Complex Event Recognition with Allen Relations

Periklis Mantenoglou$^{2,1}$  Dimitrios Kelesis$^{3,1}$  Alexander Artikis$^{4,1}$

$^1$NCSR Demokritos, Greece
$^2$National and Kapodistrian University of Athens, Greece
$^3$National Technical University of Athens, Greece
$^4$University of Piraeus, Greece

http://cer.iit.demokritos.gr/
Complex Event Recognition

INPUT ▶ RECOGNITION ▶ OUTPUT ▶
Complex
Event
Recognition
System
Complex Event
Definitions
Simple Event Stream
. . .
. . .
Complex Event Stream
. . .
. . .

https://cer.iit.demokritos.gr (maritime)
Complex Event Recognition

INPUT ▶ RECOGNITION ▶ OUTPUT ▶

Complex Event Definitions

https://cer.iit.demokritos.gr (maritime)
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.

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:
  - $F = V$ holds at a particular time-point if $F = V$ has been initiated by an event at some earlier time-point, and not terminated by another event in the meantime.

---

Run-Time Event Calculus (RTEC): Fluent Specification

Simple Fluents:

\text{initiatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{In1}, T)[, conditions].
\vdots
\text{terminatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{T1}, T)[, conditions].
\vdots

where conditions:

\begin{align*}
0-K & \ [\text{not}] \ \text{happensAt}(E_k, T), \\
0-M & \ [\text{not}] \ \text{holdsAt}(F_m = V_m, T), \\
0-N & \text{atemporal-constraint}_n
\end{align*}

Run-Time Event Calculus (RTEC): Fluent Specification

Simple Fluents:

\[
\text{initiatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{I_{n_1}}, T)[, \text{conditions}].
\]

\[
\text{terminatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{T_{T_1}}, T)[, \text{conditions}].
\]

where conditions:

\[
0^{-K} [\text{not}] \text{happensAt}(E_k, T),
\]

\[
0^{-M} [\text{not}] \text{holdsAt}(F_m = V_m, T),
\]

\[
0^{-N} \text{atemporal-constraint}_n
\]

Statically Determined Fluents:

\[
\text{holdsFor}(F = V, I) \leftarrow
\text{holdsFor}(F_1 = V_1, I_1)[, \text{conditions}],
\text{holdsFor}(F_2 = V_2, I_2), \ldots
\text{holdsFor}(F_n = V_n, I_n),
\text{intervalConstruct}(L_1, I_{n+1}), \ldots
\text{intervalConstruct}(L_m, I)].
\]

where intervalConstruct:

\text{union\_all} or \text{intersect\_all} or \text{relative\_complement\_all}

Statically Determined Fluent: Anchored or Moored

\[
\text{holdsFor}(\text{anchoredOrMoored}(Vessel) = \text{true}, I) \leftarrow \\
\text{holdsFor}(\text{stopped}(Vessel) = \text{farFromPorts}, I_{sf}), \\
\text{holdsFor}(\text{withinArea}(Vessel, \text{anchorage}) = \text{true}, I_{wa}), \\
\text{intersect\_all}([I_{sf}, I_{wa}], I_{sa}), \\
\text{holdsFor}(\text{stopped}(Vessel) = \text{nearPorts}, I_{sn}), \\
\text{union\_all}([I_{sa}, I_{sn}], I).
\]
Statically Determined Fluent:
Anchored or Moored

\[
\text{holdsFor}(\text{anchoredOrMoored}(\text{Vessel}) = \text{true}, I) \leftarrow \\
\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{farFromPorts}, I_{sf}), \\
\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{anchorage}) = \text{true}, I_{wa}), \\
\text{intersect\_all}([I_{sf}, I_{wa}], I_{sa}), \\
\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{nearPorts}, I_{sn}), \\
\text{union\_all}([I_{sa}, I_{sn}], I).
\]
Statically Determined Fluent:
Anchored or Moored

\[\text{holdsFor}(\text{anchoredOrMoored}(\text{Vessel}) = \text{true}, \ I) \leftarrow \]
\[\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{farFromPorts}, \ l_{sf}), \]
\[\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{anchorage}) = \text{true}, \ l_{wa}), \]
\[\text{intersect}_\text{all}([l_{sf}, l_{wa}], \ l_{sa}), \]
\[\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{nearPorts}, \ l_{sn}), \]
\[\text{union}_\text{all}([l_{sa}, l_{sn}], \ I).\]
Statically Determined Fluent: Anchored or Moored

\[
holdsFor(anchoredOrMoored(Vessel) = true, I) \leftarrow \\
holdsFor(stopped(Vessel) = farFromPorts, I_{sf}), \\
holdsFor(withinArea(Vessel, anchorage) = true, I_{wa}), \\
intersect_all([I_{sf}, I_{wa}], I_{sa}), \\
holdsFor(stopped(Vessel) = nearPorts, I_{sn}), \\
union_all([I_{sa}, I_{sn}], I).
\]
Statically Determined Fluent: Anchored or Moored

\[ \text{holdsFor}(\text{anchoredOrMoored}(Vessel) = \text{true}, I) \leftarrow \]
\[ \text{holdsFor}(\text{stopped}(Vessel) = \text{farFromPorts}, I_{sf}), \]
\[ \text{holdsFor}(\text{withinArea}(Vessel, anchorage) = \text{true}, I_{wa}), \]
\[ \text{intersect\_all}(I_{sf}, I_{wa}, I_{sa}), \]
\[ \text{holdsFor}(\text{stopped}(Vessel) = \text{nearPorts}, I_{sn}), \]
\[ \text{union\_all}(I_{sa}, I_{sn}, I). \]
Statically Determined Fluent: Anchored or Moored

\[
\text{holdsFor}(\text{anchoredOrMoored}(\text{Vessel}) = \text{true}, \ I) \leftarrow \\
\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{farFromPorts}, \ I_\text{sf}), \\
\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{anchorage}) = \text{true}, \ I_\text{wa}), \\
\text{intersect\_all}([I_\text{sf}, I_\text{wa}], \ I_\text{sa}), \\
\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{nearPorts}, \ I_\text{sn}), \\
\text{union\_all}([I_\text{sa}, I_\text{sn}], \ I).
\]

![Diagram showing time line with intervals for different states: I, I_sn, I_sa, I_wa, I_sf]

https://cer.iit.demokritos.gr (maritime)
Interval Constructs & Allen Relations

\[
\begin{align*}
\text{relative_complement_all} & (I_1, [I_2, I_3], I_c) \\
\text{intersect_all} & ([I_1, I_2, I_3], I_i) \\
\text{union_all} & ([I_1, I_2, I_3], I_u)
\end{align*}
\]

<table>
<thead>
<tr>
<th>Relation</th>
<th>Illustration</th>
</tr>
</thead>
<tbody>
<tr>
<td>before($i^s, i^t$)</td>
<td>$i^s$</td>
</tr>
<tr>
<td>meets($i^s, i^t$)</td>
<td>$i^s$</td>
</tr>
<tr>
<td>starts($i^s, i^t$)</td>
<td>$i^t$</td>
</tr>
<tr>
<td>finishes($i^s, i^t$)</td>
<td>$i^s$</td>
</tr>
<tr>
<td>during($i^s, i^t$)</td>
<td>$i^s$</td>
</tr>
<tr>
<td>overlaps($i^s, i^t$)</td>
<td>$i^s$</td>
</tr>
<tr>
<td>equal($i^s, i^t$)</td>
<td>$i^s$</td>
</tr>
</tbody>
</table>
RTEC_A: RTEC with Allen Relations

\[
\text{holdsFor}(\text{disappearedInArea}(\text{Vessel}, \text{Area Type}) = \text{true}, I) \leftarrow \\
\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{Area Type}) = \text{true}, S), \\
\text{holdsFor}(\text{gap}(\text{Vessel}) = \text{farFromPorts}, T), \\
\text{allen}(\text{meets}, S, T, \text{target}, I).
\]
\textbf{RTEC}_A: RTEC with Allen Relations

\begin{align*}
\text{holdsFor}(\text{disappearedInArea}(Vessel, \text{Area Type}) = \text{true}, I) \leftarrow \\
\text{holdsFor}(\text{withinArea}(Vessel, \text{Area Type}) = \text{true}, S), \\
\text{holdsFor}(\text{gap}(Vessel) = \text{farFromPorts}, T), \\
\text{allen}(\text{meets}, S, T, \text{target}, I).
\end{align*}
RTEC$_A$: RTEC with Allen Relations

\[
\text{holdsFor}(\text{disappearedInArea}(\text{Vessel}, \text{Area Type}) = \text{true}, I) \leftarrow \\
\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{Area Type}) = \text{true}, S), \\
\text{holdsFor}(\text{gap}(\text{Vessel}) = \text{farFromPorts}, T), \\
\text{allen}(\text{meets}, S, T, \text{target}, I).
\]
**RTEC\textsubscript{A}: RTEC with Allen Relations**

\[
\text{holdsFor}(\text{disappearedInArea}(Vessel, \text{Area Type}) = \text{true}, I) \leftarrow \text{holdsFor}(\text{withinArea}(Vessel, \text{Area Type}) = \text{true}, S), \text{holdsFor}(\text{gap}(Vessel) = \text{farFromPorts}, T), \text{allen}(<\text{meets}, S, T, \text{target}, I>).
\]
RTEC\(_A\): Windowing

\[
\text{holdsFor}(\text{disappearedInArea}(\text{Vessel}, \text{Area Type}) = \text{true}, I) \leftarrow \\
\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{Area Type}) = \text{true}, S), \\
\text{holdsFor}(\text{gap}(\text{Vessel}) = \text{farFromPorts}, T), \\
\text{allen}(\text{meets}, S, T, \text{target}, I).
\]

**Query time:** \(q_{81}\)

\[
\begin{align*}
&\quad S \\
&\quad T \\
&\quad \quad \quad \quad q_{81}
\end{align*}
\]
RTEC_A: Windowing

holdsFor(disappearedInArea(Vessel, Area Type) = true, I) ← holdsFor(withinArea(Vessel, Area Type) = true, S), holdsFor(gap(Vessel) = farFromPorts, T), allen(meets, S, T, target, I).

Query time: $q_{81}$

\[ w_{81} \]

\[ S \hspace{1cm} \]

\[ q_{81} \]

\[ T \hspace{1cm} \]

\[ q_{81} \]
**RTECₐ: Windowing**

\[
\text{holdsFor}(\text{disappearedInArea}(\text{Vessel}, \text{Area Type}) = \text{true}, I) \leftarrow \\
\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{Area Type}) = \text{true}, S), \\
\text{holdsFor}(\text{gap}(\text{Vessel}) = \text{farFromPorts}, T), \\
\text{allen}(\text{meets}, S, T, \text{target}, I).
\]

**Query time: \(q_{81}\)**

\[
\begin{array}{c}
\text{S} \\
\vdots_{i_1} \\
\text{T} \\
\vdots_{i_2}
\end{array}
\]

\[
\begin{array}{c}
\text{w}_{81} \\
\text{q}_{81}
\end{array}
\]

\[
\text{q}_{81}
\]
holdsFor(disappearedInArea(Vessel, AreaType) = true, I) ←
holdsFor(withinArea(Vessel, AreaType) = true, S),
holdsFor(gap(Vessel) = farFromPorts, T),
allen(meets, S, T, target, I).

Query time: $q_{81}$
\textbf{RTEC}_A: Windowing

\begin{align*}
\text{holdsFor}(\text{disappearedInArea}(\text{Vessel}, \text{Area Type}) = \text{true}, l) & \leftarrow \\
\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{Area Type}) = \text{true}, S), \\
\text{holdsFor}(\text{gap}(\text{Vessel}) = \text{farFromPorts}, T), \\
\text{allen}(\text{meets}, S, T, \text{target}, l).
\end{align*}

\textbf{Query time: } q_{82}

\begin{tikzpicture}
  \draw[->] (-2,0) -- (4,0);
  \node at (-2,0) {$S$};
  \node at (4,0) {$T$};
  \node at (-2,0.2) {$q_{81}$};
  \node at (4,0.2) {$q_{82}$};
  \draw (0,0) -- (2,0);
  \draw[red] (1,0) -- (1,0.2);
  \draw[blue] (2.5,0) -- (3.5,0);
\end{tikzpicture}
\textbf{RTEC}_A: Windowing

\begin{equation*}
\text{holdsFor}(\text{disappearedInArea}(\text{Vessel}, \text{Area Type}) = \text{true}, I) \leftarrow \\
\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{Area Type}) = \text{true}, S), \\
\text{holdsFor}(\text{gap}(\text{Vessel}) = \text{farFromPorts}, T), \\
\text{allen}(\text{meets}, S, T, \text{target}, I).
\end{equation*}

\begin{center}
Query time: $q_{82} \quad w_{82} \quad S \quad T \quad q_{81} \quad q_{82}$
\end{center}
RTEC\textsubscript{A}: Windowing

\[
\text{holdsFor}(\text{disappearedInArea}(Vessel, \text{Area Type}) = \text{true}, I) \leftarrow \\
\text{holdsFor}(\text{withinArea}(Vessel, \text{Area Type}) = \text{true}, S), \\
\text{holdsFor}(\text{gap}(Vessel) = \text{farFromPorts}, T), \\
\text{allen}(\text{meets}, S, T, \text{target}, I).
\]

\text{Query time: } q_{82}

\text{Query time: } q_{82}

\text{is, } q_{82}

\text{2, it, } q_{82}

\text{Query time: } q_{82}

\text{Query time: } q_{82}
holdsFor(disappearedInArea(Vessel, AreaType) = true, I) ←
holdsFor(withinArea(Vessel, AreaType) = true, S),
holdsFor(gap(Vessel) = farFromPorts, T),
allen(meets, S, T, target, I).

Query time: $q_{82}$

$w_{82}$
RTEC_A: Windowing

\[ \text{holdsFor}(\text{disappearedInArea}(\text{Vessel}, \text{Area Type}) = \text{true}, I) \leftarrow \text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{Area Type}) = \text{true}, S), \]
\[ \text{holdsFor}(\text{gap}(\text{Vessel}) = \text{farFromPorts}, T), \]
\[ \text{allen}(\text{meets}, S, T, \text{target}, I). \]

Query time: \( q_{82} \)
**RTEC_A: Correctness & Complexity**

<table>
<thead>
<tr>
<th>Correctness of RTEC_A</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTEC_A computes all maximal intervals of a fluent defined in terms of an Allen relation, and no other interval.</td>
</tr>
</tbody>
</table>
Correctness of RTEC$_A$

RTEC$_A$ computes all maximal intervals of a fluent defined in terms of an Allen relation, and no other interval.

Complexity of RTEC$_A$

The cost of computing the maximal intervals of a fluent defined in terms of an Allen relation is $O(n)$, where $n$ is the number of input intervals.
Experimental Evaluation

Code, Data & Temporal Specifications:

- [https://github.com/aartikis/RTEC/tree/allen](https://github.com/aartikis/RTEC/tree/allen)
### Experimental Evaluation

<table>
<thead>
<tr>
<th>Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input Intervals</th>
<th>RTEC\textsubscript{A}</th>
<th>D\textsuperscript{2}IA</th>
<th>RTEC\textsubscript{A}</th>
<th>D\textsuperscript{2}IA</th>
</tr>
</thead>
<tbody>
<tr>
<td>19K</td>
<td>40</td>
<td>410</td>
<td>6K</td>
<td>6K</td>
</tr>
<tr>
<td>37K</td>
<td>65</td>
<td>592</td>
<td>9K</td>
<td>9K</td>
</tr>
<tr>
<td>74K</td>
<td>99</td>
<td>1.1K</td>
<td>16K</td>
<td>16K</td>
</tr>
<tr>
<td>148K</td>
<td>156</td>
<td>1.6K</td>
<td>32K</td>
<td>31K</td>
</tr>
<tr>
<td>297K</td>
<td>285</td>
<td>2.7K</td>
<td>77K</td>
<td>76K</td>
</tr>
</tbody>
</table>
Summary & Further Work

RTEC_A:

- An open-source complex event recognition framework.
- Support for Allen relations in event patterns.
- Correct Allen relation computation with windowing.
- Linear time complexity.
- Reproducible empirical evaluation on large, real data streams.

Further Work:

- Support approximate Allen relations.
- Contrast Allen relation with event sequencing operators.
- Support events with delayed effects.
Summary & Further Work

RTEC_A:

- An open-source complex event recognition framework.
- Support for Allen relations in event patterns.
- Correct Allen relation computation with windowing.
- Linear time complexity.
- Reproducible empirical evaluation on large, real data streams.

Further Work:

- Support approximate Allen relations.
- Contrast Allen relation with event sequencing operators.
- Support events with delayed effects.
Run-Time Event Calculus (RTEC)

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>happensAt($E$, $T$)</td>
<td>Event $E$ occurs at time $T$</td>
</tr>
<tr>
<td>initiatedAt($F = V$, $T$)</td>
<td>At time $T$ a period of time for which $F = V$ is initiated</td>
</tr>
<tr>
<td>terminatedAt($F = V$, $T$)</td>
<td>At time $T$ a period of time for which $F = V$ is terminated</td>
</tr>
<tr>
<td>holdsFor($F = V$, $I$)</td>
<td>$I$ is the list of the maximal intervals for which $F = V$ holds continuously</td>
</tr>
<tr>
<td>holdsAt($F = V$, $T$)</td>
<td>The value of fluent $F$ is $V$ at time $T$</td>
</tr>
<tr>
<td>union_all([$J_1$, $\ldots$, $J_n$], $I$)</td>
<td>$I = (J_1 \cup \ldots \cup J_n)$</td>
</tr>
<tr>
<td>intersect_all([$J_1$, $\ldots$, $J_n$], $I$)</td>
<td>$I = (J_1 \cap \ldots \cap J_n)$</td>
</tr>
<tr>
<td>relative_complement_all($I'$, [$J_1$, $\ldots$, $J_n$], $I$)</td>
<td>$I = I' \setminus (J_1 \cup \ldots \cup J_n)$</td>
</tr>
</tbody>
</table>

## Run-Time Event Calculus (RTEC)

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>happensAt(E, T)</code></td>
<td>Event $E$ occurs at time $T$</td>
</tr>
<tr>
<td><code>initiatedAt(F = V, T)</code></td>
<td>At time $T$ a period of time for which $F = V$ is initiated</td>
</tr>
<tr>
<td><code>terminatedAt(F = V, T)</code></td>
<td>At time $T$ a period of time for which $F = V$ is terminated</td>
</tr>
<tr>
<td><code>holdsFor(F = V, I)</code></td>
<td>$I$ is the list of the maximal intervals for which $F = V$ holds continuously</td>
</tr>
<tr>
<td><code>holdsAt(F = V, T)</code></td>
<td>The value of fluent $F$ is $V$ at time $T$</td>
</tr>
<tr>
<td><code>union_all([J₁, ..., Jₙ], I)</code></td>
<td>$I = (J₁ \cup ... \cup Jₙ)$</td>
</tr>
<tr>
<td><code>intersect_all([J₁, ..., Jₙ], I)</code></td>
<td>$I = (J₁ \cap ... \cap Jₙ)$</td>
</tr>
<tr>
<td><code>relative_complement_all(I', [J₁, ..., Jₙ], I)</code></td>
<td>$I = I' \setminus (J₁ \cup ... \cup Jₙ)$</td>
</tr>
</tbody>
</table>

relative_complement_all
(I₁, [I₂], I)

\[
(I₁, [I₂], I)
\]
**RTEC\(_A\): RTEC with Allen Relations**

\[
\text{holdsFor}(\text{suspiciousRendezVous}(\text{Vessel}_1, \text{Vessel}_2) = \text{true}, I) \leftarrow \\
\text{holdsFor}(\text{gap}(\text{Vessel}_1) = \text{farFromPorts}, I_{g_1}), \\
\text{holdsFor}(\text{gap}(\text{Vessel}_2) = \text{farFromPorts}, I_{g_2}), \\
\text{holdsFor}(\text{proximity}(\text{Vessel}_1, \text{Vessel}_2) = \text{true}, T), \\
\text{union\_all}([I_{g_1}, I_{g_2}, S]), \\
\text{allen}(\text{during}, S, T, \text{target}, I).
\]
Experimental Evaluation

### Batch setting.

<table>
<thead>
<tr>
<th>Batch size</th>
<th>Reasoning Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Intervals</td>
<td>RTEC&lt;sub&gt;A&lt;/sub&gt;</td>
</tr>
<tr>
<td>200</td>
<td>1</td>
</tr>
<tr>
<td>2K</td>
<td>14</td>
</tr>
<tr>
<td>20K</td>
<td>154</td>
</tr>
<tr>
<td>200K</td>
<td>1.8K</td>
</tr>
</tbody>
</table>

### Streaming setting.

<table>
<thead>
<tr>
<th>Window size</th>
<th>Reasoning Time</th>
<th>Output Interval Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Days</td>
<td>Input Intervals</td>
<td>RTEC&lt;sub&gt;A&lt;/sub&gt;</td>
</tr>
<tr>
<td>1</td>
<td>125</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>250</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>500</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>1K</td>
<td>8</td>
</tr>
<tr>
<td>16</td>
<td>2K</td>
<td>15</td>
</tr>
</tbody>
</table>