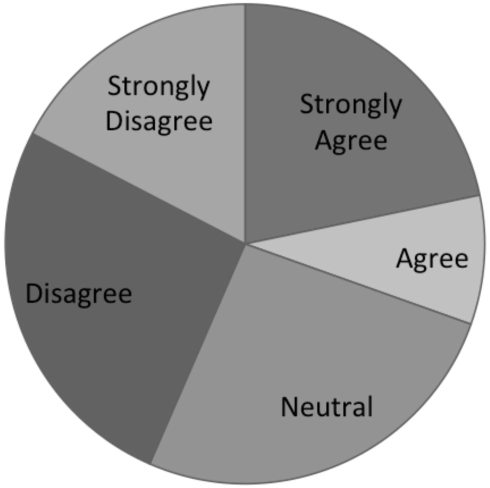
TRUE

Strongly Held Belief 2

High-level programming languages (e.g., Java) are not applicable to CPS

- Many CPS developers prefer low-level languages like nesC and other proprietary languages
- However, many also choose languages like Java, C++, Python, etc.

"A programming language like Java is not applicable to systems with hard real-time constraints"



Strongly Held Belief 3

Resource constraints (e.g., CPU, memory, and storage) are a major issue in developing and debugging CPS

- Low levels (e.g., sensor implementations) have to be concerned about resource constraints
- However, many of the tasks of CPS developers are constrained to the higher (application layers)
  - Developers assume lower levels have abstracted away resource constraint concerns

MYTH

Strongly Held Belief 4

Existing model checking and other formal techniques are insufficient to meet CPS applications' needs

- CPS developers believe that formal techniques:
  - Have learning curves that are too steep
  - Are computationally inefficient for large-scale systems

TRUE

- However, CPS developers commonly desire to use formal techniques, at least for components of the system

Strongly Held Belief 4 (More details)

There is a significant gap in between models of computing and communications and models of physics that makes applying them jointly in CPS challenging.

- CPS inherently intertwines cyber and physical
  - But tools and techniques for debugging the CPS generally focus on one or the other (often depending on the expertise of the user)
- **Teaser**: conceptually, models ought to be practically usable, e.g., for testing and debugging

Strongly Held Belief 5

An ad hoc, trial-and-error approach to development is the state of the art for CPS systems

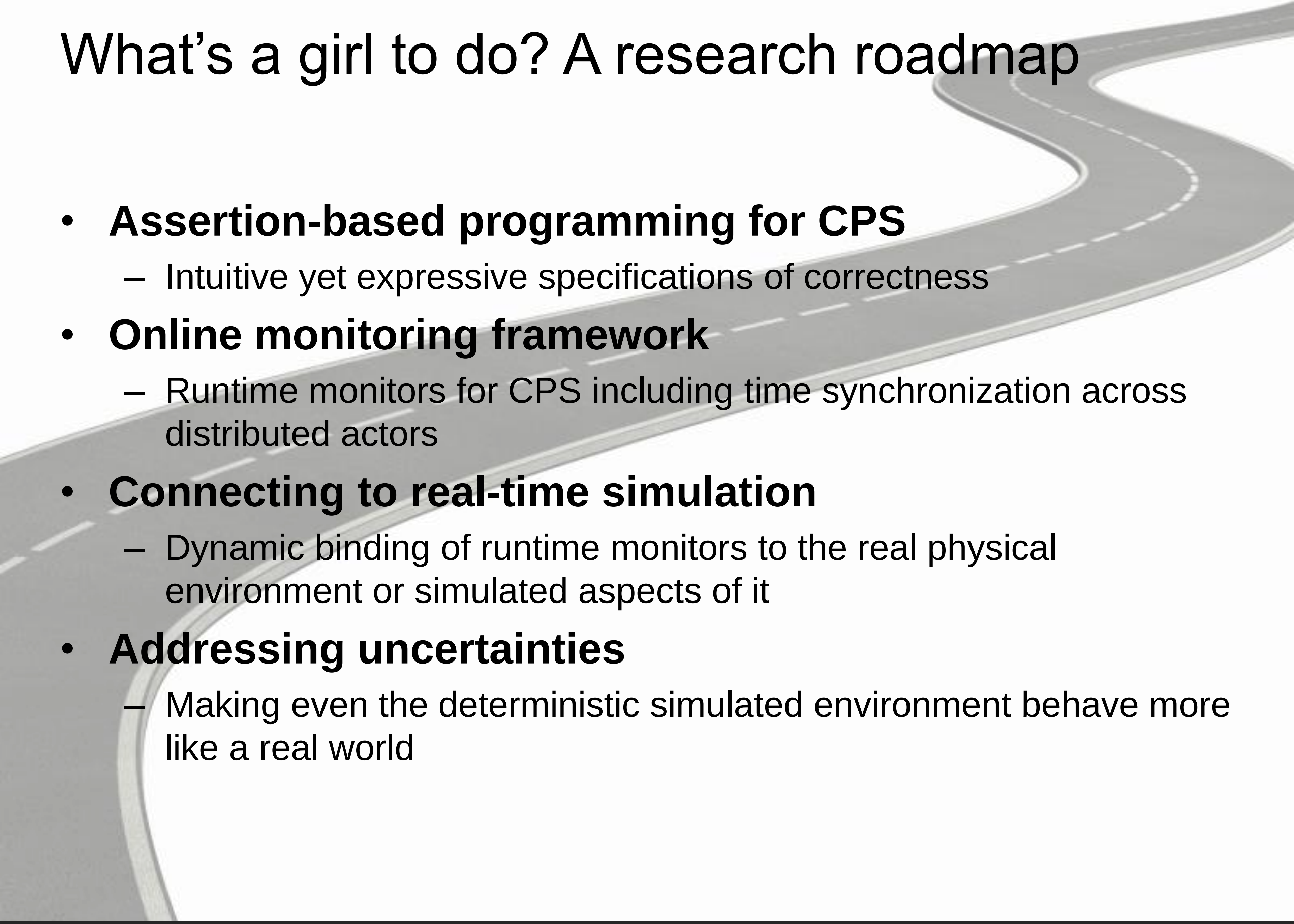
- 91.3% of the survey respondents either “Agree” or “Strongly Agree” with this statement

TRUE

Key Takeaways

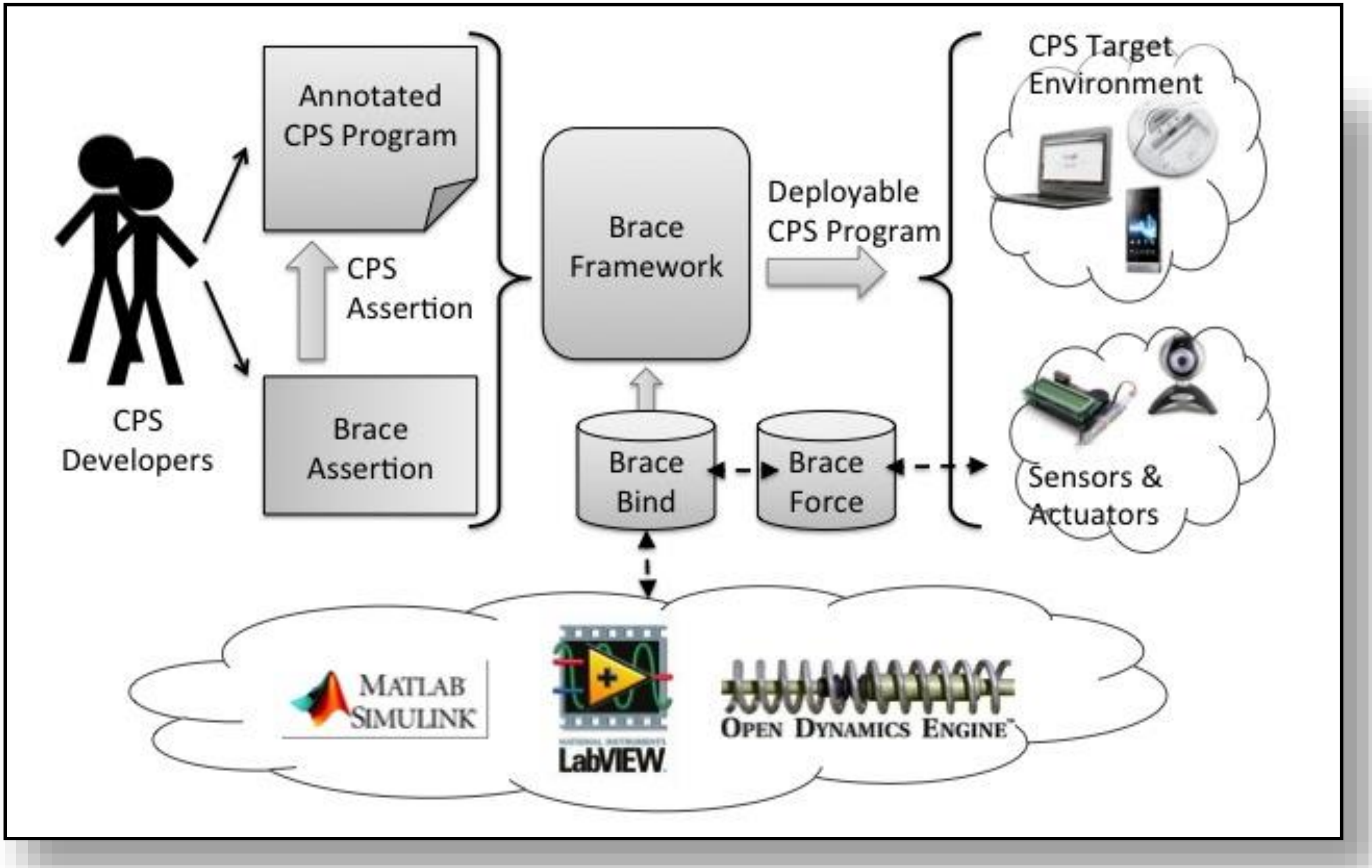
- Trial-and-error testing (which is the state of the practice) does not provide sufficient rigor in error detection
- Formal methods provide a desired level of expressiveness but are neither intuitive nor efficient
- Existing simulation tools are limited in their capabilities to jointly model physical and cyber components

What’s a girl to do? A research roadmap



- **Assertion-based programming for CPS**
  - Intuitive yet expressive specifications of correctness
- **Online monitoring framework**
  - Runtime monitors for CPS including time synchronization across distributed actors
- **Connecting to real-time simulation**
  - Dynamic binding of runtime monitors to the real physical environment or simulated aspects of it
- **Addressing uncertainties**
  - Making even the deterministic simulated environment behave more like a real world

The Brace Framework



Questions?

Xi (James) Zheng

Christine Julien (c.julien@utexas.edu)

The Center for Advanced Research in Software Engineering
The University of Texas at Austin