Reactive Rules and Applications

Katerina Ksystra

19/12/2018

WORKSHOP - Rules: Logic and Applications

What are Reactive rule-based systems?

- Agents whose behavior is defined by **Reactive rules**
  - E-commerce platforms (react to user actions, e.g. put an item in the basket)
  - Web services (react to notifications, e.g. SOAP messages)
  - Active databases

Reactive rules

- Reactive rules families:
  - Production rules (PR)
  - Event Condition Action rules (ECA)
  - Knowledge representation rules (KR)
  - Complex Event Processing (CEP)

Reactive rules

- Bridge the **gap** between existing and dynamic Web
- Used to specify reactive systems
- Support ad-hoc, flexible, dynamic workflows that change at run-time

- No formal verification tools
- **Interaction** of rules during execution
- **Unpredictable** behavior of rules

Problem

- **Not enough** appropriate formal **tools**
  - for **intelligent agents** with complex behavior
- Addressing their **characteristics** consistently is **not trivial**
- **Need** for developing methodologies **targeted** to those systems
- Meet their **challenges** and **requirements**

Formal methods

- **Unambiguous** description of a system (Specification)
- Effective **analysis** of desired properties (Verification)
- Algebraic specification languages
  - Combinations of **logical** systems
- **CafeOBJ**
  - **Specification** of complex systems as abstract **state machines**
  - **Verification** of safety properties through **theorem proving**

Why Formal methods?

- Testing and simulation **not enough** to guarantee system's behavior
- Better **understanding** of the specified system
- **Reasoning** support
- A system does **not** have **defects**
- A system does **satisfy** desirable **properties**
- Constructing **reliable** distributed systems

In a nutshell…

- Need for verifying **structure and safety** properties about reactive rules
- Exploit CafeOBJ's support for **rewriting logic** specifications
- Express reactive rules as rewrite rules

In a nutshell…

- PR and Rewriting logic / PR and Equational Logic in CafeOBJ
- ECA and Rewriting logic / ECA and Equational Logic in CafeOBJ
- Proving termination properties using rewriting logic approach
- Proving confluence properties using rewriting logic approach
- Simulating rules' execution behavior using rewriting logic approach
- Proving safety properties using both approaches

A light-control intelligent system

- **ECA** rules specifying a **light-control** intelligent system
- Energy consumption **reduction**
  - **Turning off** the lights in **unoccupied** rooms or when the occupant is **asleep** using **sensors**
- Automatic adjustment for indoor light intensity based on the **outdoor** light intensity

environmental variables	Mtn, ExtLgt, Slp
local variables	lgtsTmr, intLgts
external events	MtnOn activated when Mtn = true MtnOff activated when Mtn = false ExtLgtLow activated when ExtLgt $\leqslant$ 5
internal events	SecElp, LgtsOff, LgtsOn, ChkExtLgt, ChkMtn, ChkSlp
(R1)	When the room is unoccupied for 6 minutes, turn off lights if they are on.
r1	on MtnOff if (intLgts > 0 and lgtsTmr = 0) do set (lgtsTmr, 1) par activate (SecElp)
r2	on SecElp if (lgtsTmr $\geqslant$ 1 and lgtsTmr < 6 and Mtn = false) do increase (lgtsTmr, 1)
r3	on SecElp if (lgtsTmr = 6 and Mtn = false) do set (lgtsTmr, 0) par activate (LgtsOff)
r4	on LgtsOff do (set (intLgts, 0) par activate (ChkExtLgt))
(R2)	When lights are off, if external light intensity is below 6, turn on lights.
r5	on ChkExtLgt if (intLgts = 0 and ExtLgt $\leqslant$ 5) do activate (LgtsOn)
(R3)	When lights are on, if the room is empty or a person is asleep, turn off lights.
r6	on LgtsOn do (set (intLgts, 6) seq activate (ChkMtn))
r7	on ChkMtn if (Slp = true or (Mtn = false and intLgts >= 1)) do activate (LgtsOff)
(R4)	If the external light intensity drops below 5, set the lights intensity to 6 and check if the person is asleep. If the person is asleep, turn off the lights.
r8	on ExtLgtLow do set (intLgts, 6) par activate (ChkSlp)
r9	on (ChkSlp if (Slp = true) do set (intLgts, 0)
(R5)	If the room is occupied, set the lights intensity to 4.
r10	on MtnOn do set (intLgts, 4) par set (lgtsTmr, 0)

- Environmental variables: store the values measured by sensors
- Mtn: motion sensor detecting whether the room is occupied or not
- Slp: pressure sensor detecting whether the person is asleep or not
- ExtLgt: light sensor for monitoring the outdoor lighting
- MtnOn, MtnOff, ExtLgtLow: external events (activated by environmental variables)
- Internal events: internal actions

Desired safety property:

- *The lights cannot be turned off if someone is in the room and he/she does not sleep*
- `inv1(S) = (not((intLgts(S) = 0) and (Mtn(S) = true) and (Slp(S) = false))`

Comparison of methodologies

Equational approach

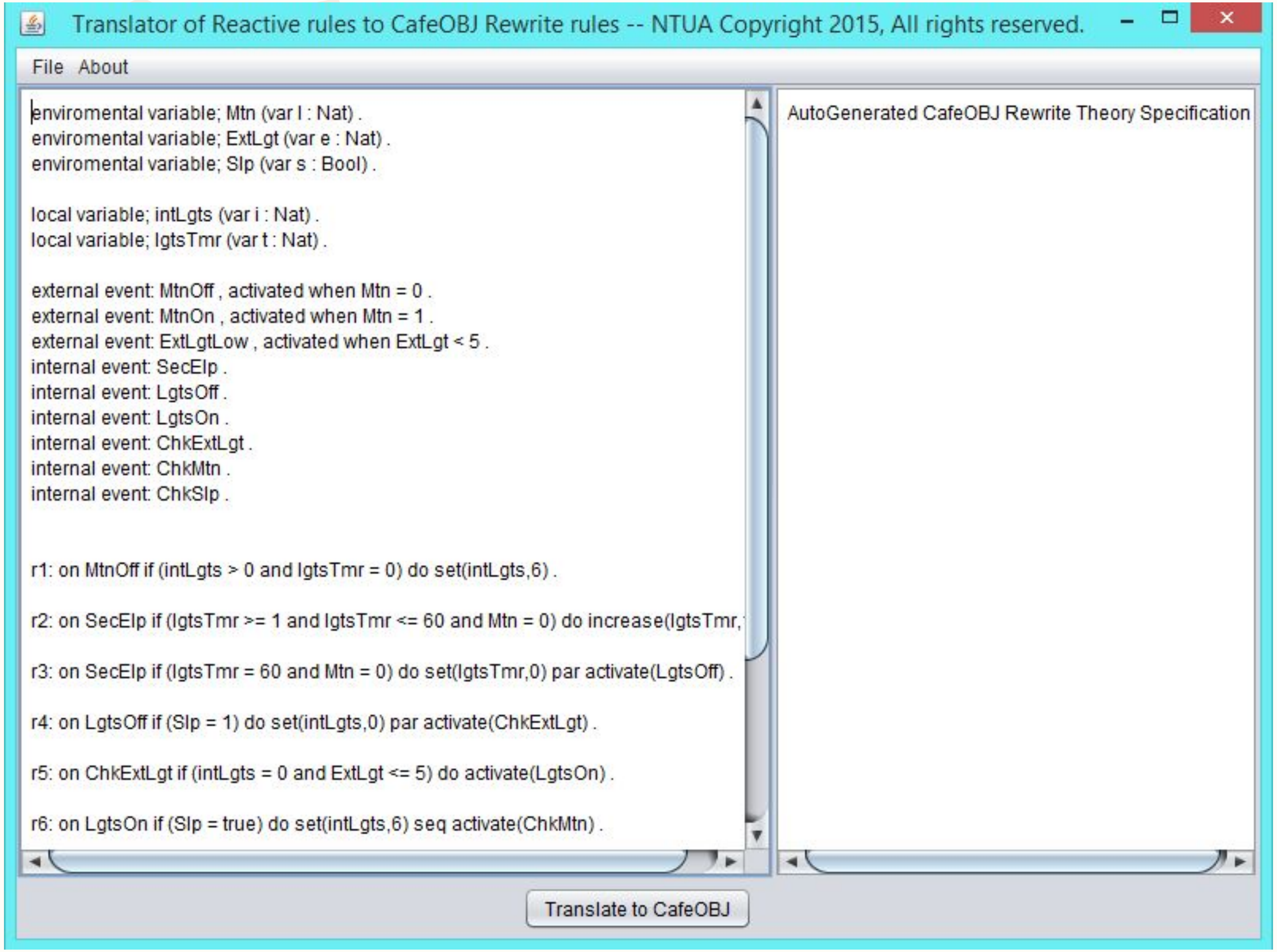
- More expressive formalism
- **Preferred** when the reactive rule-based system is complex

Rewriting approach

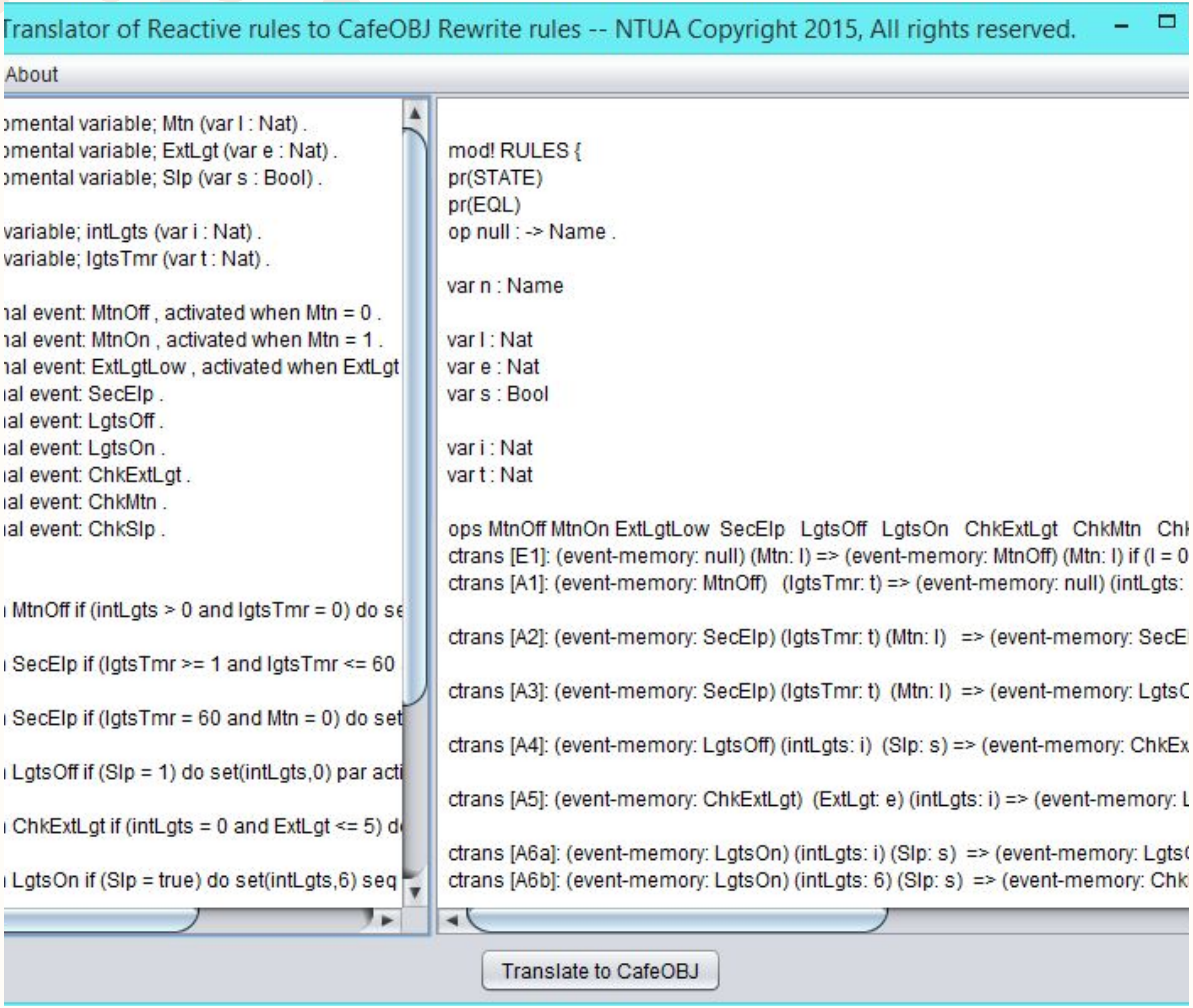
- More natural and easier to use from reactive rules researchers
- Seamless framework for verifying both safety properties and structure errors
- Supports both theorem proving and model checking techniques
- **Preferred** in most cases

Tool: from reactive to rewrite rules

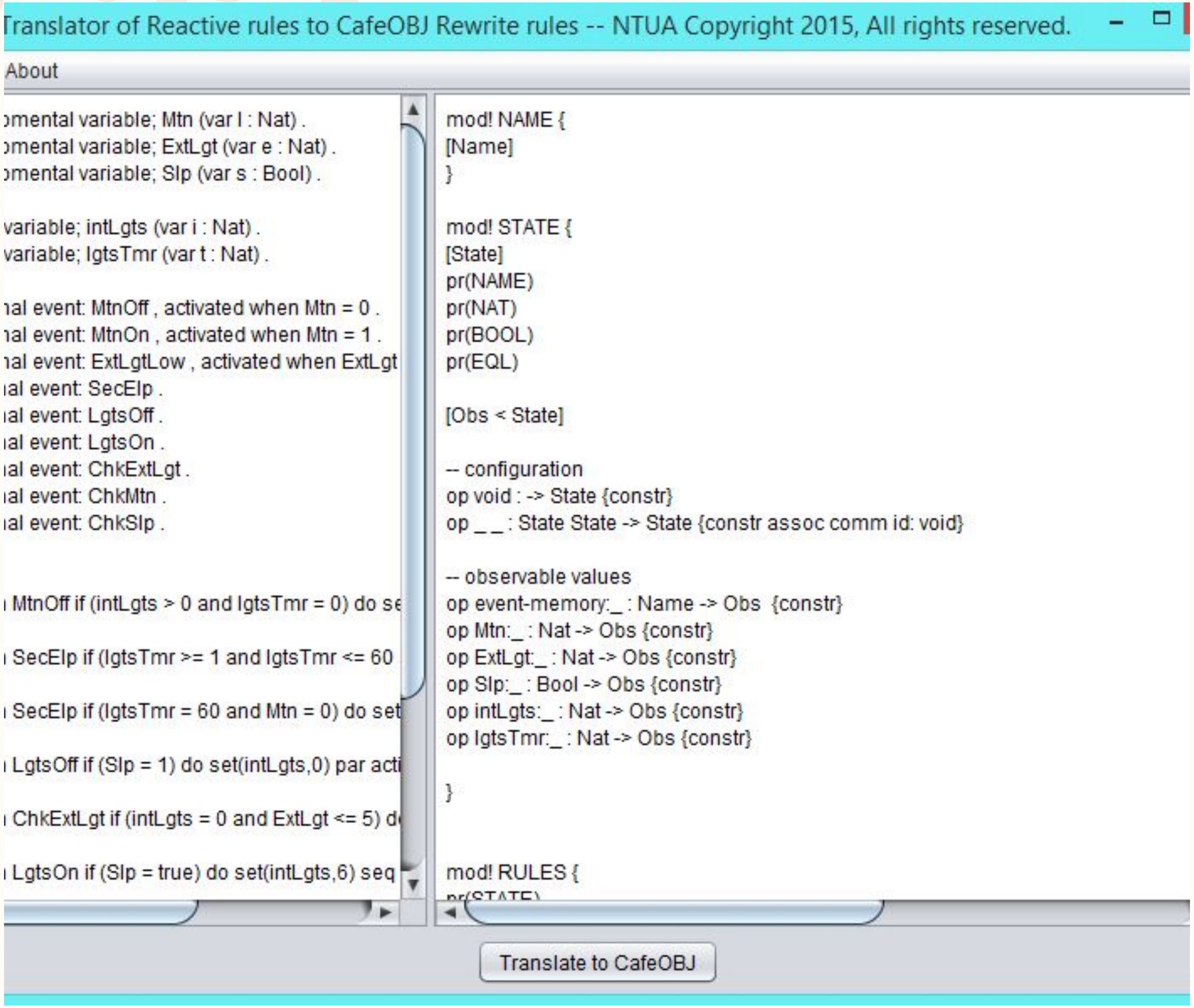
- Transforms a set of reactive rules into a CafeOBJ rewrite theory specification
- Written in Java
- Syntactic guidelines of reactive rules specification:
  - Rules declaration should start with an identification number (e.g. r1)
  - Definition of variables, events and rules should end with a fullstop (.)
  - External events should be accompanied with their detection conditions



ECA rules of the light control intelligent agent



Click on **Translate to CafeOBJ** button



Additional CafeOBJ specification

The output of the tool can be given **directly** as input to the CafeOBJ processor

Future work

- **Integrate** reactive rules verification methodology with other **rewriting** systems
  - e.g. **Maude** → better **tool** support and broader **use**

Research Interests

- Formal methods
  - Algebraic specifications
  - Automated theorem provers
- Intelligent Software Agents
- Complex Systems
  - Blockchains

Thank you!

- Questions???