CREST
A Continuous, REactive SysTemS DSL

Stefan Klikovits  Alban Linard  Didier Buchs

University of Geneva, Switzerland
Growing plants
Growing plants

(c) Debra Roby
https://www.flickr.com/photos/darinhercules/
CREST diagrams

Growing lamp

- electricity: 0 (Watt)
- switch: off (Switch)
- room temperature: 22 (Celsius)
- temperature: 22 (Celsius)
- light: 0 (Lumen)

Growing lamp

On

Off

room temperature: 22 (Celsius)

0 (Watt)

0 (Lumen)
CREST diagrams

Growing lamp

- electricity: 0 (Watt)
- switch: off (Switch)
- room temperature: 22 (Celsius)
- temperature: 22 (Celsius)
- on time: 0 (Time)
- light: 0 (Lumen)
CREST diagrams

Growing lamp

\[ switch = \text{off} \lor \ \text{electricity} < 100 \]

\[ switch = \text{on} \land \ \text{electricity} = 100 \]

- electricity: 0 (Watt)
- switch: off (Switch)
- room temperature: 22 (Celsius)
- temperature: 22 (Celsius)
- on time: 0 (Time)
- light: 0 (Lumen)

Klikovits et. al.
CREST DSL
stefan.klikovits@unige.ch
CREST diagrams

On

Off

room temperature + 4

switch = off \lor electricity < 100

switch = on \land electricity = 100

temperature: 22 (Celsius)

on time: 0 (Time)

light: 0 (Lumen)

800

0

electricity: 0 (Watt)

switch: off (Switch)

room temperature: 22 (Celsius)

Klikovits et. al. CREST DSL stefan.klikovits@unige.ch
CREST diagrams

Growing lamp

room temperature + 4

\[ \text{switch} = \text{off} \lor \text{electricity} < 100 \]

\[ \text{switch} = \text{on} \land \text{electricity} = 100 \]

- electricity: 0 (Watt)
- switch: off (Switch)
- room temperature: 22 (Celsius)
- temperature: 22 (Celsius)
- on time: 0 (Time)
- light: 0 (Lumen)

Klikovits et. al. CREST DSL stefan.klikovits@unige.ch
Composition

---

**Plant growing system**

- **Plant**
  - light 0 (Lux)
  - temperature 22 (Celsius)

- **Growing lamp**
  - room temperature 22 (Celsius)
  - switch off (Switch)
  - electricity 0 (Watt)
  - light 0 (Lumen)

- **Room temperature**
  - 22 (Celsius)

- **Switch**
  - off (Switch)

- **Electricity**
  - 0 (Watt)

- **Light**
  - 0 (Lux)

---

Klikovits et. al. CREST DSL stefan.klikovits@unige.ch
Composition

<<Plant growing system>>

<<Growing lamp>>
- Room temperature: 22°C
- Switch: off (Switch)
- Electricity: 0 Watt
- Light: 0 Lux

<<Plant>>
- Light: 0 Lux
- Temperature: 22°C

SPLIT

ADD

MAX
Composition

---

**Plant growing system**

- **room temperature 22 (Celsius)**
- **switch off (Switch)**
- **electricity 0 (Watt)**
- **light 0 (lux)**

---

**Growing lamp**

- **temperature 22 (Celsius)**
- **light 0 (Lumen)**

---

**Plant**

- **light 0 (Lux)**
- **temperature 22 (Celsius)**

---

**Klikovits et. al. CREST DSL stefan.klikovits@unige.ch**
Composition

≪Plant growing system≫

≪Growing lamp≫

room temperature 22 (Celsius)

switch off (Switch)

electricity 0 (Watt)

light 0 (Lux)

≪ADD≫

lamp 0 (Lumen)

external 0 (Lux)

add

lamp / 0.25 + external

MAX

room temperature 22 (Celsius)

switch off (Switch)

electricity 0 (Watt)

light 0 (Lux)

temperature 22 (Celsius)

light 0 (Lux)

Klikovits et. al. CREST DSL stefan.klikovits@unige.ch
class GrowingLamp(Entity):

calendar_klikovits_et_al_mono_2021 Crest DSL stefan.klikovits@unige.ch
class GrowingLamp(Entity):

electricity = Input(resource=res_electricity, init=0)
light = Output(resource=res_light_lumen, init=0)
Language Realization

class GrowingLamp(Entity):

electricity = Input(resource=res_electricity, init=0)
light = Output(resource=res_light_lumen, init=0)

on = current = State()
off = State()
Language Realization

class GrowingLamp(Entity):

electricity = Input(resource=res_electricity, init=0)
light = Output(resource=res_light_lumen, init=0)

on = current = State()
off = State()

off_to_on = Transition(source=off, target=on,
    guard=(lambda lamp: lamp.switch.value == "on" and
           lamp.electricity.value >= 100)
)
class GrowingLamp(Entity):

electricity = Input(resource=res_electricity, init=0)
light = Output(resource=res_light_lumen, init=0)

on = current = State()
off = State()

off_to_on = Transition(source=off, target=on,
guard=(lambda lamp: lamp.switch.value == "on" and
       lamp.electricity.value >= 100))

@update(state=on)
def set_light_on(lamp, dt=0):
    lamp.light.value = 800
Why not «formalism-XYZ»?

- powerful, but complex
- too generic
- feature workarounds
- architecture and behaviour
One DSL to rule them all?

ONE DOES NOT SIMPLY REPLACE EVERYTHING ELSE
\[
\langle T, R, E, e \rangle \\
e = \langle P_e, \text{resource}_e, TS_e, U_e, \text{entities}_e, \text{Inf}_e \rangle \\
P_e = I_e \sqcup O_e \sqcup L_e \\
TS = \langle S_e, \rightarrow_e \rangle; \rightarrow_e \subseteq S_e \times S_e \times G_e \\
\ldots
\]
\langle \mathbb{T}, \mathcal{R}, \mathcal{E}, e \rangle

e = \langle P_e, \text{resource}_e, TS_e, U_e, \text{entities}_e, \text{Inf}_e \rangle

P_e = I_e \sqcup O_e \sqcup L_e

TS = \langle S_e, \rightarrow_e \rangle; \rightarrow_e \subseteq S_e \times S_e \times G_e

\ldots

\text{it’s in the paper}
How to design a simulator

- advance time in constant steps?
- transition time calculation
- generate CREST diagrams
How to design a simulator

- advance time in constant steps?
- transition time calculation
- generate CREST diagrams

flow-rate: 0.5 per time step

fill-level < 10

fill-level = 10

Klikovits et. al.  stefan.klikovits@unige.ch
How to design a simulator

- advance time in constant steps?
- transition time calculation
- generate CREST diagrams

fill-level < 10

fill-level = 10

fill-level: 0.5 per time step
How to design a simulator

- advance time in constant steps?
- transition time calculation
- generate CREST diagrams

fill-level < 10

fill-level = 10

flow-rate: 0.5 per time step

fill-level

flow-rate: 0.5 per time step
Verification

- state space exploration
- zone/region-based verification
Verification

- state space exploration
- zone/region-based verification

Verification

- state space exploration
- zone/region-based verification
- requirements language
Controller synthesis

- planning + optimization + simulation
- changing parameters
- changing (sub-)systems
Applications

plant growing  home automation  job scheduling  office automation

Conclusion

CREST

- architecture & behaviour
- continuous & reactive
- formal basis
CREST
A Continuous, REactive SysTems DSL

Stefan Klikovits  Alban Linard  Didier Buchs

University of Geneva, Switzerland
Meta-Model

- Entity
- Influence
- Port
- Input
- Output
- Local
- State
- Update
- Transition
- Guard

**Relations:**
- Source
- Target
- Observe
- Write
- Transition
- Guard

**Annotations:**
- Subentity
- CREST DSL stefan.klikovits@unige.ch
Complex Ports

Requestable Resources
Complex Ports

Requestable Resources

Out → In
Req-In → Req-Out

Out → In
Complex Ports

Requestable Resources
Complex Ports

Requestable Resources

combined input & splitter
Complex Ports

Requestable Resources

combined input & splitter

Klikovits et. al.
CREST DSL
stefan.klikovits@unige.ch