Specification and Verification of Trustworthy Component-Based Real-Time Reactive Systems

Authors:
Vasu Alagar and Mubarak Mohammad
Concordia University
Montréal, Canada

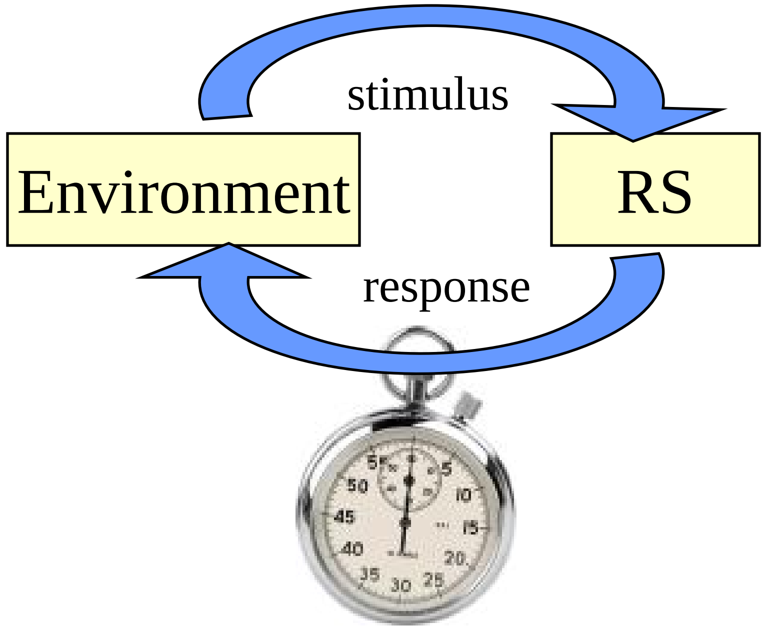
Presented by:

Mubarak Mohammad

Agenda

- Context
- Motivation
- Contributions:
  - A formal methodology for developing trustworthy RTRS
  - Automatic generation of component behavior
- Modeling Checking
- Example
- Conclusion

Real-Time Reactive Systems (RTRS)

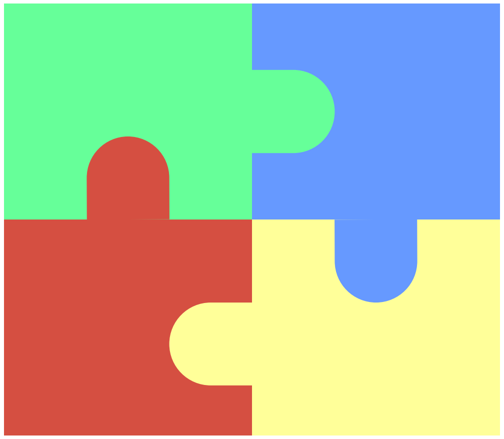


Trustworthiness

- A *trustworthy* system is a system that can be depended upon for quality of service.
- RTRS are required to be trustworthy due to:
  - Their non-terminating behavior
  - The critical contexts it operate in
- In order to trust, the *credentials* of trust should be defined and examined:
  - Safety
  - Security

Component-Based Development (CBD)

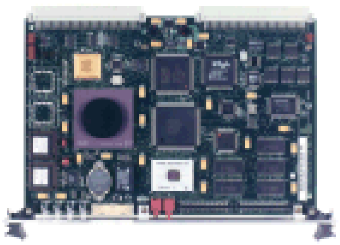
- Advantages [1]:
  - Reusability
  - Managing design complexity
  - Reducing time and effort
  - Increasing productivity



- Trustworthy component: a component that guarantees safe and secure interactions.

[1] Ivica Crnkovic and Magnus Larsson, editors. *building reliable component-based Software Systems*. Artech House Publishers, 2002.

Motivation

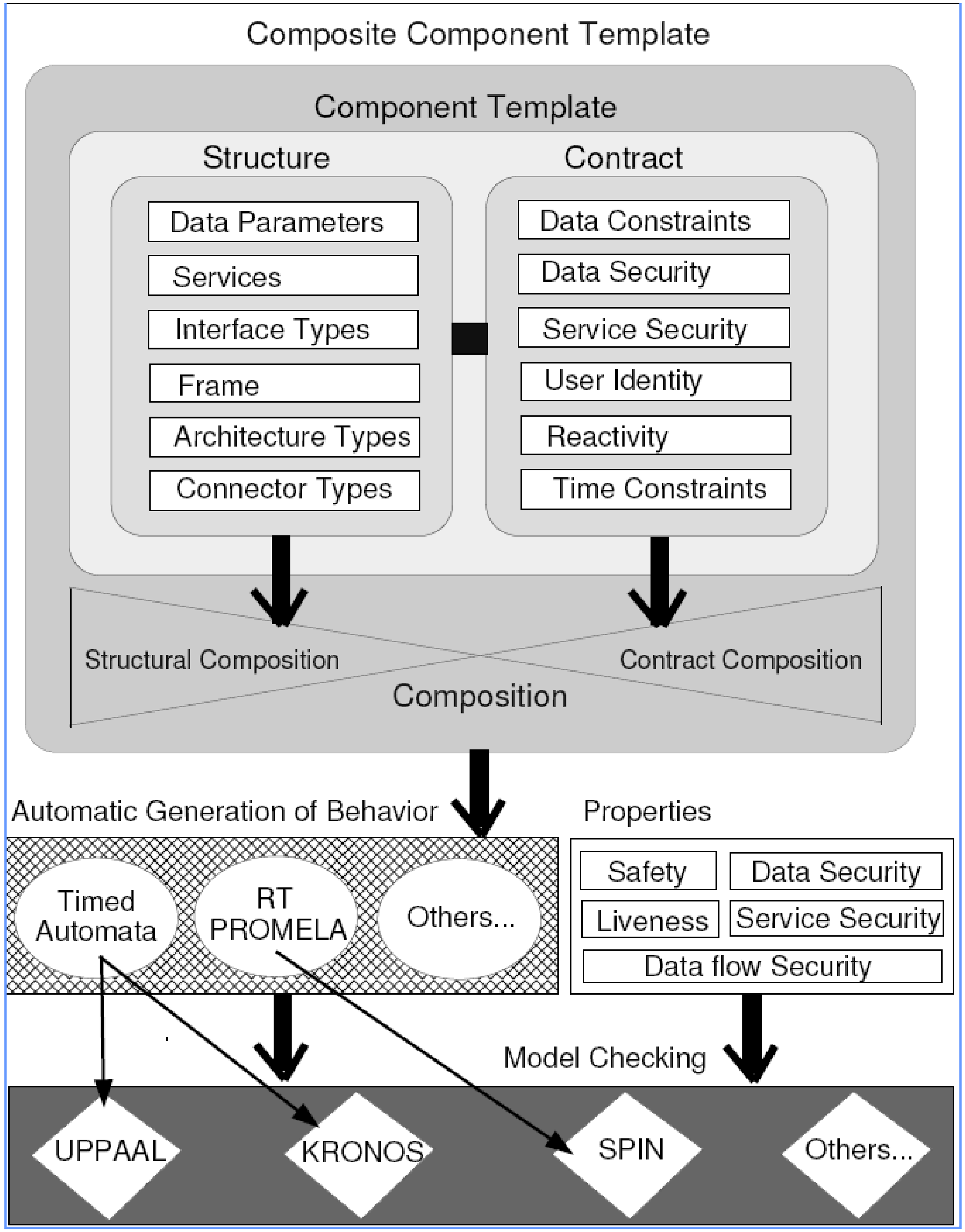
- The design of RTRS should rely on rigorous formal model to be formally verifiable.

- Provide a formal approach for the development of trustworthy component-based RTRS.

Formal Methodology

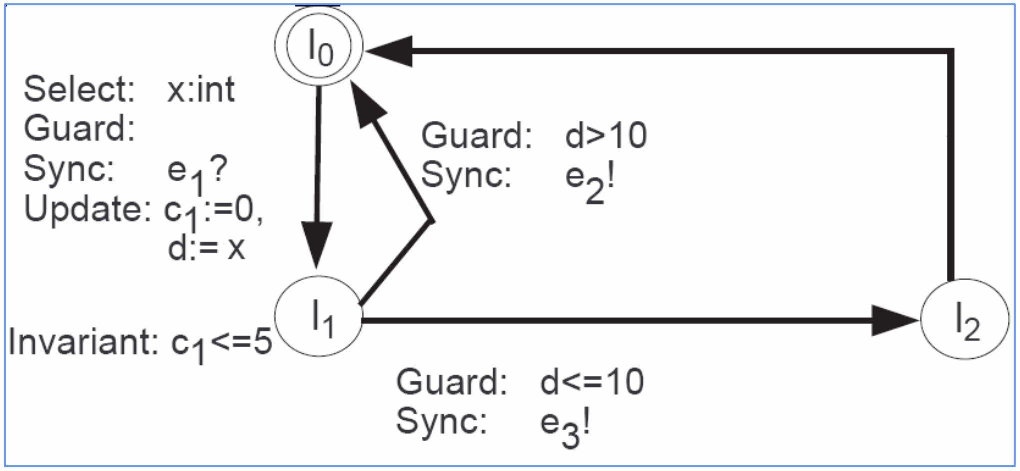
- Verification-oriented design methodology that involves:
  1. Formal specification of component structure and functional/nonfunctional (trustworthiness) properties [2] ;
  2. Automatic generation of component behavior; and
  3. Verification of functional/nonfunctional component behavior using model checking.

[2] Vasu Alagar and Mubarak Mohammad. A component model for Trustworthy Real-Time Reactive Systems Development. *In Proceedings of Formal Aspects of Component Systems*, Sophia-Antipolis, France, Sept 2007.


Composite Component Template
Component Template
Structure
Contract
Data Parameters
Services
Interface Types
Frame
Architecture Types
Connector Types
Data Constraints
Data Security
Service Security
User Identity
Reactivity
Time Constraints
Structural Composition
Contract Composition
Composition
Automatic Generation of Behavior
Properties
Timed Automata
RT PROMELA
Others...
Safety
Data Security
Liveness
Service Security
Data flow Security
Model Checking
UPPAAL
KRONOS
SPIN
Others...

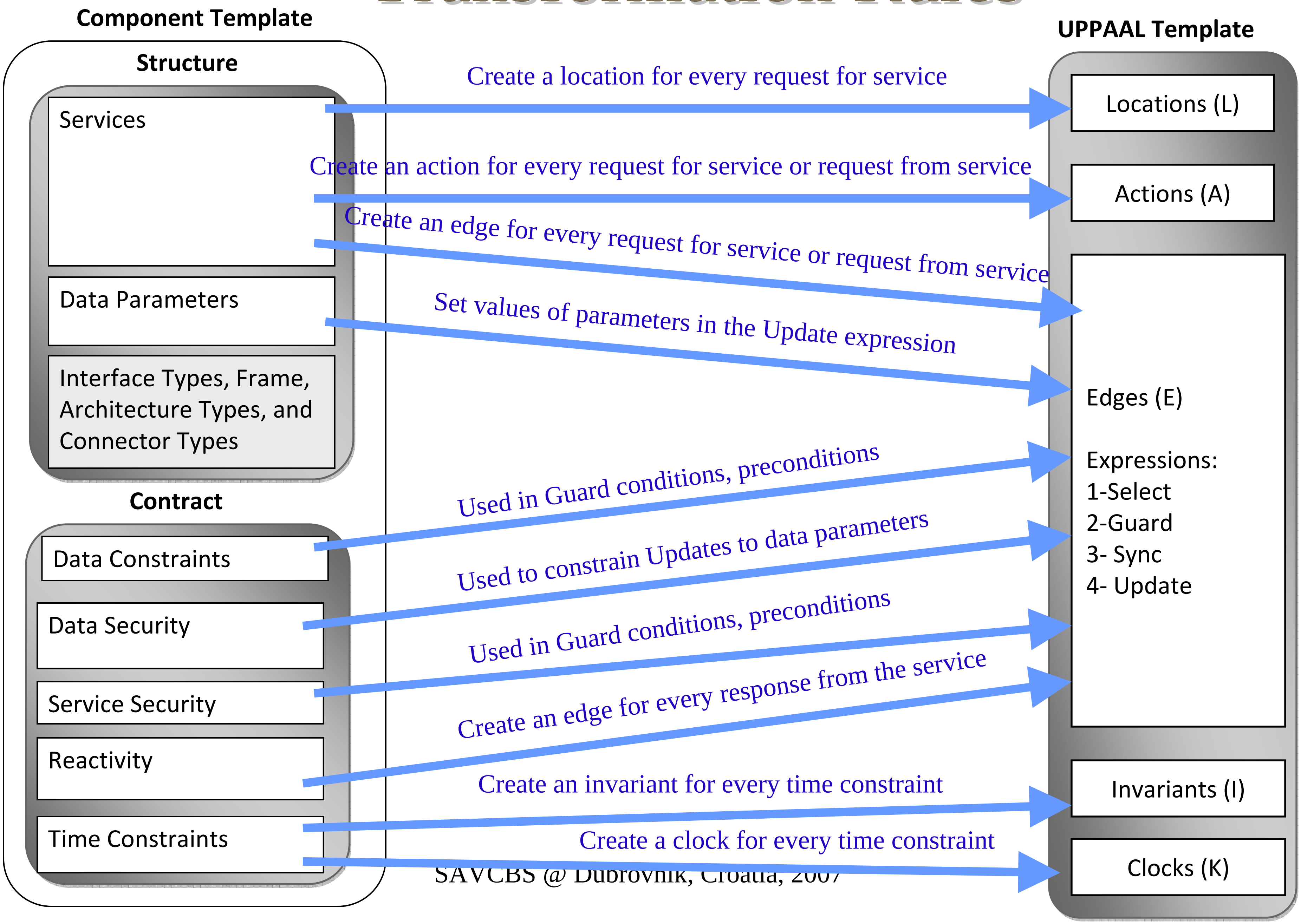
UPPAAL Modeling Language [3]

- Time Automata (L, l_0, K, A, E, I)
  - L is a set of locations denoting states;
  - l_0 is the initial location;
  - K is a set of clocks;
  - A is a set of actions, events causing transitions;
  - E is a set of edges, transition specifications; and
  - I is a function assigning clock constraints to locations as invariants.

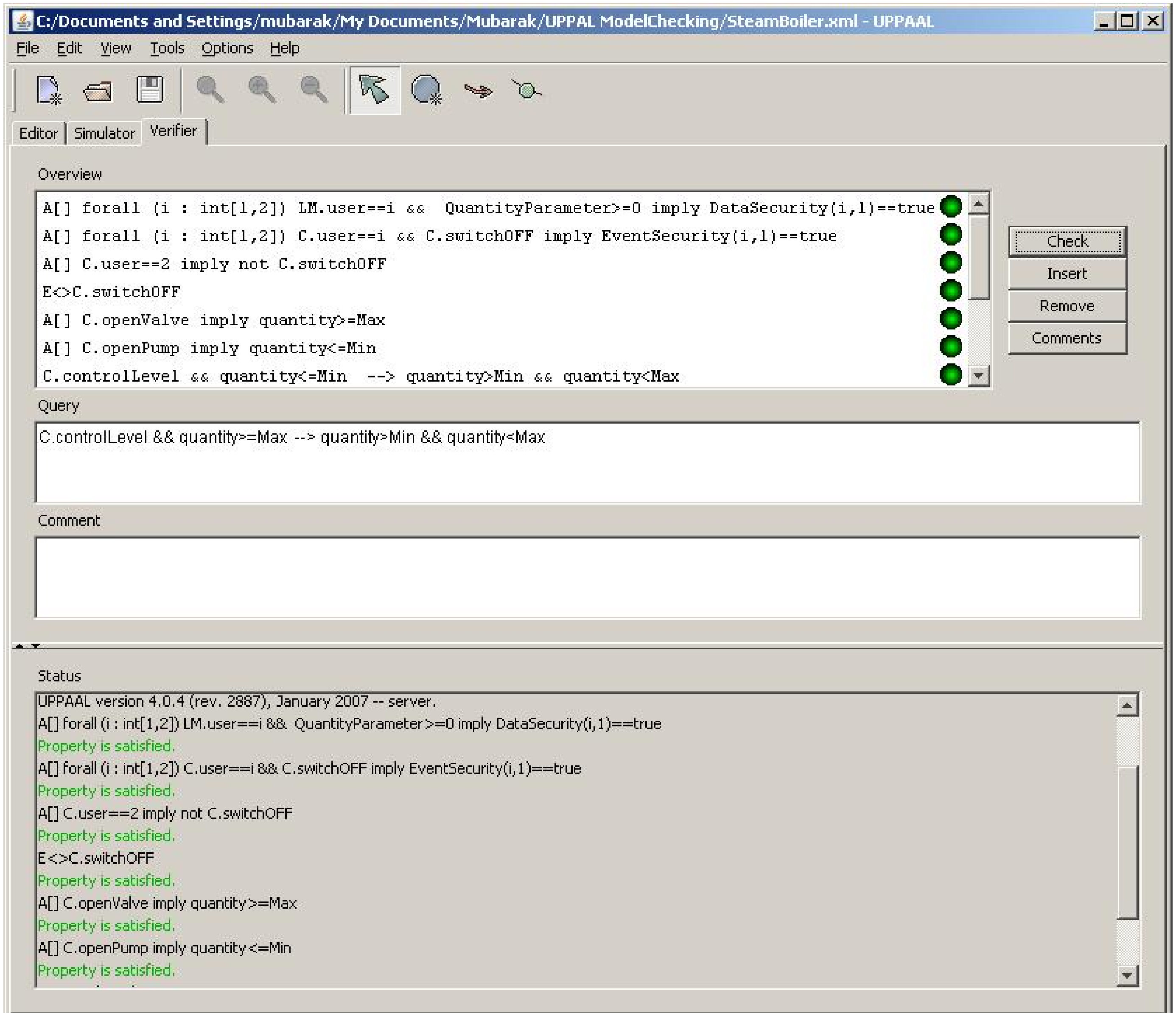


[3] Gerd Behrmann, Alexandre David, and Kim G Larsen. A tutorial on UPPAAL. In *Proceedings of SFM-RT'04*, 2004.

Transformation Rules



Model Checking



Example

Events = {e1:*Stimulus*, e2:*Response*, e3:*Request*},

Data Parameters(e1)={d:Int},

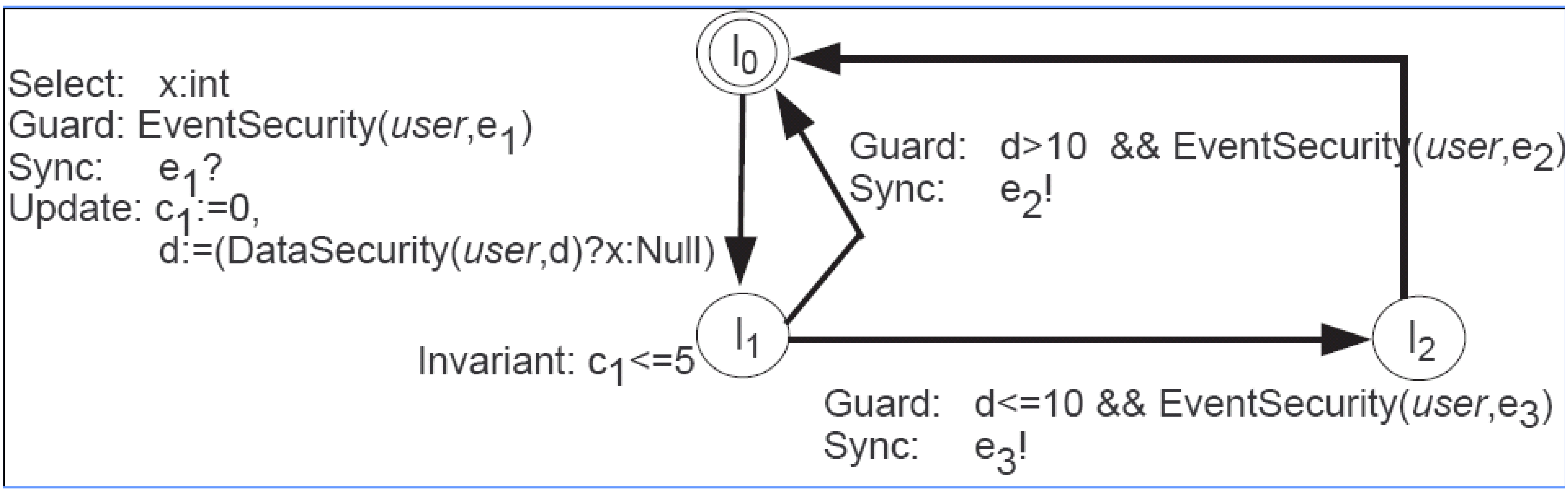
Reactivity(e1)={e2,e3},

Data Constraint(e1,e2): d>10,

Data Constraint(e1,e3): d<=10,

Time Constraint(e1,e2)=[0,5],

Time Constraint(e1,e3)=[0,5]



Conclusion

- We plan to evaluate our method on problems from different domains where safety and security are critical.
- We are investigating the requirements of a trustworthy ADL.
- We are building a visual interface tool for designing trustworthy RTRS.