Stream Reasoning with Cycles

Periklis Mantenoglou\textsuperscript{2,1}  Manolis Pitsikalis\textsuperscript{4}  
Alexander Artikis\textsuperscript{3,1}

\textsuperscript{1}NCSR Demokritos, Greece  
\textsuperscript{2}National and Kapodistrian University of Athens, Greece  
\textsuperscript{3}University of Piraeus, Greece  
\textsuperscript{4}University of Liverpool, UK

http://cer.iit.demokritos.gr/
Stream Reasoning

**INPUT** ▶ **RECOGNITION** ▶ **OUTPUT**

Stream Reasoning System

Streams of Simple Events

Temporal (+Spatial) Specifications

Recognised Complex Events
Stream Reasoning

INPUT ▶ RECOGNITION ▶ OUTPUT ▶
Stream Reasoning System
Temporal (+Spatial) Specifications
Streams of Simple Events
Recognised Complex Events

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

<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>
</tbody>
</table>

Fluent-Value Pair Specification

Definition:

\[
\text{initiatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{ln_1}, T),
\quad [\text{conditions}]
\]

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

\[
\text{initiatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{ln_i}, T),
\quad [\text{conditions}]
\]

\[
\text{terminatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{T_j}, T),
\quad [\text{conditions}]
\]

where

\[
\text{conditions:} \quad 0^K \text{happensAt}(E_k, T),
0^-M \text{holdsAt}(F_m = V_m, T),
0^-N \text{atemporal-constraint}_n
\]
Fluent-Value Pair Computation

Definition:

\[ \text{initiatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{In}, T), \quad \text{[conditions]} \]

\[ \quad \ldots \]

\[ \text{initiatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{In_i}, T), \quad \text{[conditions]} \]

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

\[ \quad \ldots \]

\[ \text{terminatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{T_j}, T), \quad \text{[conditions]} \]

Reasoning:
Fluent-Value Pair Computation

Definition:

\[ \text{initiatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{ln_1}, T), \]
\[ \text{[conditions]} \]

\[ \cdots \]

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

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

\[ \cdots \]

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

Reasoning:

\[ \text{time} \]

\[ 0 \]
Fluent-Value Pair Computation

Definition:

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

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

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

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

Reasoning:
Fluent-Value Pair Computation

Definition:

\[
\text{initiatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{ln1}, T), \text{[conditions]}
\]

\[
\ldots
\]

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

Reasoning: \text{holdsFor}(F = V, I)

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

\[
\ldots
\]

\[
\text{terminatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{Tj}, T), \text{[conditions]}
\]
Cyclic Dependencies in Temporal Specifications

\[
\text{initiatedAt}(\text{status}(M) = \text{proposed}, T) \leftarrow \\
\text{happensAt}(\text{propose}(P, M), T), \\
\text{holdsAt}(\text{status}(M) = \text{null}, T).
\]
Cyclic Dependencies in Temporal Specifications

\[
\text{initiatedAt}(status(M) = \text{proposed}, T) \leftarrow \\
\text{happensAt}(\text{propose}(P, M), T), \\
\text{holdsAt}(status(M) = \text{null}, T).
\]

\[
\text{initiatedAt}(status(M) = \text{voting}, T) \leftarrow \\
\text{happensAt}(\text{second}(S, M), T), \\
\text{holdsAt}(status(M) = \text{proposed}, T).
\]
Cyclic Dependencies in Temporal Specifications

\[
\text{initiatedAt}(\text{status}(M) = \text{proposed}, T) \leftarrow \\
\text{happensAt}(\text{propose}(P, M), T), \\
\text{holdsAt}(\text{status}(M) = \text{null}, T).
\]

\[
\text{initiatedAt}(\text{status}(M) = \text{voting}, T) \leftarrow \\
\text{happensAt}(\text{second}(S, M), T), \\
\text{holdsAt}(\text{status}(M) = \text{proposed}, T).
\]

\[
\text{initiatedAt}(\text{status}(M) = \text{voted}, T) \leftarrow \\
\text{happensAt}(\text{close_ballot}(C, M), T), \\
\text{holdsAt}(\text{status}(M) = \text{voting}, T).
\]
Cyclic Dependencies in Temporal Specifications

\[
\text{initiatedAt}(\text{status}(M) = \text{proposed}, T) \leftarrow \\
\text{happensAt}(\text{propose}(P, M), T), \\
\text{holdsAt}(\text{status}(M) = \text{null}, T).
\]

\[
\text{initiatedAt}(\text{status}(M) = \text{voting}, T) \leftarrow \\
\text{happensAt}(\text{second}(S, M), T), \\
\text{holdsAt}(\text{status}(M) = \text{proposed}, T).
\]

\[
\text{initiatedAt}(\text{status}(M) = \text{voted}, T) \leftarrow \\
\text{happensAt}(\text{close\_ballot}(C, M), T), \\
\text{holdsAt}(\text{status}(M) = \text{voting}, T).
\]

\[
\text{initiatedAt}(\text{status}(M) = \text{null}, T) \leftarrow \\
\text{happensAt}(\text{declare}(C, M, \text{Res}), T), \\
\text{holdsAt}(\text{status}(M) = \text{voted}, T).
\]
Cyclic Dependencies in Temporal Specifications

\[
\begin{align*}
\text{initiatedAt}(status(M) = \textit{proposed}, T) & \leftarrow \\
\text{happensAt}(\textit{propose}(P, M), T), \\
\text{holdsAt}(status(M) = \textit{null}, T).
\end{align*}
\]

\[
\begin{align*}
\text{initiatedAt}(status(M) = \textit{voting}, T) & \leftarrow \\
\text{happensAt}(\textit{second}(S, M), T), \\
\text{holdsAt}(status(M) = \textit{proposed}, T).
\end{align*}
\]

\[
\begin{align*}
\text{initiatedAt}(status(M) = \textit{voted}, T) & \leftarrow \\
\text{happensAt}(\textit{close\_ballot}(C, M), T), \\
\text{holdsAt}(status(M) = \textit{voting}, T).
\end{align*}
\]

\[
\begin{align*}
\text{initiatedAt}(status(M) = \textit{null}, T) & \leftarrow \\
\text{happensAt}(\textit{declare}(C, M, Res), T), \\
\text{holdsAt}(status(M) = \textit{voted}, T).
\end{align*}
\]
Cyclic Dependencies in Temporal Specifications

\[
\begin{align*}
\text{initiatedAt}(status(M) = \text{proposed}, T) & \leftarrow \\
\text{happensAt}(\text{propose}(P, M), T), \\
\text{holdsAt}(status(M) = \text{null}, T).
\end{align*}
\]

\[
\begin{align*}
\text{initiatedAt}(status(M) = \text{voting}, T) & \leftarrow \\
\text{happensAt}(\text{second}(S, M), T), \\
\text{holdsAt}(status(M) = \text{proposed}, T).
\end{align*}
\]

\[
\begin{align*}
\text{initiatedAt}(status(M) = \text{voted}, T) & \leftarrow \\
\text{happensAt}(\text{close_ballot}(C, M), T), \\
\text{holdsAt}(status(M) = \text{voting}, T).
\end{align*}
\]

\[
\begin{align*}
\text{initiatedAt}(status(M) = \text{null}, T) & \leftarrow \\
\text{happensAt}(\text{declare}(C, M, Res), T), \\
\text{holdsAt}(status(M) = \text{voted}, T).
\end{align*}
\]
Cyclic Dependencies in Temporal Specifications

\[
\text{initiatedAt}(\text{status}(M) = \text{proposed}, T) \leftarrow \\
\text{happensAt}(\text{propose}(P, M), T), \\
\text{holdsAt}(\text{status}(M) = \text{null}, T).
\]

\[
\text{initiatedAt}(\text{status}(M) = \text{voting}, T) \leftarrow \\
\text{happensAt}(\text{second}(S, M), T), \\
\text{holdsAt}(\text{status}(M) = \text{proposed}, T).
\]

\[
\text{initiatedAt}(\text{status}(M) = \text{voted}, T) \leftarrow \\
\text{happensAt}(\text{close_ballot}(C, M), T), \\
\text{holdsAt}(\text{status}(M) = \text{voting}, T).
\]

\[
\text{initiatedAt}(\text{status}(M) = \text{null}, T) \leftarrow \\
\text{happensAt}(\text{declare}(C, M, \text{Res}), T), \\
\text{holdsAt}(\text{status}(M) = \text{voted}, T).
\]
Cyclic Dependencies in Temporal Specifications

\[
\text{initiatedAt}(\text{status}(M) = \text{proposed}, T) \leftarrow \\
\text{happensAt}(\text{propose}(P, M), T), \\
\text{holdsAt}(\text{status}(M) = \text{null}, T).
\]

\[
\text{initiatedAt}(\text{status}(M) = \text{voting}, T) \leftarrow \\
\text{happensAt}(\text{second}(S, M), T), \\
\text{holdsAt}(\text{status}(M) = \text{proposed}, T).
\]

\[
\text{initiatedAt}(\text{status}(M) = \text{voted}, T) \leftarrow \\
\text{happensAt}(\text{close_ballot}(C, M), T), \\
\text{holdsAt}(\text{status}(M) = \text{voting}, T).
\]

\[
\text{initiatedAt}(\text{status}(M) = \text{null}, T) \leftarrow \\
\text{happensAt}(\text{declare}(C, M, \text{Res}), T), \\
\text{holdsAt}(\text{status}(M) = \text{voted}, T).
\]
RTEC°: Run-Time Event Calculus for Cyclic Dependencies

RTEC°: an extension of RTEC for efficient reasoning over specifications with cyclic dependencies

- Formal & open-source computational framework
- Locally stratified specifications
- Incremental caching of intermediate computations
- Scalable in large, real-world data streams
Semantics

An event description of RTEC

- status = proposed
- status = voting
- status = voted
- status = null

- auxPer = true
- outcome = carried
- permission = true
- sanctioned = true
- power = true
- obligation = true
- voted = aye/nay
- voted = null

- status = true
- outcome = carried
- permission = true
- sanctioned = true
- power = true
- obligation = true
- voted = aye/nay
- voted = null
An event description of RTEC₀ is a locally stratified logic program.
Handling Cyclic Dependencies

\[ F_1 = V_1 \]
\[ F_2 = V_2 \]
Handling Cyclic Dependencies

\[ F_1 = V_1 \]
\[ F_2 = V_2 \]
Handling Cyclic Dependencies

\[ F_1 = V_1, \quad F_2 = V_2 \]

\[ q_i - \omega \quad \omega \quad q_i \]

\[ \text{time} \]
Handling Cyclic Dependencies

\[ F_1 = V_1 \]
\[ F_2 = V_2 \]
Handling Cyclic Dependencies

\[ F_1 = V_1, \quad F_2 = V_2 \]

\[ q_i - \omega \quad \omega \quad q_i \]

\[ \text{time} \]
Handling Cyclic Dependencies

\[ F_1 = V_1 \]
\[ F_2 = V_2 \]

\[ \omega \]

\[ q_i - \omega \]
\[ q_i \]

\[ time \]
Handling Cyclic Dependencies

\( F_1 = V_1 \)
\( F_2 = V_2 \)

\( q_i - \omega \quad \omega \quad q_i \)

time
Handling Cyclic Dependencies

\[ \omega q_i - \omega F_1 = V_1 \]
\[ \omega q_i - \omega F_2 = V_2 \]
Handling Cyclic Dependencies

\[ F_1 = V_1, \quad F_2 = V_2 \]

\[ q_i - \omega \quad \omega \quad q_i \]

\[ time \]
Handling Cyclic Dependencies

\[
\begin{align*}
F_1 &= V_1 \\
F_2 &= V_2
\end{align*}
\]
Handling Cyclic Dependencies

\[ F_1 = V_1, \quad F_2 = V_2 \]

\[ \omega \]

\[ q_i - \omega \quad q_i \]

\[ time \]
Handling Cyclic Dependencies

\[
F_1 = V_1 \\
F_2 = V_2
\]
Handling Cyclic Dependencies

\[ F_1 = V_1, \quad F_2 = V_2 \]

\[ q_i - \omega, \quad q_i \]
Handling Cyclic Dependencies

\[ F_1 = V_1 \]
\[ F_2 = V_2 \]
Handling Cyclic Dependencies

$$\begin{align*}
F_1 &= V_1 \\
F_2 &= V_2
\end{align*}$$
Handling Cyclic Dependencies

\[
\begin{align*}
F_1 &= V_1 \\
F_2 &= V_2
\end{align*}
\]
Handling Cyclic Dependencies

\[ F_1 = V_1 \]
\[ F_2 = V_2 \]

\[ q_i - \omega \]
\[ q_i \]
Handling Cyclic Dependencies

RTEC_0

Event Calculus
Handling Cyclic Dependencies

RTEC_o

Event Calculus
Handling Cyclic Dependencies

\[ \text{RTEC}_c \]

Event Calculus

\[ \text{F}_1 = V_1 \]
\[ \text{F}_2 = V_2 \]
Handling Cyclic Dependencies

RTEC

\[ F_1 = V_1 \]
\[ F_2 = V_2 \]

Event Calculus

\[ F_1 = V_1 \]
\[ F_2 = V_2 \]
Handling Cyclic Dependencies

RTEC

Event Calculus
Handling Cyclic Dependencies

**RTEC**

\[
\begin{align*}
F_1 &= V_1 \\
F_2 &= V_2
\end{align*}
\]

**Event Calculus**

\[
\begin{align*}
F_1 &= V_1 \\
F_2 &= V_2
\end{align*}
\]
Handling Cyclic Dependencies

**RTEC**

\[
\begin{align*}
F_1 &= V_1 \\
F_2 &= V_2
\end{align*}
\]

time

Event Calculus

\[
\begin{align*}
F_1 &= V_1 \\
F_2 &= V_2
\end{align*}
\]

time
Handling Cyclic Dependencies

RTEC

Event Calculus
Handling Cyclic Dependencies

\[ F_1 = V_1 \]
\[ F_2 = V_2 \]

\[ \omega \]

Event Calculus
Handling Cyclic Dependencies

RTEC_0

Event Calculus
Handling Cyclic Dependencies

RTEC

Event Calculus
Handling Cyclic Dependencies

RTEC

Event Calculus
Handling Cyclic Dependencies

\[ \text{RTEC}_o \]

Event Calculus

\[ \text{F}_1 = V_1 \]
\[ \text{F}_2 = V_2 \]
Handling Cyclic Dependencies

\[ \text{RTEC}_0 \]

\[ F_1 = V_1, \quad F_2 = V_2 \]

Event Calculus

\[ \text{time} \]

\[ \omega \]

\[ q_i - \omega \quad q_i \]

\[ \text{time} \]
Handling Cyclic Dependencies

\[ \text{RTEC}_\circ \]

\[ \omega \]

Event Calculus

\[ \mathcal{O}(m^2 \omega k) \]

\[ \mathcal{O}(m \omega k) \]
Experimental Setup

Multi-Agent Systems: Voting & NetBill

- Compute, e.g., normative positions of agents.
Experimental Setup

Multi-Agent Systems: Voting & NetBill

- Compute, e.g., normative positions of agents.

Maritime Situational Awareness

- Recognise dangerous, illegal and suspicious vessel activity.

[Map images showing maritime activity]
Experimental Setup

Multi-Agent Systems: Voting & NetBill

• Compute, e.g., normative positions of agents.

Maritime Situational Awareness

• Recognise dangerous, illegal and suspicious vessel activity.

Code, Data & Temporal Specifications

https://github.com/aartikis/RTEC
Experimental Results

Voting & NetBill

![Graph showing experimental results for Voting & NetBill]
Experimental Results

Voting & NetBill

Voting: status fluent
Experimental Results

Voting & NetBill

Voting: status fluent

Brest
Experimental Results

Voting & NetBill

Voting: status fluent

Brest

European seas
**Summary & Further Work**

**RTEC**

- A formal, open-source stream reasoning system
- Efficient treatment of cyclic dependencies
- Locally stratified specifications
- Reproducible empirical evaluation on large data streams

**Further Work**

- Efficient treatment of deadlines
- Integrating RTEC in neuro-symbolic frameworks

**Resources**

https://cer.iit.demokritos.gr/
Summary & Further Work

RTEC

• A formal, open-source stream reasoning system
• Efficient treatment of cyclic dependencies
• Locally stratified specifications
• Reproducible empirical evaluation on large data streams

Further Work

• Efficient treatment of deadlines
• Integrating RTEC in neuro-symbolic frameworks

Resources

https://cer.iit.demokritos.gr/
Summary & Further Work

RTEC\textsuperscript{o}.

- A formal, open-source stream reasoning system
- Efficient treatment of cyclic dependencies
- Locally stratified specifications
- Reproducible empirical evaluation on large data streams

Further Work

- Efficient treatment of deadlines
- Integrating RTEC\textsuperscript{o} in neuro-symbolic frameworks

Resources

https://cer.iit.demokritos.gr/