A Calculus of Macro-Events: *Progress Report*

Iliano Cervesato — *ITT Industries*

Angelo Montanari — *University of Udine*

Outline

- EC
- Case study
- Explicit time & Event duration
- Macro-Events
- Implementation
- Future work

Event Calculus [Kowalski & Sergot, 86]

- Simple model of time and change
  - Event occurrences start/end properties
  - Does p hold at time t?
  - Constraints
- Given as a logic program
- Extensively studied

Limitations

No independent semantics

- Difficult to study
- Non portable
- *Ad hoc* extensions

On-Going Project

- Abstract specification of EC
  - Connectives and quantifiers
  - Preconditions
  - Relative time
- Implementation correctness
- Complexity study
- Extensions

This work

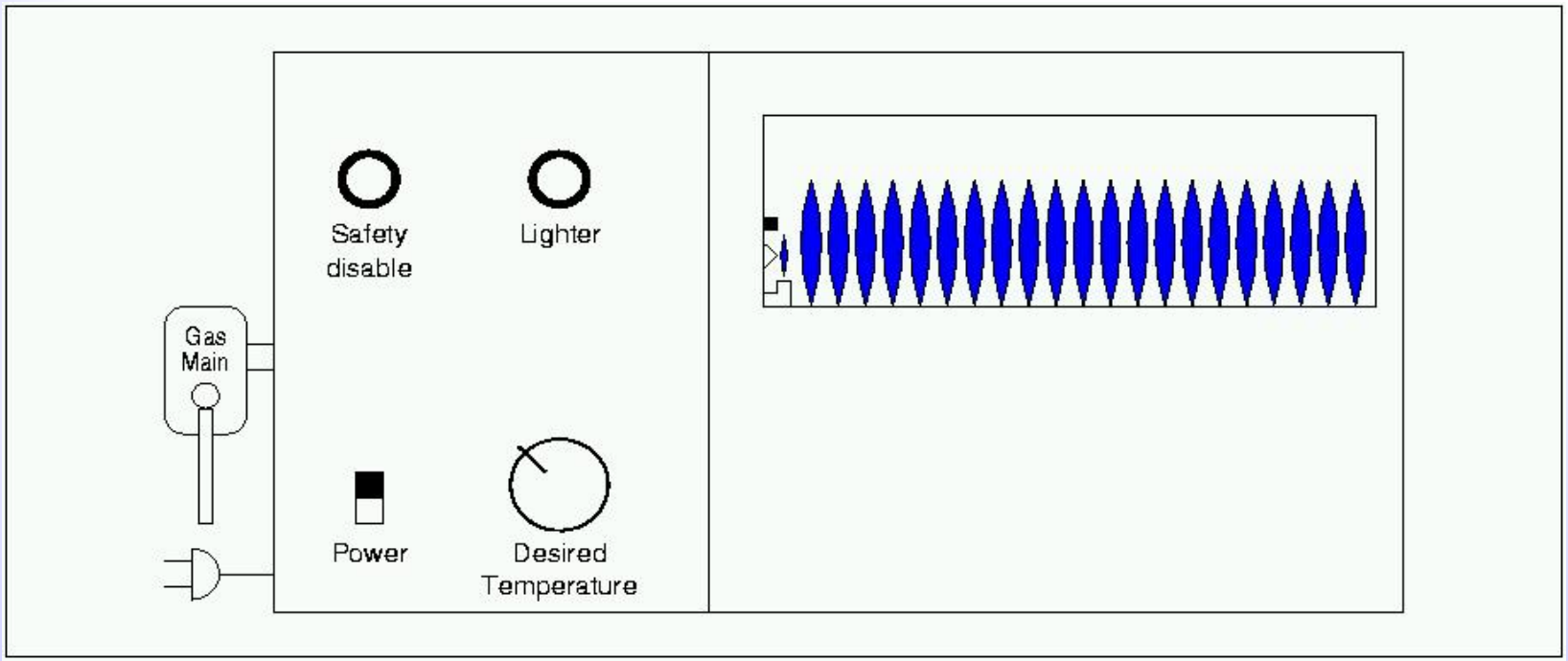
- Properties due to event cooperation
  - Serial
  - Parallel
  - Alternative
  - Iterated
- Explicit time, event duration

Case Study

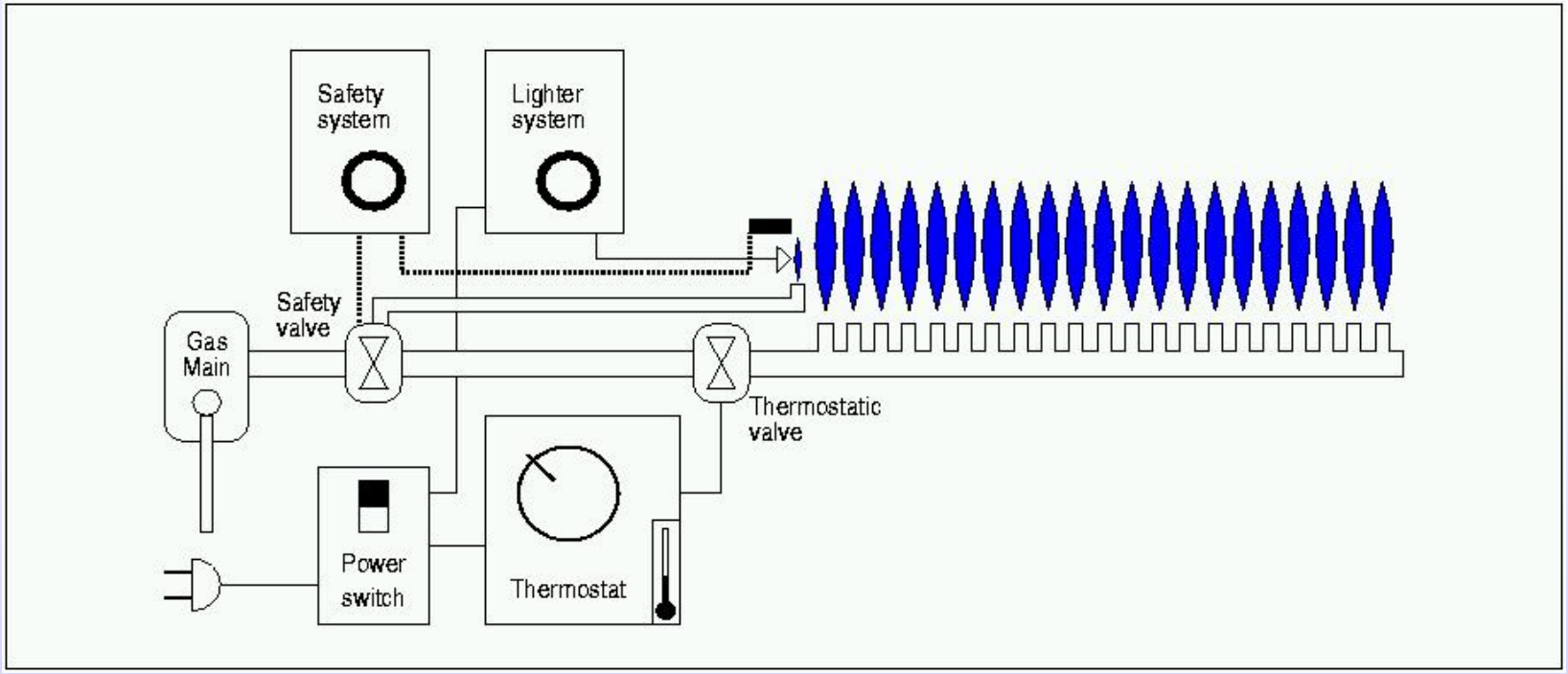
A simple gas heater

- Concrete
- Modeling benchmark
- Extensible
  - Dagstuhl steam-boiler problem

Gas Heater: a User's Perspective

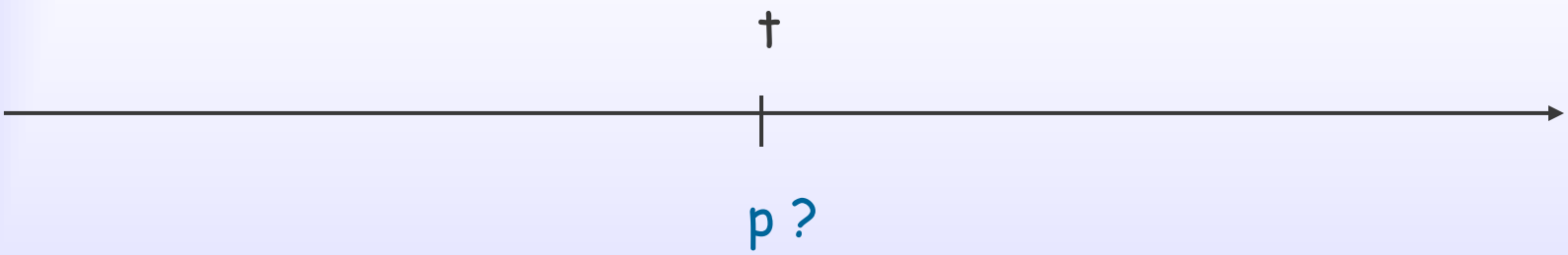


Gas Heater: Engineer's View

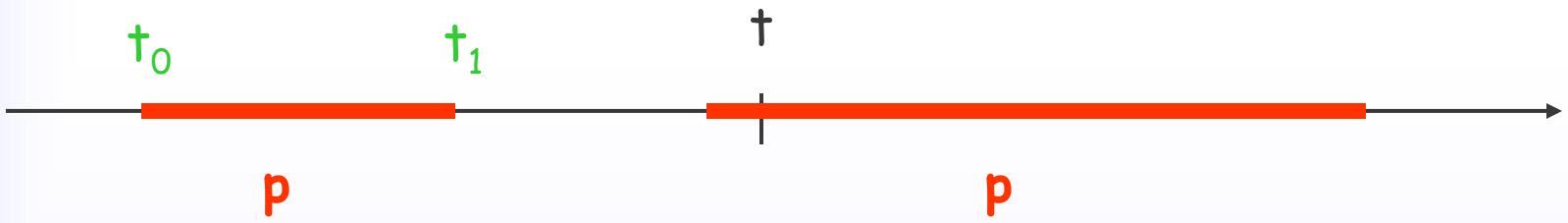


EC with Explicit Time

- Does p hold at time t?



Where does p hold?



- MVI
  - e_i happens at t_i
  - e_0 initiates p, e_1 terminates p
  - Preconditions hold
  - No interrupting events

ECT-structures

(E, P, [-|-), (-|-], T)

- E: events
- P: properties
- [-|-): initiation map
- (-|-]: termination map
- T: temporal domain

Time Structure

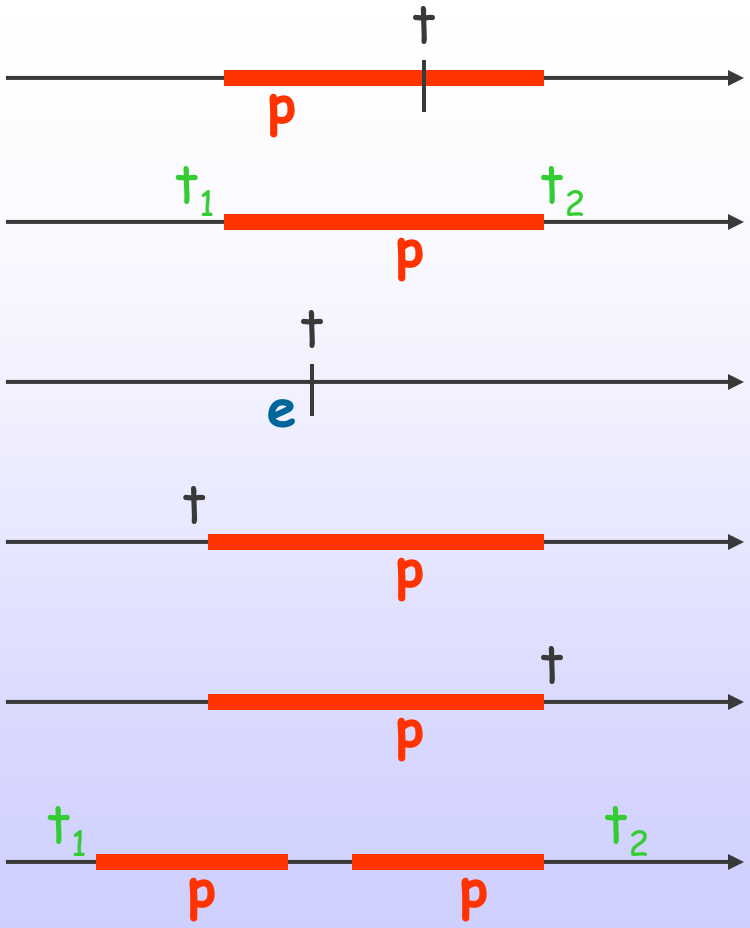
Sequence T of pairs

(e, t)

- e: event
- t: time

Building Blocks

- holdsAt (p,t)
- mvi (p, t_1, t_2)
- happens (e, t)
- initiate (p, t)
- terminate (p, t)
- broken (p, t_1, t_2)



Event Duration

Non-instantaneous event occurrences

$$(E,\ P,\ [\text{-}|\text{-}),\ (\text{-}|\text{-}],\ T,\ \delta)$$

- …
- δ : duration function

Specification

Changes are localized to

- happens (e, t, d)
- initiate (p, t)
- terminate (p, t)

Case Study – Events

- gasOn
- powerOn
- pressDisable
- pressLighter
- coolDown
- gasOff
- powerOff
- releaseDisable
- releaseLighter
- warmUp

- start

Case Study – Time Structure

- (start, 0)
- (coolDown, 0)
- (gasOn, 1)
- (powerOn, 2)
- (pressDisable, 3)
- (pressLighter, 3)
- (releaseLighter, 5)
- (releaseDisable, 6)
- (warmUp, 8)
- (coolDown, 10)
- (warmUp, 11)
- (powerOff, 18)
- (gasOff, 19)
- (coolDown, 25)

Case Study – Properties

- gas
- power
- cold
- thermoValveOpen
- safetyValveOpen
- sparkling
- pilotOn
- pilotOff
- burning

Case Study – Basic Specs.

- [gas|-) = gasOn
- [cold|-) = coolDown
- [thermoValveOpen|-) = coolDown

- (gas|-] = gasOff
- (cold|-] = warmUp
- (thermoValveOpen|-] = warmUp

Case Study – Preconditions

- [safetyValveOpen|-) = pressDisable

- (safetyValveOpen | pilotOn] = gasOff
- (safetyValveOpen | pilotOff] = relDisable

Case Study – Shortcomings

- pressLighter and pressDisable must occur <u>together</u> for pilotOn to hold
- Cannot be expressed in EC

Macro-Events – Inspiration

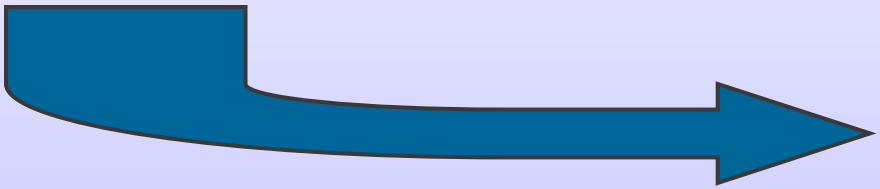
- Process algebra
- Path expression

Related work within EC:

- Chittaro & Montanari
- Mizzaro Jr.
- Evans
- Belegrinos & Georgeff
- Lesperance & al.
- Lin & Dean

Macro-Events – Syntax

- e
- $m_1 ;_d^D m_2$
- $m_1 + m_2$
- $m_1 \,||\, m_2$
- m^*



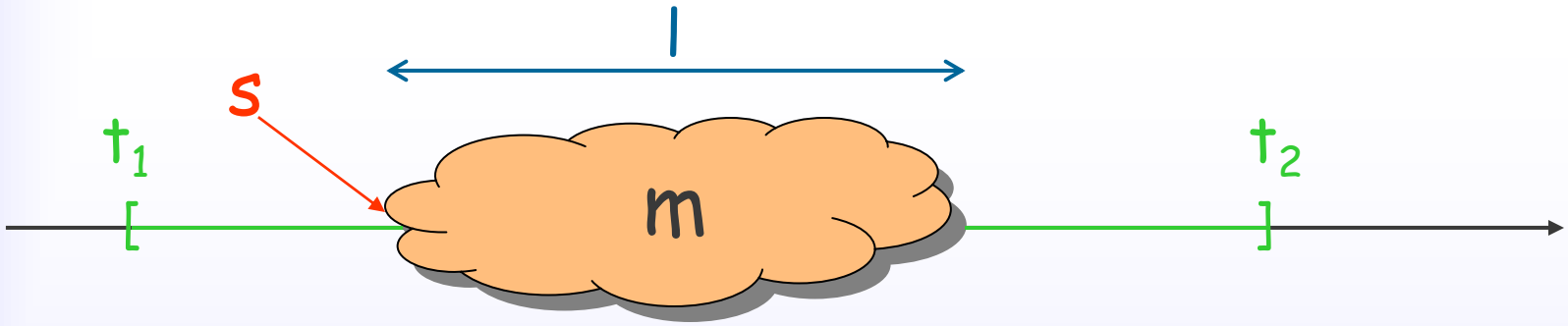
- $m_1 ; m_2$
- $m_1 ;; m_2$
- m^+
- m^n

MECTD-Structure

$$(E,\ P,\ M,\ [-|-),\ (-|-],\ T,\ \delta)$$

- M: set of macro-events
- $[-|-)$: initiates macro-events in M
- $(-|-]$: terminates macro-events in M

Macro-Events – me



me (m, $[t_1,t_2]$, s, l)

- m: macro-event
- $[t_1,t_2]$: reference interval
- s: starting time
- l: duration

me $(e, [t_1,t_2], s, l)$

$$(e, s) \in T \quad \wedge$$
$$l = \delta(e) \quad \wedge$$
$$[s, s+l] \leq [t_1,t_2]$$

$me\ (m_1 \parallel m_2, [t_1,t_2], s, l)$

$$
\begin{aligned}
\exists\ & s_1, l_1, s_2, l_2. \\
& me\ (m_1, [t_1,t_2], s_1, l_1) \quad \wedge \\
& me\ (m_2, [t_1,t_2], s_2, l_2) \quad \wedge \\
& s = \min(s_1, s_2) \quad \wedge \\
& l = \max(s_1 + l_1, s_2 + l_2) - s
\end{aligned}
$$

me (m*, $[t_1,t_2]$, s, l)

$$
\begin{aligned}
\exists\ & l_1, s_2, l_2, t. \\
& (s = t_1 \wedge l = 0) \qquad\qquad \vee \\
& (\mathrm{me}(m, [t_1,t], s, l_1) \qquad \wedge \\
& \ \mathrm{me}(m^*, [t,t_2], s_2, l_2) \qquad \wedge \\
& \ s + l_1 \leq t \leq s_2 \qquad \wedge \\
& \ l = s_2 + l_2 - s)
\end{aligned}
$$

Monitoring

Did m occur in $[t_1,t_2]$?

check $(m, [t_1,t_2])$

iff

$\exists\ s, l.\ me\ (m,\ [t_1,t_2],\ s,\ l)$

Evaluation – Does p hold at t ?

- Redefine happens

happens (m, t, d)

iff

me (m, [0,∞], s, d)

Case Study – Macro-Events

- [pilotOn | power,*gas*) =
  pressLighter || pressDisable

- (pilotOn | -] = *gasOff*

Implementation

- Logical specification is executable
- Written in Prolog
- Sound and complete

Future Work

- More powerful primitives
  - Recursion, negation, …
- Cooperation
- Interference