

The Run-Time Event Calculus (RTEC)

Periklis Mantenoglou Manolis Pitsikalis
Alexander Artikis

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



Stream Reasoning

Problem:

- Continuous pattern matching over data streams
- Reasoning over complex temporal specifications:
 - cyclic dependencies
 - deadlines
- Low latency

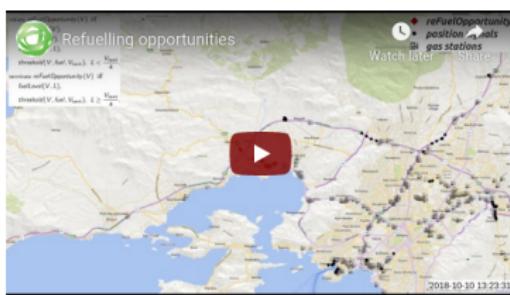
Approach:

- Complex temporal specifications \implies Event Calculus
- Efficient stream reasoning \implies Run-Time Event Calculus

Applications

- Multi-Agent Systems (MAS) for:
 - Voting
 - Negotiation
 - Argumentation
- Complex Event Recognition (CER) for:
 - Maritime Situational Awareness
 - Commercial Fleet Management
 - Activity Recognition
 - City Transport Management

Complex Event Recognition



Event Calculus¹

- 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.

¹Robert A. Kowalski, Marek J. Sergot: A Logic-based Calculus of Events. New Gener. Comput. 4(1): 67-95 (1986)

Event Calculus¹

- 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.

¹ Robert A. Kowalski, Marek J. Sergot: A Logic-based Calculus of Events. New Gener. Comput. 4(1): 67-95 (1986)

Run-Time Event Calculus (RTEC)²

A stream reasoning system based on the Event Calculus:

- windows
- caching and indexing
- interval manipulation constructs
- efficient handling of cyclic dependencies
- efficient handling of deadlines

²Artikis A., Sergot M. and Palouras G., An Event Calculus for Event Recognition. In IEEE Transactions on Knowledge and Data Engineering (TKDE), 27(4), 895–908, 2015.

RTEC: Main Predicates

Predicate	Meaning
$\text{happensAt}(E, T)$	Event E occurs at time T
$\text{initiatedAt}(F = V, T)$	At time T a period of time for which $F = V$ is initiated
$\text{terminatedAt}(F = V, T)$	At time T a period of time for which $F = V$ is terminated
$\text{holdsFor}(F = V, I)$	I is the list of the maximal intervals for which $F = V$ holds continuously
$\text{holdsAt}(F = V, T)$	The value of fluent F is V at time T
$\text{unionall}([J_1, \dots, J_n], I)$	$I = (J_1 \cup \dots \cup J_n)$
$\text{intersectall}([J_1, \dots, J_n], I)$	$I = (J_1 \cap \dots \cap J_n)$
$\text{complementall}(I', [J_1, \dots, J_n], I)$	$I = I' \setminus (J_1 \cup \dots \cup J_n)$

RTEC: Main Predicates

Predicate	Meaning
$\text{happensAt}(E, T)$	Event E occurs at time T
$\text{initiatedAt}(F = V, T)$	At time T a period of time for which $F = V$ is initiated
$\text{terminatedAt}(F = V, T)$	At time T a period of time for which $F = V$ is terminated
$\text{holdsFor}(F = V, I)$	I is the list of the maximal intervals for which $F = V$ holds continuously
$\text{holdsAt}(F = V, T)$	The value of fluent F is V at time T
$\text{unionall}([J_1, \dots, J_n], I)$	$I = (J_1 \cup \dots \cup J_n)$
$\text{intersectall}([J_1, \dots, J_n], I)$	$I = (J_1 \cap \dots \cap J_n)$
$\text{complementall}(I', [J_1, \dots, J_n], I)$	$I = I' \setminus (J_1 \cup \dots \cup J_n)$

Fluent-Value Pair Specification

Definition:

$\text{initiatedAt}(F = V, T) \leftarrow$
 $\text{happensAt}(E_{In_1}, T),$
 [conditions]

...

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

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

...

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

where

conditions: ${}^{0-K}\text{happensAt}(E_k, T),$
 ${}^{0-M}\text{holdsAt}(F_m = V_m, T),$
 ${}^{0-N}\text{atemporal-constraint}_n$

Example: Leaving Object

Definition:

initiatedAt(*leaving_object(P, Obj) = true, T*) ←
happensAt(*appear(Obj), T*),
holdsAt(*inactive(Obj) = true, T*),
holdsAt(*close(P, Obj) = true, T*),
holdsAt(*person(P) = true, T*).

terminatedAt(*leaving_object(P, Obj) = true, T*) ←
happensAt(*disappear(Obj), T*).

Computing the maximal intervals of Fluent-Value Pairs*

Definition:

$\text{initiatedAt}(F = V, T) \leftarrow$
 $\text{happensAt}(E_{In_1}, T),$
[conditions]

...

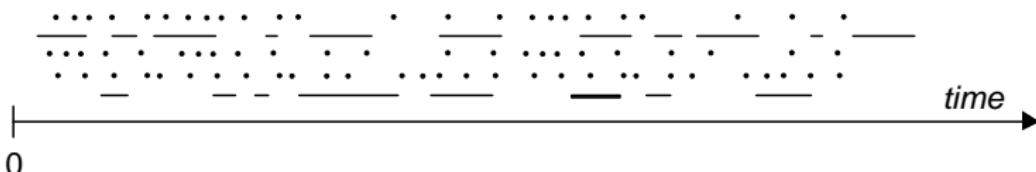
$\text{initiatedAt}(F = V, T) \leftarrow$
 $\text{happensAt}(E_{In_i}, T),$
[conditions]

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

...

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

Computation:



Computing the maximal intervals of Fluent-Value Pairs*

Definition:

$\text{initiatedAt}(F = V, T) \leftarrow$
 $\text{happensAt}(E_{In_1}, T),$
[conditions]

...

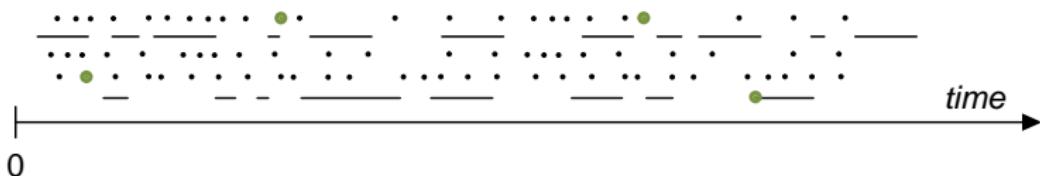
$\text{initiatedAt}(F = V, T) \leftarrow$
 $\text{happensAt}(E_{In_i}, T),$
[conditions]

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

...

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

Computation:



Computing the maximal intervals of Fluent-Value Pairs*

Definition:

$\text{initiatedAt}(F = V, T) \leftarrow$
 $\text{happensAt}(E_{In_1}, T),$
[conditions]

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

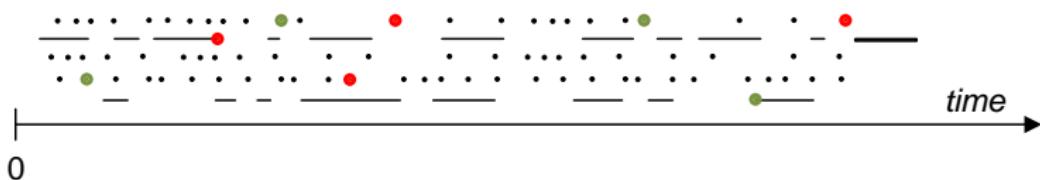
...

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

...

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

Computation:



Computing the maximal intervals of Fluent-Value Pairs*

Definition:

$\text{initiatedAt}(F = V, T) \leftarrow$
 $\text{happensAt}(E_{In_1}, T),$
[conditions]

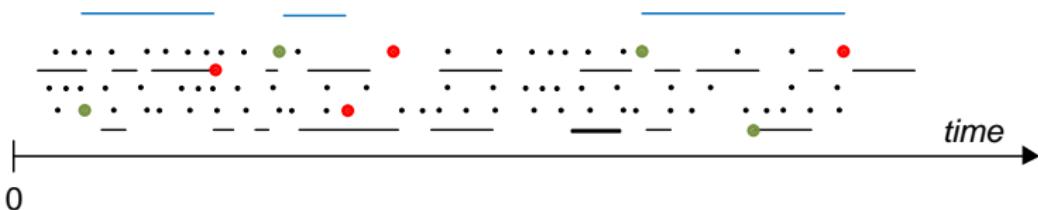
$\text{terminatedAt}(F = V, T) \leftarrow$
 $\text{happensAt}(E_{T_1}, T),$
[conditions]

...

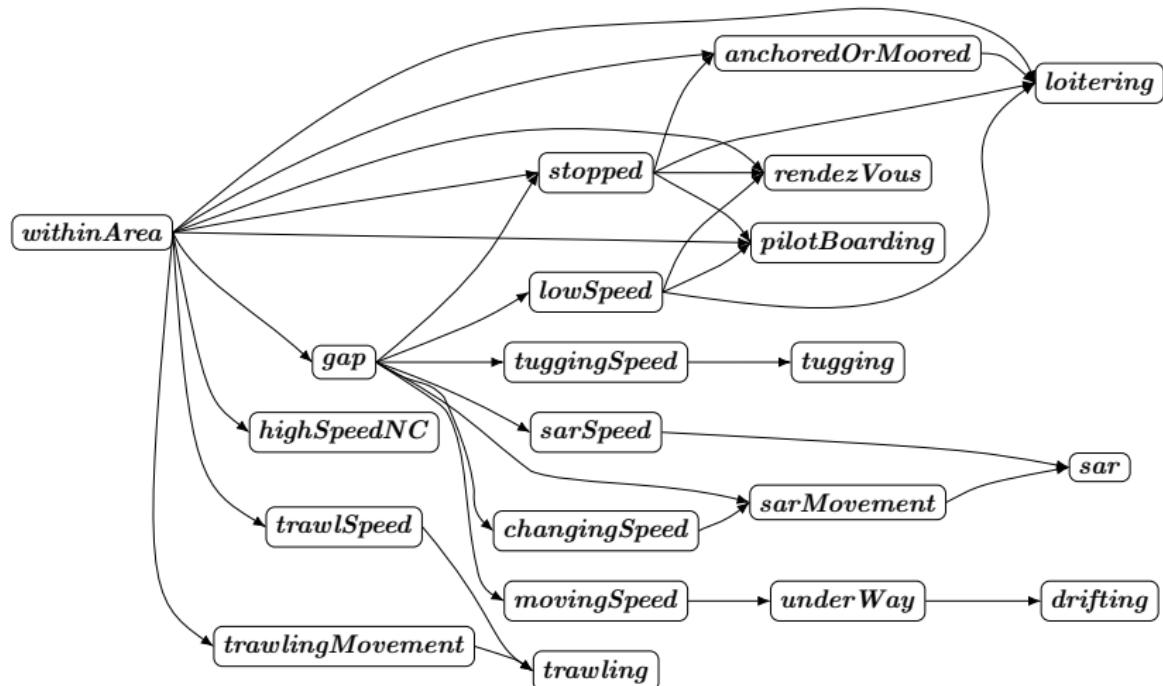
$\text{initiatedAt}(F = V, T) \leftarrow$
 $\text{happensAt}(E_{In_i}, T),$
[conditions]

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

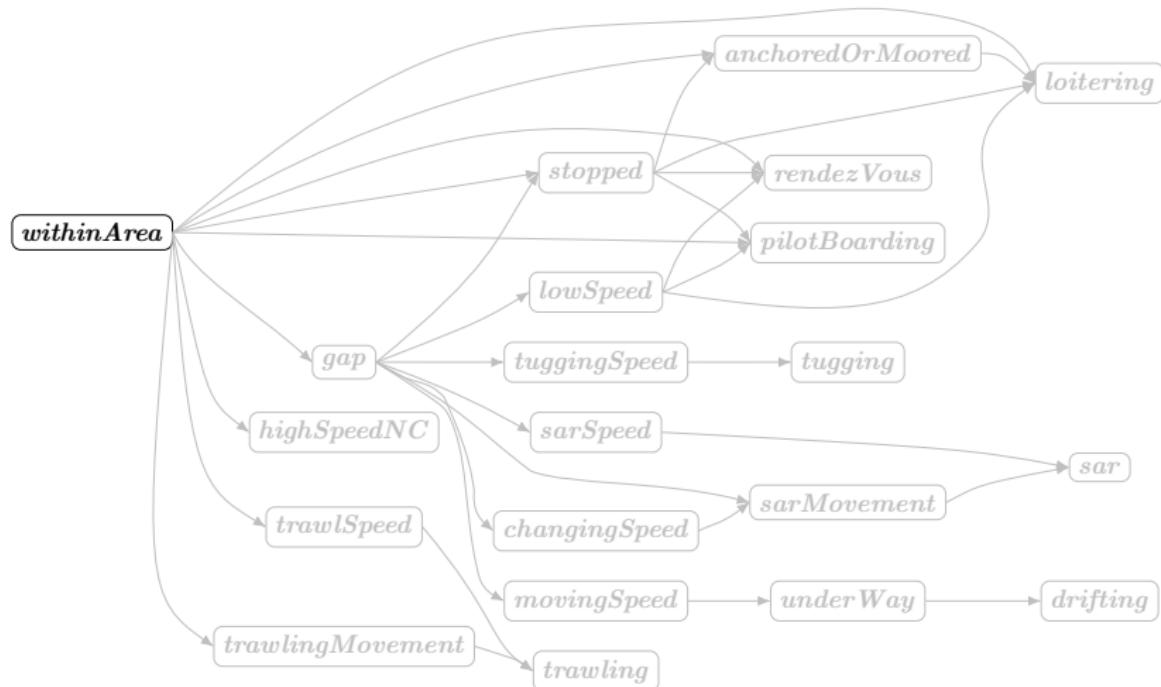
Computation: $\text{holdsFor}(F = V, I)$



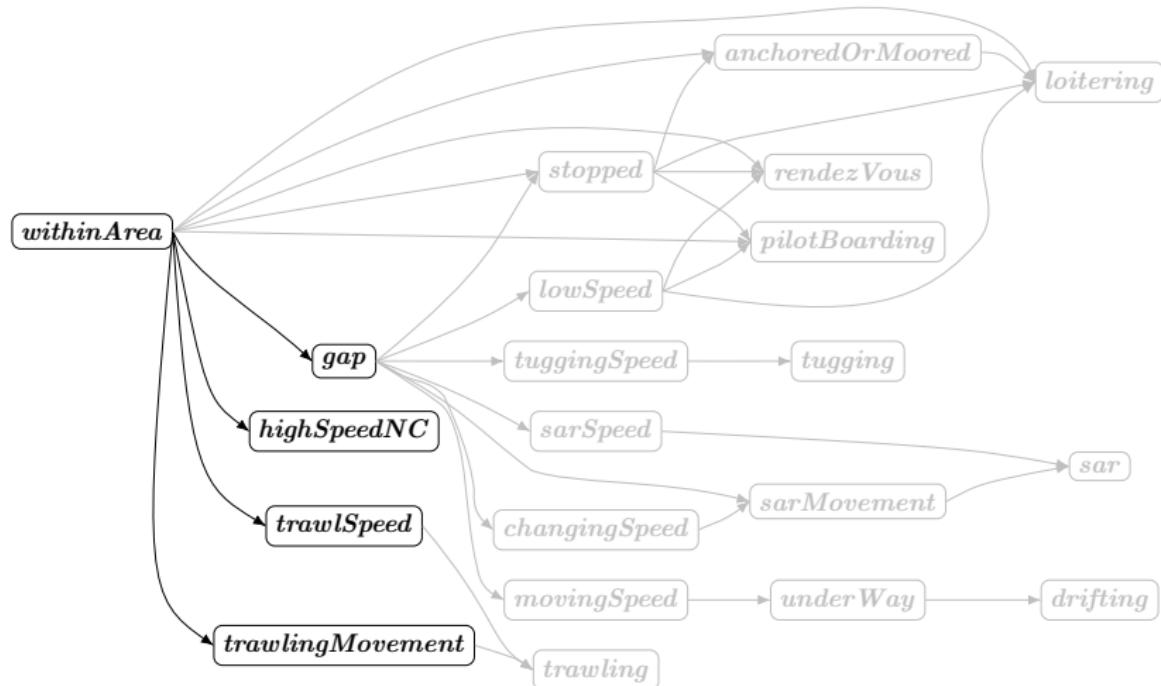
Hierarchical Knowledge Bases (Example)



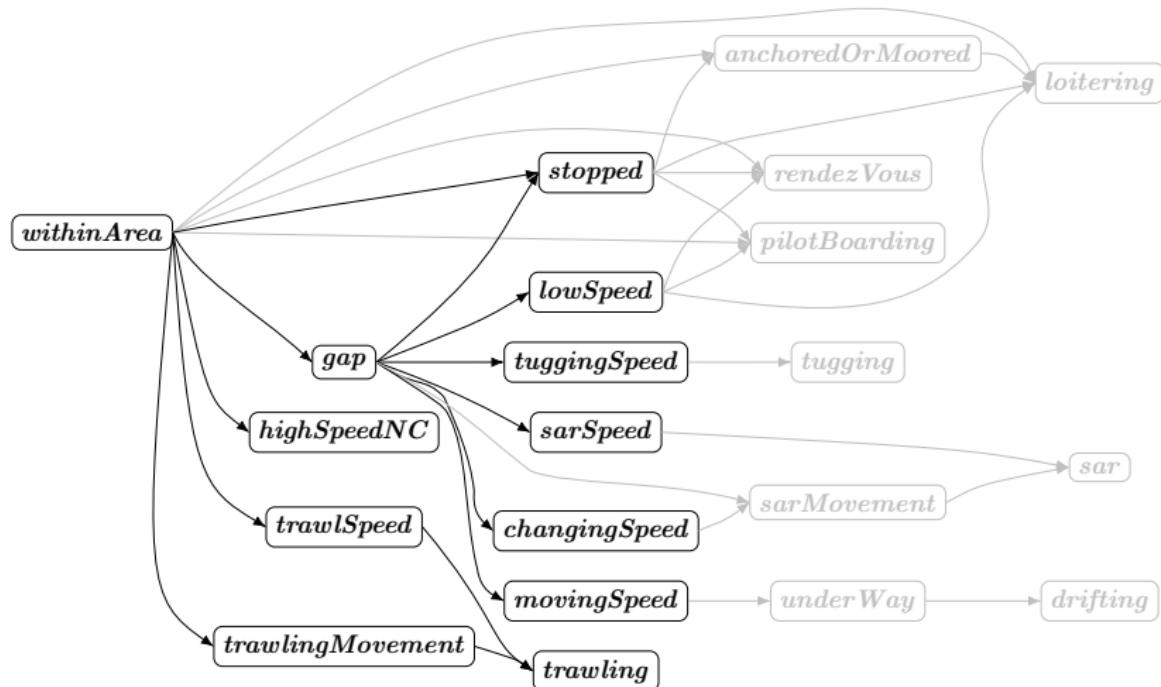
Hierarchical Knowledge Bases (Example)



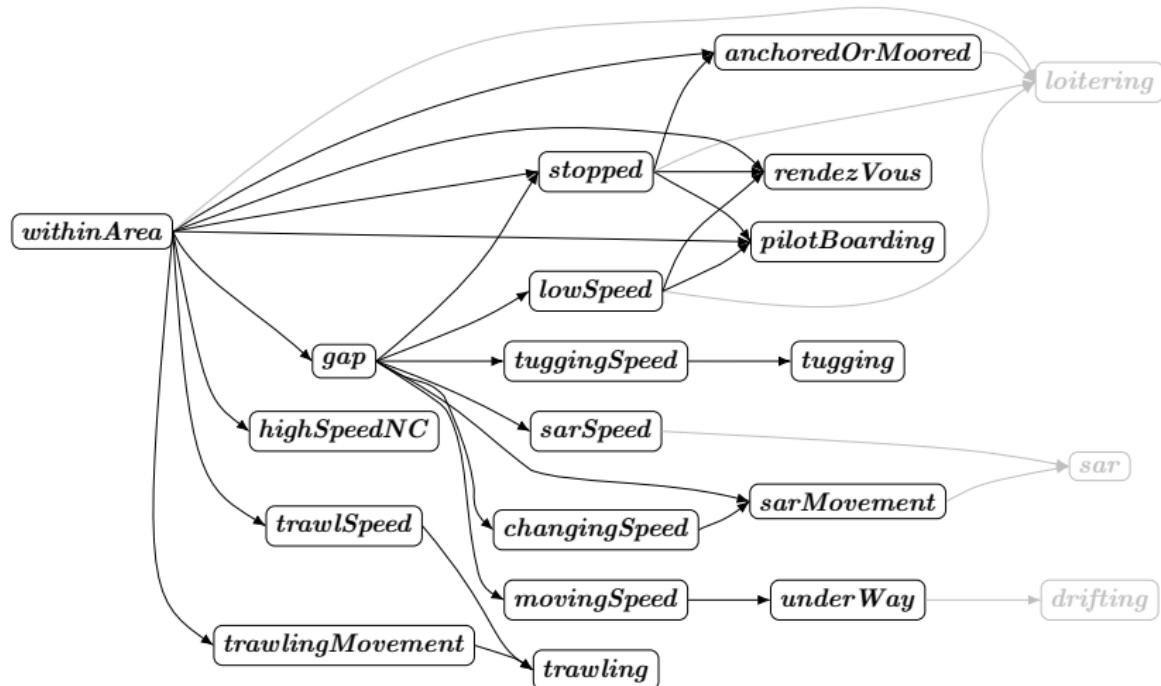
Hierarchical Knowledge Bases (Example)



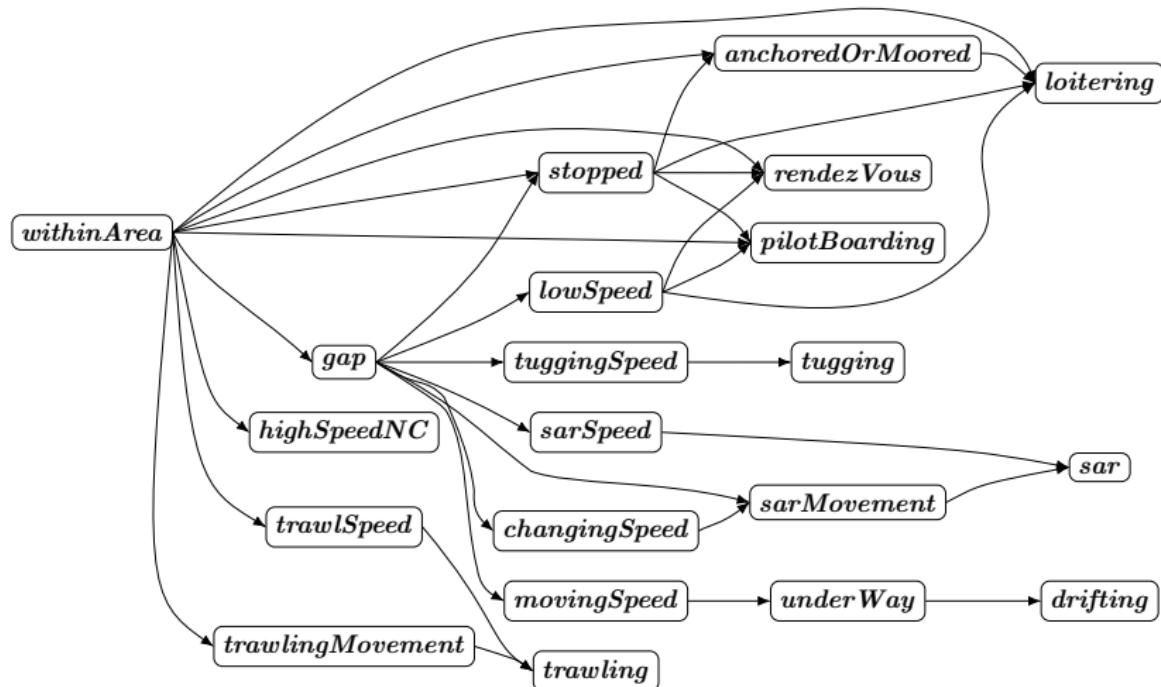
Hierarchical Knowledge Bases (Example)



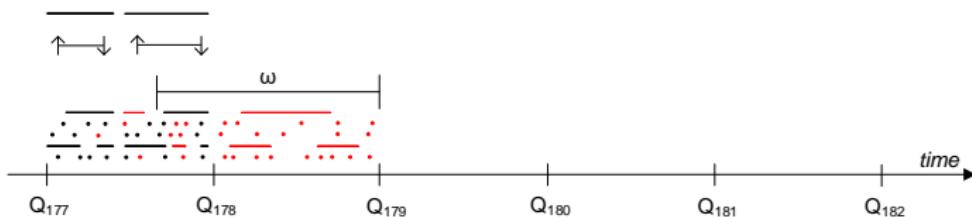
Hierarchical Knowledge Bases (Example)



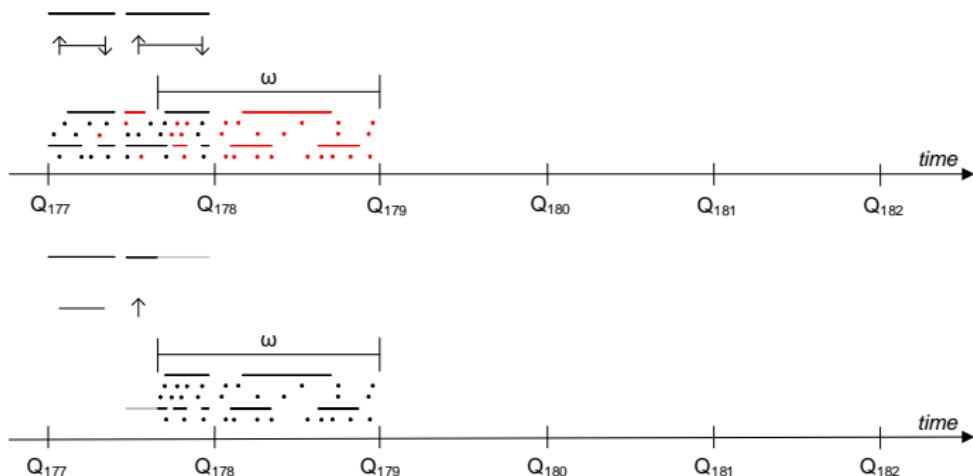
Hierarchical Knowledge Bases (Example)



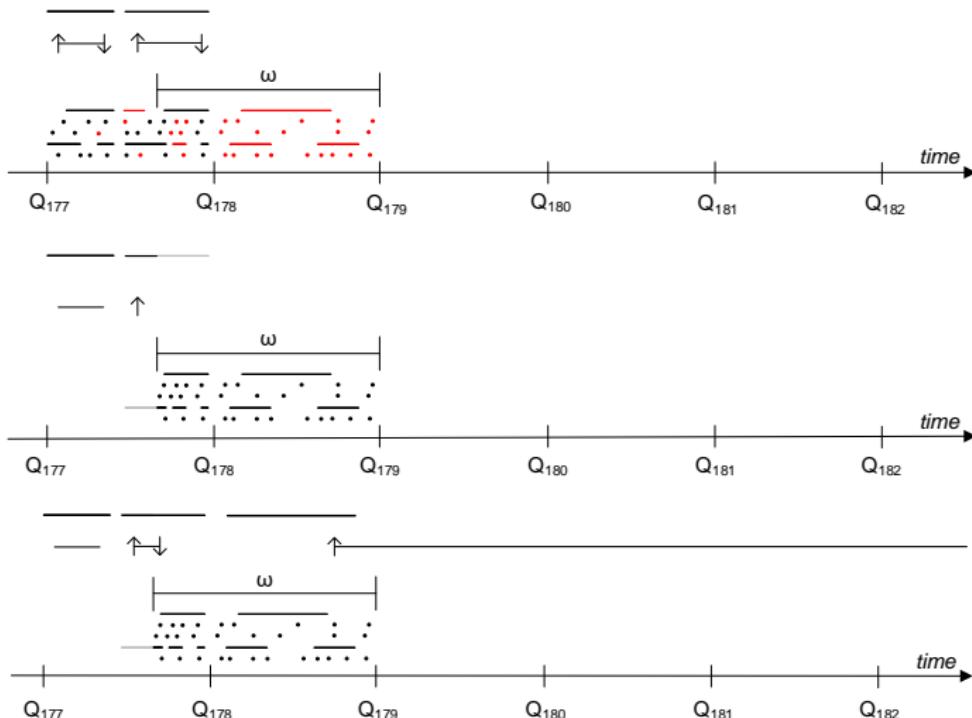
RTEC: Windowing



RTEC: Windowing



RTEC: Windowing



Deadlines

- Fluent-value pairs may be subject to deadlines
- E-commerce: an offer may be accepted at most by a specified time after being presented
- Maritime Situational Awareness: a fishing activity is terminated at a specified time after multiple changes in heading
- Multi-Agent Systems: agents may be suspended temporarily. Further violations may extend the period of suspension

Deadlines

initiatedAt($F = V_1, T$) \Rightarrow initiatedAt($F = V_2, T+R$)
if not brokenBetween($F = V_1, T, T+R$)

Deadlines

initiatedAt($F = V_1, T$) \Rightarrow initiatedAt($F = V_2, T+R$)
if not brokenBetween($F = V_1, T, T+R$)

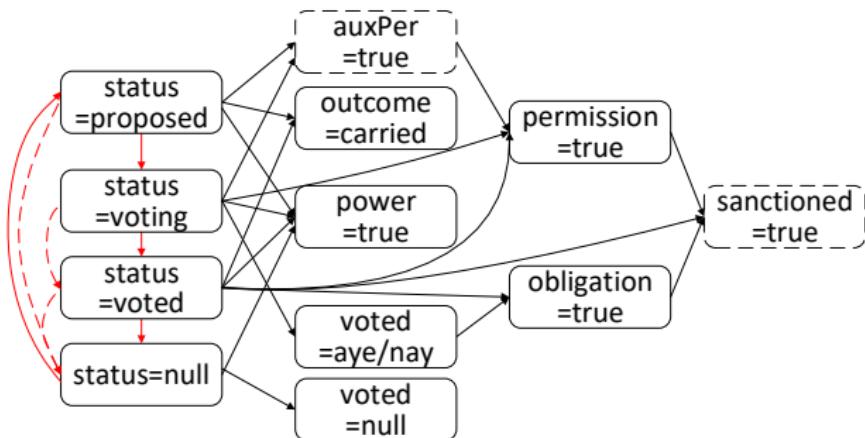
initiatedAt($F = V_1, T$) \rightsquigarrow initiatedAt($F = V_2, T+R$)
if not brokenBetween($F = V_1, T, T+R$) and
not initiatedAt($F = V_1, T'$), $T < T' < T+R$

Example: Voting³

- Agents propose, second and vote for motions
- Chairs close ballots and declare the outcome
- RTEC monitors the normative positions of agents, e.g.:
 - obligation
 - permission
 - institutionalised power
- Agents may be sanctioned for defying the rules of the game

³Pitt J., Kamara L., Sergot M. and Artikis A., Voting in multi-agent systems. Computer Journal 49(2), 156–170, 2006.

Voting: Hierarchical Knowledge Base



Voting: status fluent

```
initiatedAt(status(M) = proposed, T) ←  
    happensAt(propose(P, M), T),  
    holdsAt(status(M) = null, T).
```

Voting: status fluent

```
initiatedAt( $status(M) = proposed$ ,  $T$ ) ←  
    happensAt( $propose(P, M)$ ,  $T$ ),  
    holdsAt( $status(M) = null$ ,  $T$ ).  
  
initiatedAt( $status(M) = voting$ ,  $T$ ) ←  
    happensAt( $second(S, M)$ ,  $T$ ),  
    holdsAt( $status(M) = proposed$ ,  $T$ ).
```

Voting: status fluent

```
initiatedAt( $status(M) = proposed$ ,  $T$ ) ←  
    happensAt( $propose(P, M)$ ,  $T$ ),  
    holdsAt( $status(M) = null$ ,  $T$ ).  
  
initiatedAt( $status(M) = voting$ ,  $T$ ) ←  
    happensAt( $second(S, M)$ ,  $T$ ),  
    holdsAt( $status(M) = proposed$ ,  $T$ ).  
  
initiatedAt( $status(M) = voted$ ,  $T$ ) ←  
    happensAt( $close\_ballot(C, M)$ ,  $T$ ),  
    holdsAt( $status(M) = voting$ ,  $T$ ).
```

Voting: status fluent

initiatedAt($status(M) = proposed$, T) \leftarrow
 happensAt($propose(P, M)$, T),
 holdsAt($status(M) = null$, T).

initiatedAt($status(M) = voting$, T) \leftarrow
 happensAt($second(S, M)$, T),
 holdsAt($status(M) = proposed$, T).

initiatedAt($status(M) = voted$, T) \leftarrow
 happensAt($close_ballot(C, M)$, T),
 holdsAt($status(M) = voting$, T).

initiatedAt($status(M) = null$, T) \leftarrow
 happensAt($declare(C, M, Res)$, T),
 holdsAt($status(M) = voted$, T).

Voting: Deadlines

$\text{initiatedAt}(\text{status}(M) = \text{voting}, T) \implies$
 $\text{initiatedAt}(\text{status}(M) = \text{voted}, T + r_{\text{voting}})$

$\text{initiatedAt}(\text{status}(M) = \text{proposed}, T) \implies$
 $\text{initiatedAt}(\text{status}(M) = \text{null}, T + r_{\text{proposed}})$

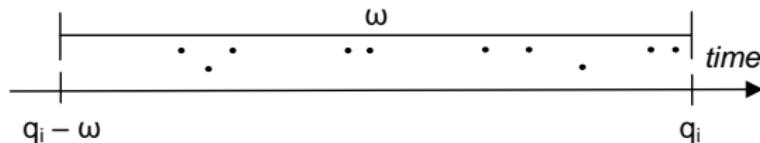
$\text{initiatedAt}(\text{status}(M) = \text{voted}, T) \implies$
 $\text{initiatedAt}(\text{status}(M) = \text{null}, T + r_{\text{voted}})$

Voting: Deadlines

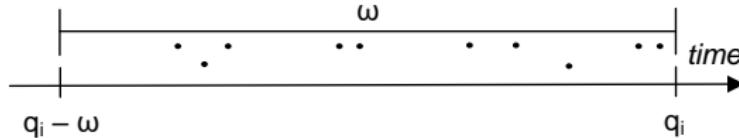
$\text{initiatedAt}(\textit{sanctioned}(C) = \text{true}, T) \rightsquigarrow$
 $\text{initiatedAt}(\textit{sanctioned}(C) = \text{false}, T + r_{\textit{sanctioned}})$

Handling Deadlines

non-extensible

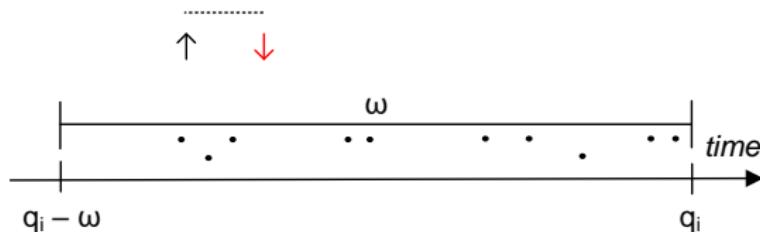


extensible

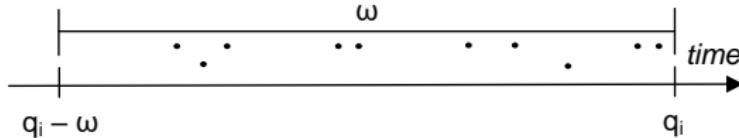


Handling Deadlines

non-extensible

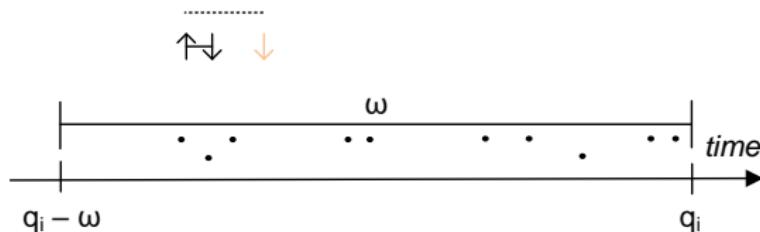


extensible

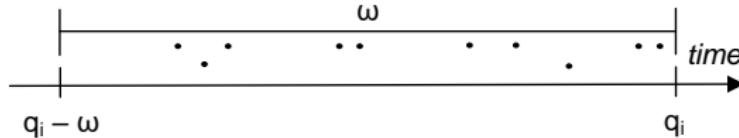


Handling Deadlines

non-extensible

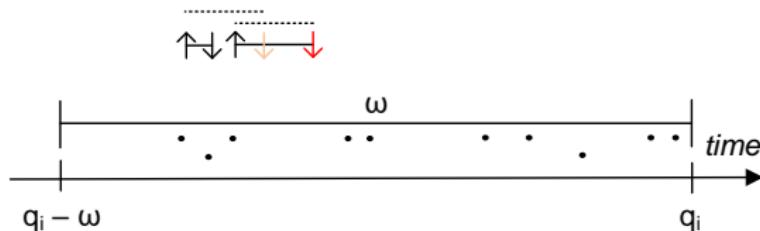


extensible

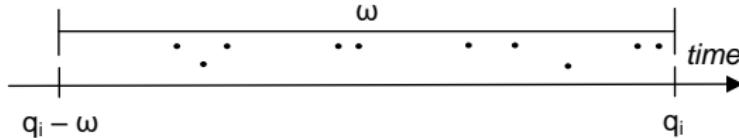


Handling Deadlines

non-extensible

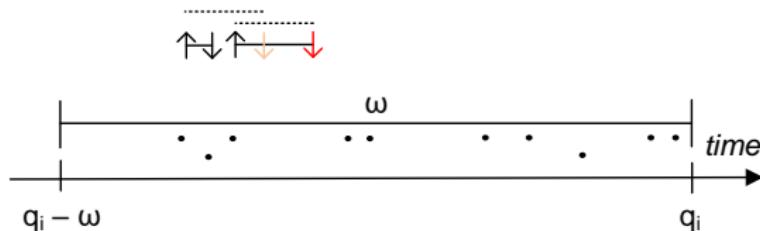


extensible

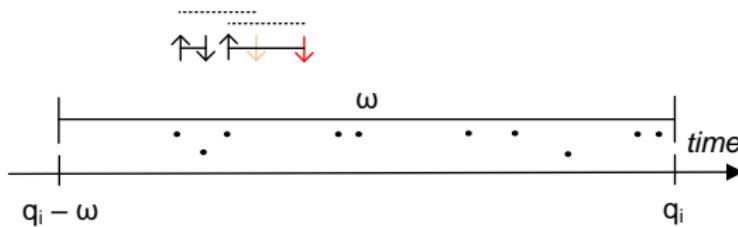


Handling Deadlines

non-extensible

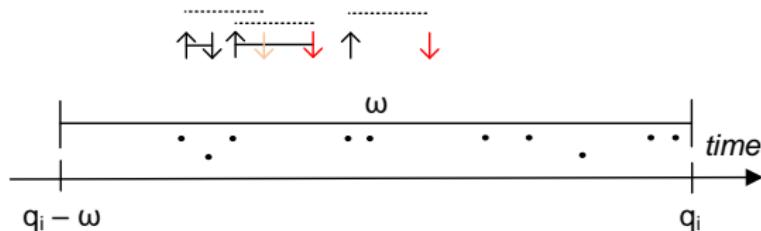


extensible

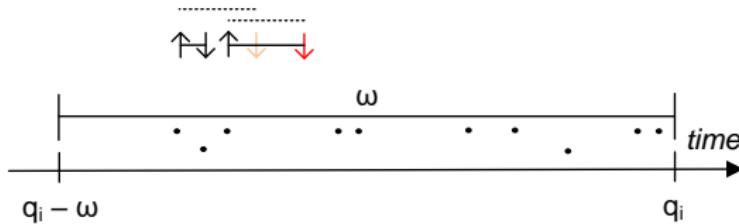


Handling Deadlines

non-extensible

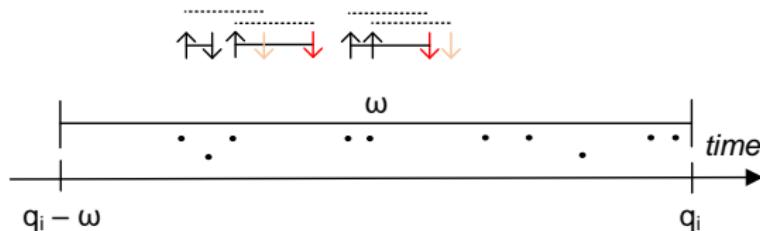


extensible

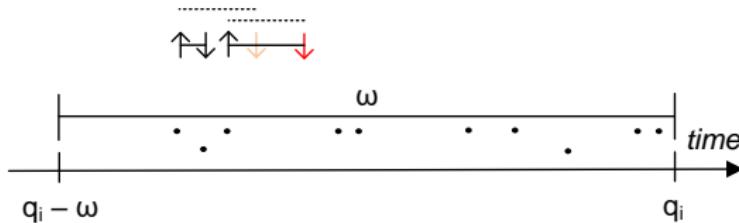


Handling Deadlines

non-extensible

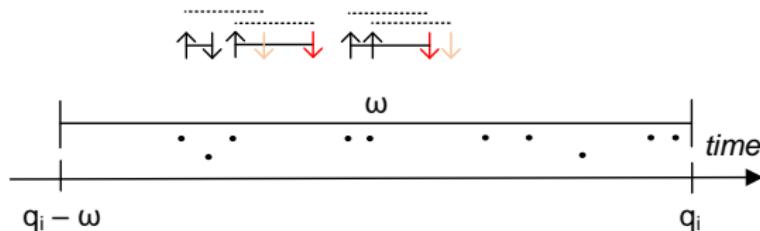


extensible

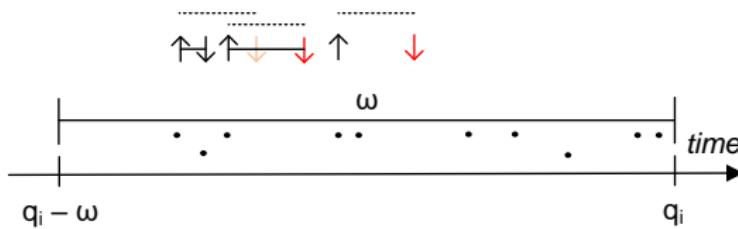


Handling Deadlines

non-extensible

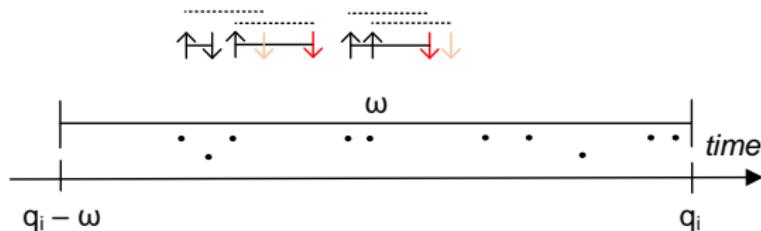


extensible

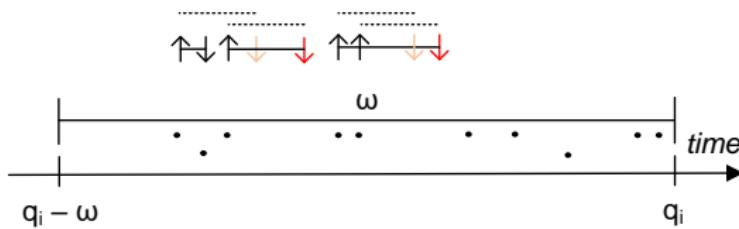


Handling Deadlines

non-extensible

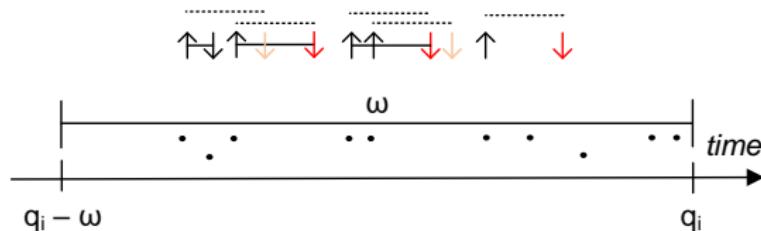


extensible

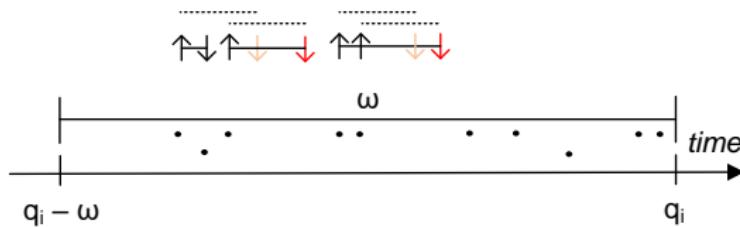


Handling Deadlines

non-extensible

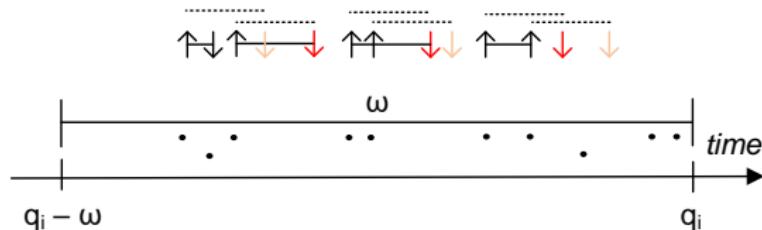


extensible

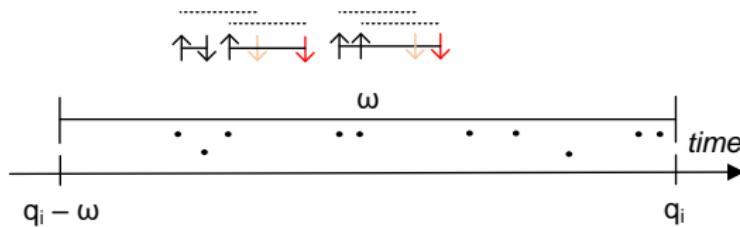


Handling Deadlines

non-extensible

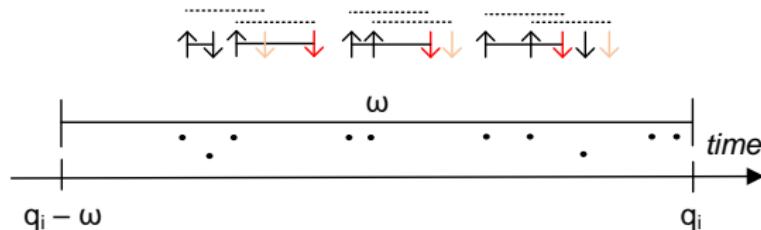


extensible

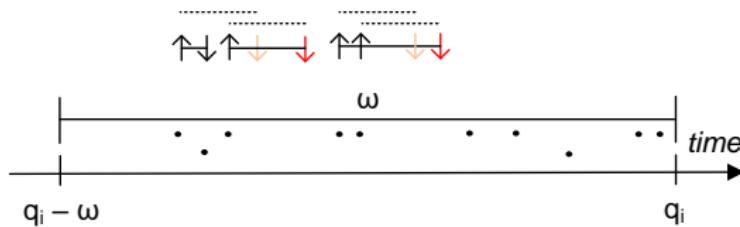


Handling Deadlines

non-extensible

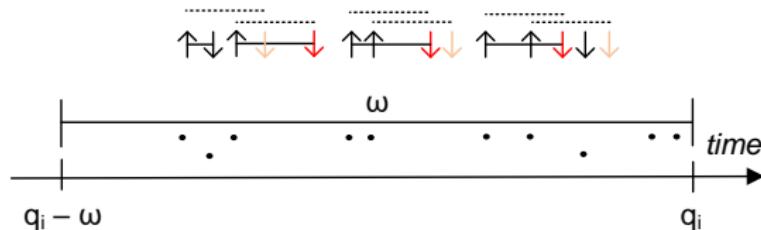


extensible

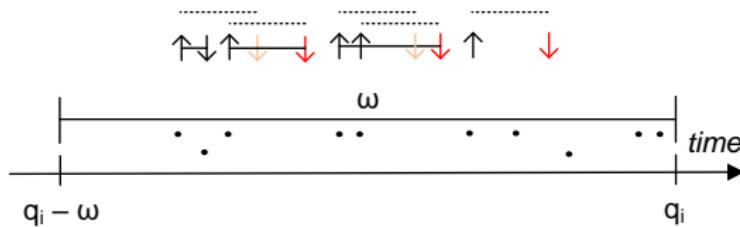


Handling Deadlines

non-extensible

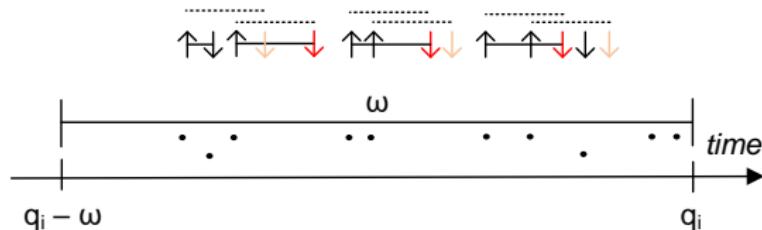


extensible

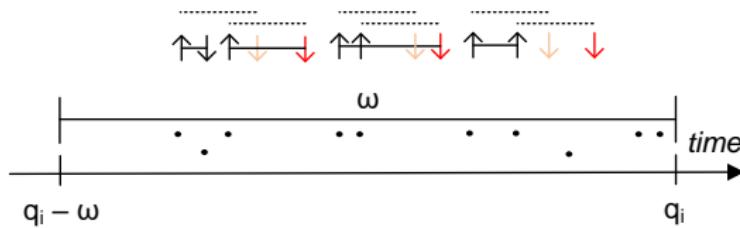


Handling Deadlines

non-extensible

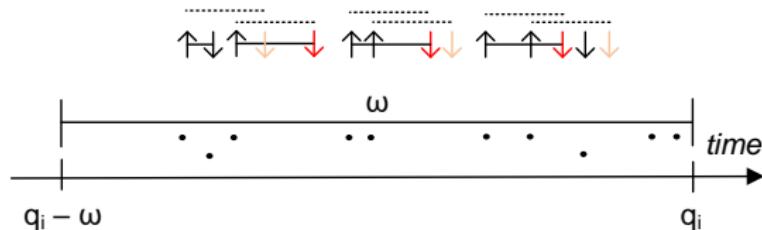


extensible

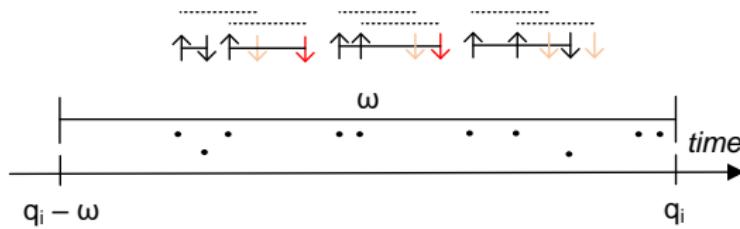


Handling Deadlines

non-extensible

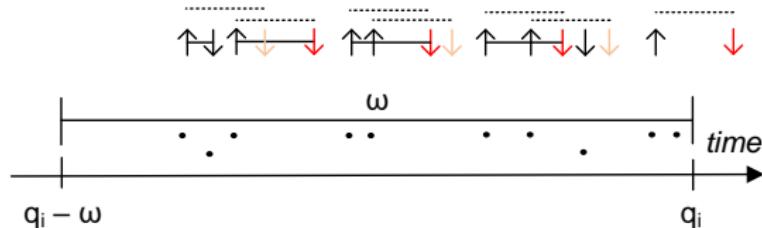


extensible

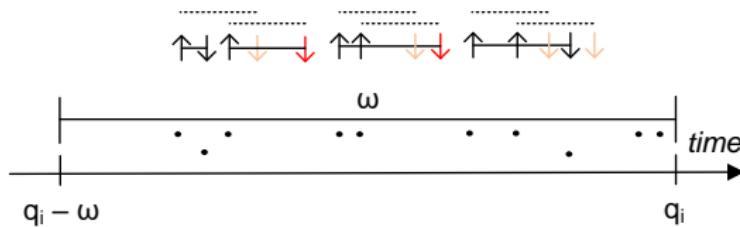


Handling Deadlines

non-extensible

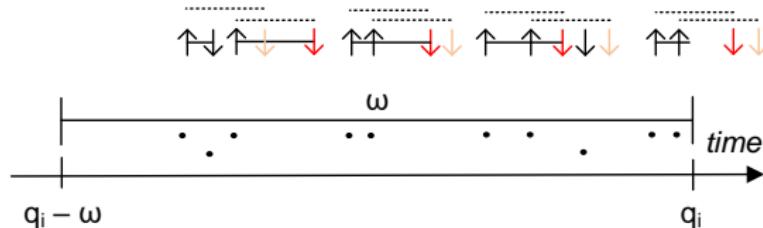


extensible

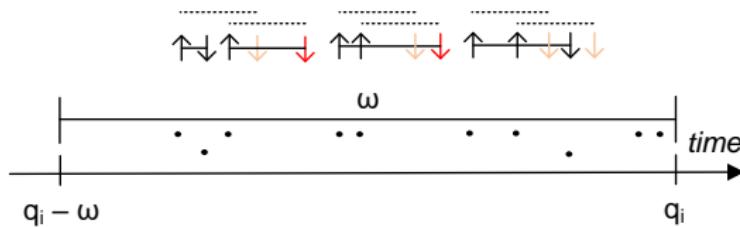


Handling Deadlines

non-extensible

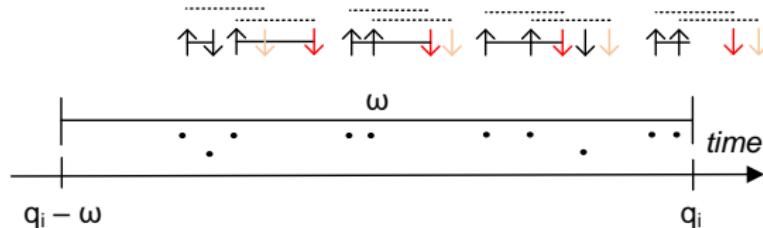


extensible

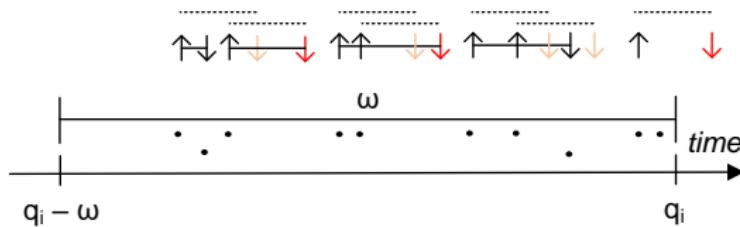


Handling Deadlines

non-extensible

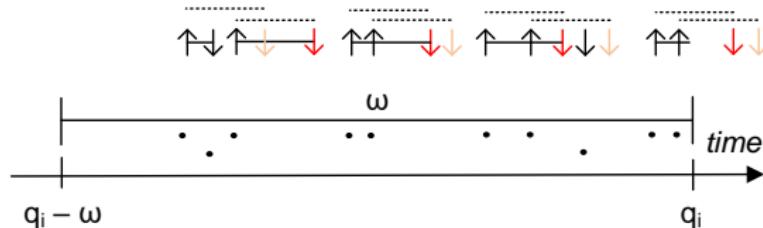


extensible

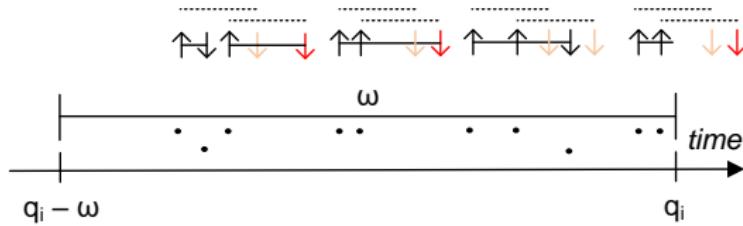


Handling Deadlines

non-extensible

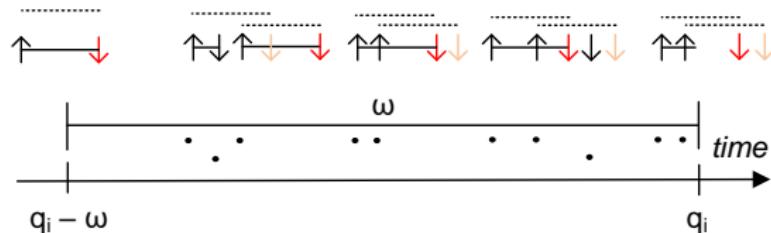


extensible

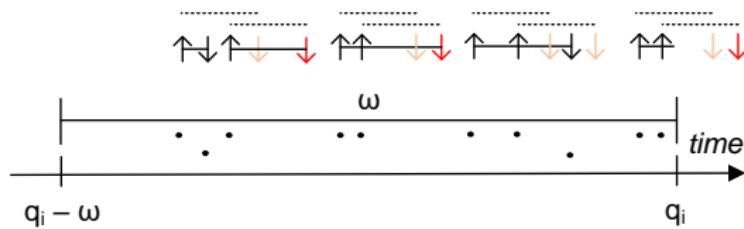


Handling Deadlines

non-extensible

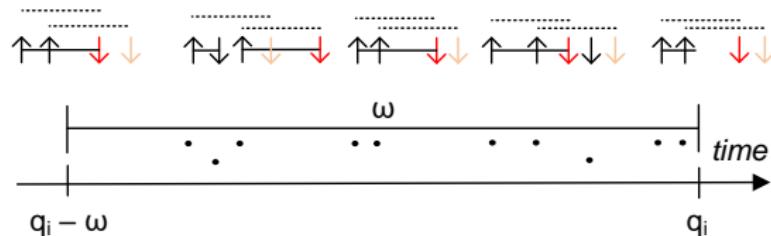


extensible

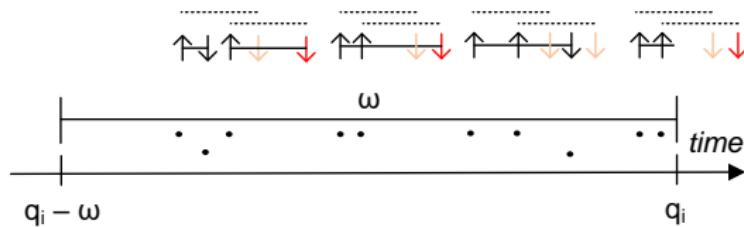


Handling Deadlines

non-extensible

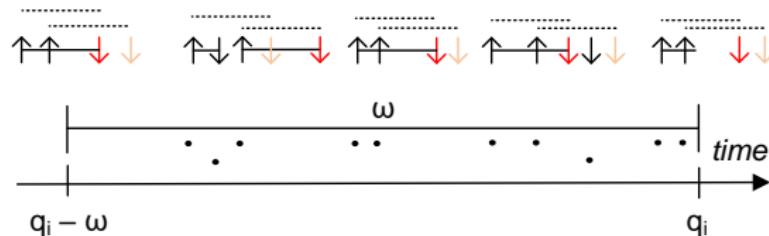


extensible

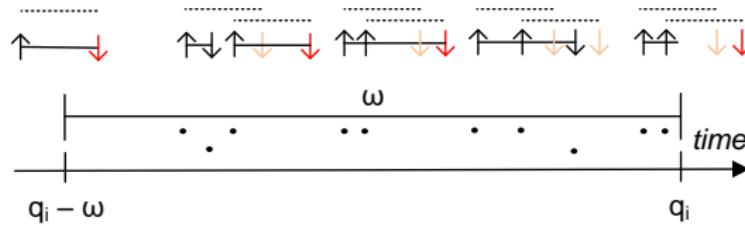


Handling Deadlines

non-extensible

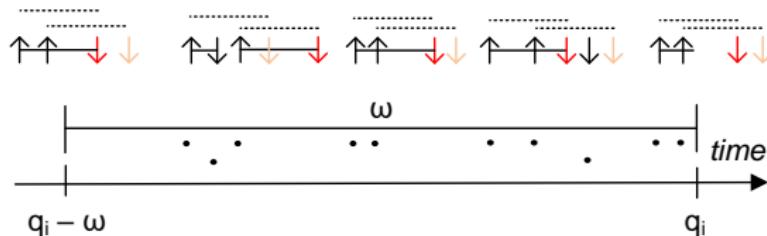


extensible

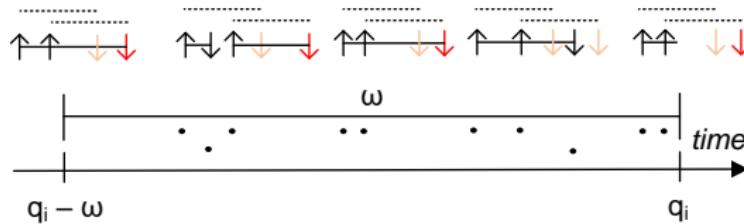


Handling Deadlines

non-extensible

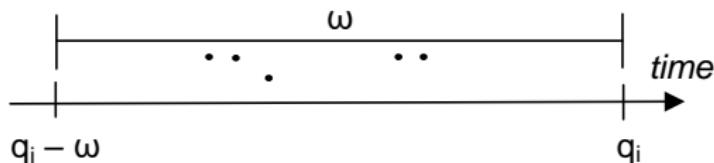


extensible

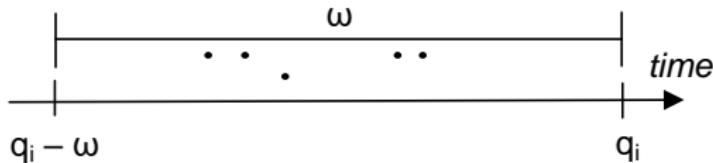


Handling Deadlines: Caching

With Caching (RTEC)

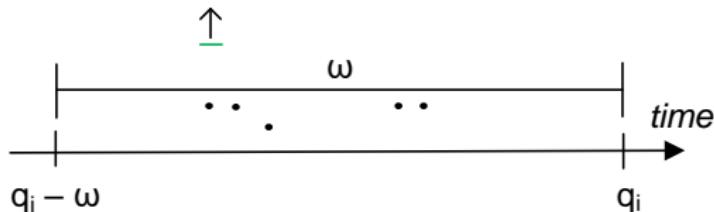


Without Caching

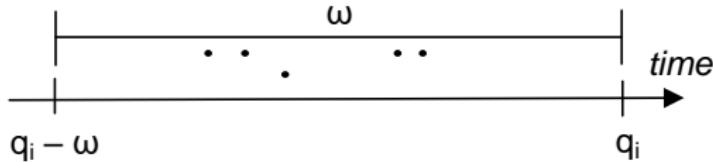


Handling Deadlines: Caching

With Caching (RTEC)

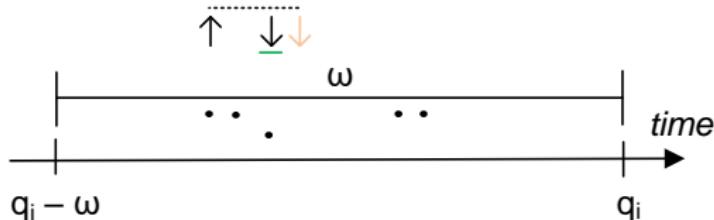


Without Caching

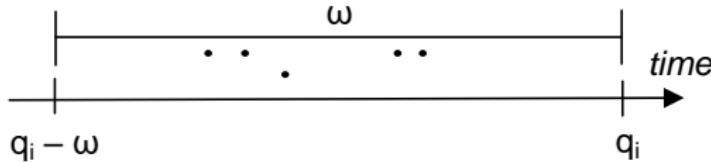


Handling Deadlines: Caching

With Caching (RTEC)

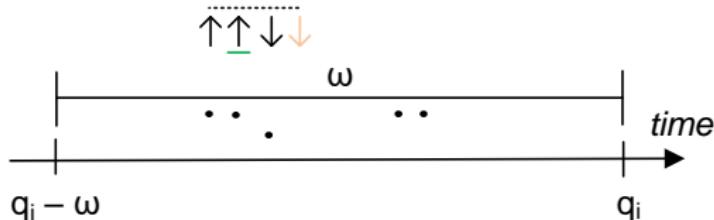


Without Caching

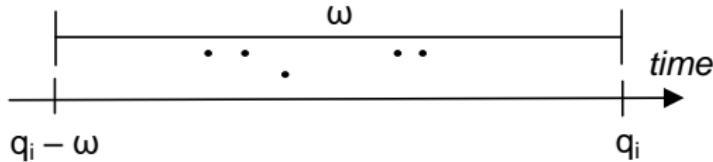


Handling Deadlines: Caching

With Caching (RTEC)

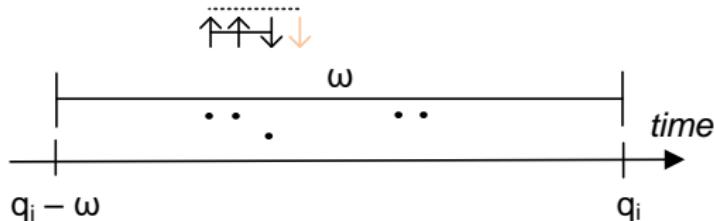


Without Caching

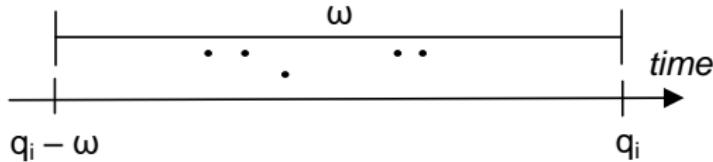


Handling Deadlines: Caching

With Caching (RTEC)

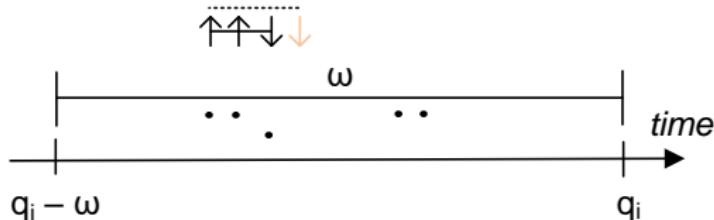


Without Caching

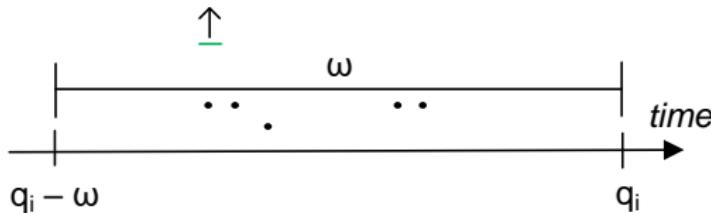


Handling Deadlines: Caching

With Caching (RTEC)

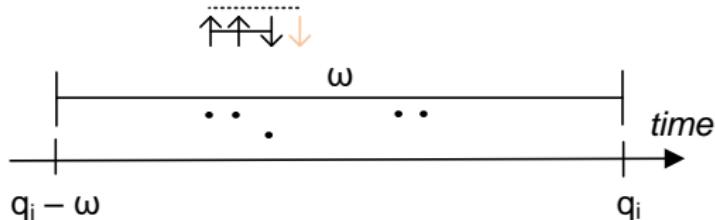


Without Caching

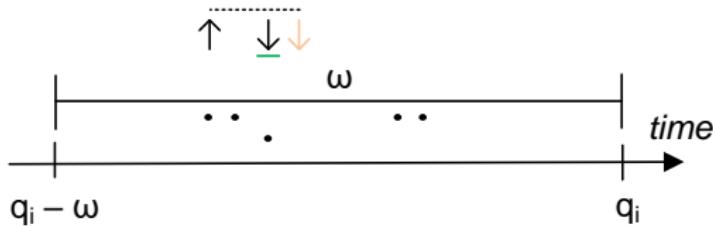


Handling Deadlines: Caching

With Caching (RTEC)

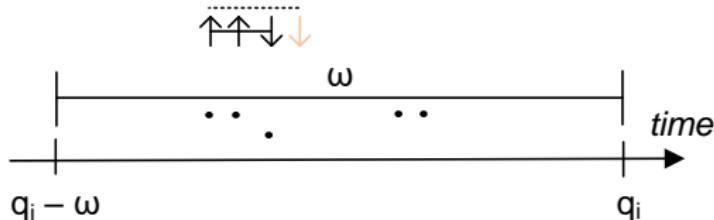


Without Caching

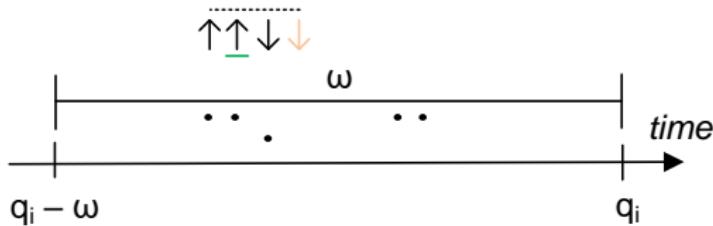


Handling Deadlines: Caching

With Caching (RTEC)

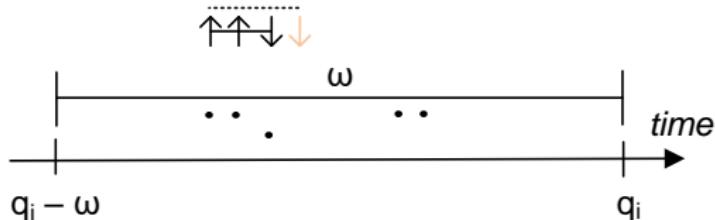


Without Caching

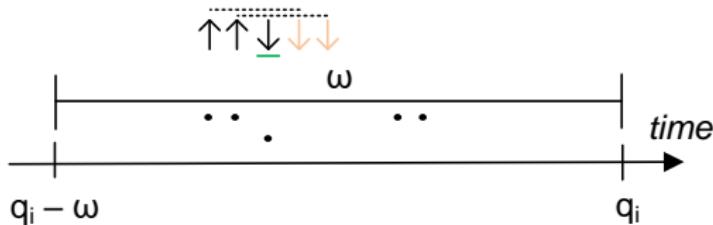


Handling Deadlines: Caching

With Caching (RTEC)

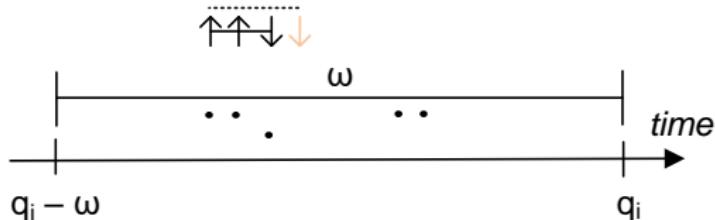


Without Caching

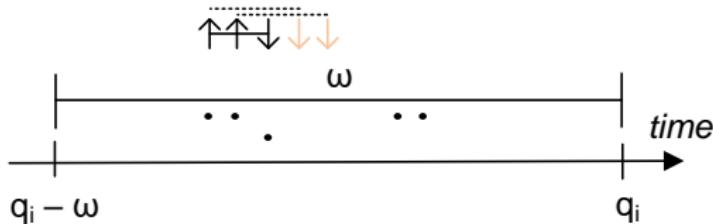


Handling Deadlines: Caching

With Caching (RTEC)

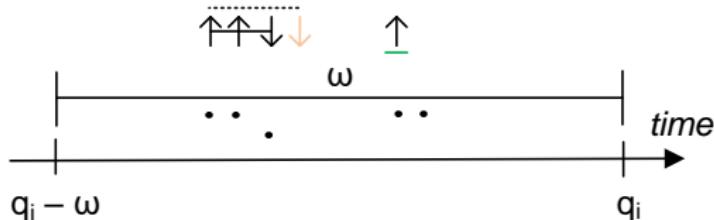


Without Caching

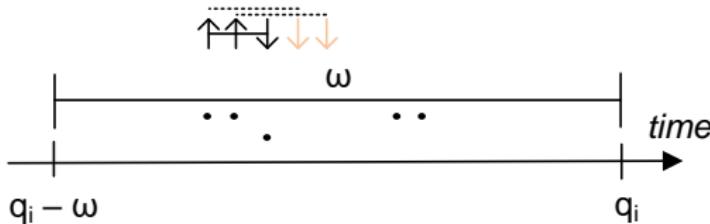


Handling Deadlines: Caching

With Caching (RTEC)

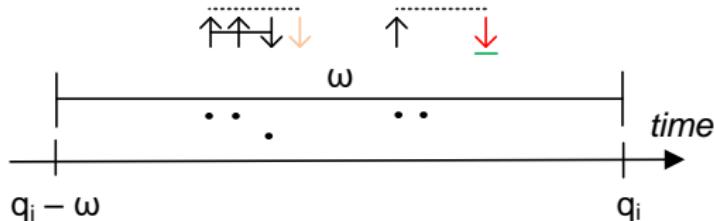


Without Caching

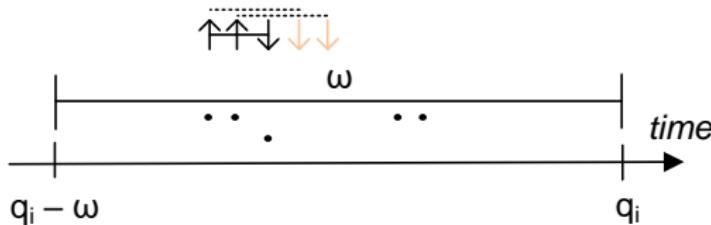


Handling Deadlines: Caching

With Caching (RTEC)

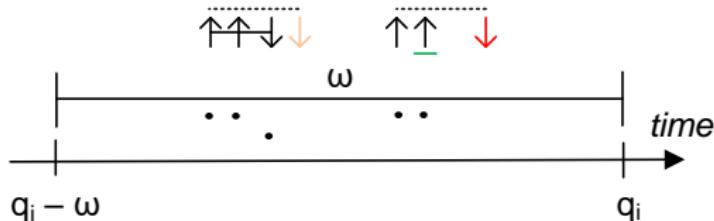


Without Caching

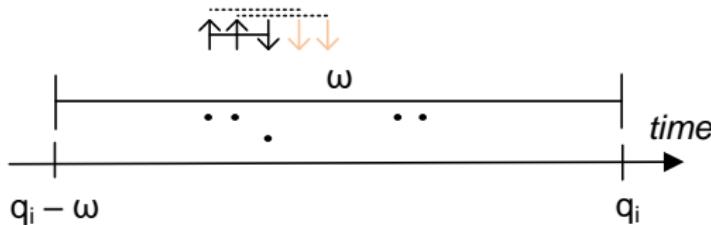


Handling Deadlines: Caching

With Caching (RTEC)

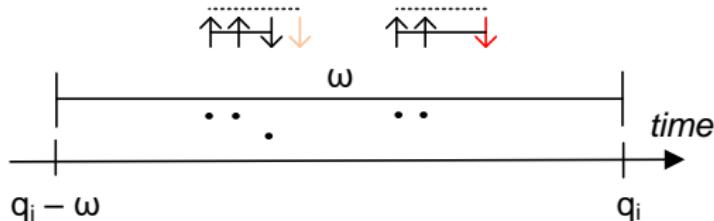


Without Caching

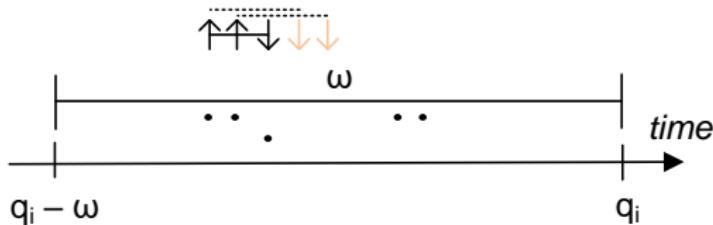


Handling Deadlines: Caching

With Caching (RTEC)

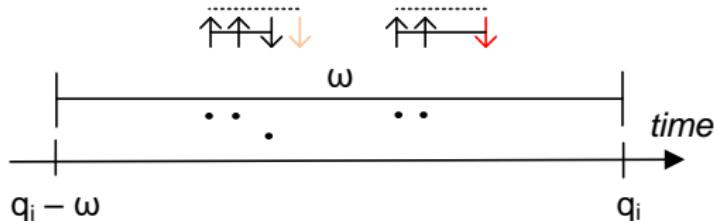


Without Caching

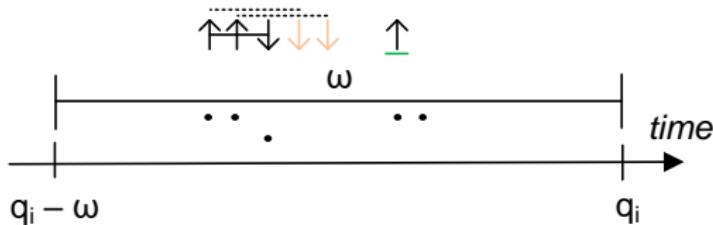


Handling Deadlines: Caching

With Caching (RTEC)

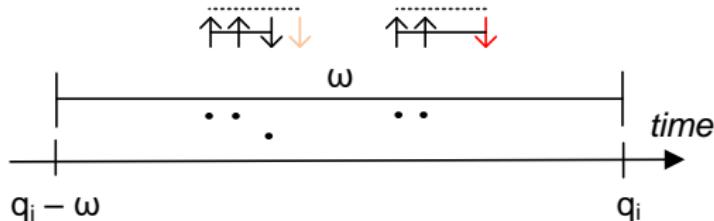


Without Caching

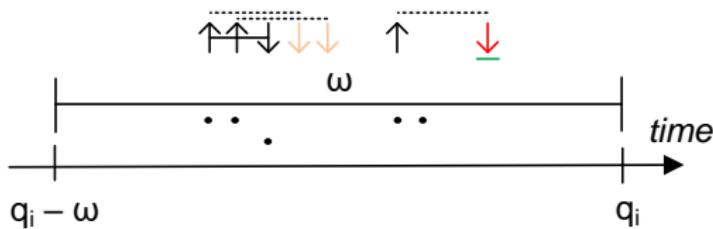


Handling Deadlines: Caching

With Caching (RTEC)

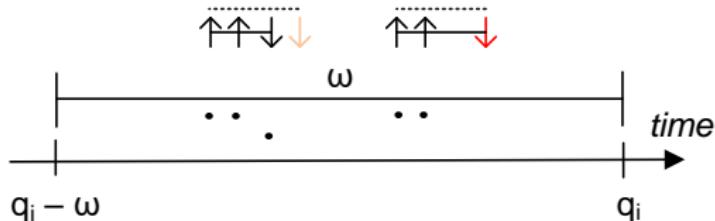


Without Caching

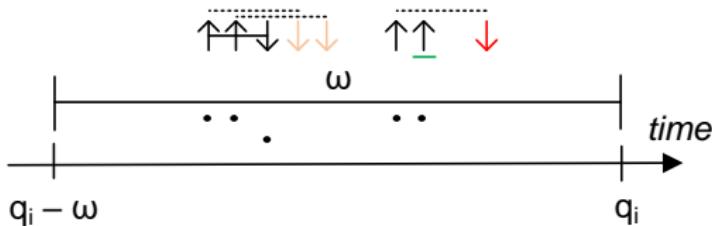


Handling Deadlines: Caching

With Caching (RTEC)

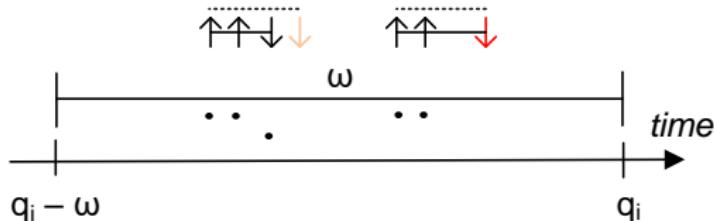


Without Caching

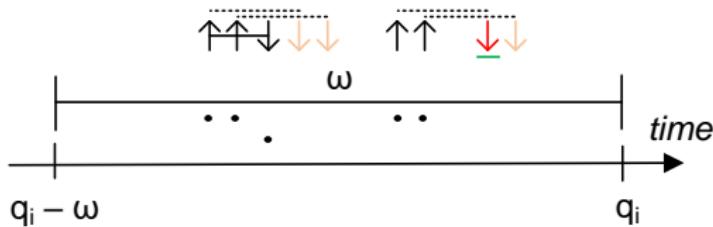


Handling Deadlines: Caching

With Caching (RTEC)

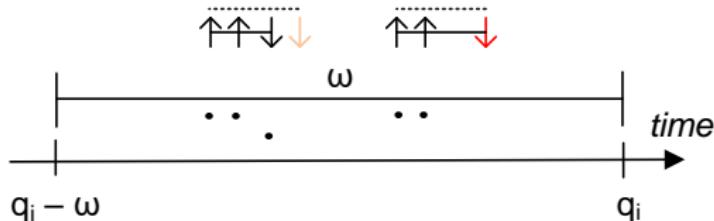


Without Caching

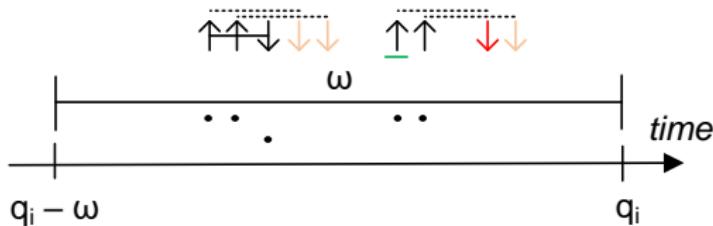


Handling Deadlines: Caching

With Caching (RTEC)

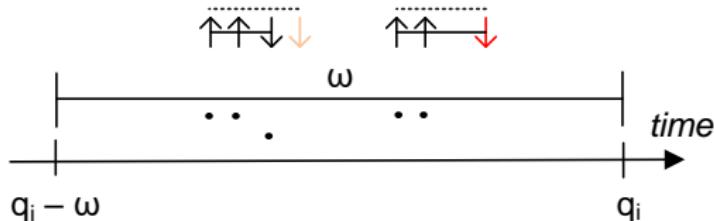


Without Caching

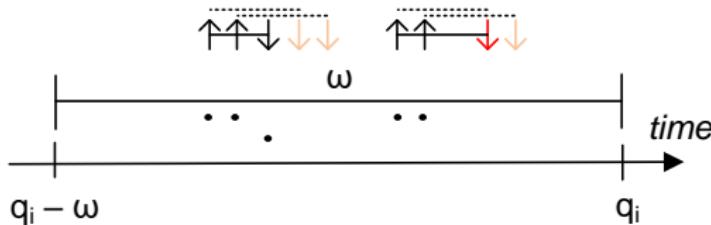


Handling Deadlines: Caching

With Caching (RTEC)

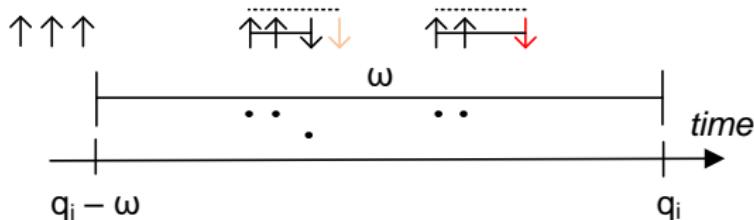


Without Caching

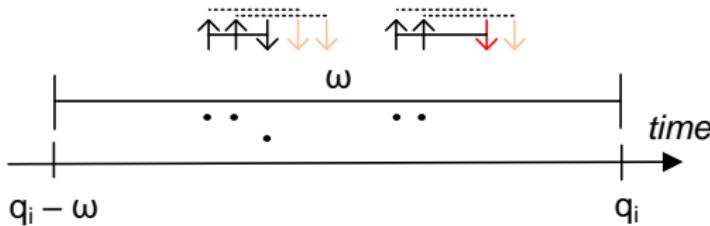


Handling Deadlines: Caching

With Caching (RTEC)

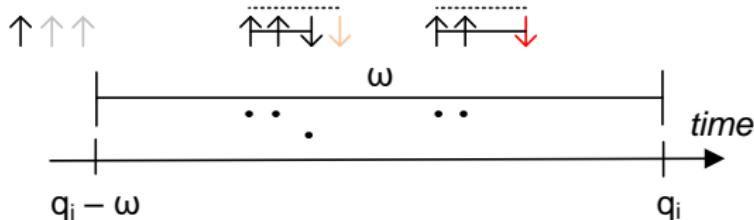


Without Caching

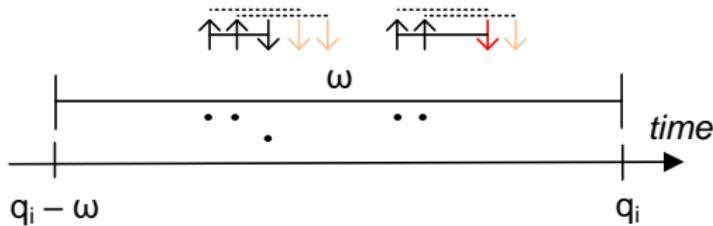


Handling Deadlines: Caching

With Caching (RTEC)

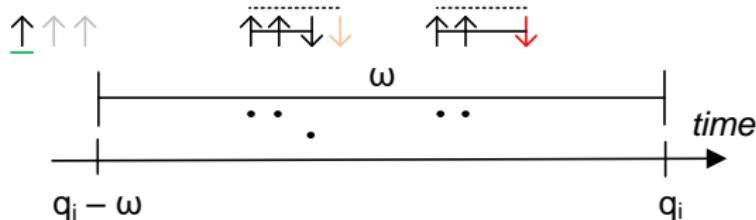


Without Caching

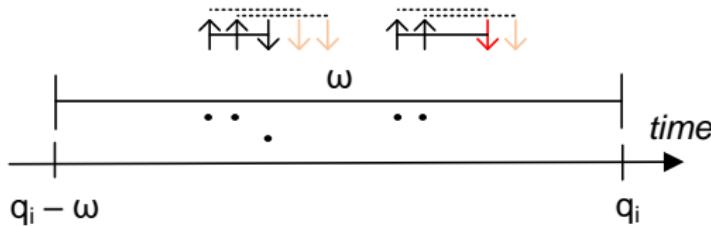


Handling Deadlines: Caching

With Caching (RTEC)

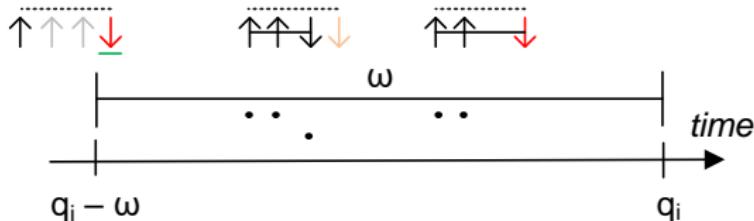


Without Caching

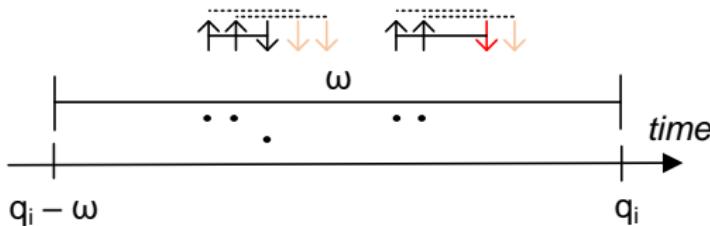


Handling Deadlines: Caching

With Caching (RTEC)

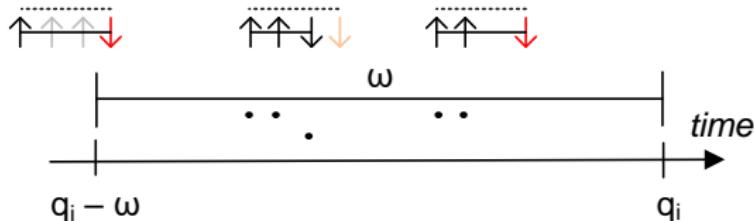


Without Caching

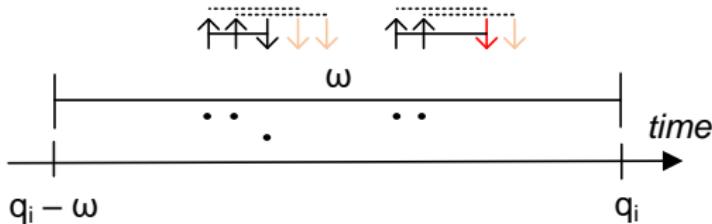


Handling Deadlines: Caching

With Caching (RTEC)

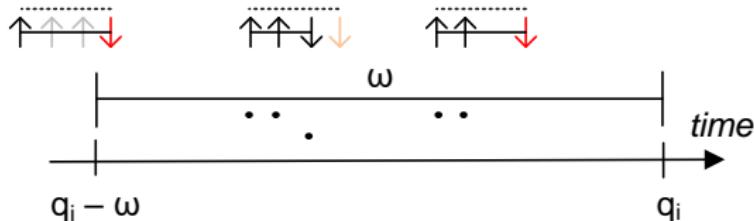


Without Caching

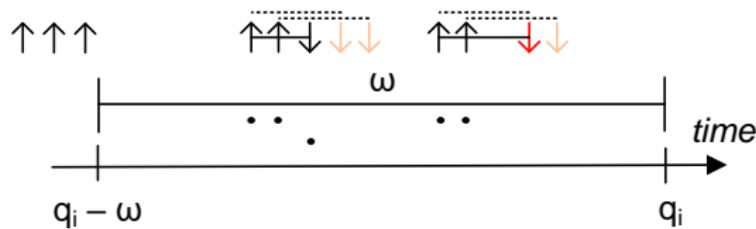


Handling Deadlines: Caching

With Caching (RTEC)

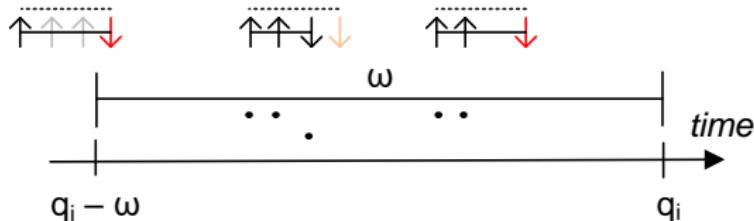


Without Caching

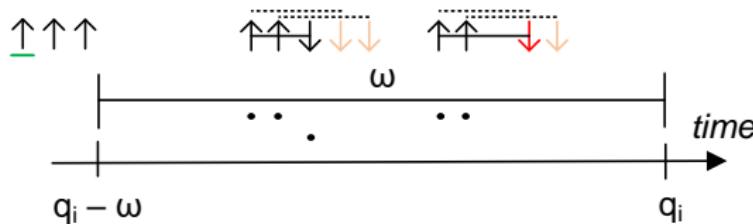


Handling Deadlines: Caching

With Caching (RTEC)

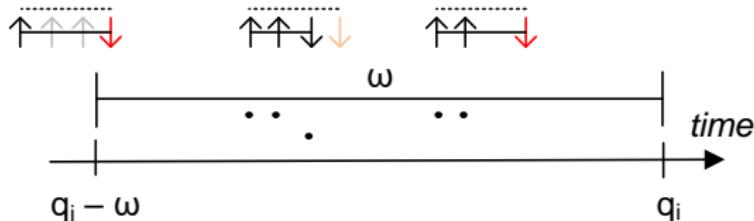


Without Caching

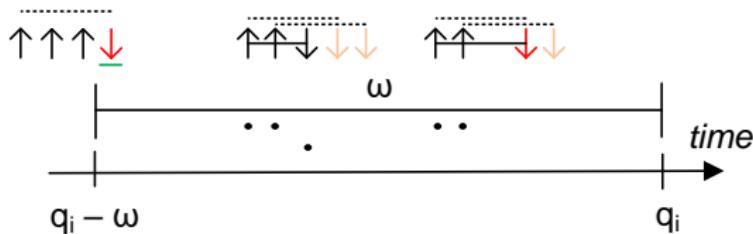


Handling Deadlines: Caching

With Caching (RTEC)

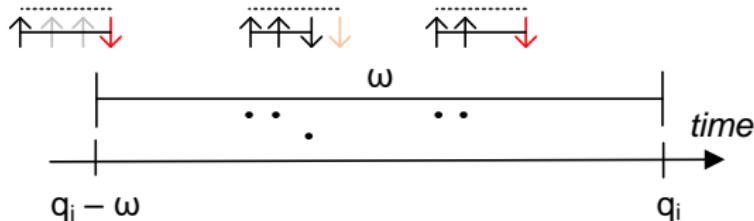


Without Caching

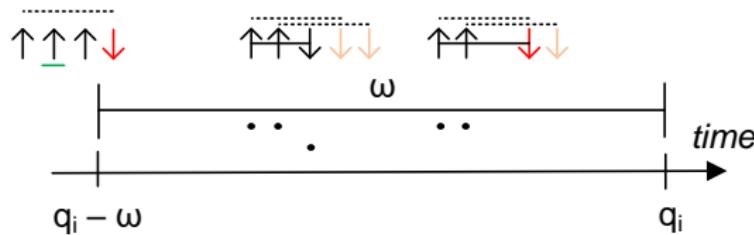


Handling Deadlines: Caching

With Caching (RTEC)

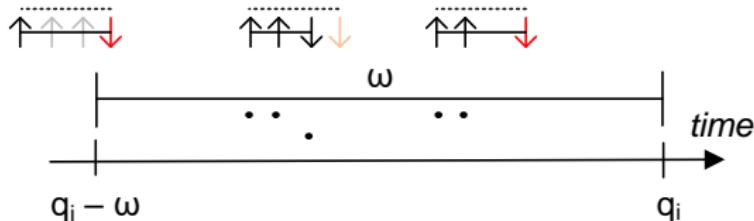


Without Caching

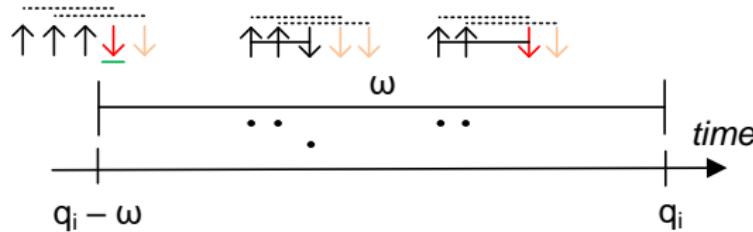


Handling Deadlines: Caching

With Caching (RTEC)

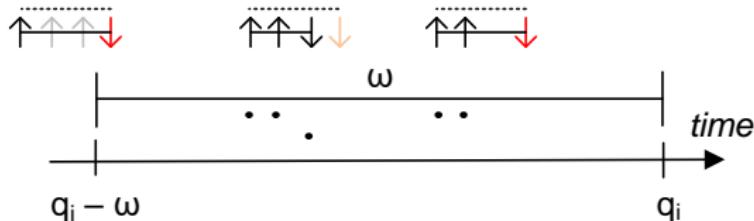


Without Caching

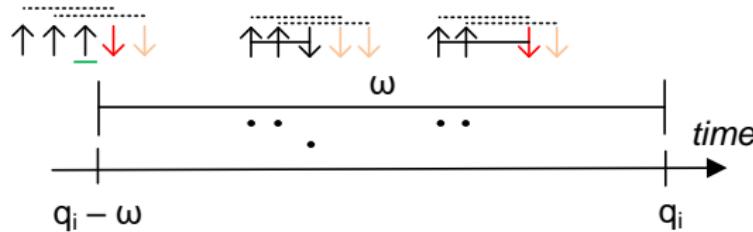


Handling Deadlines: Caching

With Caching (RTEC)

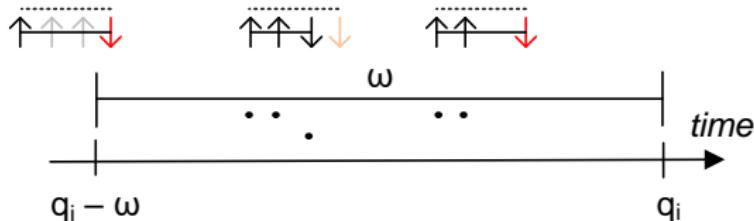


Without Caching

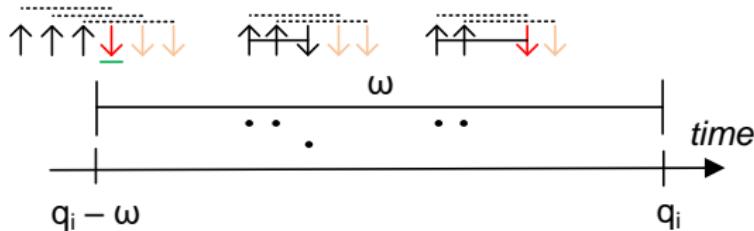


Handling Deadlines: Caching

With Caching (RTEC)

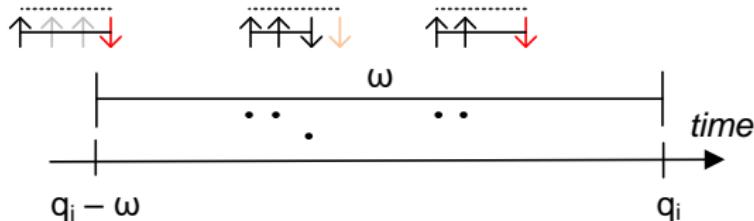


Without Caching

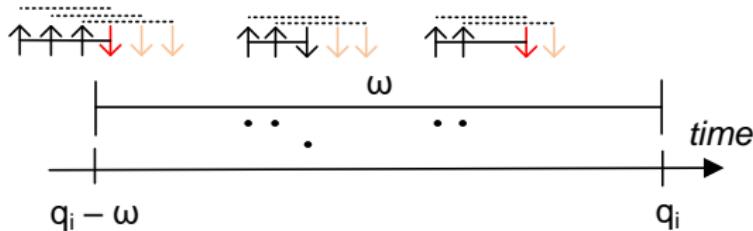


Handling Deadlines: Caching

With Caching (RTEC)



Without Caching



Handling Deadlines: Complexity

- Without Caching:

$$\mathcal{O}(vkm_{\omega}^2)$$

- With Caching (RTEC):

$$\mathcal{O}(rvk(m_{\omega}-r))$$

where

- m_{ω} : window size
- r : temporal distance between initiation and deadline
- v : number of fluent values
- k : cost of computing an initiation/termination point

Open-Source Code

Screenshot of a GitHub repository page for 'aartikis / RTEC'.

The top navigation bar includes links for Why GitHub?, Team, Enterprise, Explore, Marketplace, Pricing, Search, Sign in, and Sign up.

The repository header shows 'aartikis / RTEC' (Public), Notifications (6), Star (50), Fork (6), and Insights.

Code navigation tabs include Code (selected), Issues, Pull requests, Actions, Projects, Wiki, Security, and Insights.

Code status indicators: master (green), 3 branches (green), 0 tags (grey).

Code dropdown menu: Go to file, Code ->.

Commit list:

File	Message	Date
docs	added yap installation instructions	7 days ago
examples	Update README.md	3 days ago
execution scripts	updated cli script and file structure	3 days ago
src	minor	15 days ago
.gitignore	updated structure and readme files	3 days ago
LICENSE	updated RTEC manual	9 months ago
README.md	Update README.md	3 days ago
RTEC_manual.pdf	minor in manual	7 months ago
setup.py	added command line interface	7 days ago

File list:

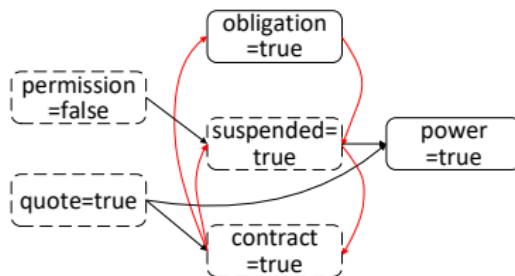
- README.md

Bottom of the page displays the RTEC logo and the tagline "Event Calculus".

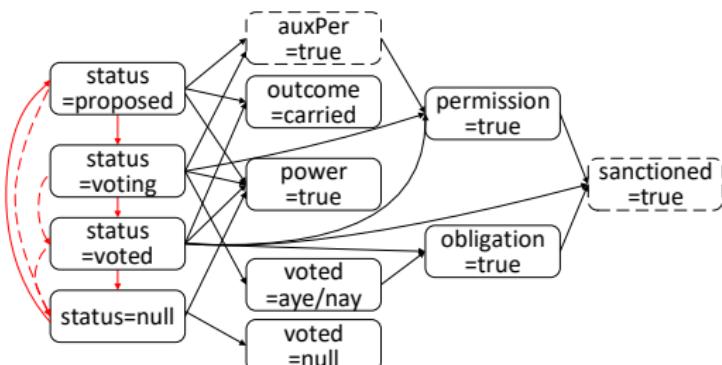
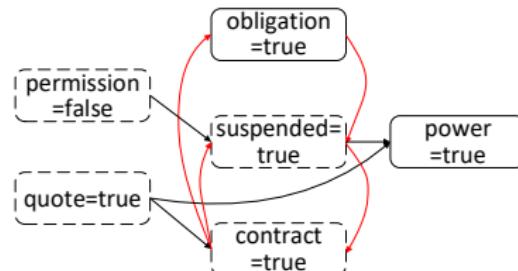
Right sidebar:

- About: RTEC is an Event Calculus implementation optimised for stream reasoning.
- Tags: data-science, cep, prolog, artificial-intelligence, stream-processing, logic-programming, complex-event-processing, multi-agent-systems, data-stream-processing, event-calculus, stream-reasoning, complex-event-recognition.
- Readme
- LGPL-3.0 License
- Releases: No releases published

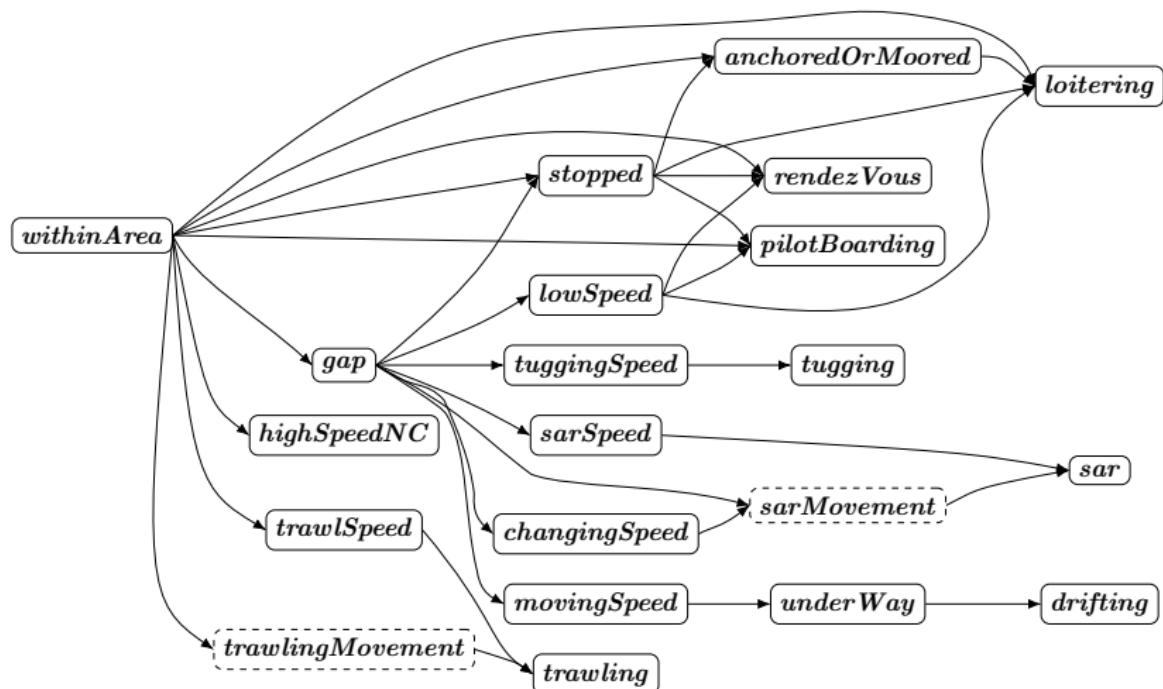
NetBill & Voting



NetBill & Voting



Maritime Situational Awareness

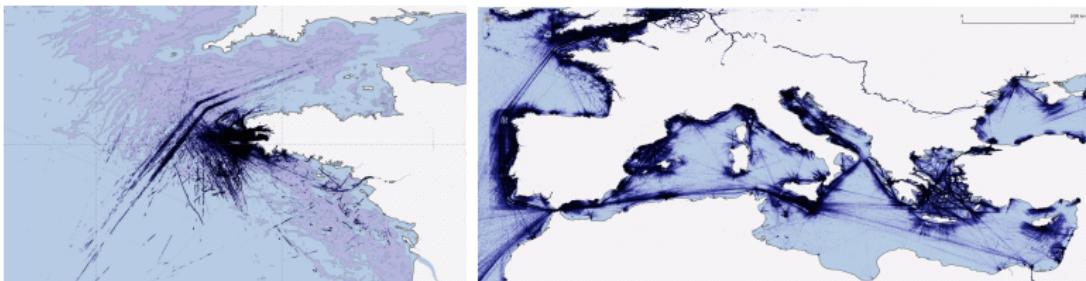


Experimental Evaluation: Datasets

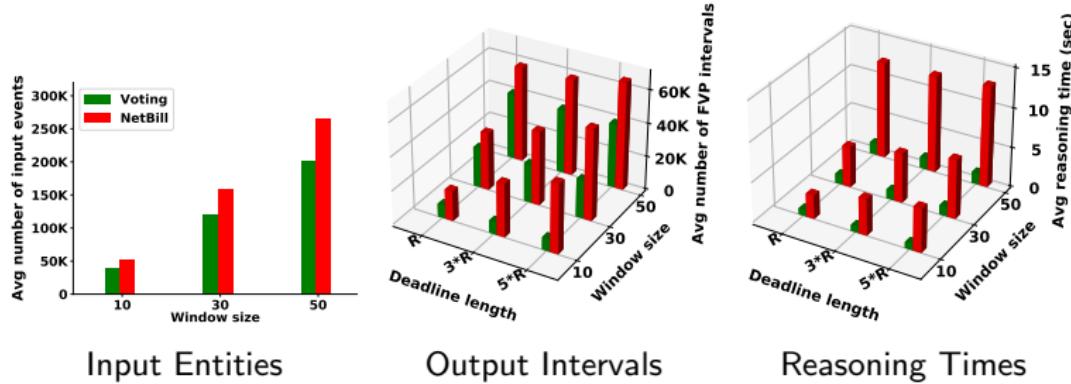
- Synthetic data streams for Multi-Agent Systems:
 - Voting → 400K events for 4000 agents
 - Netbill → 530K events for 4000 agents
- Task:
 - Computation of normative positions (e.g. permission, obligation, etc.) at run-time.

Experimental Evaluation: Datasets

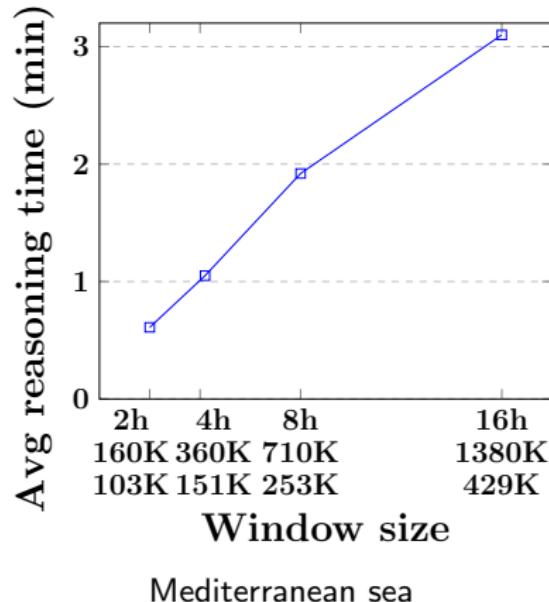
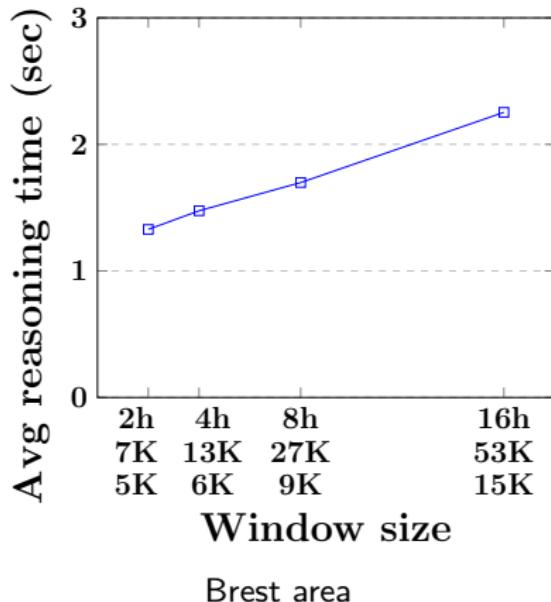
- Real data streams for Maritime Situational Awareness:
 - Brest, France → 18M events/6 months
 - Mediterranean sea → 55M events/1 month
- Task:
 - Computation of the maximal intervals of complex events (e.g. trawling, tugging, loitering, sar, drifting, etc.)



Experimental Results: NetBill & Voting



Experimental Results: Maritime Situational Awareness



Summary

- RTEC: an open-source stream reasoning system based on the Event Calculus.
- Optimisation techniques → e.g. ‘forgetting’ and caching
- Efficient treatment of:
 - cyclic dependencies
 - deadlines
- Not discussed here: Incremental RTEC for handling delays in the input stream⁴

⁴Tsiliannis E., Artikis A. and Palouras G., Incremental Event Calculus for Run-Time Reasoning. In 13th International Conference on Distributed and Event-Based Systems (DEBS), pp. 79–90, 2019.

Further Work

- Handling **uncertainty** in:
 - The input stream (imprecise or incomplete indications)
 - Domain-dependent definitions
- Expressing **query language operators** (e.g. 'sequence')
- Integrating RTEC in **neuro-symbolic frameworks**
- Providing **personalised explanations**

Appendix

Cyclic Dependencies

Cyclic dependencies:

```
initiatedAt( $F_1 = V_1, T$ ) ←  
happensAt( $E_1, T$ ),  
holdsAt( $F_2 = V_2, T$ ).
```

```
initiatedAt( $F_2 = V_2, T$ ) ←  
happensAt( $E_2, T$ ),  
holdsAt( $F_1 = V_1, T$ ).
```

⋮

Voting: Cyclic Dependencies

```
initiatedAt( $status(M) = proposed$ ,  $T$ ) ←  
    happensAt( $propose(P, M)$ ,  $T$ ),  
    holdsAt( $status(M) = null$ ,  $T$ ).
```

Voting: Cyclic Dependencies

```
initiatedAt( $status(M) = proposed$ ,  $T$ ) ←  
    happensAt( $propose(P, M)$ ,  $T$ ),  
    holdsAt( $status(M) = null$ ,  $T$ ).  
  
initiatedAt( $status(M) = voting$ ,  $T$ ) ←  
    happensAt( $second(S, M)$ ,  $T$ ),  
    holdsAt( $status(M) = proposed$ ,  $T$ ).
```

Voting: Cyclic Dependencies

```
initiatedAt( $status(M) = proposed$ ,  $T$ ) ←  
    happensAt( $propose(P, M)$ ,  $T$ ),  
    holdsAt( $status(M) = null$ ,  $T$ ).  
  
initiatedAt( $status(M) = voting$ ,  $T$ ) ←  
    happensAt( $second(S, M)$ ,  $T$ ),  
    holdsAt( $status(M) = proposed$ ,  $T$ ).  
  
initiatedAt( $status(M) = voted$ ,  $T$ ) ←  
    happensAt( $close\_ballot(C, M)$ ,  $T$ ),  
    holdsAt( $status(M) = voting$ ,  $T$ ).
```

Voting: Cyclic Dependencies

```
initiatedAt( $status(M) = proposed$ ,  $T$ ) ←  
    happensAt( $propose(P, M)$ ,  $T$ ),  
    holdsAt( $status(M) = null$ ,  $T$ ).  
  
initiatedAt( $status(M) = voting$ ,  $T$ ) ←  
    happensAt( $second(S, M)$ ,  $T$ ),  
    holdsAt( $status(M) = proposed$ ,  $T$ ).  
  
initiatedAt( $status(M) = voted$ ,  $T$ ) ←  
    happensAt( $close\_ballot(C, M)$ ,  $T$ ),  
    holdsAt( $status(M) = voting$ ,  $T$ ).  
  
initiatedAt( $status(M) = null$ ,  $T$ ) ←  
    happensAt( $declare(C, M, Res)$ ,  $T$ ),  
    holdsAt( $status(M) = voted$ ,  $T$ ).
```

Voting: Cyclic Dependencies

```
initiatedAt(status(M) = proposed, T) ←  
    happensAt(propose(P, M), T),  
    holdsAt(status(M) = null, T).  
  
initiatedAt(status(M) = voting, T) ←  
    happensAt(second(S, M), T),  
    holdsAt(status(M) = proposed, T).  
  
initiatedAt(status(M) = voted, T) ←  
    happensAt(close_ballot(C, M), T),  
    holdsAt(status(M) = voting, T).  
  
initiatedAt(status(M) = null, T) ←  
    happensAt(declare(C, M, Res), T),  
    holdsAt(status(M) = voted, T).
```

⋮

Voting: Cyclic Dependencies

```
initiatedAt(status(M) = proposed, T) ←  
    happensAt(propose(P, M), T),  
    holdsAt(status(M) = null, T).  
  
initiatedAt(status(M) = voting, T) ←  
    happensAt(second(S, M), T),  
    holdsAt(status(M) = proposed, T).  
  
initiatedAt(status(M) = voted, T) ←  
    happensAt(close_ballot(C, M), T),  
    holdsAt(status(M) = voting, T).  
  
initiatedAt(status(M) = null, T) ←  
    happensAt(declare(C, M, Res), T),  
    holdsAt(status(M) = voted, T).
```

⋮

Voting: Cyclic Dependencies

```
initiatedAt(status(M) = proposed, T) ←  
    happensAt(propose(P, M), T),  
    holdsAt(status(M) = null, T).  
  
initiatedAt(status(M) = voting, T) ←  
    happensAt(second(S, M), T),  
    holdsAt(status(M) = proposed, T).  
  
initiatedAt(status(M) = voted, T) ←  
    happensAt(close_ballot(C, M), T),  
    holdsAt(status(M) = voting, T).  
  
initiatedAt(status(M) = null, T) ←  
    happensAt(declare(C, M, Res), T),  
    holdsAt(status(M) = voted, T).
```

⋮

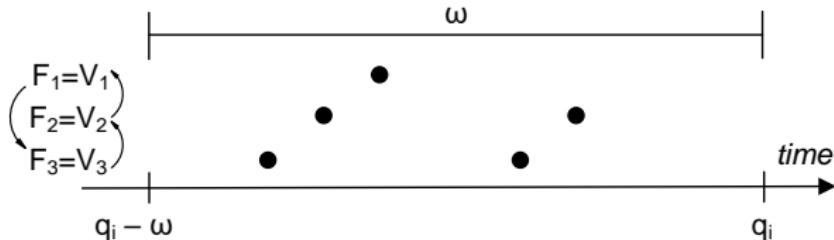
Voting: Cyclic Dependencies

```
initiatedAt(status(M) = proposed, T) ←  
    happensAt(propose(P, M), T),  
    holdsAt(status(M) = null, T).  
  
initiatedAt(status(M) = voting, T) ←  
    happensAt(second(S, M), T),  
    holdsAt(status(M) = proposed, T).  
  
initiatedAt(status(M) = voted, T) ←  
    happensAt(close_ballot(C, M), T),  
    holdsAt(status(M) = voting, T).  
  
initiatedAt(status(M) = null, T) ←  
    happensAt(declare(C, M, Res), T),  
    holdsAt(status(M) = voted, T).
```

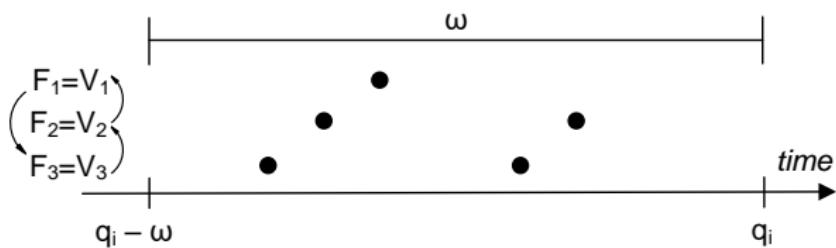
⋮

Handling Cyclic Dependencies

With Caching (RTEC)

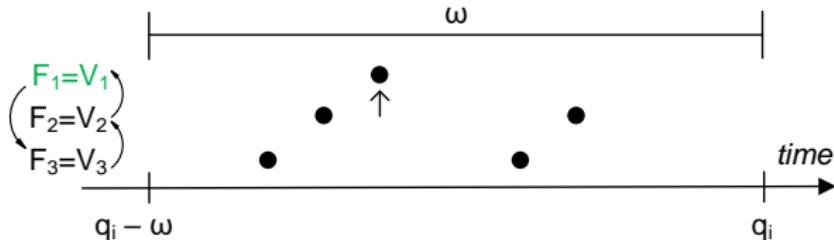


Without Caching

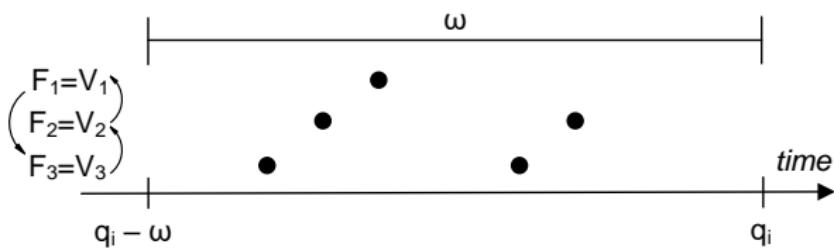


Handling Cyclic Dependencies

With Caching (RTEC)

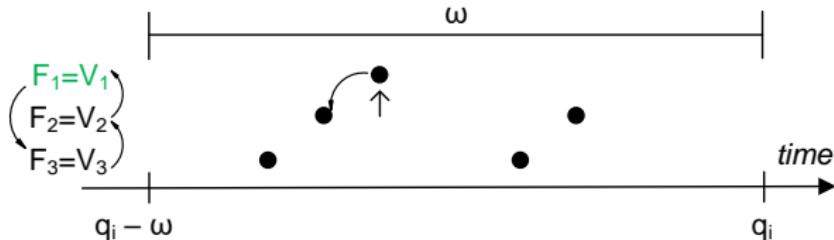


Without Caching

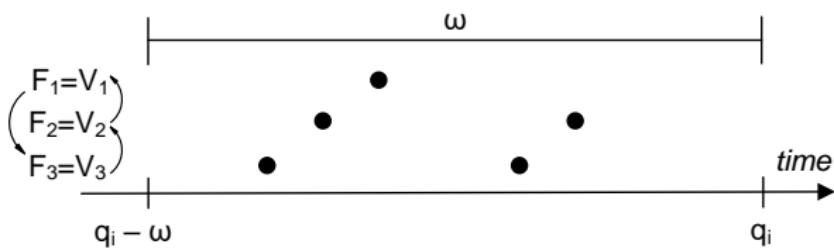


Handling Cyclic Dependencies

With Caching (RTEC)

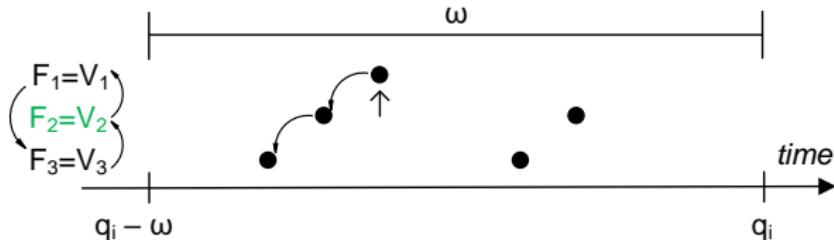


Without Caching

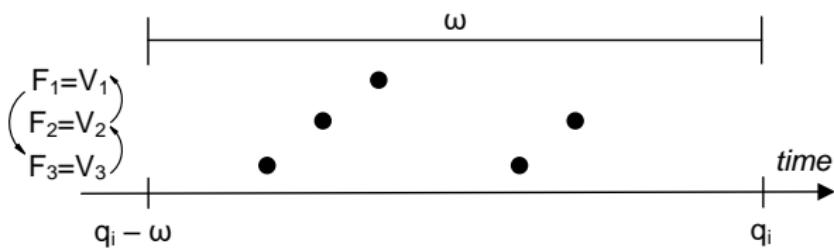


Handling Cyclic Dependencies

With Caching (RTEC)

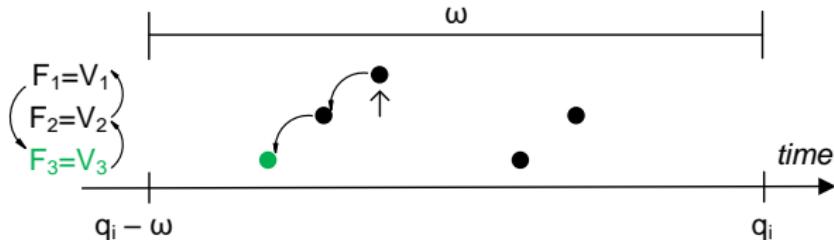


Without Caching

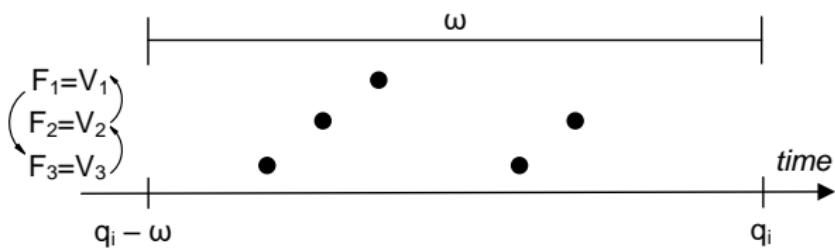


Handling Cyclic Dependencies

With Caching (RTEC)

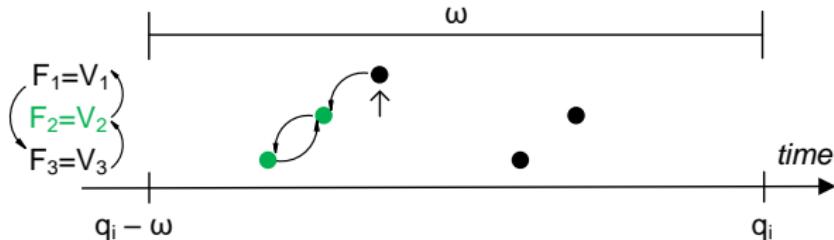


Without Caching

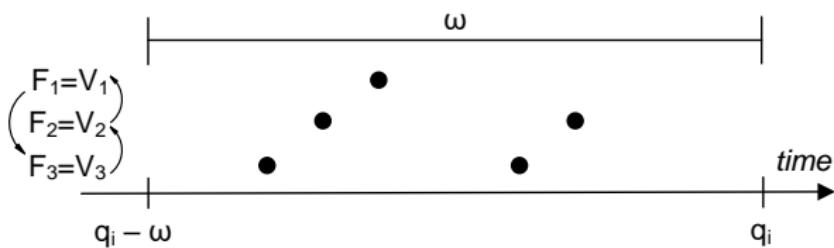


Handling Cyclic Dependencies

With Caching (RTEC)

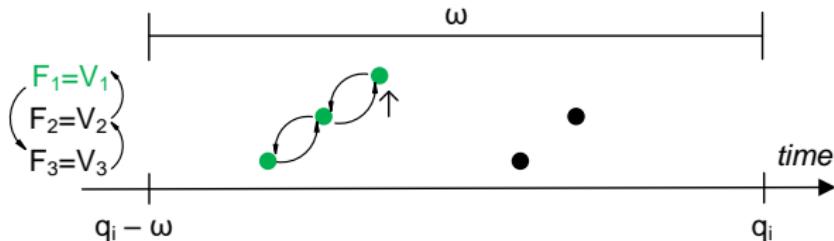


Without Caching

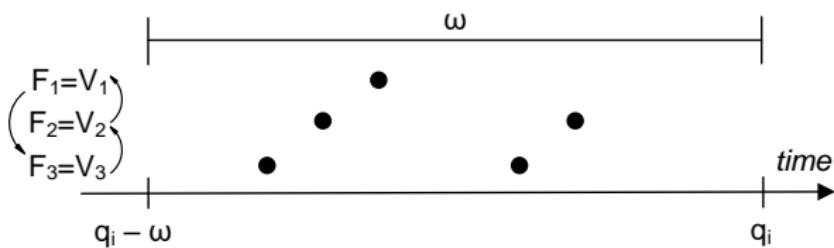


Handling Cyclic Dependencies

With Caching (RTEC)

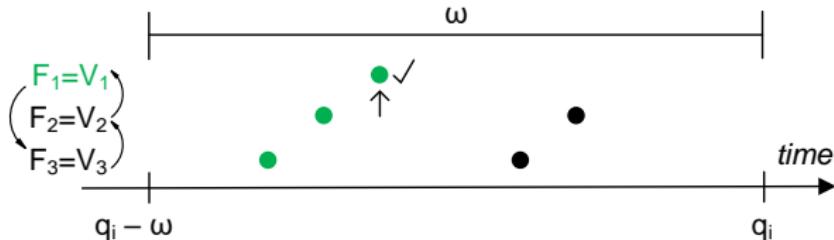


Without Caching

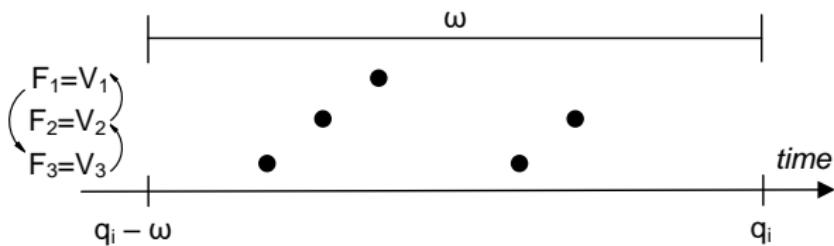


Handling Cyclic Dependencies

With Caching (RTEC)

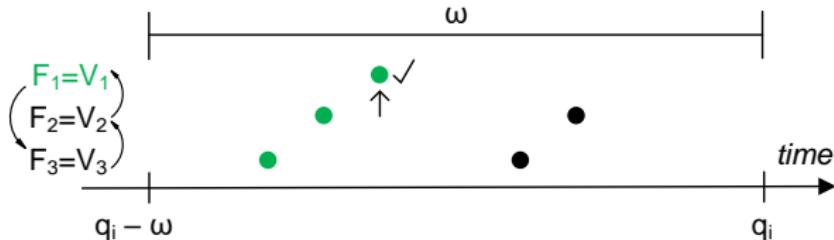


Without Caching

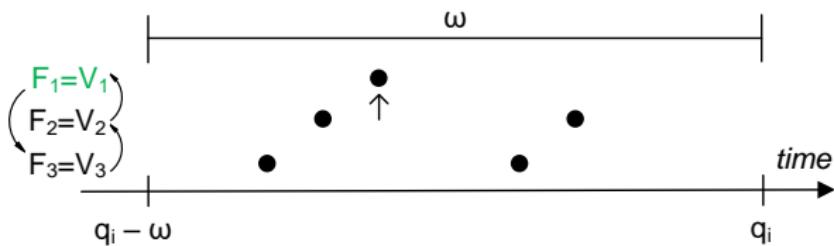


Handling Cyclic Dependencies

With Caching (RTEC)

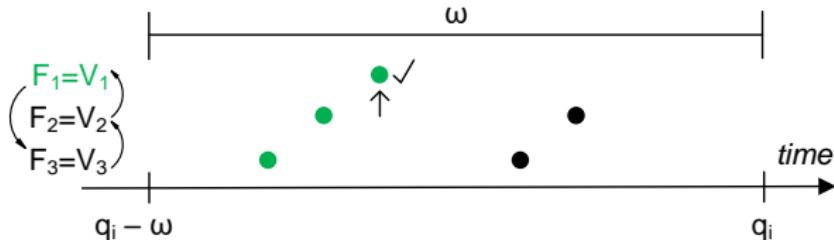


Without Caching

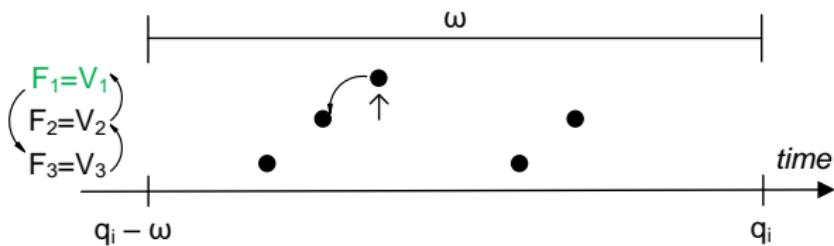


Handling Cyclic Dependencies

With Caching (RTEC)

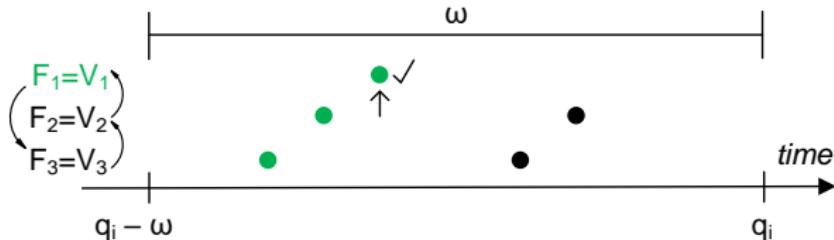


Without Caching

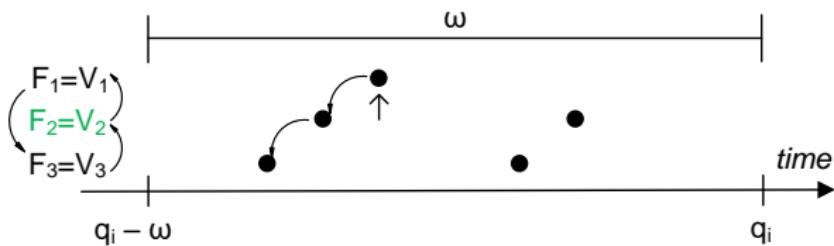


Handling Cyclic Dependencies

With Caching (RTEC)

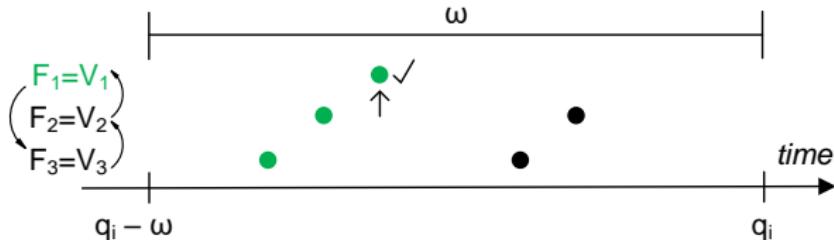


Without Caching

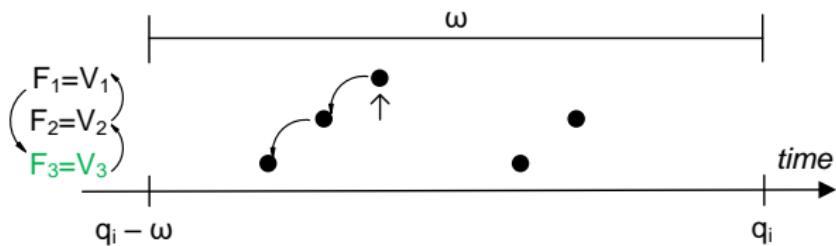


Handling Cyclic Dependencies

With Caching (RTEC)

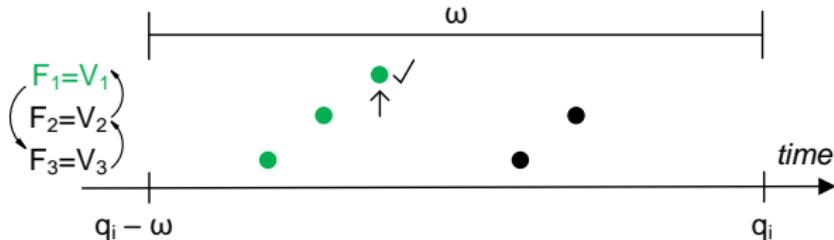


Without Caching

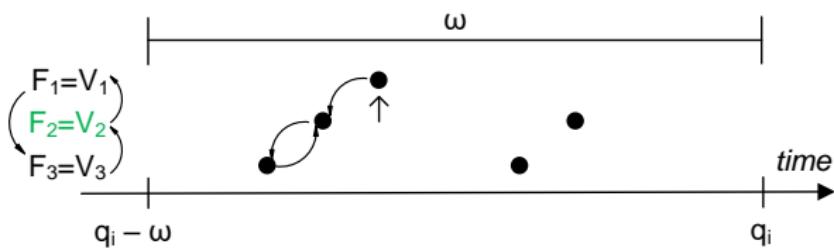


Handling Cyclic Dependencies

With Caching (RTEC)

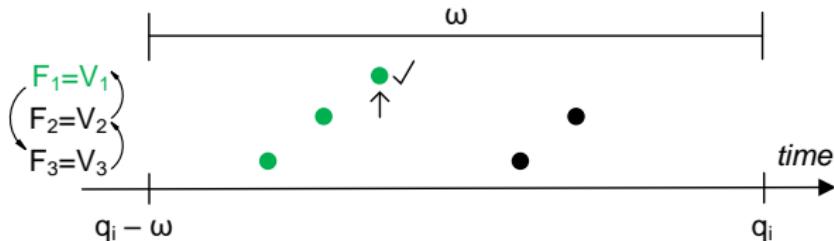


Without Caching

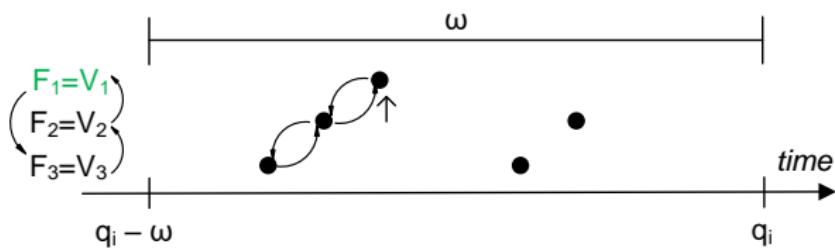


Handling Cyclic Dependencies

With Caching (RTEC)

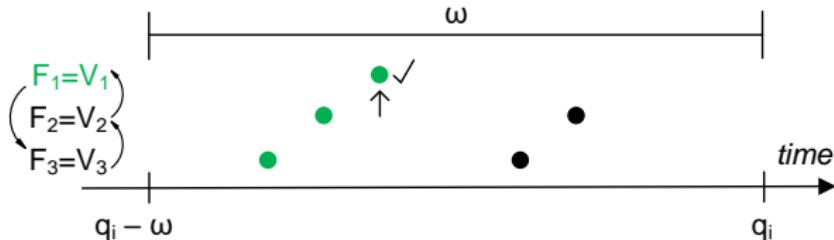


Without Caching

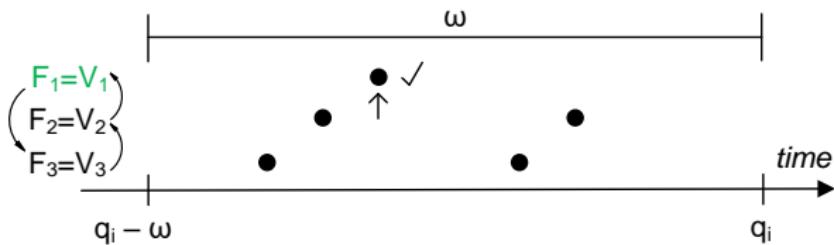


Handling Cyclic Dependencies

With Caching (RTEC)

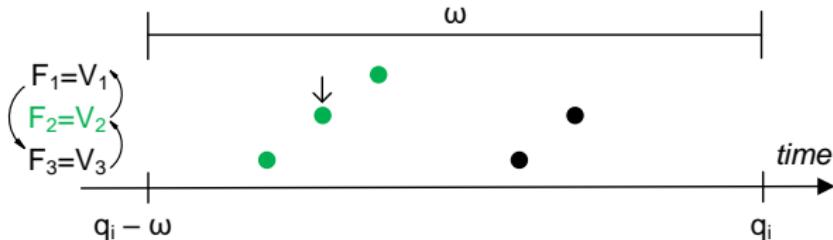


Without Caching

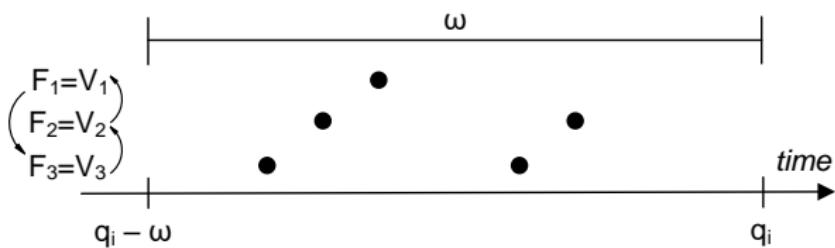


Handling Cyclic Dependencies

With Caching (RTEC)

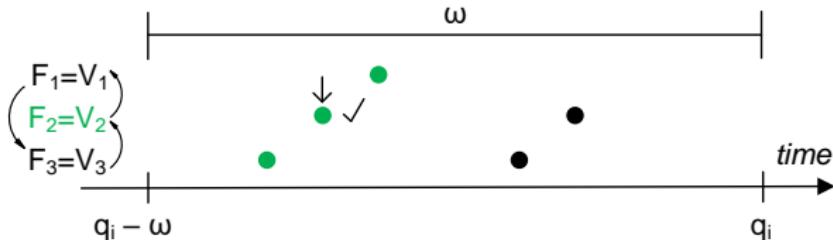


Without Caching

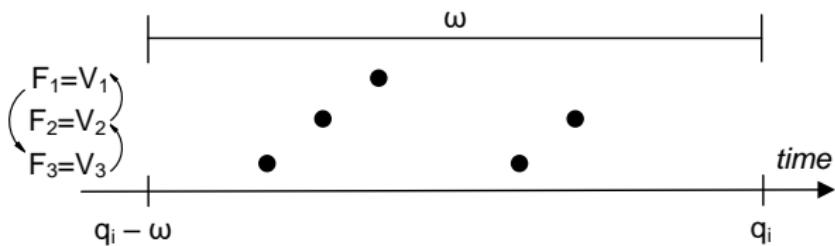


Handling Cyclic Dependencies

With Caching (RTEC)

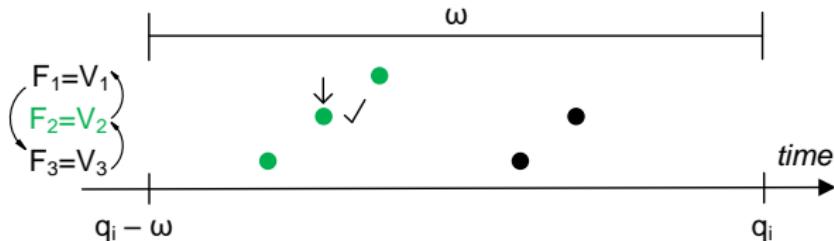


Without Caching

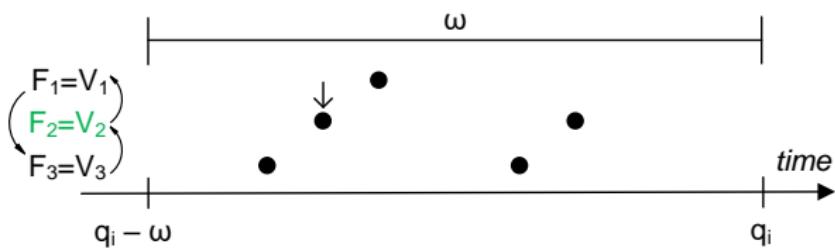


Handling Cyclic Dependencies

With Caching (RTEC)

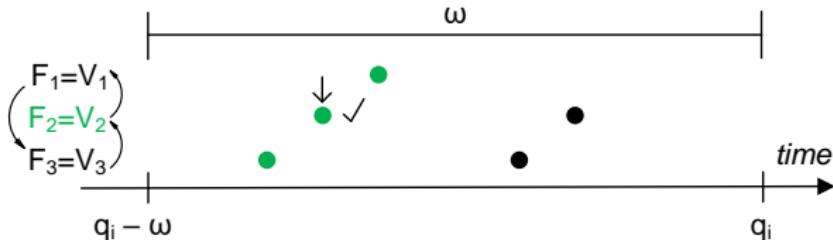


Without Caching

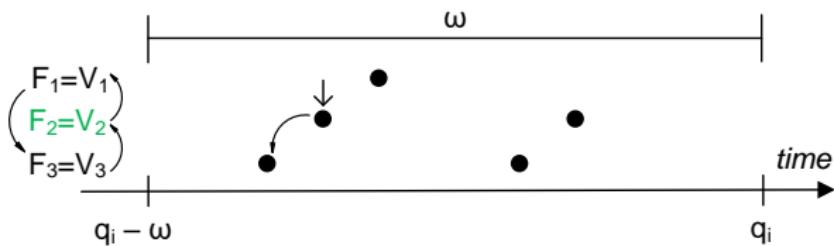


Handling Cyclic Dependencies

With Caching (RTEC)

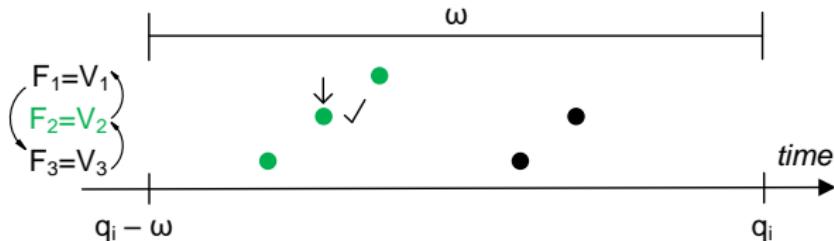


Without Caching

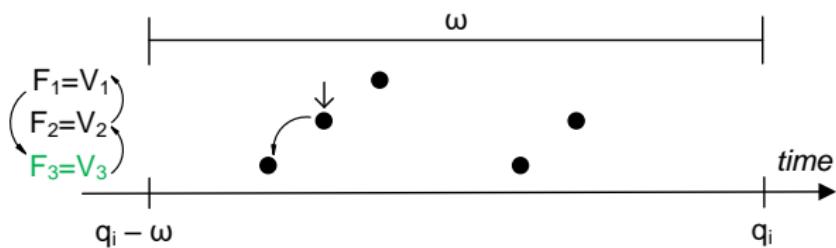


Handling Cyclic Dependencies

With Caching (RTEC)

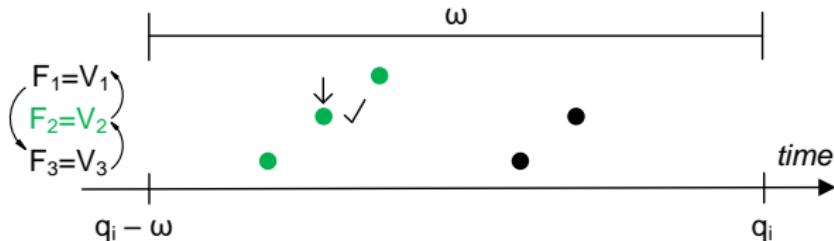


Without Caching

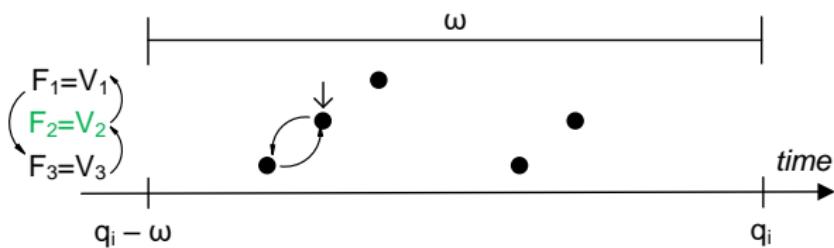


Handling Cyclic Dependencies

With Caching (RTEC)

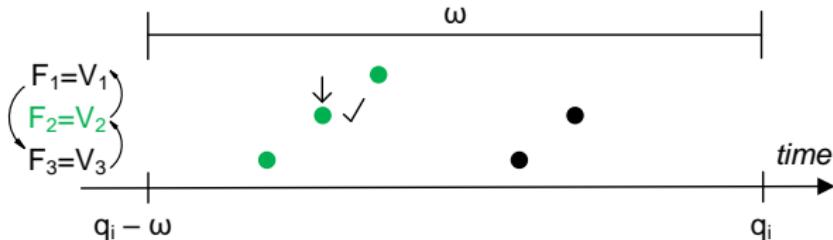


Without Caching

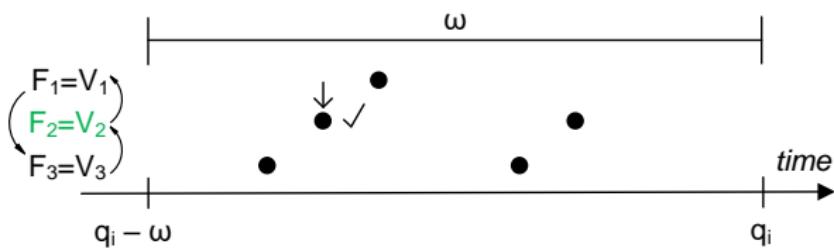


Handling Cyclic Dependencies

With Caching (RTEC)

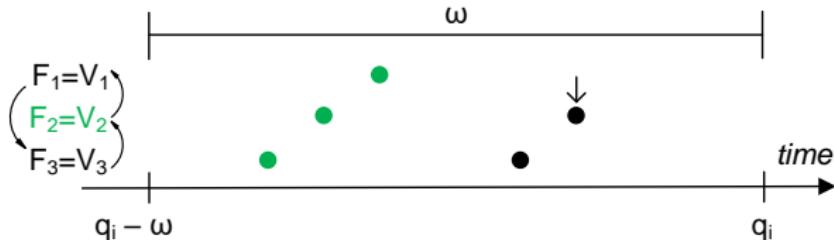


Without Caching

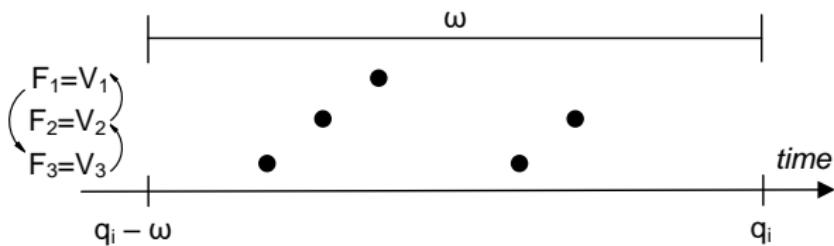


Handling Cyclic Dependencies

With Caching (RTEC)

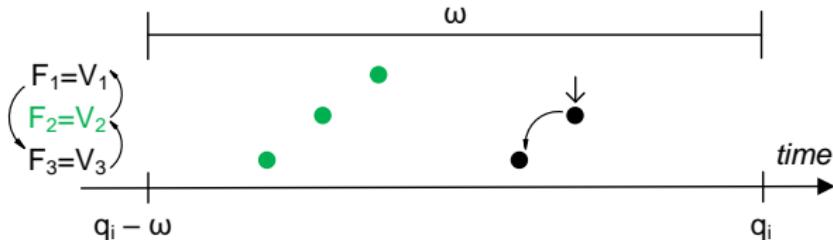


Without Caching

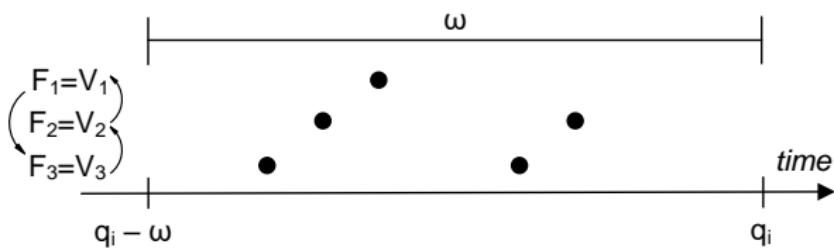


Handling Cyclic Dependencies

With Caching (RTEC)

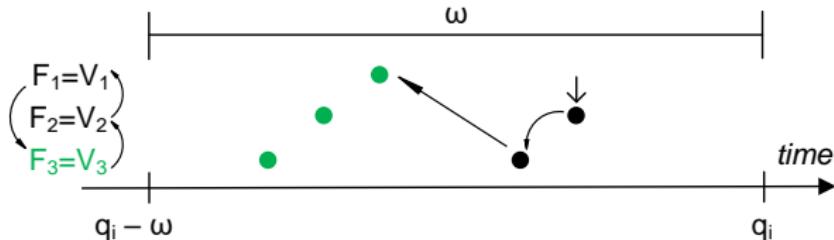


Without Caching

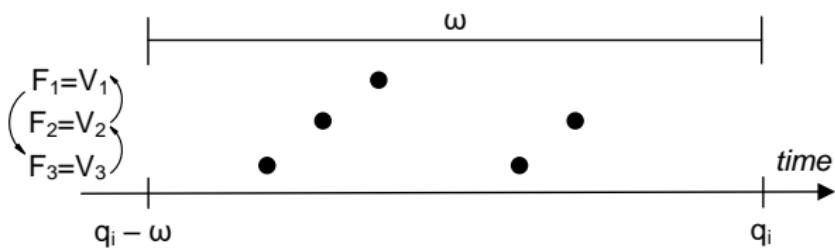


Handling Cyclic Dependencies

With Caching (RTEC)

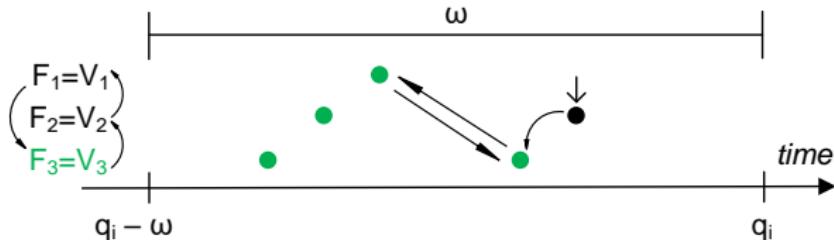


Without Caching

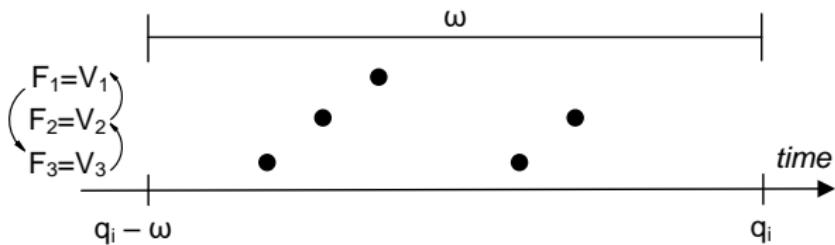


Handling Cyclic Dependencies

With Caching (RTEC)

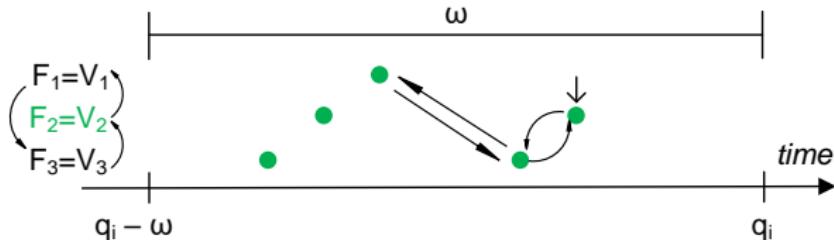


Without Caching

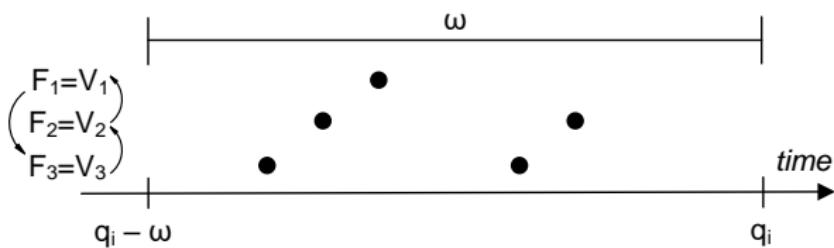


Handling Cyclic Dependencies

With Caching (RTEC)

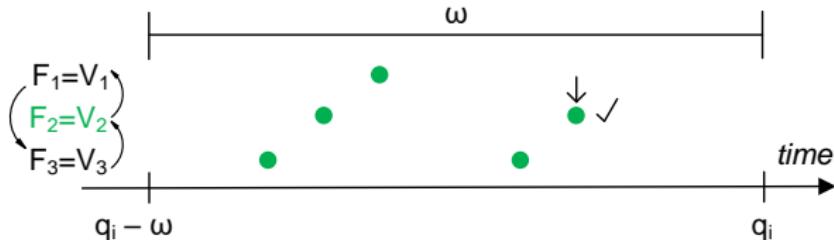


Without Caching

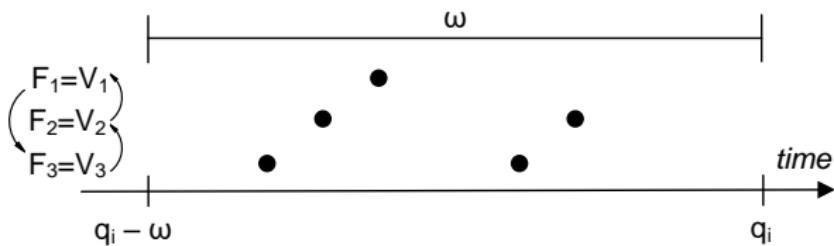


Handling Cyclic Dependencies

With Caching (RTEC)

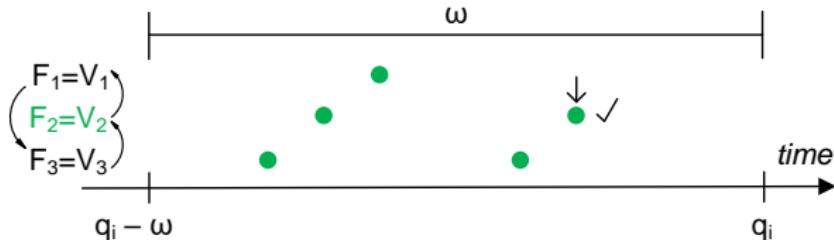


Without Caching

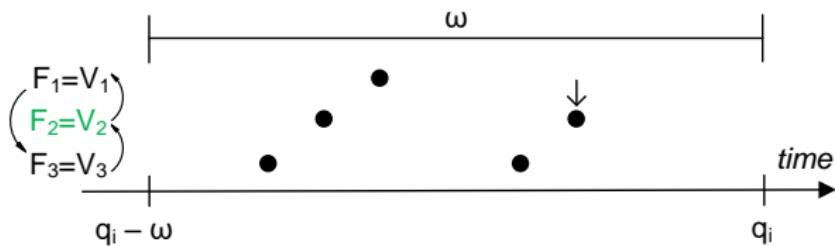


Handling Cyclic Dependencies

With Caching (RTEC)

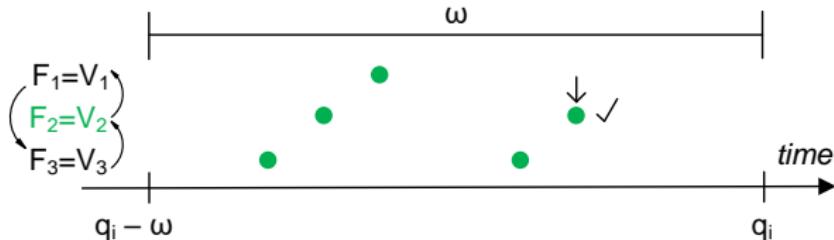


Without Caching

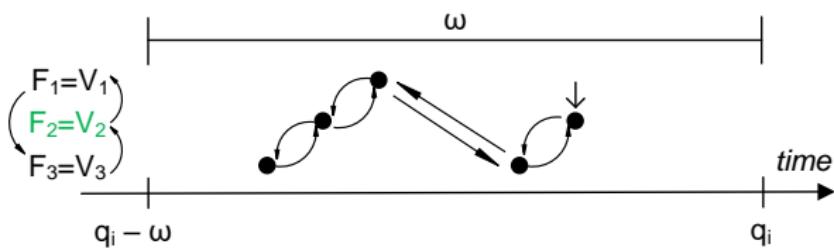


Handling Cyclic Dependencies

With Caching (RTEC)

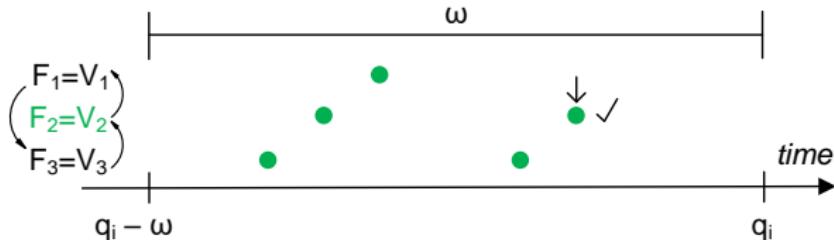


Without Caching

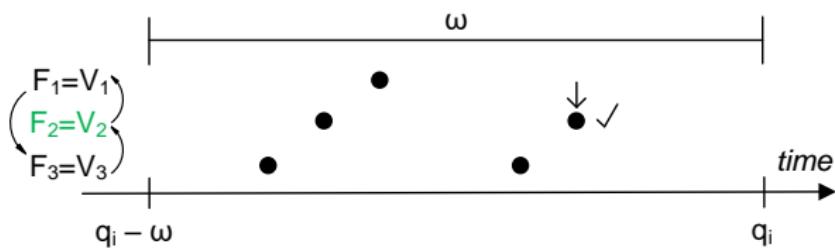


Handling Cyclic Dependencies

With Caching (RTEC)

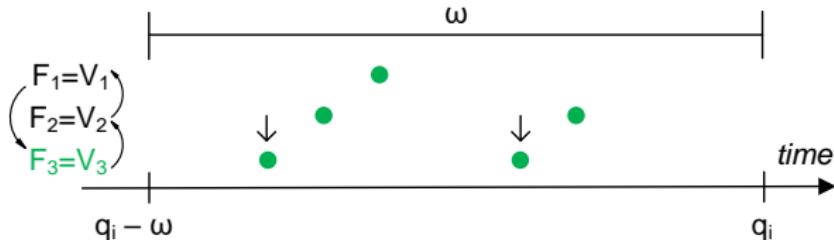


Without Caching

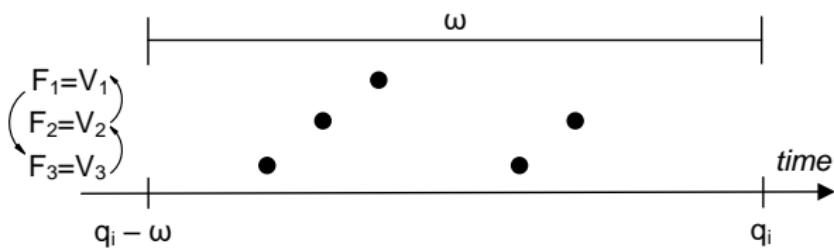


Handling Cyclic Dependencies

With Caching (RTEC)

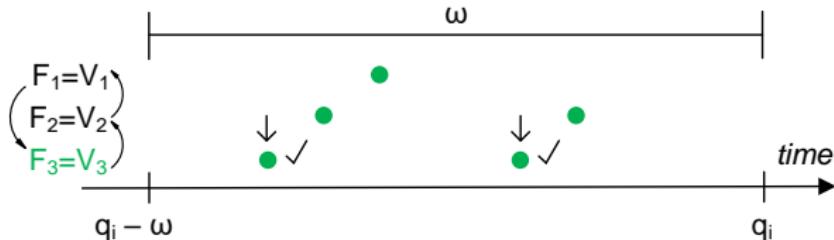


Without Caching

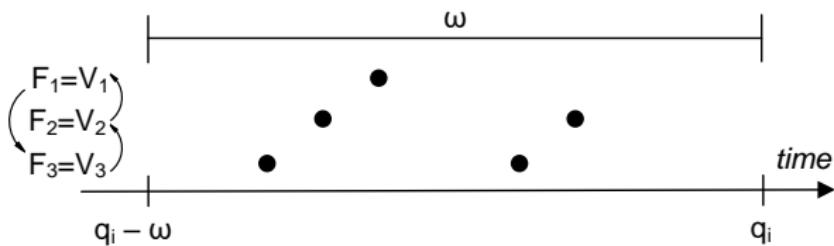


Handling Cyclic Dependencies

With Caching (RTEC)

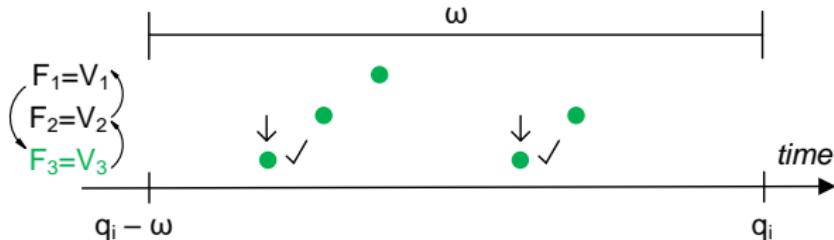


Without Caching

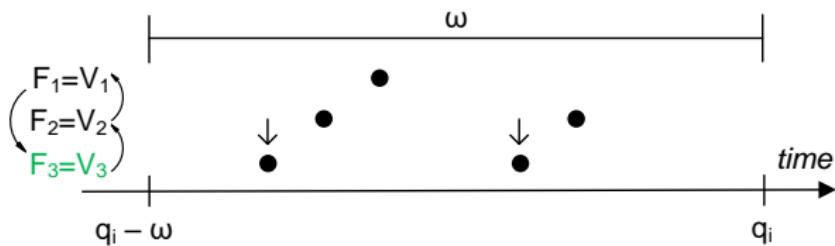


Handling Cyclic Dependencies

With Caching (RTEC)

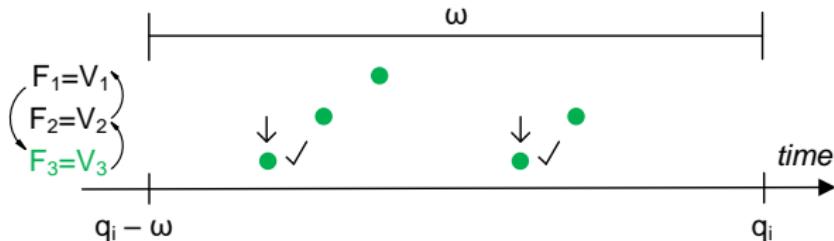


Without Caching

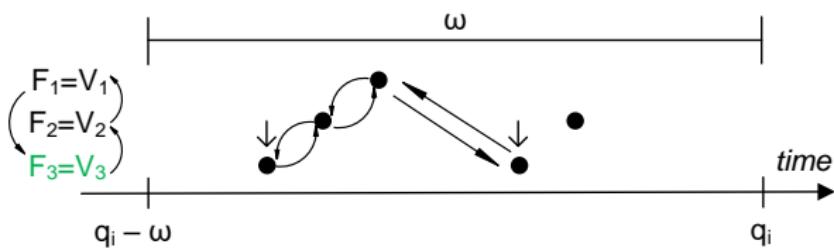


Handling Cyclic Dependencies

With Caching (RTEC)

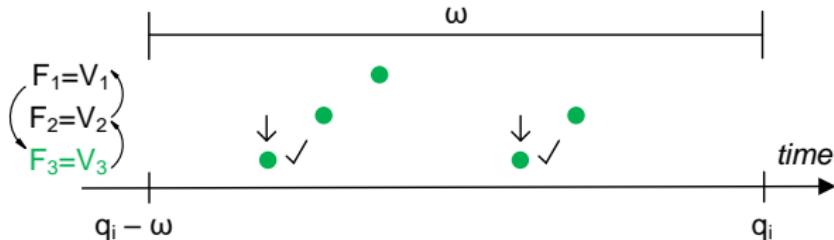


Without Caching

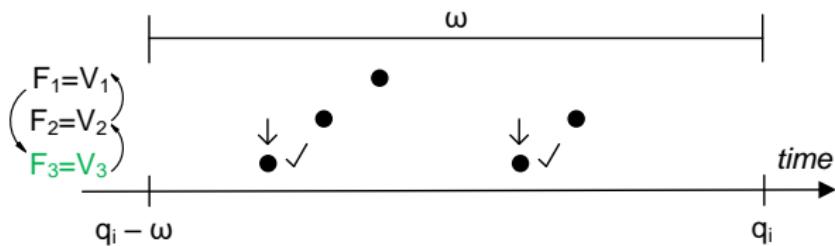


Handling Cyclic Dependencies

With Caching (RTEC)



Without Caching



Handling Cyclic Dependencies: Complexity

- Without Caching:

$$\mathcal{O}(m_\omega^2 l^2 k)$$

- With Caching (RTEC):

$$\mathcal{O}(m_\omega lk)$$

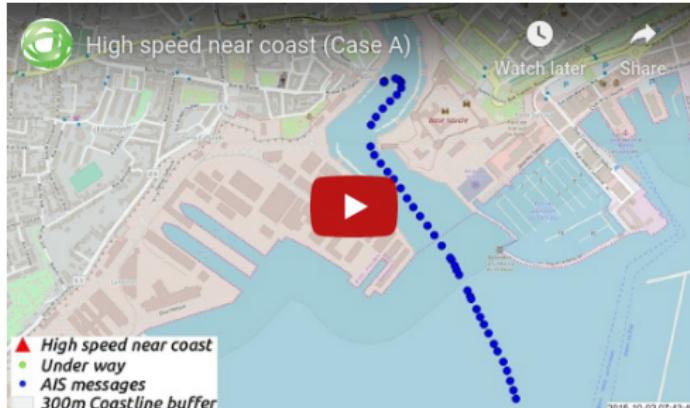
where

- m_ω : window size
- l : cycle length
- k : cost of computing an initiation/termination point

Example: High Speed Near Coast

initiatedAt(*highSpeedNC(Vessel)* = true, T) \leftarrow
happensAt(*velocity(Vessel, Speed, _CoG, _TrueHeading)*, T),
holdsAt(*withinArea(Vessel, nearCoast)* = true, T),
threshold(v_{hs}, V_{hs}),
Speed > V_{hs}.

terminatedAt(*highSpeedNC(Vessel)* = true, T) \leftarrow
happensAt(*Vessel, end(withinArea(Vessel, nearCoast))* = true), T).



Voting: Fluents

initiatedAt($\text{obligation}(\text{declare}(C, M, \text{carried})) = \text{true}, T$) \leftarrow
happensAt($\text{start}(\text{status}(M) = \text{voted}), T$),
 $\text{countVotes}(\text{voted}(V, M) = \text{aye}, \text{Ayes})$,
 $\text{countVotes}(\text{voted}(V, M) = \text{aye}, \text{Nays})$, $\text{Ayes} \geq \text{Nays}$.

initiatedAt($\text{suspended}(C) = \text{true}, T$) \leftarrow
happensAt($\text{end}(\text{status}(M) = \text{voted}), T$),
not happensAt($\text{declare}(C, M, \text{carried}), T$),
holdsAt($\text{obligation}(\text{declare}(C, M, \text{carried})) = \text{true}, T$).

Voting: Fluents

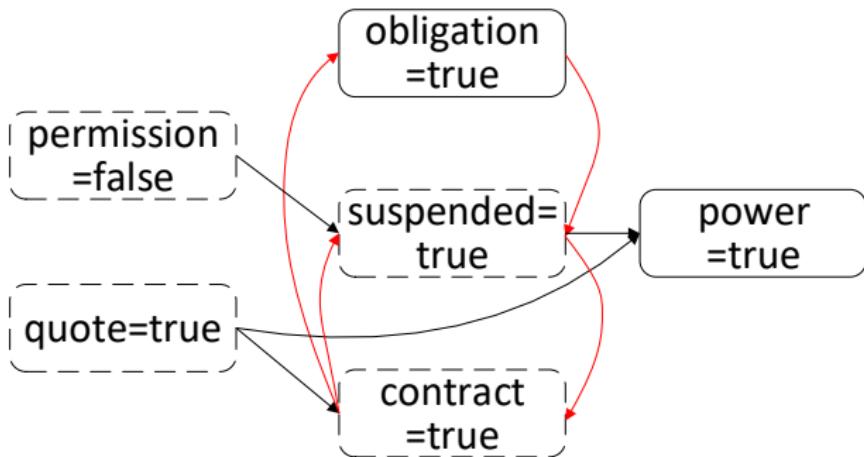
initiatedAt($\text{obligation}(\text{declare}(C, M, \text{carried})) = \text{true}, T$) \leftarrow
happensAt($\text{start}(\text{status}(M) = \text{voted}), T$),
 $\text{countVotes}(\text{voted}(V, M) = \text{aye}, \text{Ayes})$,
 $\text{countVotes}(\text{voted}(V, M) = \text{aye}, \text{Nays})$, $\text{Ayes} \geq \text{Nays}$.

initiatedAt($\text{suspended}(C) = \text{true}, T$) \leftarrow
happensAt($\text{end}(\text{status}(M) = \text{voted}), T$),
not happensAt($\text{declare}(C, M, \text{carried}), T$),
holdsAt($\text{obligation}(\text{declare}(C, M, \text{carried})) = \text{true}, T$).

Example Application: NetBill

- NetBill: a protocol for exchanging digital goods
- Agents (consumers and merchants) form contracts and perform transactions
- RTEC monitors normative positions of agents, e.g.:
 - obligation,
 - permission or
 - institutionalised powerto perform some action (e.g., to present an offer)
- Agents may be suspended for defying the rules of NetBill

Netbill: Entity Dependency Graph



Netbill: Fluents

```
initiatedAt(quote(M, C, G)=true, T) ←  
    happensAt(present_quote(M, C, G, P), T).  
  
terminatedAt(quote(M, C, G)=true, T) ←  
    happensAt(accept_quote(C, M, G), T).
```

Netbill: Fluents

$\text{initiatedAt}(\text{permission}(\text{present_quote}(M, C)) = \text{true}, T) \leftarrow$
 $\text{happensAt}(\text{request_quote}(C, M, G), T).$

$\text{initiatedAt}(\text{permission}(\text{present_quote}(M, C)) = \text{false}, T) \leftarrow$
 $\text{happensAt}(\text{present_quote}(M, C, G, P), T).$

$\text{initiatedAt}(\text{suspended}(M) = \text{true}, T) \leftarrow$
 $\text{happensAt}(\text{present_quote}(M, C, G, P), T),$
 $\text{holdsAt}(\text{permission}(\text{present_quote}(M, C)) = \text{false}, T).$

Netbill: Fluents

$\text{initiatedAt}(\text{permission}(\text{present_quote}(M, C)) = \text{true}, T) \leftarrow$
 $\text{happensAt}(\text{request_quote}(C, M, G), T).$

$\text{initiatedAt}(\text{permission}(\text{present_quote}(M, C)) = \text{false}, T) \leftarrow$
 $\text{happensAt}(\text{present_quote}(M, C, G, P), T).$

$\text{initiatedAt}(\text{suspended}(M) = \text{true}, T) \leftarrow$
 $\text{happensAt}(\text{present_quote}(M, C, G, P), T),$
 $\text{holdsAt}(\text{permission}(\text{present_quote}(M, C)) = \text{false}, T).$

Netbill: Cyclic Definitions

```
initiatedAt(contract( $M, C, G = \text{true}$ ,  $T$ )  $\leftarrow$ 
  happensAt(accept-quote( $C, M, G$ ),  $T$ ),
  holdsAt(quote( $M, C, G = \text{true}$ ,  $T$ ),
  not holdsAt(suspended( $M = \text{true}$ ),  $T$ ),
  not holdsAt(suspended( $C = \text{true}$ ),  $T$ ).
```

```
maxDuration(contract( $M, C, G = \text{true}$ ,  $r_{contr}$ )).
```

```
initiatedAt(suspended( $M = \text{true}$ ),  $T$ )  $\leftarrow$ 
  initiatedAt(contract( $M, C, G = \text{false}$ ,  $T$ ),
  holdsAt(obligation(send-goods( $M, S, G = \text{true}$ ),  $T$ )).
```

```
maxDuration(suspended( $M = \text{true}$ ),  $r_{susp}$ ).
```

Netbill: Deadlines

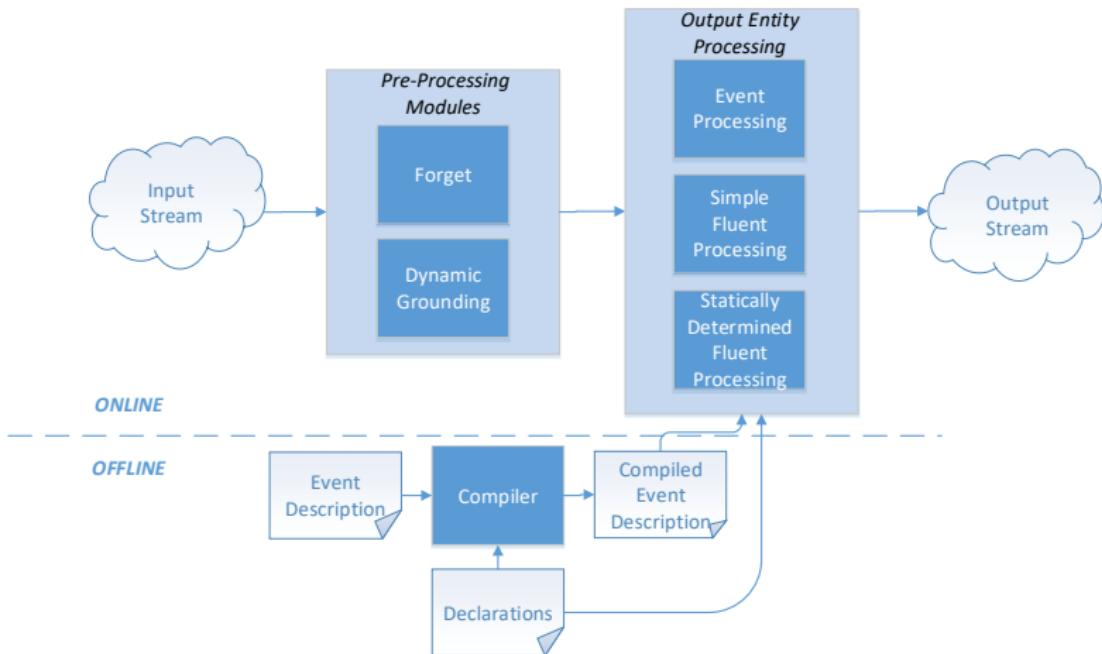
initiatedAt($\text{contract}(M, C, G) = \text{true}$, T) \leftarrow
 happensAt($\text{accept_quote}(C, M, G)$, T),
 holdsAt($\text{quote}(M, C, G) = \text{true}$, T),
 not holdsAt($\text{suspended}(M) = \text{true}$, T),
 not holdsAt($\text{suspended}(C) = \text{true}$, T).

maxDuration($\text{contract}(M, C, G) = \text{true}$, r_{contr}).

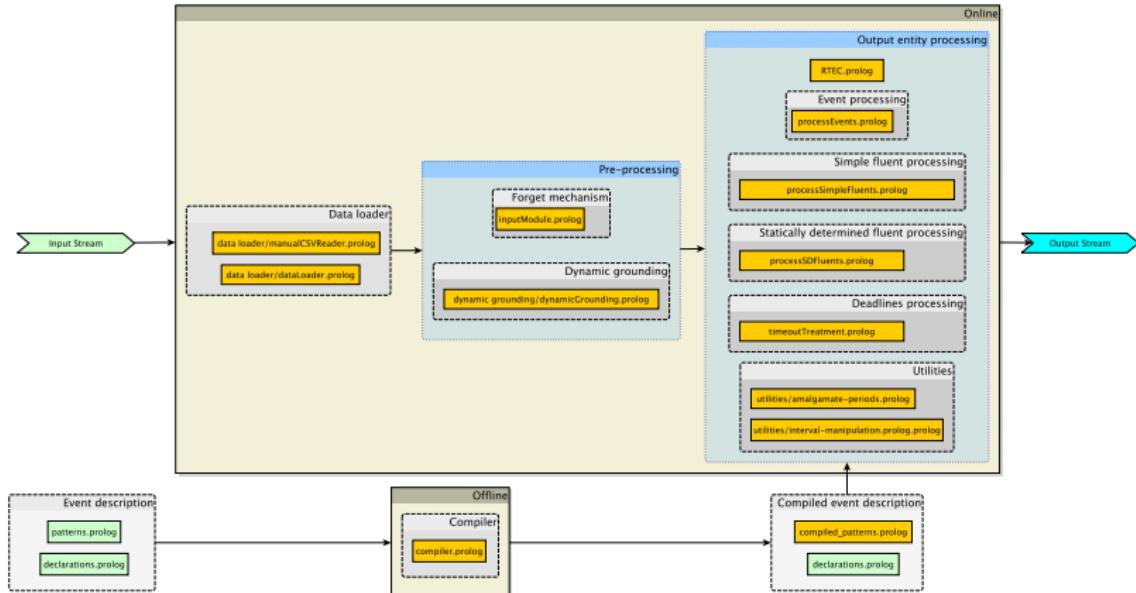
initiatedAt($\text{suspended}(M) = \text{true}$, T) \leftarrow
 initiatedAt($\text{contract}(M, C, G) = \text{false}$, T),
 holdsAt($\text{obligation}(\text{send_goods}(M, S, G) = \text{true})$, T).

maxDuration($\text{suspended}(M) = \text{true}$, r_{susp}).

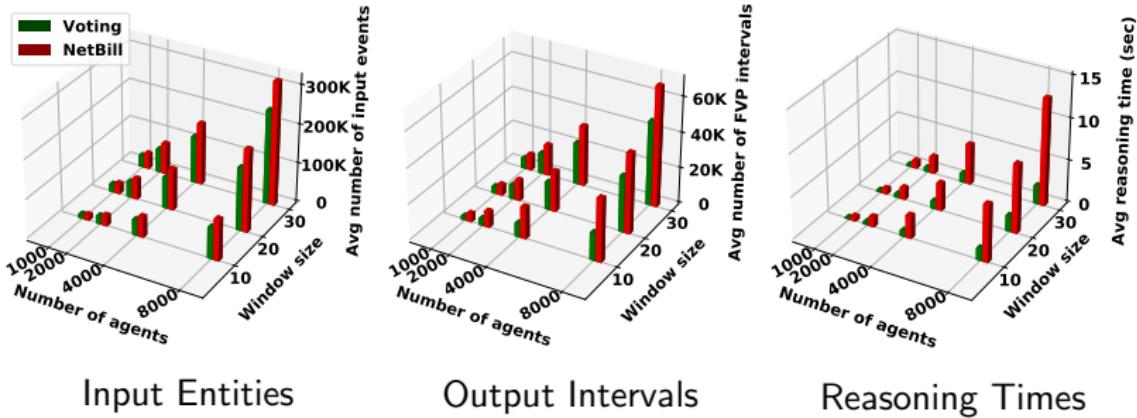
Run-Time Event Calculus (RTEC)



Architecture



Experimental Results: NetBill and Voting, Cycles

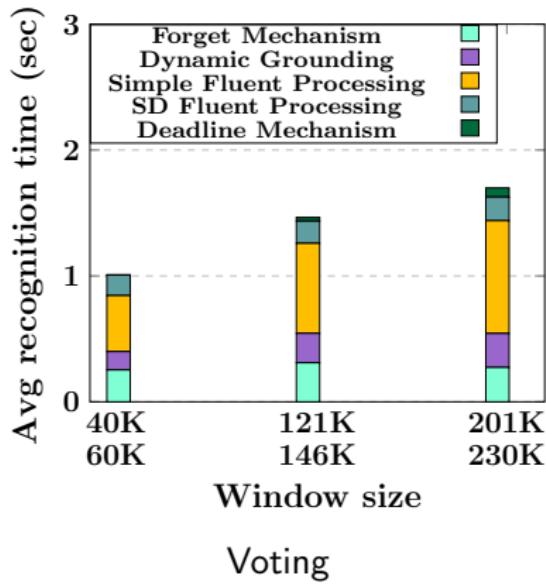
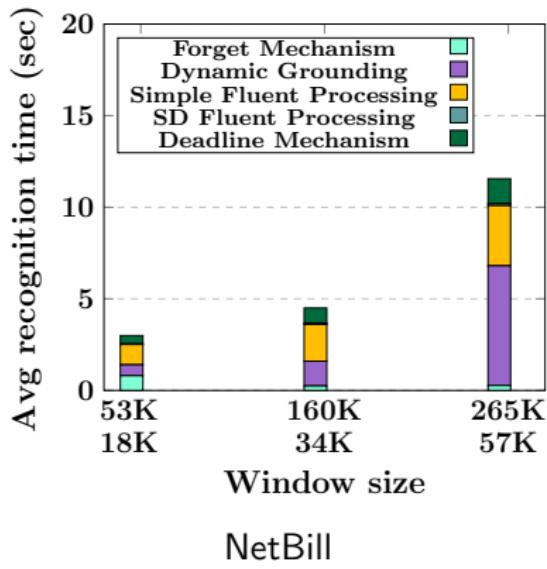


Input Entities

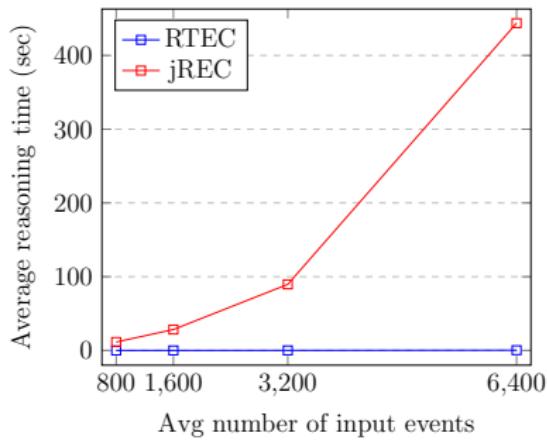
Output Intervals

Reasoning Times

Experimental Results: NetBill and Voting



Comparison of RTEC and jREC on cyclic dependencies



Voting, cyclic ‘status’ pattern