Safe Runtime Verification of Real-Time Properties

Christian Colombo
Gordon J. Pace
Gerardo Schneider

FORMATS'09

Motivation

- A lot of applications have real-time properties
  - Web sites (expiration of cookies)
  - Financial systems (detect inactive users)
  - User interfaces (max. response time)
  - Quality of service (reply to queries)
  - Communication protocols

The Challenges

- Not easy to check for such properties without cluttering the code

  (requires more than one point in the program)
- Not trivial to represent real-time properties
- Time passes while events are timestamped and communicated to the monitor
- During monitoring (the time required for checking), properties can be violated

The Challenges

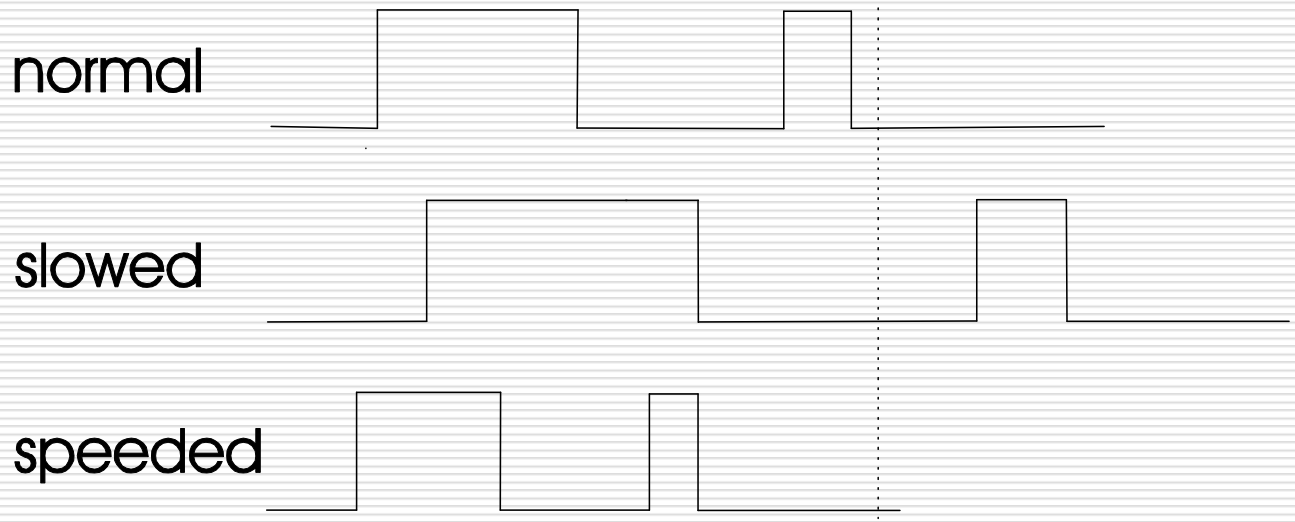
- Bugs can (dis)appear by changing the speed of the program
  - Debugging / Live deployment
  - Better (faster) hardware
  - Server / Backup server
- Monitoring – usually slows down a system
  - Introducing monitors – 3rd party s/w
  - Removing monitors – after testing

Slowdown/Speedup Sensitivity

- There are two main aspects of slowdown/speedup sensitivity:
  - Changes in the system being observed
  - Changes in the satisfaction of properties
- By making assumptions on the changes in the system we can give guarantee on the satisfaction of the properties

Assumptions

- Events do not change their order when adding changing speed
- The system slows down everywhere: i.e. Each interval becomes longer



Main Idea

- Classify properties as:
  - **Slowdown truth preserving** – slowing down the system, the property is not broken
  - **Speedup truth preserving** – speeding up the system, the property is not broken
  - **Slowdown false preserving** – slowing down the system, the property is not "fixed"
  - Etc.

Examples

- No more than 3 bad logins in 10 minutes
- Assume the program is correct
- No bugs will be introduced by slowing it down (eg: by debugging or monitoring)

- If a user tries to access restricted content, an alert should be send within 10 seconds
- No bugs will be introduced by speeding it up (eg: by removing monitors)

Properties of Properties

- Real-time properties can be classified under:
  - Slowdown truth preserving
    - P holds on an interpretation I

    => P holds on any slowdown of I
  - Speedup truth preserving
    - P holds on an interpretation I

    => P holds on any speedup of I

An Interesting Relationship

- The negation of a formula which is slowdown truth preserving is speed up truth preserving (and vice-versa)

Duration Calculus Syntax

- DC ::= ∫P > n integral
  - | ⌈P⌉ almost everywhere
  - | ¬D negation
  - | D ^ D conjunction
  - | D v D disjunction
  - | D ; D chop
  - | □ D always
  - | ◇ D eventually

Duration Calculus

- An interpretation

  $\mathbf{I} \in \text{Time} \times \text{BoolVar} \rightarrow \text{Bool}$

- Satisfaction

  $\mathbf{I} \vdash_{[b,e]} D$

- Validity

  $\mathbf{I} \vdash D$ defined by $\forall n \,.\, \mathbf{I} \vdash_{[0,n]} D$

- Stretching Invariant

  $\mathbf{I} \vdash D \Rightarrow \mathbf{I}_{[\text{stretched}]} \vdash D$

Main Result

Property\ Fragment	$\int > c$	$\int < c$	$\neg$	;	$\wedge$	$\vee$	$\updownarrow$	$\not\updownarrow$
interval-stretch truth preserving	✓	×	☒	✓	✓	✓	✓	✓
interval-stretch false preserving	×	✓	☒	✓	✓	✓	✓	✓
interval-compress truth preserving	×	✓	☒	✓	✓	✓	✓	✓
interval-compress false preserving	✓	×	☒	✓	✓	✓	✓	✓

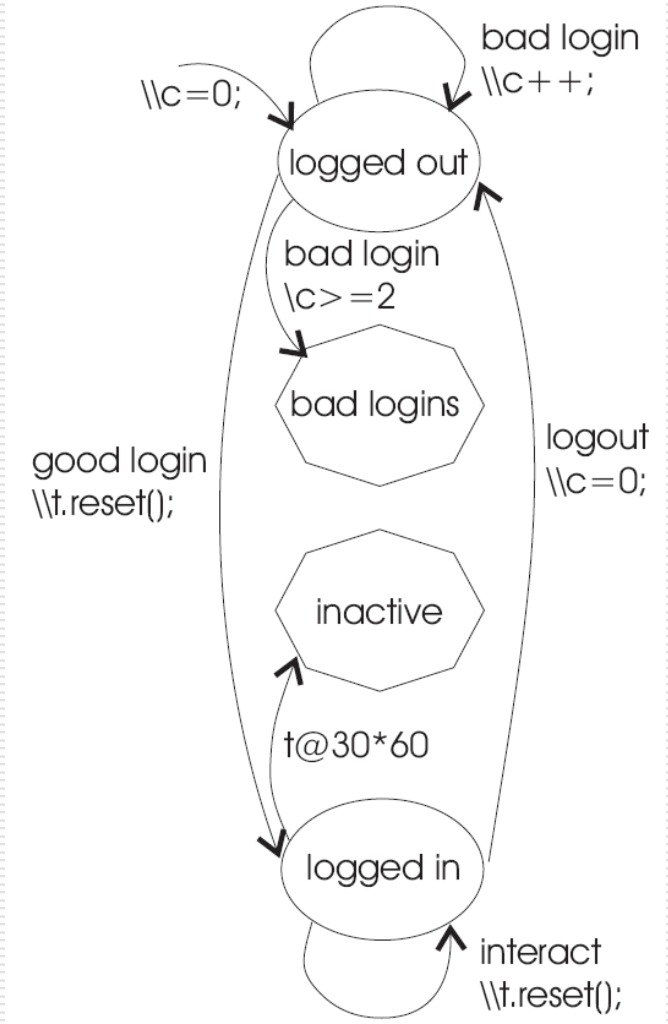
Monitoring & Slowing/Speeding

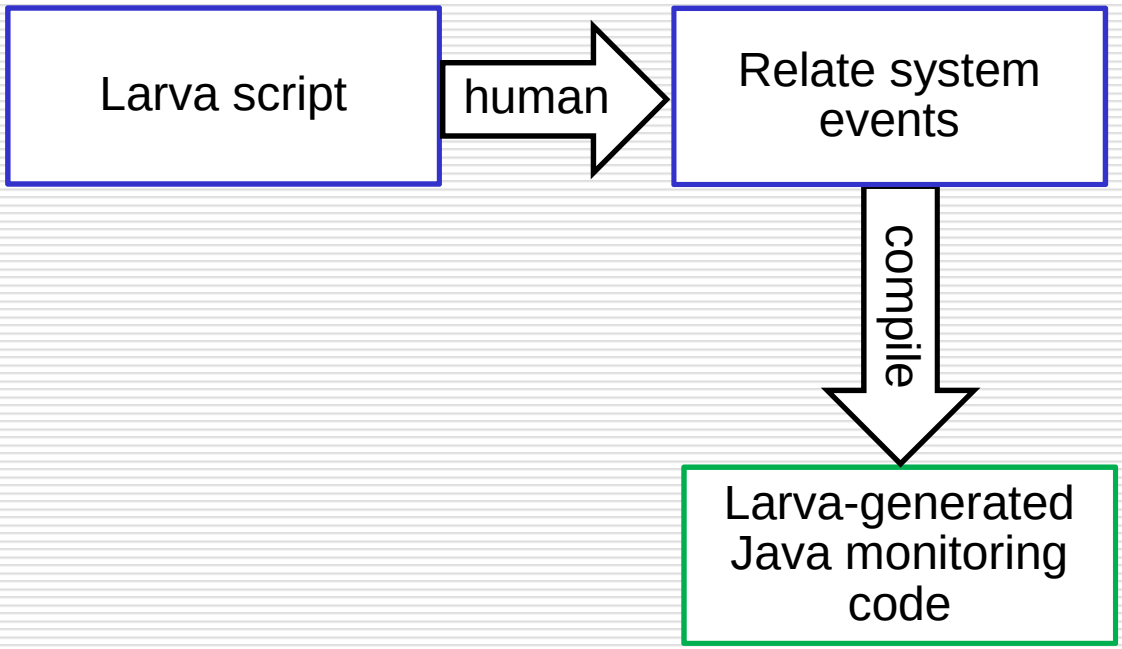
- No false positives
  - If a property holds when monitoring, it still holds without monitors

    (Speedup truth preserving)
- No false negatives
  - If a program is well implemented, inserting monitors will not cause a violation

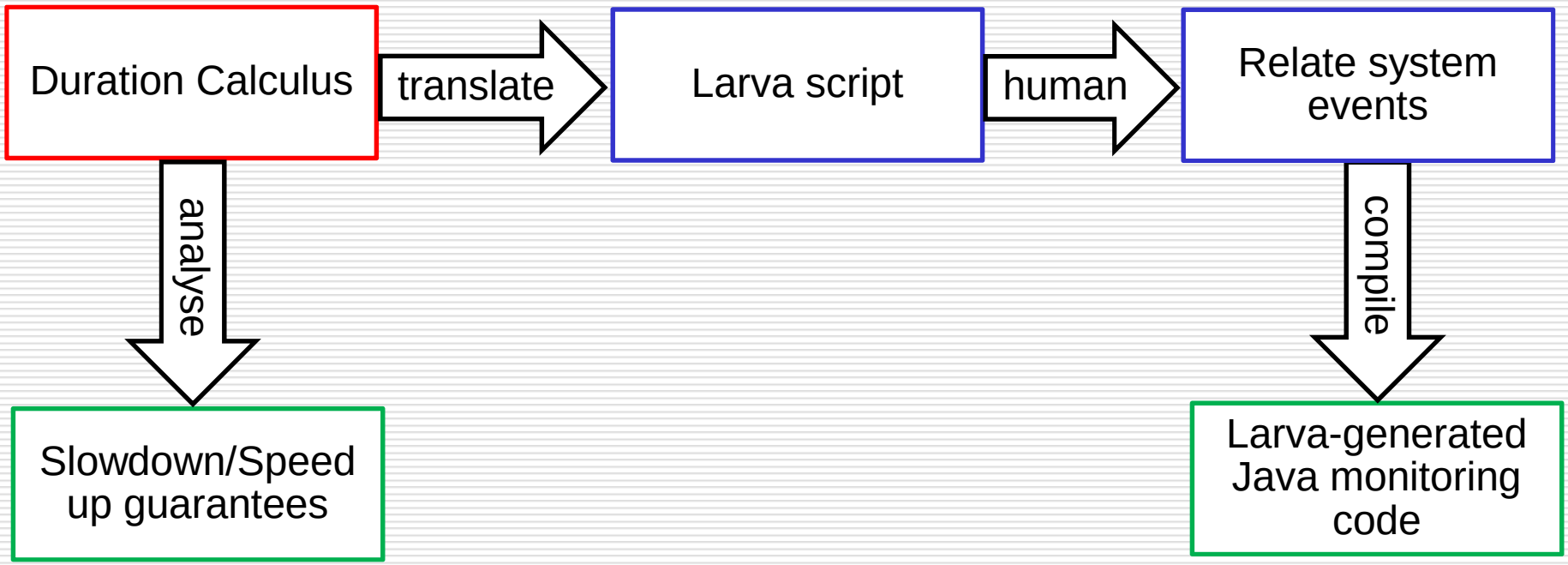
    (Slowdown truth preserving)

LARVA Architecture





Extended Architecture



Case Study

- An intrusion detection system based on TCP packet transmission patterns
- Properties:
  - No incoming connections
  - Low frequency of redirect messages
  - Low frequency of connection failures
- Example:

$\neg(\text{true}; \text{msg}; \ell<2; \text{msg}; \ell<2; \text{msg}; \text{true})$

Conclusions & Future Directions

- An attempt to make runtime monitoring of real-time properties more predictable
- Theory not only for monitoring and not only for duration calculus
  - Done on Timed Regular Expressions
  - To be carried out on Timed Automata
- Different time transforms on different variables
- Bounded stretching and compressing

?

General Architecture