Composite Activity Definition Construction with Large Larnguage Models

Andreas Kouvaras
a.kouvaras@unipi.gr
University of Piraeus
Greece

Periklis Mantenoglou
pmantenoglou@iit.demokritos.gr
NCSR "Demokritos"
Greece

Alexander Artikis
a.artikis@unipi.gr
University of Piraeus & NCSR "Demokritos"
Greece

Abstract

We use LLMs to construct event descriptions for RTEC.

CCS Concepts

• **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

Keywords

Do, Not, Us, This, Code, Put, the, Correct, Terms, for, Your, Paper

ACM Reference Format:
Andreas Kouvaras, Periklis Mantenoglou, and Alexander Artikis. 2018. Composite Activity Definition Construction with Large Larnguage Models. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

2 Background

2.1 Run-Time Event Calculus

The Event Calculus is a logic programming formalisms for representing events and reasoning about their effects over time [3]. The Run-Time Event Calculus (RTEC) is an extension of the Event Calculus that is optimised for composite event recognition over large event streams [1, 4, 5].

Representation. The language of RTEC is many-sorted, including sorts for representing time, instantaneous events and fluents. RTEC employs a linear time-line with non-negative integer time-points. A 'fluent-value pair' (FVP) $F{=}V$ denotes that fluent F has value V. $\mathsf{happensAt}(E, T)$ signifies that event E occurs at time-point T. $\mathsf{initiatedAt}(F = V, T)$ (resp. $\mathsf{terminatedAt}(F = V, T)$) expresses that a time period during which a fluent F has the value V continuously is initiated (terminated) at T. $\mathsf{holdsAt}(F = V, T)$ states that F has value V at T, while $\mathsf{holdsFor}(F = V, I)$ expresses that $F{=}V$ holds continuously in the intervals included in list I.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

A formalisation of the temporal specifications of a domain in RTEC is called *event description*.

Definition 1 (Event Description)**.** An event description is a set of:

- ground $\mathsf{happensAt}(E, T)$ facts, expressing an input stream of event instances,
- rules with head $\mathsf{initiatedAt}(F = V, T)$ or $\mathsf{terminatedAt}(F = V, T)$, expressing the effects of events on FVP $F{=}V$, and
- rules with head $\mathsf{holdsFor}(F = V, I)$, defining FVP $F{=}V$ based on other FVPs. ■

RTEC features two types of FVPs: 'simple' and 'statically determined'. Simple FVPs are defined using a set of $\mathsf{initiatedAt}$ and $\mathsf{terminatedAt}$ rules, and are subject to the commonsense law of inertia, i.e., a FVP $F{=}V$ holds at a time-point T, if $F{=}V$ has been 'initiated' by an event at a time-point earlier than T, and not 'terminated' by another event in the meantime.

Example 1 (Within area)**.** In maritime monitoring, various areas, e.g., fisheries restricted areas, disallow certain activities. It is thus useful to compute the intervals during which a vessel is in such an area. See the formalisation below:

$$\begin{aligned}&\mathsf{initiatedAt}(\mathit{withinArea}(\mathit{Vl}, \mathit{AreaType}) = \mathsf{true},\ T) \leftarrow\\ &\quad \mathsf{happensAt}(\mathit{entersArea}(\mathit{Vl}, \mathit{AreaID}),\ T),\\ &\quad \mathit{areaType}(\mathit{AreaID}, \mathit{AreaType}).\end{aligned} \tag{1}$$

$$\begin{aligned}&\mathsf{terminatedAt}(\mathit{withinArea}(\mathit{Vl}, \mathit{AreaType}) = \mathsf{true},\ T) \leftarrow\\ &\quad \mathsf{happensAt}(\mathit{leavesArea}(\mathit{Vl}, \mathit{AreaID}),\ T),\\ &\quad \mathit{areaType}(\mathit{AreaID}, \mathit{AreaType}).\end{aligned} \tag{2}$$

$$\begin{aligned}&\mathsf{terminatedAt}(\mathit{withinArea}(\mathit{Vl}, \mathit{AreaType}) = \mathsf{true},\ T) \leftarrow\\ &\quad \mathsf{happensAt}(\mathit{gapStart}(\mathit{Vl}),\ T).\end{aligned} \tag{3}$$

$\mathit{withinArea}(\mathit{Vl}, \mathit{AreaType})$ is a Boolean simple fluent denoting that a vessel Vl is in some area of $\mathit{AreaType}$, while $\mathit{entersArea}(\mathit{Vl}, \mathit{AreaID})$, $\mathit{leavesArea}(\mathit{Vl}, \mathit{AreaID})$ and $\mathit{gapStart}(\mathit{Vl})$ are input events, derived by the online processing of vessel position signals, and their spatial relations with areas of interest [7]. $\mathit{areaType}(\mathit{AreaID}, \mathit{AreaType})$ is an atemporal predicate storing background knowledge concerning the types of areas in a dataset. Rules (1) and (2) state that $\mathit{withinArea}(\mathit{Vl}, \mathit{AreaType})$ is initiated (resp. terminated) as soon as vessel Vl enters (leaves) an area AreaID, whose type is $\mathit{AreaType}$. According to rule (3), $\mathit{withinArea}(\mathit{Vl}, \mathit{AreaType})$ is terminated when there is a communication gap, i.e., when Vl stops transmitting its position. In this case, we are uncertain of the vessel's whereabouts. Using rules (1)-(3), RTEC computes, with the use of application-independent rules, $\mathsf{holdsFor}(\mathit{withinArea}(\mathit{Vl}, \mathit{AreaType}) = \mathsf{true}, I)$, i.e., the list of maximal intervals I during which Vl is in $\mathit{AreaType}$. ◊

Definition 2 (Syntax of Rules Defining Simple FVPs)**.** Consider a simple FVP $F = V$. The $\mathsf{initiatedAt}(F = V, T)$ rules of the event description have the following syntax:

$$\begin{aligned}&\mathsf{initiatedAt}(F=V,\ T) \leftarrow\\ &\quad \mathsf{happensAt}(E_1,\ T)[[, [\mathsf{not}]\ \mathsf{happensAt}(E_2,\ T),\ \dots,\\ &\quad [\mathsf{not}]\ \mathsf{happensAt}(E_n,\ T), [\mathsf{not}]\ \mathsf{holdsAt}(F_1=V_1,\ T),\ \dots,\\ &\quad [\mathsf{not}]\ \mathsf{holdsAt}(F_k=V_k,\ T)]].\end{aligned}$$

The first body literal of an initiatedAt rule is a positive happensAt predicate; this is followed by a possibly empty set, denoted by '[[ ]]', of positive/negative happensAt and holdsAt predicates. 'not' expresses negation-by-failure [2], while '[not]' denotes that 'not' is optional. All (head and body) predicates are evaluated on the same time-point T. The bodies of terminatedAt$(F=V, T)$ rules have the same form. ■

A statically determined FVP $F=V$ is defined via a rule with head holdsFor$(F=V, I)$, which computes maximal interval during which $F=V$ holds continuously based on the maximal intervals of other FVPs.

Example 2 (Anchored and moored vessels)**.** Consider the following example from maritime situational awareness:

$$\begin{aligned}&\mathsf{holdsFor}(\mathit{anchoredOrMoored}(\mathit{Vl})=\mathsf{true}, I) \leftarrow\\ &\quad \mathsf{holdsFor}(\mathit{stopped}(\mathit{Vl})=\mathit{farFromPorts}, I_{sf}),\\ &\quad \mathsf{holdsFor}(\mathit{withinArea}(\mathit{Vl}, \mathit{anchorage})=\mathsf{true}, I_a),\\ &\quad \mathsf{intersect\_all}([I_{sf}, I_a], I_{sfa}),\\ &\quad \mathsf{holdsFor}(\mathit{stopped}(\mathit{Vl})=\mathit{nearPorts}, I_{sn}),\\ &\quad \mathsf{union\_all}([I_{sfa}, I_{sn}], I).\end{aligned} \tag{4}$$

anchoredOrMoored(*Vl*) is a Boolean statically determined fluent; it is defined in terms of three other FVPs: $\mathit{stopped}(\mathit{Vl})=\mathit{farFromPorts}$, $\mathit{stopped}(\mathit{Vl})=\mathit{nearPorts}$ and $\mathit{withinArea}(\mathit{Vl}, \mathit{anchorage})=\mathsf{true}$. The multi-valued fluent *stopped*(*Vl*) expresses the periods during which vessel *Vl* is idle near some port or far from all ports. The specification of this fluent is available with the complete event description of maritime situational awareness[1]. Rule (4) derives the intervals during which vessel *Vl* is both stopped far from all ports and within an anchorage area, by applying the intersect_all operation on the lists of maximal intervals I_{sf} and I_a. The output of this operation is list I_{sfa}. Subsequently, list I is derived by applying union_all on lists I_{sfa} and I_{sn}. In this way, list I contains the maximal intervals during which vessel *Vl* has stopped near some port or within an anchorage area. ◊

Definition 3 (Syntax of Rules Defining Statically Determined FVPs)**.** The definition of statically determined FVP $F=V$ is a rule that has the following syntax:

$$\begin{aligned}&\mathsf{holdsFor}(F=V,\ I_{n+m}) \leftarrow\\ &\quad \mathsf{holdsFor}(F_1=V_1,\ I_1)[[, \mathsf{holdsFor}(F_2=V_2,\ I_2),\ \dots\\ &\quad \mathsf{holdsFor}(F_n=V_n,\ I_n), \mathsf{intervalConstruct}(L_1,\ I_{n+1}),\ \dots\\ &\quad \mathsf{intervalConstruct}(L_m,\ I_{n+m})]].\end{aligned}$$

The first body literal of a holdsFor rule defining $F=V$ is a holdsFor predicate expressing the maximal intervals of an FVP other than $F=V$. This is followed by a possibly empty list, denoted by '[[ ]]', of holdsFor predicates and interval manipulation constructs, expressed by intervalConstruct. intervalConstruct(L_j, I_{n+j}) may be union_all(L_j, I_{n+j}), intersect_all(L_j, I_{n+j}) or relative_complement_all(I_k, L_j, I_{n+j}). I_k, where $k < n+j$, is a list of maximal intervals appearing earlier in the body of the rule, and list L_j contains a subset of these lists. The output list I_{n+m} contains the maximal intervals during which $F=V$ holds continuously. ■

union_all(L, I) (resp. intersect_all(L, I)) computes the list of maximal intervals I as the union (intersection) of all lists of maximal intervals of list L. relative_complement_all(I', L, I) computes the list of maximal intervals I by removing from the maximal intervals of list I' all interval segments included in an interval of some list in L.

Reasoning. The key reasoning task of RTEC is the computation the maximal intervals of the FVPs in the event description. For a statically determined FVP $F=V$, RTEC derives the list of maximal intervals I of $F=V$ by evaluating the conditions of the rule with head holdsFor$(F=V, I)$. For a simple FVP $F=V$, which is defined via a set of initiatedAt and terminatedAt rules, RTEC operates as follows. First, RTEC computes the initiations of $F=V$. If there is at least one initiation, then RTEC computes all time-points where $F=V$ is 'broken', i.e., $F=V$ is terminated or F is initiated with a value other than V. These are the terminations of $F=V$. Subsequently, RTEC computes the maximal intervals of $F=V$ by matching each initiation T_s of F=V with the first termination T_e of F=V after T_s, ignoring every intermediate initiation between T_s and T_e. RTEC may then derive holdsAt$(F=V, T)$ by checking whether T belongs to one of the maximal intervals of F=V.

RTEC employs a simple caching mechanism to avoid unnecessary re-computations, according to which the FVPs of an event description are processed in an order specified by its dependency graph. RTEC processes FVPs in a bottom-up manner, computing and caching their intervals level-by-level. This way, the intervals of the FVPs that are required for the processing of a FVP of level n are fetched from the cache without the need for re-computation.

3 Method

Our goal is to construct RTEC event descriptions with the use of LLMs. We compare these event descriptions with hand-crafted event descriptions in RTEC, constructed by domain experts, using a similarity metric. We employed a similarity metric for event descriptions, which is an extension of the similarity metric for collections of ground atoms used in [6] We define the simil

Our goal is to evaluate the event description

Definition 4 (Ground Atom Distance)**.** ■

Definition 5 (Rule Distance)**.** ■

Based on Definition 5, we handle non-ground atoms as follows...

Definition 6 (Event Description Distance)**.** ■

The similarity of two event descriptions with distance d is $1-d$.

4 Experimental Evaluation

5 Conclusion

Acknowledgments

Thanks to...

References

[1] Alexander Artikis, Marek J. Sergot, and Georgios Paliouras. 2015. An Event Calculus for Event Recognition. *IEEE Trans. Knowl. Data Eng.* 27, 4 (2015), 895–908.

[1] https://github.com/aartikis/RTEC

[2] Keith L. Clark. 1977. Negation as Failure. In *Logic and Data Bases*. Plemum Press, 293–322.
[3] R. Kowalski and M. Sergot. 1986. A Logic-Based Calculus of Events. *New Gen. Computing* 4, 1 (1986), 67–96.
[4] Periklis Mantenoglou, Dimitrios Kelesis, and Alexander Artikis. 2023. Complex Event Recognition with Allen Relations. In *KR*. 502–511.
[5] Periklis Mantenoglou, Manolis Pitsikalis, and Alexander Artikis. 2022. Stream Reasoning with Cycles. In *KR*. 544–553.
[6] Evangelos Michelioudakis, Alexander Artikis, and Georgios Paliouras. 2019. Semi-supervised online structure learning for composite event recognition. *Mach. Learn.* 108, 7 (2019), 1085–1110.
[7] Georgios M. Santipantakis, Akrivi Vlachou, Christos Doulkeridis, Alexander Artikis, Ioannis Kontopoulos, and George A. Vouros. 2018. A Stream Reasoning System for Maritime Monitoring. In *TIME*, Vol. 120. 20:1–20:17.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009