Run-Time Composite Event Recognition

Alexander Artikis, Marek Sergot and George Paliouras

Institute of Informatics & Telecommunications,
NCSR "Demokritos", Greece
Department of Computing, Imperial College London, UK

a.artikis@iit.demokritos.gr
m.sergot@imperial.ac.uk
paliourg@iit.demokritos.gr

Event Recognition

Problem:

- Event recognition (event pattern matching):
  - Input: Simple, derived events (SDE) coming from various types of sensor.
  - Output: Composite events (CE), ie collections of SDE and CE that satisfy some pattern.

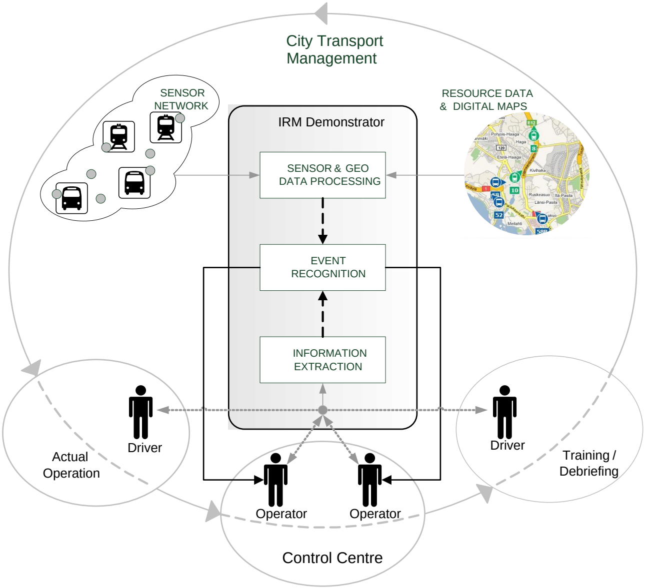
Aim:

- Real-time CE recognition in large-scale DEBS.
- Formal & declarative semantics.

Approach:

- Highly efficient logic programming: Event Calculus.

Event Recognition for City Transport Management



Event Recognition for City Transport Management

	Input	Output
200	scheduled stop enter	
215	late stop leave	
$[215, 400]$	abrupt acceleration	
$[350, 600]$	sharp turn	
700	scheduled stop enter	
705	passenger density change to high	
820	scheduled stop leave	
...		

Event Recognition for City Transport Management

Input		Output	
200	scheduled stop enter		
215	late stop leave	*since*(215)	non-punctual
[215, 400]	abrupt acceleration		
[350, 600]	sharp turn	[215, 600]	uncomfortable driving
700	scheduled stop enter		
705	passenger density change to high		
820	scheduled stop leave		
...			

Event Recognition for City Transport Management

Input		**Output**	
200	scheduled stop enter		
215	late stop leave	*since*(215)	non-punctual
[215, 400]	abrupt acceleration		
[350, 600]	sharp turn	[215, 600]	uncomfortable driving
700	scheduled stop enter		
705	passenger density change to high		
820	scheduled stop leave	[215,820]	non-punctual
...			

Event Calculus

- A logic programming language for representing and reasoning about events and their effects.
- Key components:
  - event (typically instantaneous).
  - fluent: a property that may have different values at different points in time.
- Built-in representation of inertia:
  - F holds at a particular time-point if F has been *initiated* by an event at some earlier time-point, and not *terminated* by another event in the meantime.

CE Definitions in the Event Calculus

CE definition:

$punctuality(ID) = non\_punctual$ **initiated** iff
$enter\_stop(ID, StopCode, late)$ **happens** or
$leave\_stop(ID, StopCode, early)$ **happens**

$punctuality(ID) = non\_punctual$ **terminatedAt** T iff
$enter\_stop(ID, StopCode, scheduled)$ **happensAt** T',
$leave\_stop(ID, StopCode, scheduled)$ **happensAt** T

CE recognition:

- $punctuality(ID) = non\_punctual$ **holdsFor** I

CE Definitions in the Event Calculus

CE definition:

$$
\begin{aligned}
&\mathit{driving\_quality}(ID) = \mathit{low} \text{ iff} \\
&\quad \mathit{punctuality}(ID) = \mathit{non\_punctual} \text{ or} \\
&\quad \mathit{driving\_style}(ID) = \mathit{unsafe}
\end{aligned}
$$

Compiled CE definition:

$$
\begin{aligned}
&\mathit{driving\_quality}(ID) = \mathit{low} \ \ \textbf{holdsFor} \ \ I_1 \cup I_2 \ \ \text{iff} \\
&\quad \mathit{punctuality}(ID) = \mathit{non\_punctual} \ \ \textbf{holdsFor} \ \ I_1, \\
&\quad \mathit{driving\_style}(ID) = \mathit{unsafe} \ \ \textbf{holdsFor} \ \ I_2
\end{aligned}
$$

CE Definitions in the Event Calculus

CE definition:

$$\begin{aligned}&\mathit{driving\_quality}(ID) = \mathit{medium} \text{ iff}\\ &\quad \mathit{punctuality}(ID) = \mathit{punctual},\\ &\quad \mathit{driving\_style}(ID) = \mathit{uncomfortable}\end{aligned}$$

Compiled CE definition:

$$\begin{aligned}&\mathit{driving\_quality}(ID) = \mathit{medium} \ \ \textbf{holdsFor} \ \ I_1 \cap I_2 \ \text{ iff}\\ &\quad \mathit{punctuality}(ID) = \mathit{punctual} \ \ \textbf{holdsFor} \ \ I_1,\\ &\quad \mathit{driving\_style}(ID) = \mathit{uncomfortable} \ \ \textbf{holdsFor} \ \ I_2\end{aligned}$$

CE Definitions in the Event Calculus

CE definition:

$$\begin{aligned}&\mathit{driving\_quality}(ID) = \mathit{high} \text{ iff}\\&\quad \mathit{punctuality}(ID) = \mathit{punctual},\\&\quad \mathit{driving\_style}(ID) \neq \mathit{unsafe},\\&\quad \mathit{driving\_style}(ID) \neq \mathit{uncomfortable}\end{aligned}$$

Compiled CE definition:

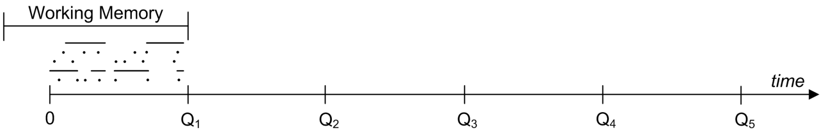
$$\begin{aligned}&\mathit{driving\_quality}(ID) = \mathit{high} \;\; \textbf{holdsFor} \;\; I_1 \setminus I_2 \cup I_3 \;\; \text{iff}\\&\quad \mathit{punctuality}(ID) = \mathit{punctual} \;\; \textbf{holdsFor} \;\; I_1,\\&\quad \mathit{driving\_style}(ID) = \mathit{unsafe} \;\; \textbf{holdsFor} \;\; I_2,\\&\quad \mathit{driving\_style}(ID) = \mathit{uncomfortable} \;\; \textbf{holdsFor} \;\; I_3\end{aligned}$$

Run-Time Event Recognition

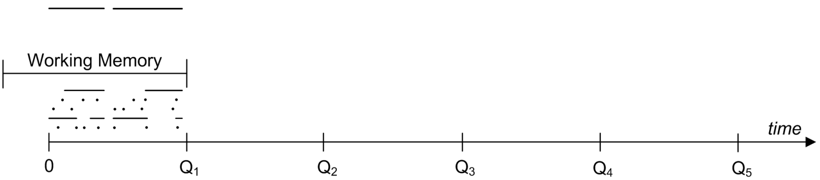
Real-time decision-support in the presence of:

- Very large SDE streams.
- Non-sorted SDE streams.
- SDE revision.
- Very large CE numbers.

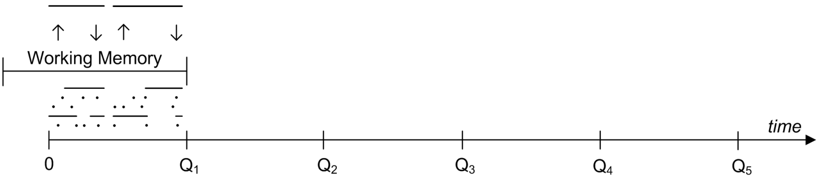
Event Calculus: Run-Time Event Recognition



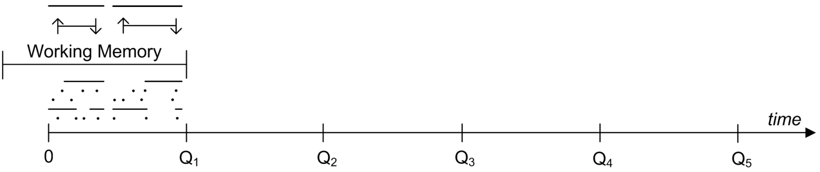
Event Calculus: Run-Time Event Recognition



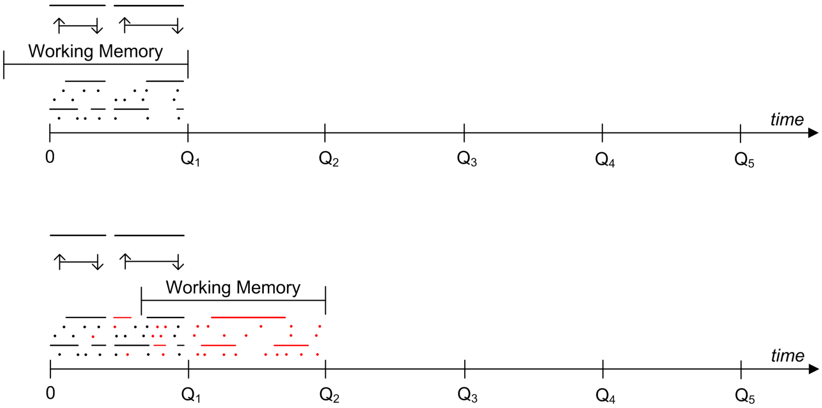
Event Calculus: Run-Time Event Recognition



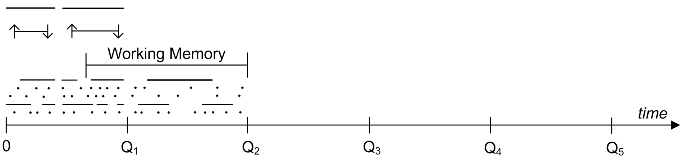
Event Calculus: Run-Time Event Recognition



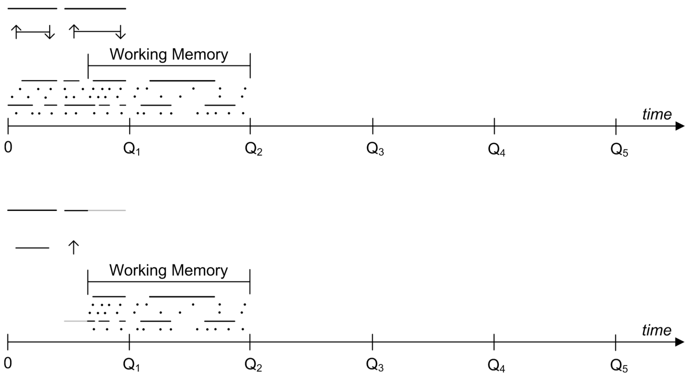
Event Calculus: Run-Time Event Recognition



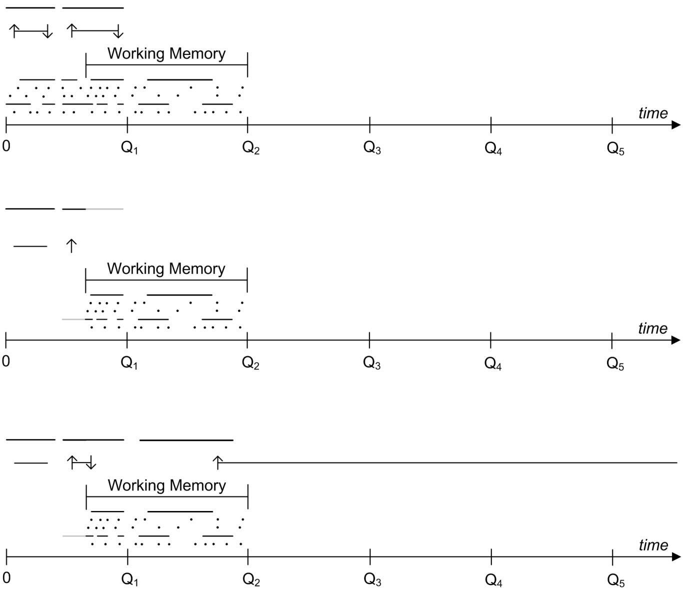
Event Calculus: Run-Time Event Recognition



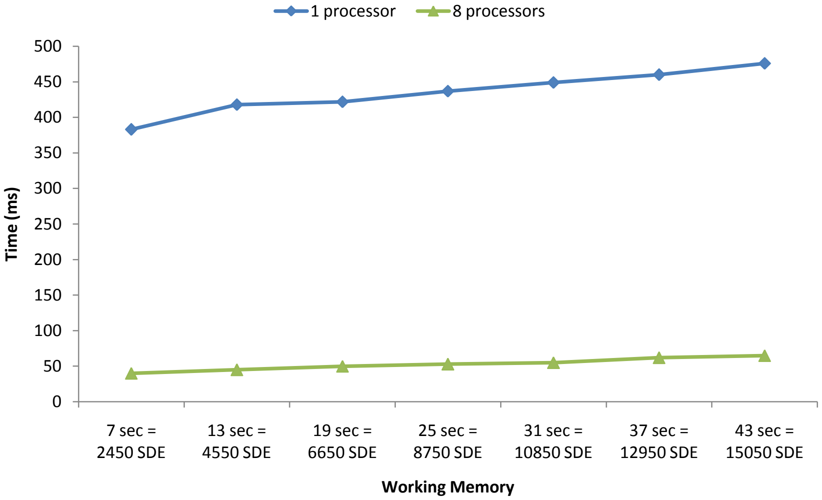
Event Calculus: Run-Time Event Recognition



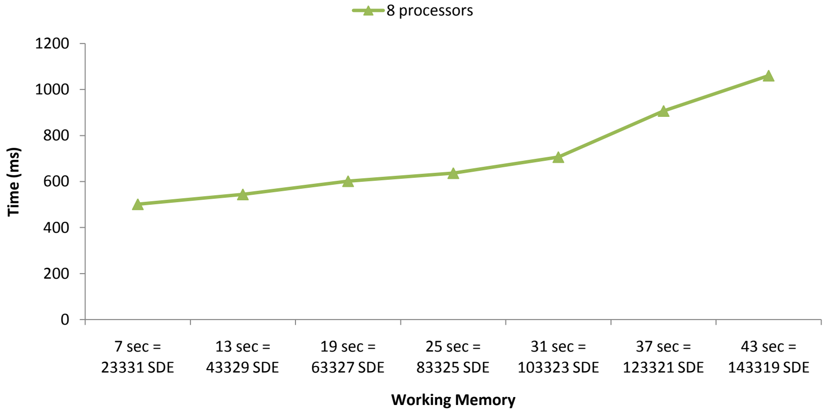
Event Calculus: Run-Time Event Recognition



City Transport Management in Helsinki



City Transport Management in Very Big Cities



Summary

Event Calculus for real-time CE recognition:

- ‘Windowing’ mechanism.
- A simple indexing mechanism means that we do not have to rely on SDE filtering modules.
- A form of caching stores the results of sub-computations in order to avoid unnecessary recomputations.
- A set of interval manipulation constructs simplify CE definitions and improve reasoning efficiency.

Summary

- Complex temporal representation:
  - Succinct representation $\longrightarrow$ code maintenance.
  - Intuitive representation $\longrightarrow$ facilitates interaction with domain experts unfamiliar with programming.
- Formal & declarative semantics.

Further Work

- Event recognition under uncertainty in the Event Calculus:
  - Erroneous SDE detection.
  - Incomplete SDE stream.
  - Imprecise CE definition.
- Machine learning in the Event Calculus:
  - Automated generation of CE definitions.