

Incremental Event Calculus for Run-Time Reasoning

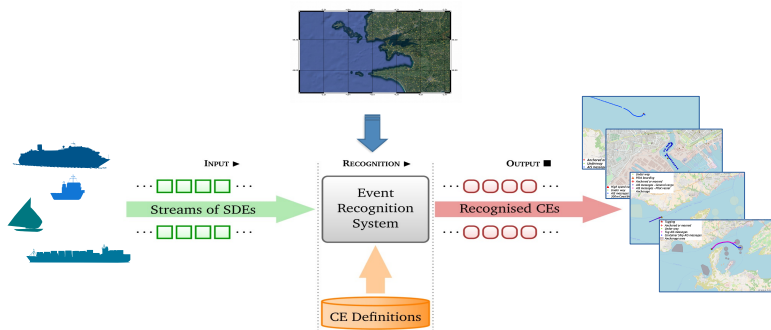
Efthimis Tsilionis, Alexander Artikis, Georgios Paliouras

Institute of Informatics and Telecommunications,
NCSR Demokritos, Greece

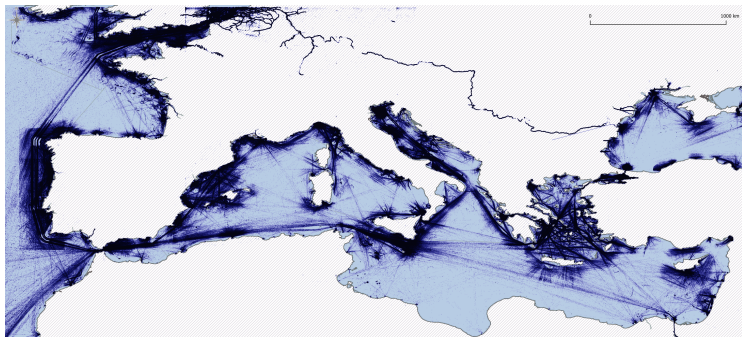
<http://cer.iit.demokritos.gr/>

27 June 2019

Composite Event Recognition (CER)



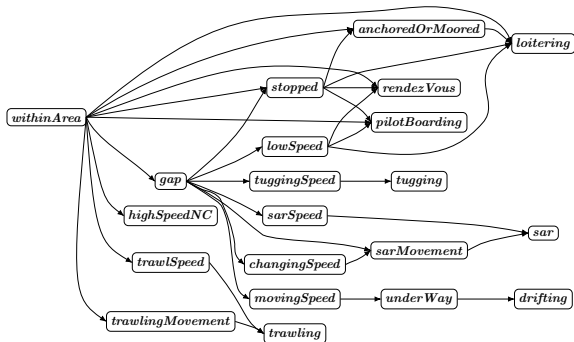
Motivation for Incremental CER



- ▶ Delayed events (e.g., satellite GPS messages)
- ▶ Overlapping temporal windows

Motivation for Incremental CER

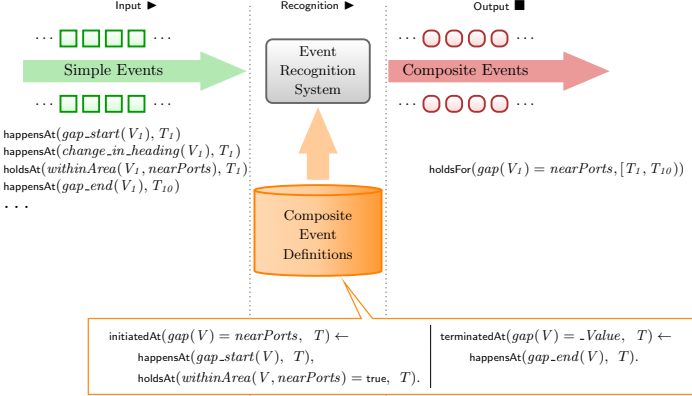
Propagation of changes



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.
- ▶ RTEC is a CER system based on the Event Calculus formalism

Run-Time Event Calculus

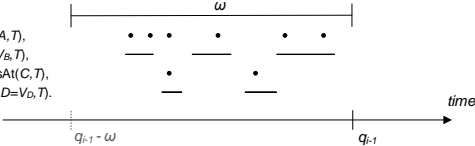


Problem Statement

initiatedAt($F=V, T$) ←

↑ ↑ ↑ ↑ ↑

happensAt(A, T),
holdsAt($B=V_B, T$),
not happensAt(C, T),
not holdsAt($D=V_D, T$).

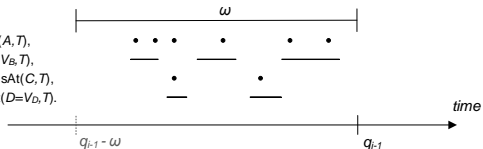


Problem Statement

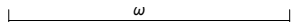
initiatedAt($F=V, T$) ←



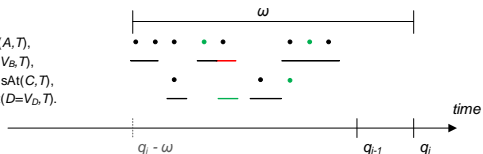
happensAt(A, T),
holdsAt($B=V_B, T$),
not happensAt(C, T),
not holdsAt($D=V_D, T$).



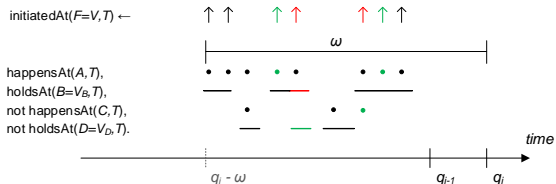
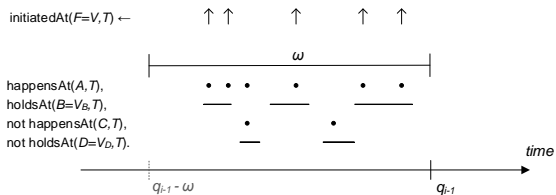
initiatedAt($F=V, T$) ←



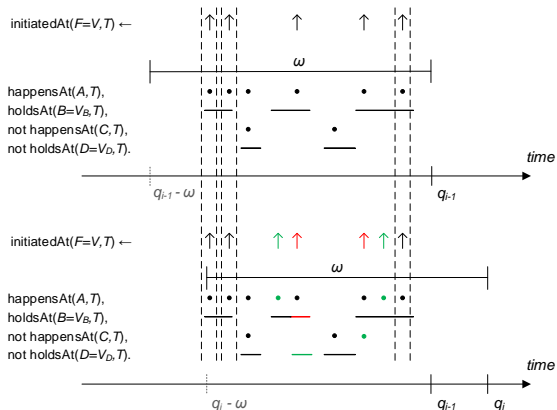
happensAt(A, T),
holdsAt($B=V_B, T$),
not happensAt(C, T),
not holdsAt($D=V_D, T$).



Problem Statement



Problem Statement: Inefficiency

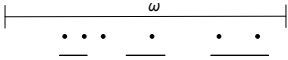


Incremental RTEC

initiatedAt($F=V, T$) ←



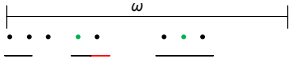
happensAt(A, T),
holdsAt($B=V_B, T$),
not happensAt(C, T),
not holdsAt($D=V_D, T$).



initiatedAt($F=V, T$) ←



happensAt(A, T),
holdsAt($B=V_B, T$),
not happensAt(C, T),
not holdsAt($D=V_D, T$).



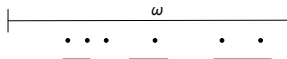
$$S_F^{Q_i} = S_F^{Q_{i-1}} + \Delta S_F(Q_{i-1}, \{ins, del\})$$

Incremental RTEC

initiatedAt($F=V, T$) ←



happensAt(A, T),
holdsAt($B=V_B, T$),
not happensAt(C, T),
not holdsAt($D=V_D, T$).



time

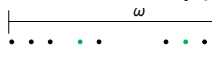
$q_{i-1} - \omega$

q_{i-1}

initiatedAt($F=V, T$) ←



happensAt(A, T),
holdsAt($B=V_B, T$),
not happensAt(C, T),
not holdsAt($D=V_D, T$).



time

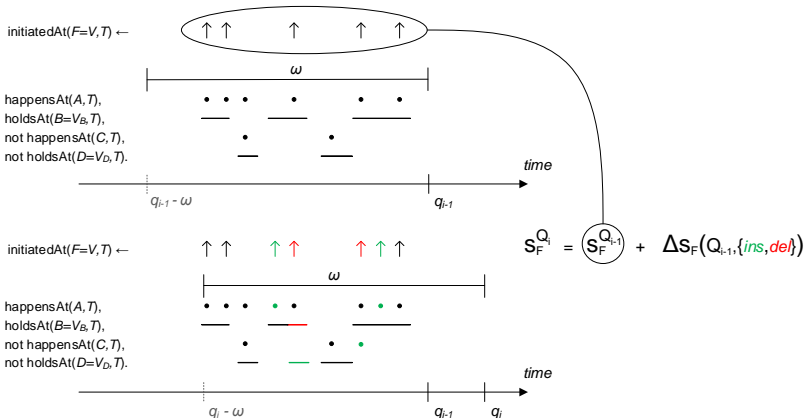
$q_i - \omega$

q_{i-1}

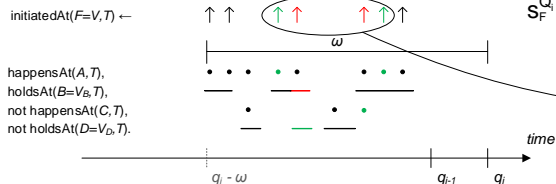
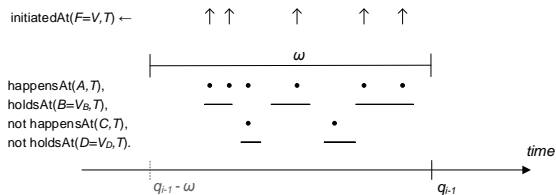
q_i



Incremental RTEC

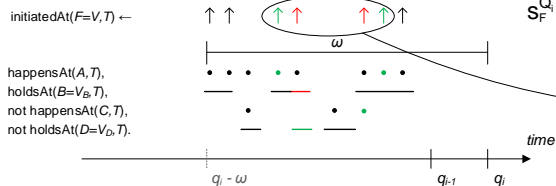
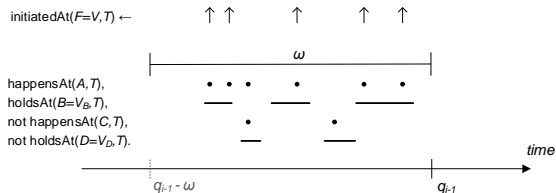


Incremental RTEC



$$S_F^{Q_i} = S_F^{Q_{i-1}} + \Delta S_F(Q_{i-1}, \{\text{ins}, \text{del}\})$$

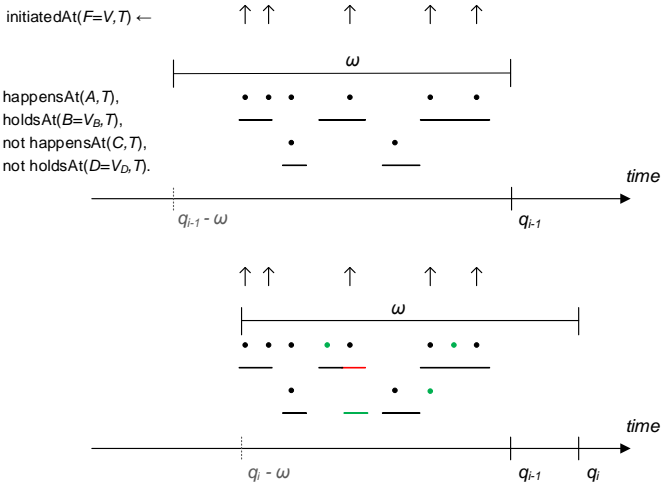
Incremental RTEC



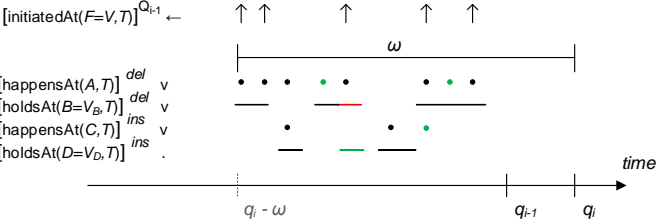
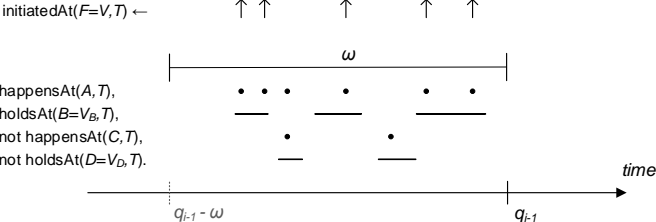
$$S_F^{Q_i} = S_F^{Q_{i-1}} + \Delta S_F(Q_{i-1}, \{ins, del\})$$

- ▶ Two phases:
 - ▶ Deletion phase
 - ▶ Addition phase

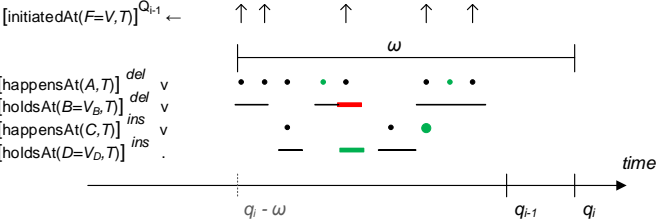
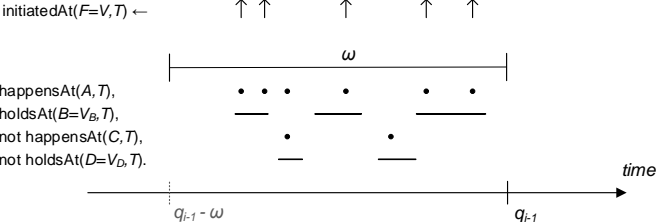
Incremental RTEC - Deletion phase



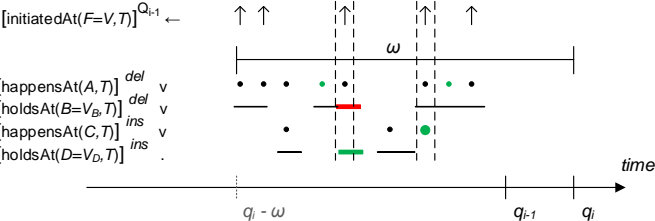
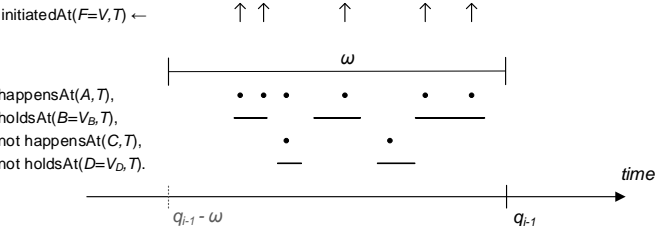
Incremental RTEC - Deletion phase



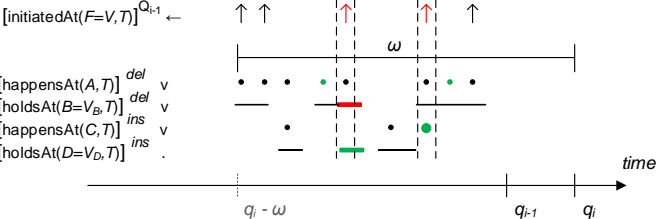
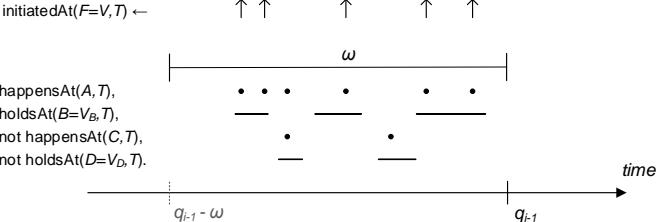
Incremental RTEC - Deletion phase



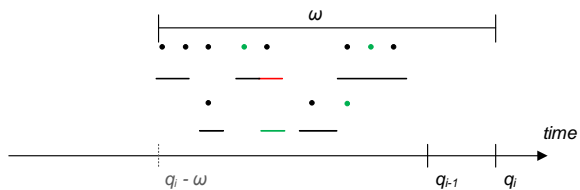
Incremental RTEC - Deletion phase



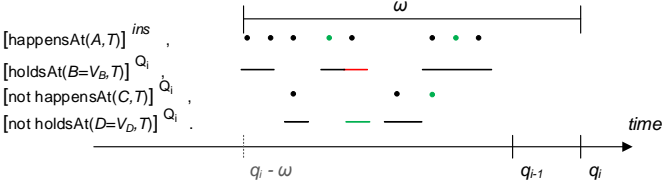
Incremental RTEC - Deletion phase



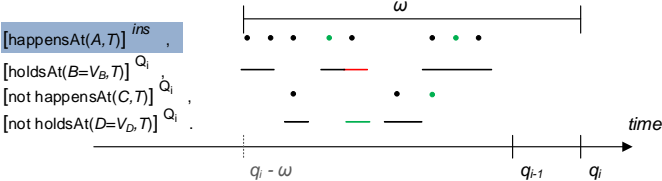
Incremental RTEC - Addition phase



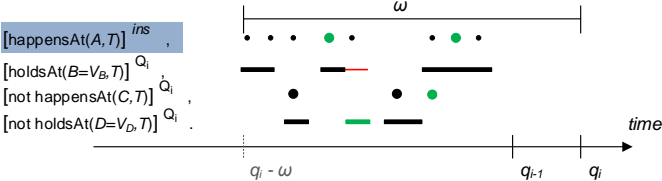
Incremental RTEC - Addition phase



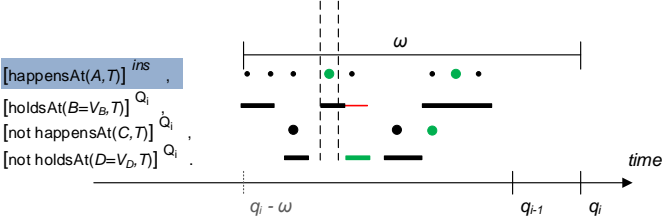
Incremental RTEC - Addition phase



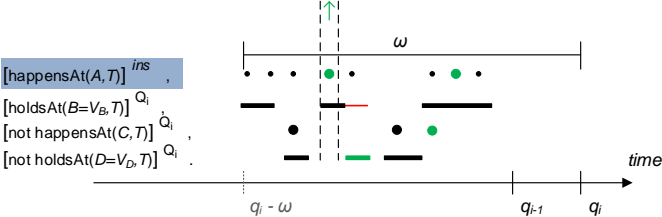
Incremental RTEC - Addition phase



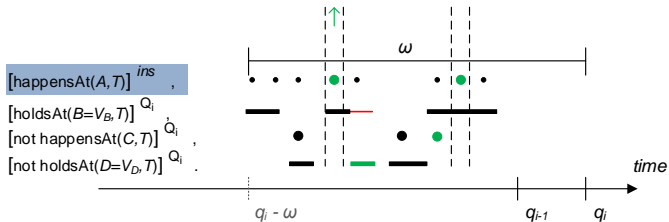
Incremental RTEC - Addition phase



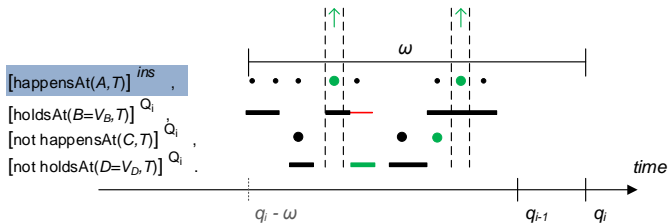
Incremental RTEC - Addition phase



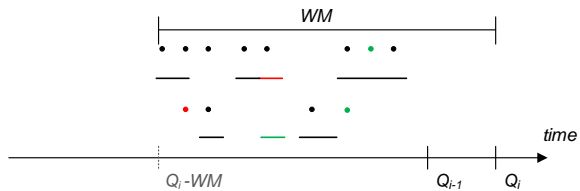
Incremental RTEC - Addition phase



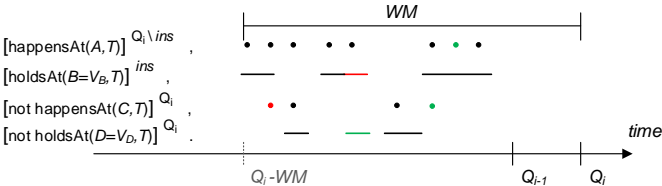
Incremental RTEC - Addition phase



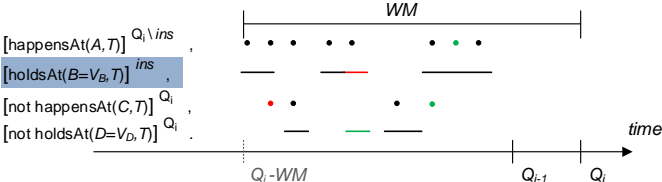
Incremental RTEC - Addition phase



Incremental RTEC - Addition phase

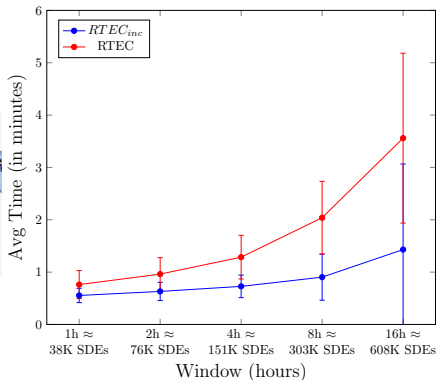
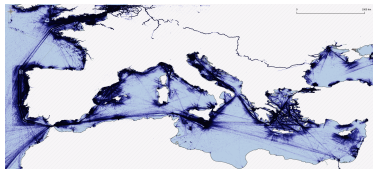


Incremental RTEC - Addition phase



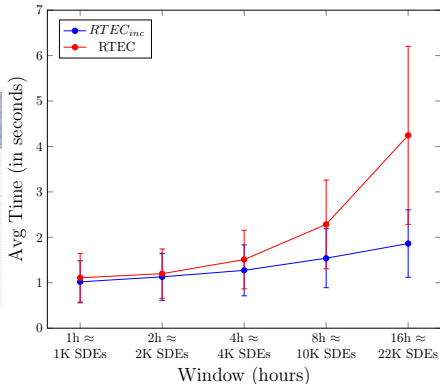
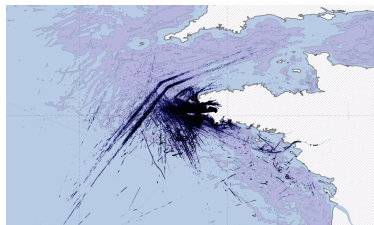
Incremental RTEC - Evaluation (Natural Delays)

- ▶ Delays up to 16 hours
- ▶ 17M position signals, 34K vessels
- ▶ European seas
- ▶ January 2016



Incremental RTEC - Evaluation (Synthetic Delays)

- ▶ 5M position signals, 5K vessels
- ▶ 40% of input events delayed
- ▶ Brest Area
- ▶ October 2015 — March 2016



Summary

- ▶ Properties of the algorithm:
 - ▶ Evaluation of small sets early
 - ▶ Optimal rule rewriting
 - ▶ Can handle retractions in the input
- ▶ We have performed a complexity analysis of the incremental version and have discovered the conditions that lead to better performance
- ▶ Evaluation of the algorithm in a fleet management application
- ▶ Future work:
 - ▶ Probabilistic version of the incremental algorithm