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

A Calculus of Macro-Events: Progress Report
A Calculus of Macro-Events: Progress Report

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

A Calculus of Macro-Events: Progress Report
Gas Heater: Engineer’s View

A Calculus of Macro-Events: Progress Report
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

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

- ...
- \(\delta: duration\ function\)
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

- \((\text{start, } 0)\)
- \((\text{coolDown, } 0)\)
- \((\text{gasOn, } 1)\)
- \((\text{powerOn, } 2)\)
- \((\text{pressDisable, } 3)\)
- \((\text{pressLighter, } 3)\)
- \((\text{releaseLighter, } 5)\)
- \((\text{releaseDisable, } 6)\)
- \((\text{warmUp, } 8)\)
- \((\text{coolDown, } 10)\)
- \((\text{warmUp, } 11)\)
- \((\text{powerOff, } 18)\)
- \((\text{gasOff, } 19)\)
- \((\text{coolDown, } 25)\)
Case Study – Properties

- gas
- power
- cold
- thermoValveOpen
- safetyValveOpen
- sparkling
- pilotOn
- pilotOff
- burning
Case Study – Basic Specs.

- \([\text{gas}](-)\) = \text{gasOn}
- \([\text{cold}](-)\) = \text{coolDown}
- \([\text{thermoValveOpen}](-)\) = \text{coolDown}

- \(\text{(gas)}(-)\) = \text{gasOff}
- \(\text{(cold)}(-)\) = \text{warmUp}
- \(\text{(thermoValveOpen)}(-)\) = \text{warmUp}
Case Study – Preconditions

- \([\text{safetyValveOpen} \rightarrow -] = \text{pressDisable}\)
- \(\text{safetyValveOpen} \mid \text{pilotOn} = \text{gasOff}\)
- \(\text{safetyValveOpen} \mid \text{pilotOff} = \text{relDisable}\)
Case Study – Shortcomings

- `pressLighter` and `pressDisable` must occur together 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

- \( e \)
- \( m_1 ;_{d}^D m_2 \)
- \( m_1 + m_2 \)
- \( m_1 \parallel m_2 \)
- \( m^{*} \)
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, [t₁,t₂], s, l)

- m: macro-event
- [t₁,t₂]: reference interval
- s: starting time
- l: duration

A Calculus of Macro-Events: Progress Report
\[ me(e, [t_1, t_2], s, l) \]

\[ (e, s) \in T \quad \wedge \]
\[ l = \delta(e) \quad \wedge \]
\[ [s, s+l] \leq [t_1, t_2] \]
\[ \exists s_1, l_1, s_2, l_2. \]
\[ \begin{align*}
\text{me} &\left( m_1 \ || \ m_2, [t_1,t_2], s, l \right) \\
\text{me} &\left( m_1, [t_1,t_2], s_1, l_1 \right) \\
\text{me} &\left( m_2, [t_1,t_2], s_2, l_2 \right) \\
s &\ = \ min \left( s_1, s_2 \right) \\
l &\ = \ max \left( s_1 + l_1, s_2 + l_2 \right) - s
\end{align*} \]
me (m*, [t₁,t₂], s, l)

∃ l₁, s₂, l₂, t.

(s = t₁ ∧ l = 0) ∨

(me (m, [t₁,t], s, l₁) ∧
me (m*, [t,t₂], s₂, l₂) ∧

s + l₁ ≤ t ≤ s₂ ∧

l = s₂ + l₂ - s)
Monitoring

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

$$\text{check} \ (m, [t_1, t_2])$$

iff

$$\exists \ s, \ l. \ \text{me} \ (m, [t_1, t_2], \ s, \ l)$$
Evaluation – Does \( p \) hold at \( t \)?

- Redefine \textit{happens}

\[
\text{happens} \ (m, \ t, \ d) \\
\text{iff} \\
\text{me} \ (m, \ [0,\infty[, \ s, \ d)
\]
Case Study – Macro-Events

• \([\text{pilotOn} \mid \text{power},\text{gas}) = \text{pressLighter} \parallel \text{pressDisable}\]

• \((\text{pilotOn} \mid \text{-}) = \text{gasOff}\)
Implementation

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

• More powerful primitives
  ➢ Recursion, negation, ...

• Cooperation

• Interference