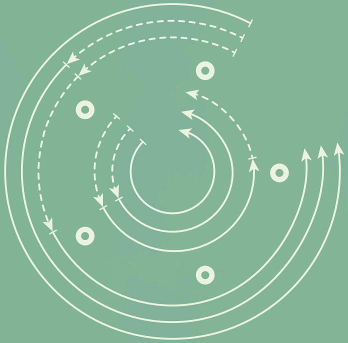
Bridging the Implementation Gap:

Advancements in Model-Based Concurrent Program Verification



Robert Benjamin Rubbens

Bridging the Implementation Gap: Advancements in Model-Based Concurrent Program Verification

Robert Benjamin Rubbens

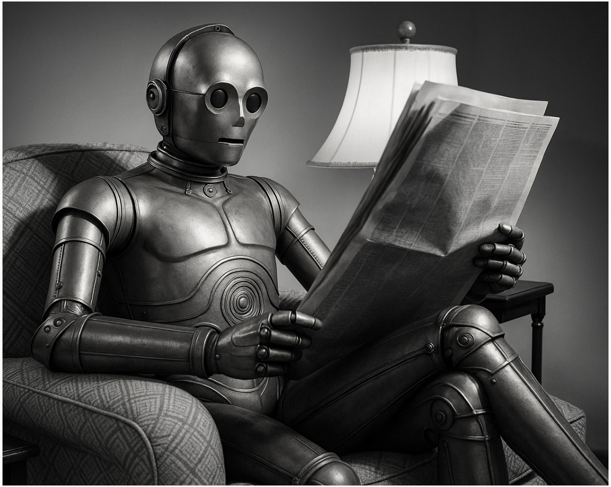
Formal Methods and Tools, University of Twente

2025-10-15

Imagine, starting your day, reading a newspaper...

Imagine, starting your day, reading a newspaper...

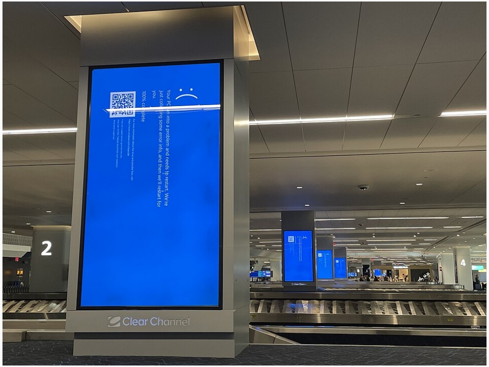
Imagine, starting your day, reading a newspaper...



Source: ChatGPT

Crowdstrike 2024 outage

- Crowdstrike: airport IT security software, used globally
- 19 July 2024: Crowdstrike software update
- Problem: at start-up, software received 20 instead of 21 pieces of data
- *Expensive* logic bug: possibly 5 billion dollars lost[1]

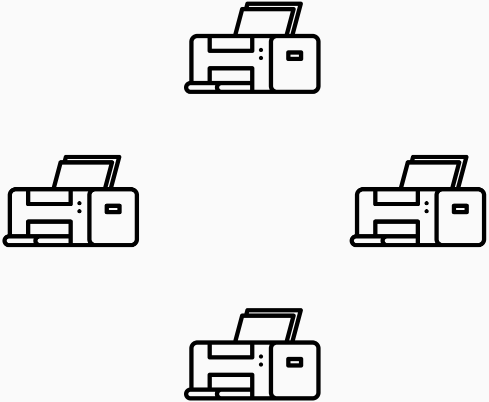


Source: Smishra1, Wikimedia Commons

[1] `https://edition.cnn.com/2024/07/24/tech/crowdstrike-outage-cost-cause`

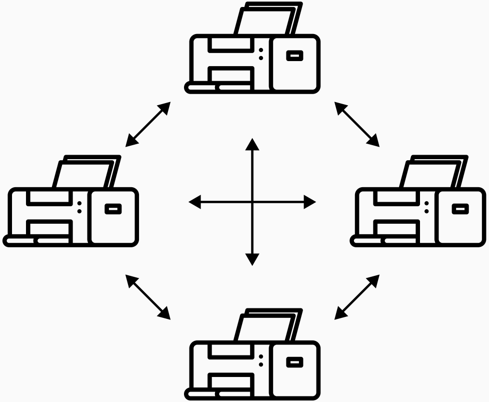
Software and concurrency

- Logic bugs are not the only problem
- Trend: demands for faster software are growing
- Solution: more things at the same time



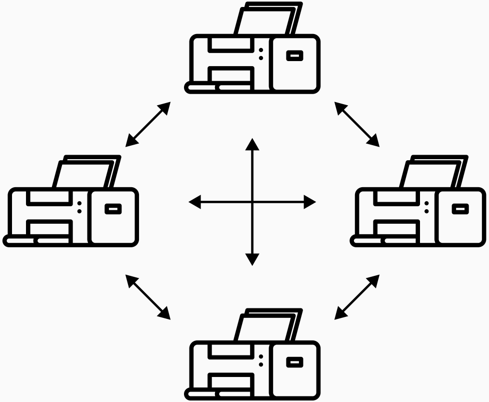
Software and concurrency

- Logic bugs are not the only problem
- Trend: demands for faster software are growing
- Solution: more things at the same time



Software and concurrency

- Logic bugs are not the only problem
- Trend: demands for faster software are growing
- Solution: more things at the same time
- Concurrency bugs are hard



Formal methods

- One possible solution: formal methods
- Mathematical techniques to prevent faults
- Insight: saying *what* you want is easier than saying *how*
- Can prevent logic *and* concurrency bugs

$$\frac{\Gamma \vdash \Delta, A \qquad A, \Sigma \vdash \Pi}{\Gamma, \Sigma \vdash \Delta, \Pi}$$

$$\{\ P\ \}\ \mathsf{c}\ \{\ Q\ \}$$

Formal methods and industry

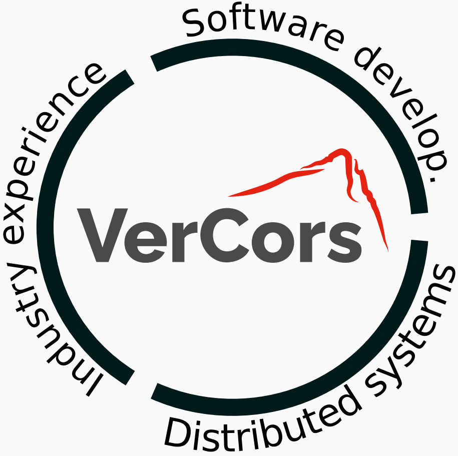
- Formal methods are promising
- But: limited uptake
- Some successes: TLA$^+$ @ AWS, Pulse/Infer @ FB, ...
- No standard tool *yet*
- Barrier to adoption still too high
- Difficult to express mental models in formal notation

Narrowing the gap

How to narrow the gap between mental models and formal methods?

Contributions

- Starting point: program verifier VerCors
- Industry experience
- Formal methods and software development
- Formal methods and distributed systems



Part 1/3: industry experience

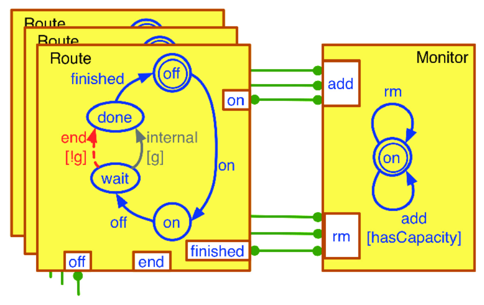
- Applied VerCors at Technolution to tunnel control software
- Found two problems: relevance
- But, also: difficult to apply, difficult to explain
- Goals:
  - Bring VerCors closer to developer mental models
  - Improve language support

Technolution

VerCors

Part 2/3: Formal methods and software development

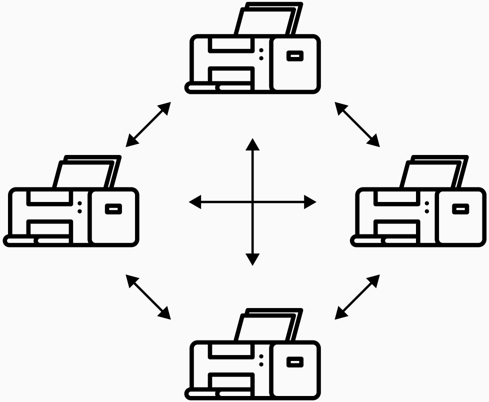
- Unify formal methods and software development
- Novel combination:
  - JavaBIP: component-based software development
  - VerCors: concurrent program verifier
- Showed feasibility, effectiveness, reuse



Source: Anastasia Mavridou

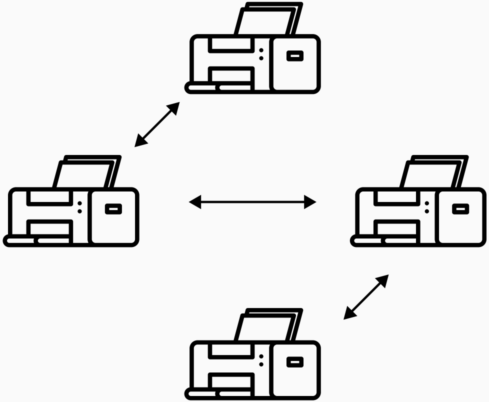
Part 3/3: Formal methods and distributed systems

- Choreographies are a DSL for designing distributed systems
- We extended existing verification tool VeyMont:
    - Shared memory
    - Parameterization
- Case studies done with VeyMont can now be more realistic



Part 3/3: Formal methods and distributed systems

- Choreographies are a DSL for designing distributed systems
- We extended existing verification tool VeyMont:
  - Shared memory
  - Parameterization
- Case studies done with VeyMont can now be more realistic

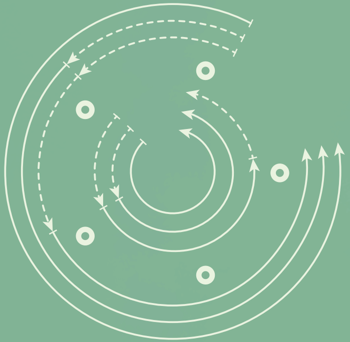


Conclusion

- Formal methods are promising
- But: uptake is limited
- Improved insights into the needs of the industry
- Created and improved two tools to better cater to industry needs
- Also, showing their effectiveness

Bridging the Implementation Gap:

Advancements in Model-Based Concurrent Program Verification



Robert Benjamin Rubbens