Runtime Verification of Contracts for Java Programs

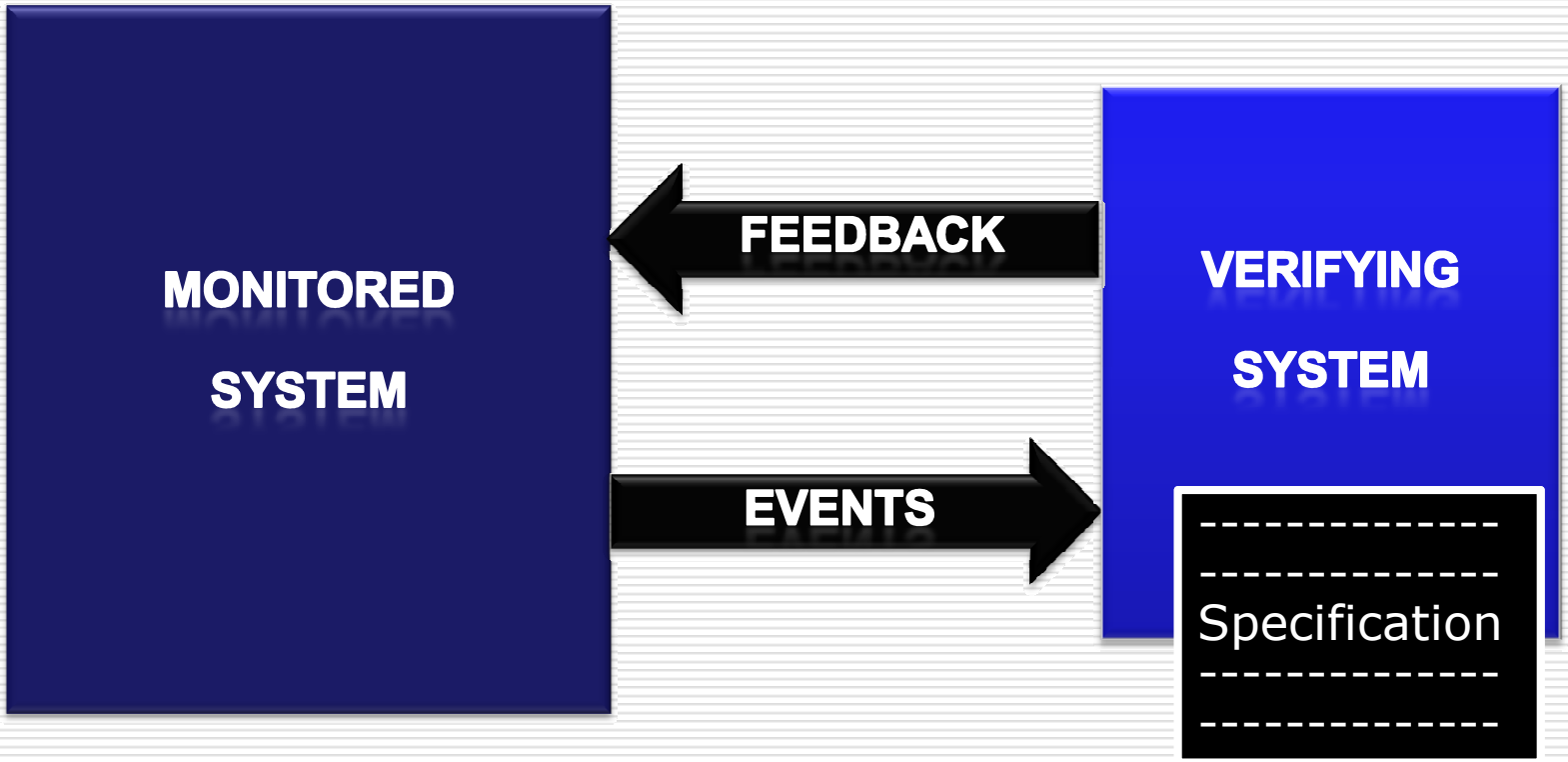
Christian Colombo
Gordon J. Pace
Gerardo Schneider

FLACOS - November 2008

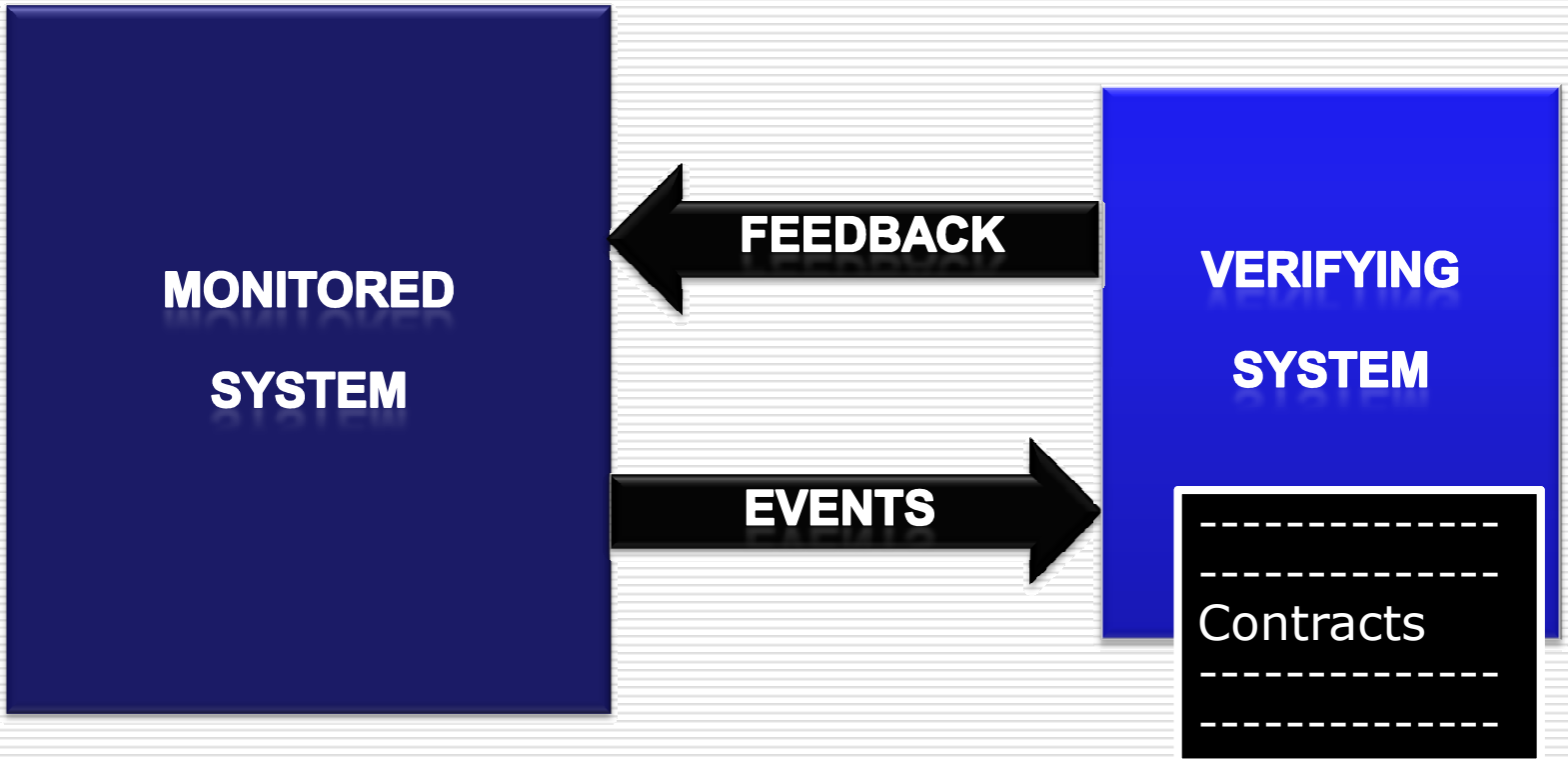
RV & Contracts

- In SOA we are concerned with security and trust.
- Model checking is not scalable.
- Testing lacks coverage.
- Particular behaviour only emerges during normal use of service composition.
- Runtime verification monitors the behaviour during runtime, scales up.
- Real-time properties / overheads.
- Contracts may have conflicts.

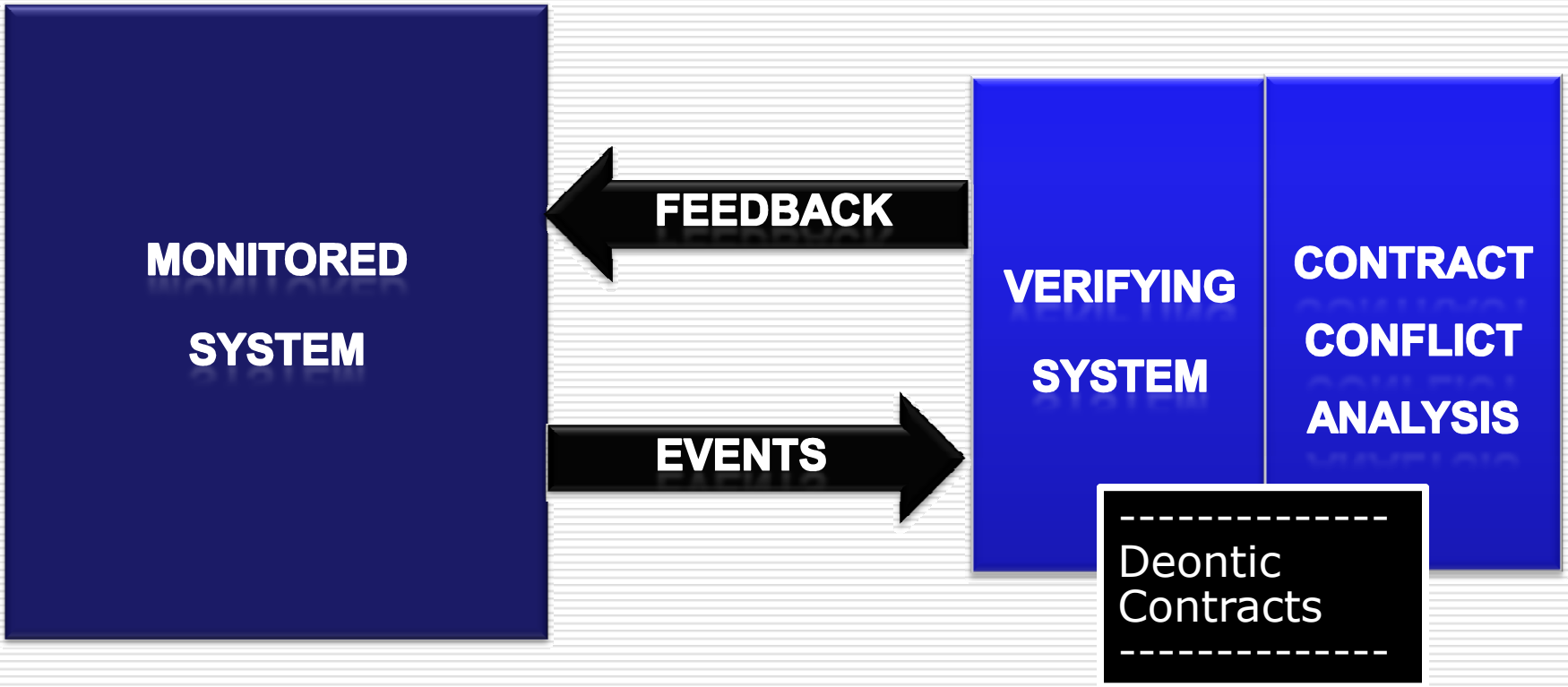
Runtime Verification



Runtime Verification



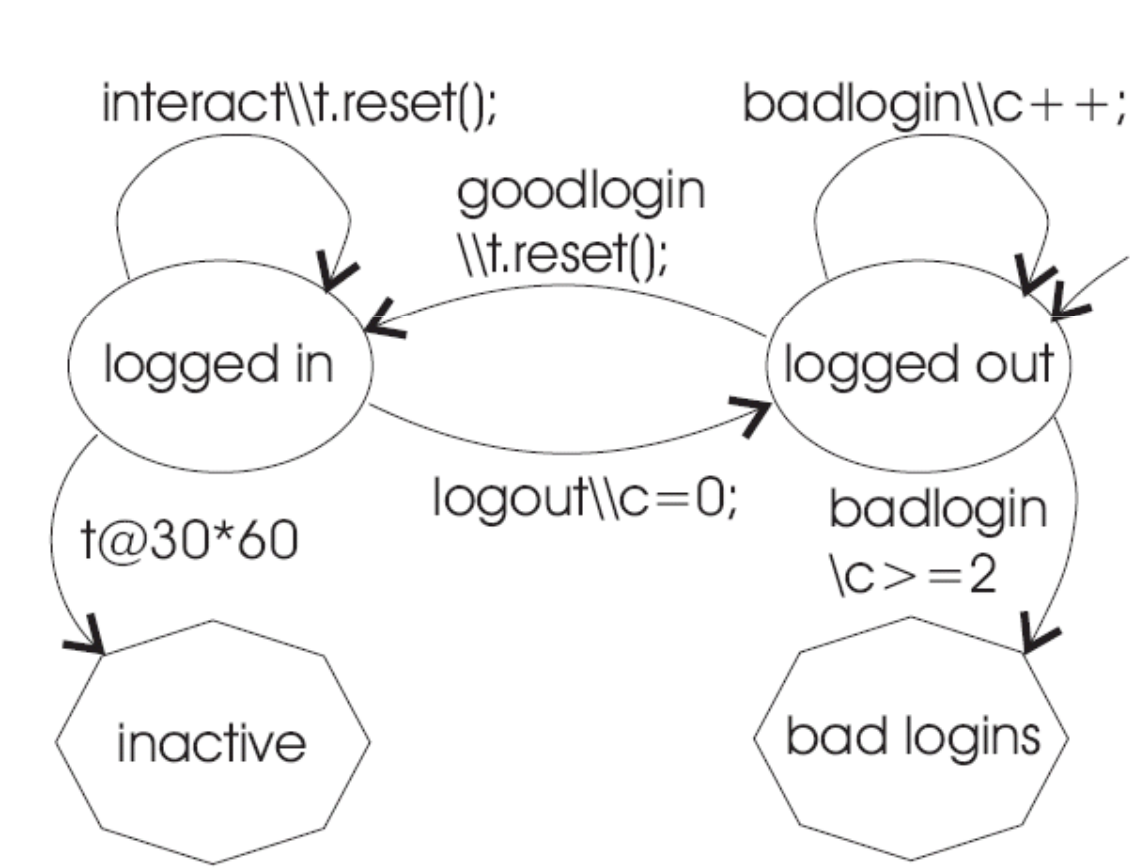
Runtime Verification



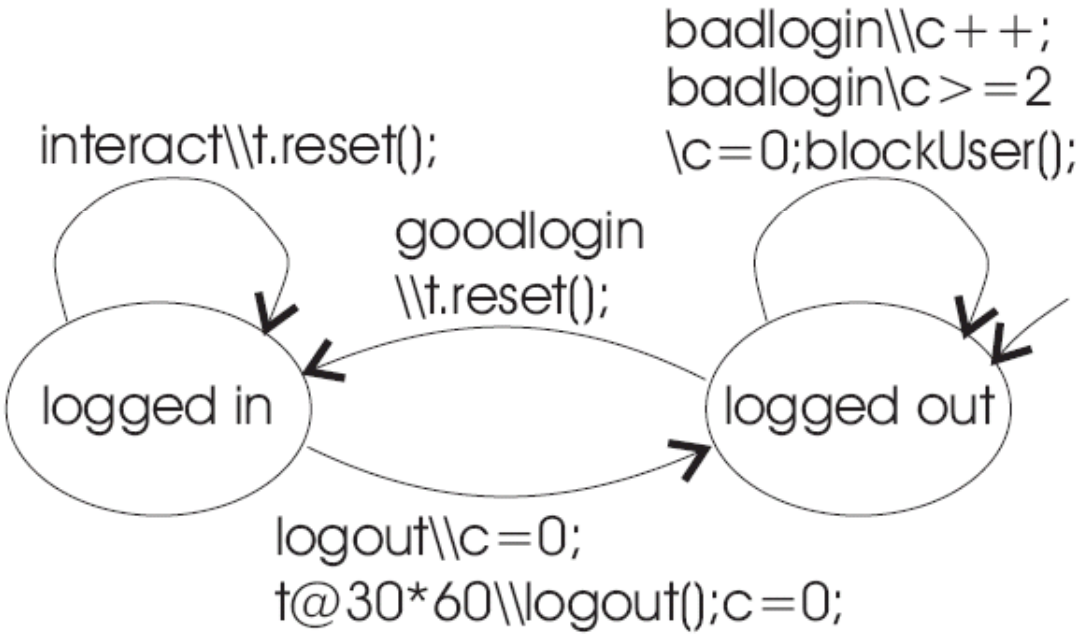
Dynamic Automata with Timers & Events (DATE)

- Communicating symbolic automata enriched with **events** and **timers**.
- Automata are automatically replicated according to context: hence **dynamic**.
- Supports:
  - Conditions and actions on transitions
  - Real-time
  - Communication between automata

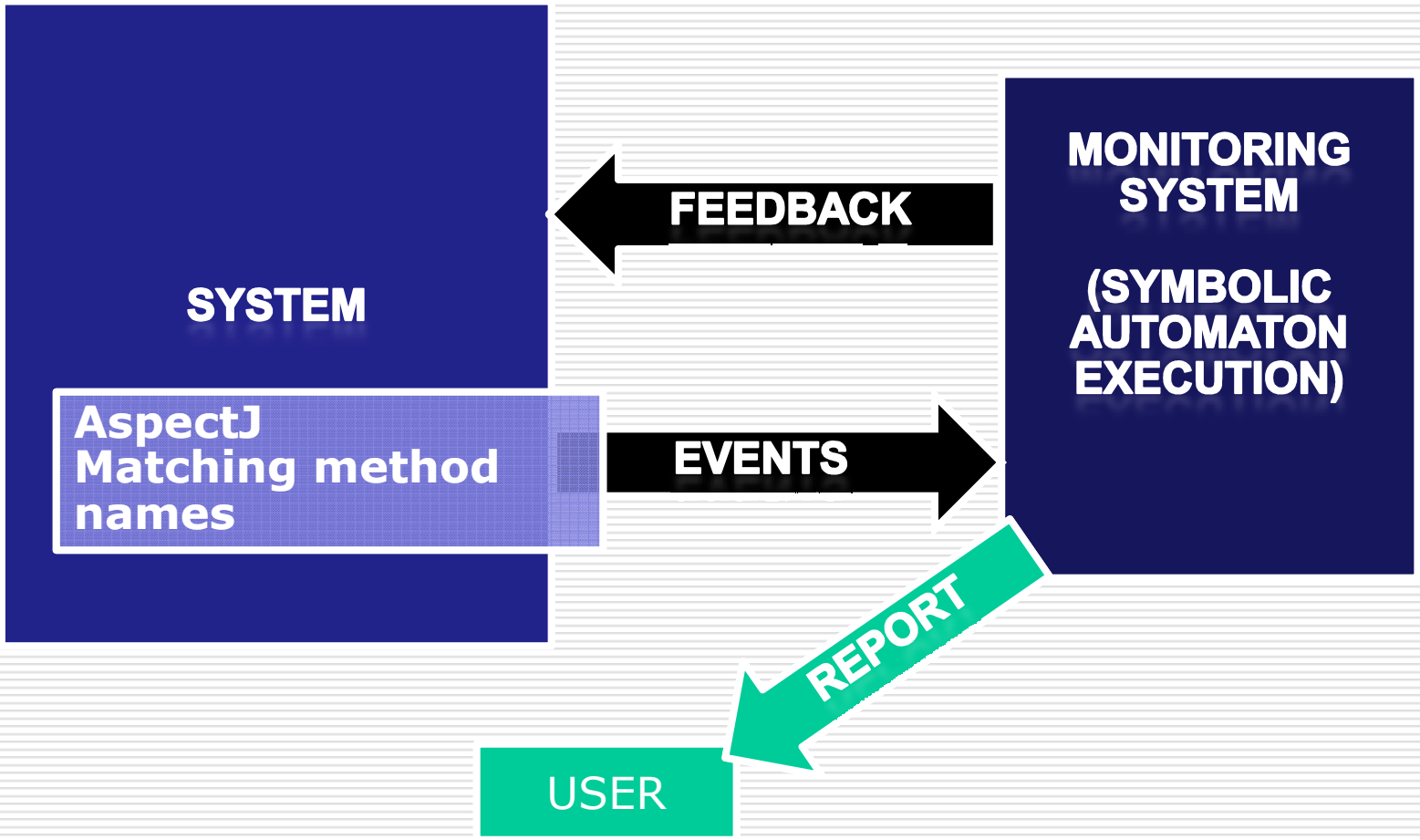
An Example (1)


interact\\t.reset();
badlogin\\c++;
goodlogin
\\t.reset();
logged in
logged out
logout\\c=0;
t@30*60
badlogin
\c>=2
inactive
bad logins

An Example (2)


badlogin\\c++;
badlogin\c>=2
\c=0;blockUser();
interact\\t.reset();
goodlogin
\\t.reset();
logged in
logged out
logout\\c=0;
t@30*60\\logout();c=0;

LARVA - Architecture



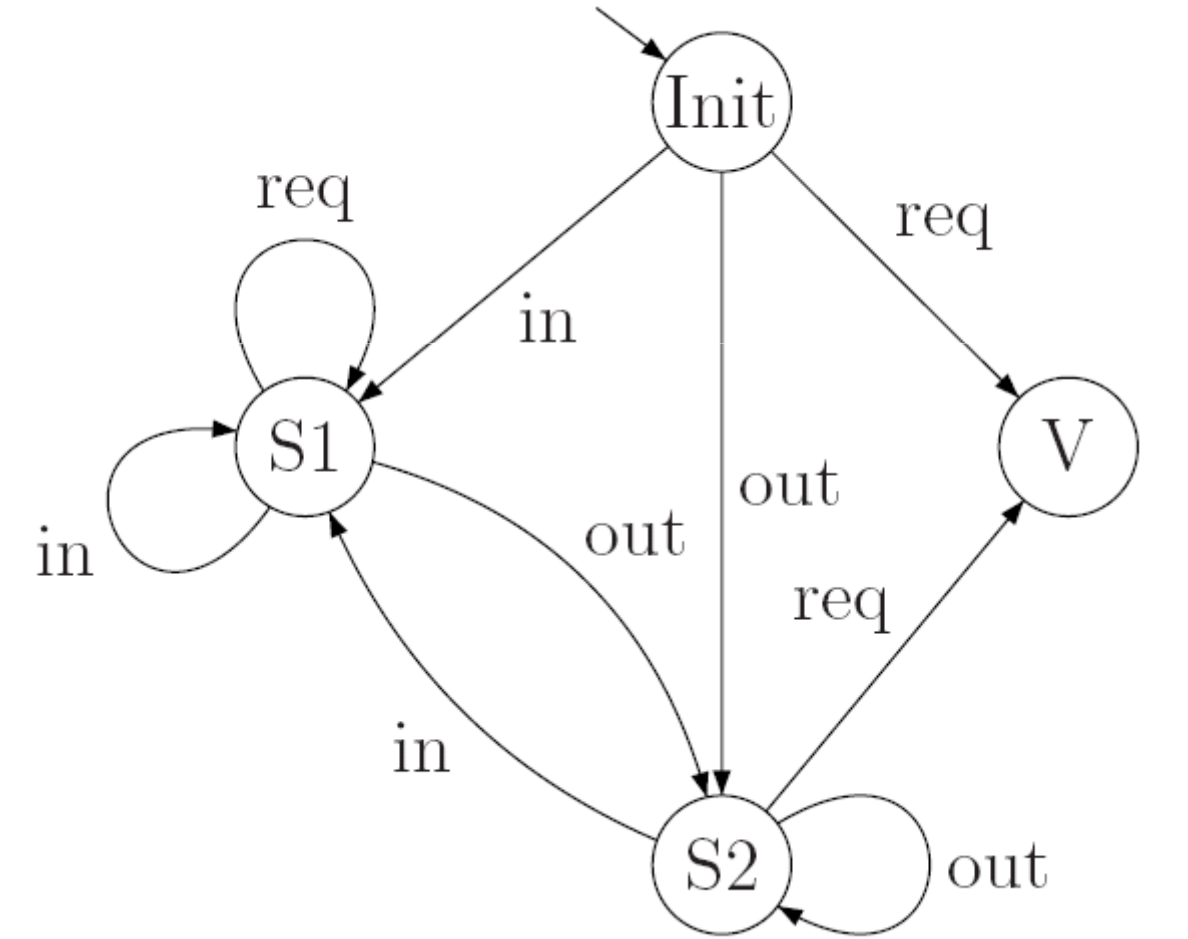
Contract Language Example

$$[\overline{login}^*]F(request)$$

$$\wedge\,[1^*][logout][\overline{login}^*]F(request)$$

$$\wedge\,F(request)$$

Contract Language to Automata



Contract Language to LARVA

```
EVENTS {
  login = {*.login()}
  logout= {*.logout()}
  request= {*.requestItem()}
}

PROPERTY clcontract {
  STATES {
    BAD { V }
    NORMAL { S1 S2 }
    STARTING { Init }
   }
```

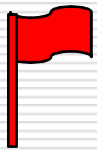
```
TRANSITIONS {
  Init -> S1 [login]
  Init -> V [request]
  Init -> S2 [logout]
  S1 -> S1 [login]
  S1 -> S1 [request]
  S1 -> S2 [logout]
  S2 -> S2 [logout]
  S2 -> V [request]
  S2 -> S1 [login]
 }
}
```

Contradictions in Contracts

$$[\overline{login}^*]F(request)$$

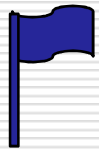
$$\wedge\,[1^*][logout][\overline{login}^*]F(request)$$

$$\wedge\,F(request)$$



Contradiction Detected!

$$O(request)$$



Ongoing Work

- Working closely with industry
- Guarantees on the effect of monitoring – memory and time
- Identifying better notations
- Investigating compensable actions

Conclusions

- Mathematical framework – DATE
- Implemented useable tool – LARVA
- Highly expressive (incl. real-time)
- Evolving theory with practical guarantees
- Can monitor contracts
- Find contradictions in contracts
- Future prospects of collaboration and improvement of current framework

Questions

- ?