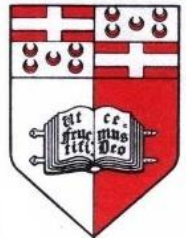


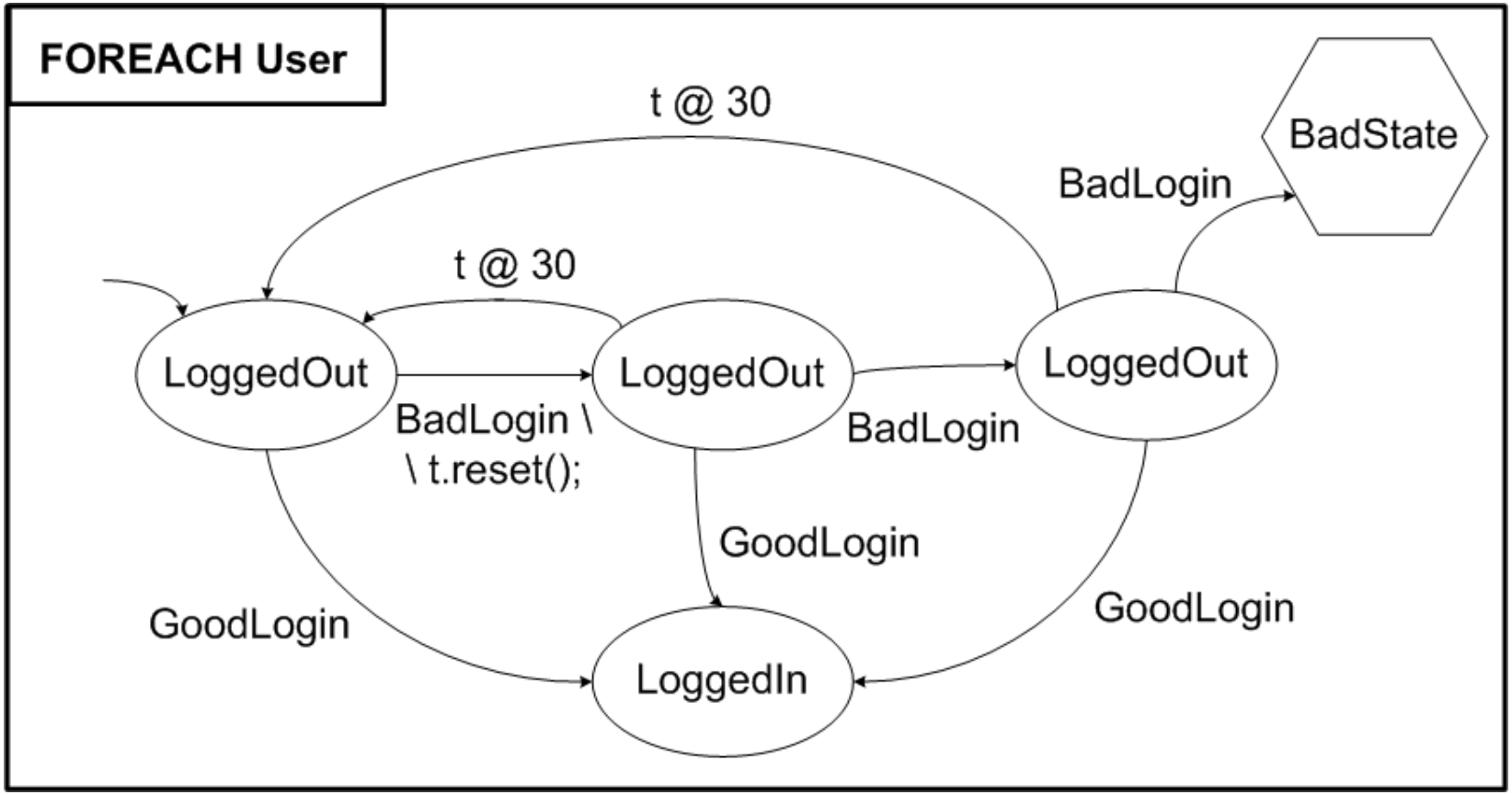
LarvaStat: Monitoring of Statistical Properties

Christian Colombo
Andrew Gauci
Gordon J. Pace

Introduction

- Most RV tools focus on the verification of functional system properties.
- Collecting statistics over system traces allows for the verification of **non-functional requirements**.
- LarvaStat: extending LARVA with statistical capabilities.

LARVA


FOREACH User
t @ 30
BadState
BadLogin
t @ 30
LoggedOut
LoggedOut
LoggedOut
BadLogin \
\ t.reset();
BadLogin
GoodLogin
GoodLogin
GoodLogin
LoggedIn

LarvaStat

- An event based runtime verification framework
  - Adds statistical constructs to LARVA
- Facilitates statistics collection at runtime
- Allows for runtime verification based on statistical information
- Addition of the **statistical event**

Collecting Statistics

- The **statistical aggregator** specification entails:
  - (i) Initial memory valuation
  - (ii) A statistic update rule with:
    - (i) The event and condition upon which the update is triggered
    - (ii) A function on how to update the value of the statistic
    - (iii) The event on which to signal the updated value of the statistic

- Allows for **multilayered statistics**.
  - e.g. the largest maximum, or the average minimum

Collecting Statistics

- Example statistic aggregator:

```
POINTSTAT UsersLoggedIn : Integer {

    INIT { UsersLoggedIn.setValue(new Integer(0)); }

    EVENTS { successfulLogin() }

    CONDITION { }

    UPDATE { UsersLoggedIn.setValue(
                    UsersLoggedIn.getValue() + 1); }

}
```

Intervals of Interest

- The **Statistic aggregator over intervals of interest** is specified through:
  - (i) A statistic aggregator
  - (ii) An event and condition marking an **interval opening**
  - (iii) An event and condition marking an **interval closing**

- Statistics collection over time intervals.

Intervals of Interest

- An example:

```
INTERVALSTAT byteCount : Integer {

    INIT {  byteCount.setValue(new Integer(0)); }

    EVENTS { sendInfo() }

    CONDITION { }

    INTERVAL {
        OPEN [ downloadStarting() ]
        CLOSE [ downloadComplete() ]
    }

    UPDATE { byteCount.setValue(
                byteCount.getValue() + bufferSize); }

}
```

Case Study (1)

- A probabilistic intrusion detection system and system profiler over an ftpd implementation.

- System profiler quantifies system performance
  - measures current system load
  - analyses system load history

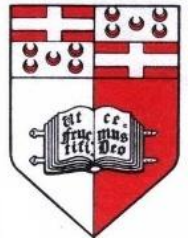
- IDS identifies incorrect or suspicious behaviour.
  - Markov chain analysis of user action sequence
  - analysis of user download and upload behaviour

Case Study (2)

- Choice of monitored users based on **user risk factor** and **system load**.
- Case study involves 20 incrementally computable statistics; implementation did not alter the original ftpd code.
- An approximate 9% overhead was measured.

Conclusions

- LarvaStat: extending LARVA with statistical capabilities.
- Makes the collection of statistics over system traces easier.
- Integrates statistics collection with runtime verification, facilitating the verification of **non-functional requirements**.
- The probabilistic intrusion detection system and integrated system profiler case study.



Thank you!

Christian Colombo
Andrew Gauci
Gordon J. Pace