Presentation Proposal: 'Situation Recognition in Complex Operational Domains using Temporal and Description Logics – A Motivation from the Automotive Domain'

Lukas Westhofen[1][0000−0003−1065−4182], Christian Neurohr[1][0000−0001−8847−5147], Daniel Neider[2][0000−0001−9276−6342], and Jean Christoph Jung[2][0000−0002−4159−2255]

[1] German Aerospace Center (DLR e.v.)
Institute of Systems Engineering for Future Mobility
Escherweg 2, Oldenburg
`{lukas.westhofen,christian.neurohr}@dlr.de`
[2] Technical University Dortmund
August-Schmidt-Straße 1, Dortmund
`{daniel.neider,jean.jung}@tu-dortmund.de`

Abstract. When confronted with automated systems operating in highly complex domains, such as urban road traffic, situations relevant to the automation must be recognized in data. Our proposed presentation makes a case for temporal querying over expressive description logics as a suitable solution to this task. To highlight its benefits, we contribute a practical motivation for temporal querying from the field of scenario-based assessment of automated driving systems. We identify desired properties of such temporal queries and devise a tailor-made language for them, based on top of Mission-Time Linear Temporal Logic and Conjunctive Queries. Finally, we present a summary of our ongoing work regarding an implementation of this language.

Keywords: Temporal Conjunctive Queries · Description Logics · Situation Recognition · Automated Systems.

1 Safety-Critical Automated Systems in Complex Domains

Recent technological advances in, e.g., sensors and computer vision, gave updraft to the development of automated systems performing safety-critical tasks in complex domains. These systems are expected to operate safely without human intervention in these contexts. Consider, for example, automated driving systems (ADSs), where the responsibility of navigating the environment safely lies fully with the system [12]. However, the combination of their safety-critical nature and the complex operational domain makes it hard to guarantee the absence of unreasonable risks. To understand this complexity, we again consider the example of ADSs, which are confronted with a variety of concepts including legal terminology – what is an on-ramp? what is the meaning of yellow line markings and traffic signs? – and a multitude of entity types – a pedestrian does not behave the same as a bicyclist, and a sports car may be reacted to differently than a police vehicle. A precise model of these concepts is required due to their implications on the system's behavior. This is essential both for an increased understanding of the engineer at design time and the system's situational awareness at run time. For example, the system may be expected to safely yield for oncoming police vehicles while ignoring yellow line markings but still considering actions of nearby pedestrians and bicyclists. As we highlight in this talk, ontologies are a suitable tool for this task.

2 Description Logics for Modeling Complex Domains

In computer science, an ontology represents a formalized conceptualization of a domain of discourse and is an established means in systems engineering. A well-examined family of formalisms to specify ontologies are Description Logics (DLs) [3]. The central concept in DLs is a so-called knowledge base $\mathcal{K}$, consisting of a TBox $\mathcal{T}$ modeling the background knowledge (an ontology) and an ABox $\mathcal{A}$ with assertional knowledge on the environment. In our setting, DLs enable semantic inference and querying on perceptions of the system's environment that is modeled as an ABox. For example, a perceived pedestrian $\mathtt{p}$ is asserted as $\mathtt{Pedestrian(p)} \in \mathcal{A}$. The ontology can then assert that each pedestrian is a vulnerable road user (VRU): $\mathtt{Pedestrian} \sqsubseteq \mathtt{VRU} \in \mathcal{T}$, enabling the inference of $\mathtt{VRU(p)}$.

However, any perception is only partial w.r.t the actual state of the world due to occlusions on the electromagnetic spectrum of the sensor, technical failures, or missing labels from the employed classification algorithm. Despite this partial information, we still aspire semantic reasoning without drawing incorrect conclusions. For this, DLs incorporate the so-called *open world assumption* (OWA), a distinguishing feature in contrast to the closed world database setting. For example, if we perceive a bicyclist $\mathtt{Bicyclist(b)} \in \mathcal{A}$ with the previous background knowledge, we can not infer that $\mathtt{b}$ is not a $\mathtt{VRU}$.

When tasked with accurately modeling complex domains, however, having only simple subsumption axioms in an OWA setting are insufficient. For this, we require more expressive DLs and their powerful inferences. For our ADS example, constraints of roles as well as intersection and union of concepts are vital: Using this expressiveness, we can model two-lane roads to have exactly two lanes (by the role constraint $=2\,\mathtt{has\_lane.Lane}$) and be a road (by the concept intersection $\mathtt{Road} \sqcap {=2}\,\mathtt{has\_lane.Lane}$). Moreover, parking vehicles are standing dynamical objects being on a parking spot. In a DL ontology, this is expressed by the following axioms:

- $\mathtt{2\_Lane\_Road} \equiv \mathtt{Road} \sqcap {=2}\,\mathtt{has\_lane.Lane}$
- $\mathtt{Vehicle} \sqcap \mathtt{Standing\_Dynamical\_Object} \sqcap$
  $\exists \mathtt{intersects.Parking\_Spot} \sqsubseteq \mathtt{Parking\_Vehicle}$

- $\texttt{Parking\_Spot} \equiv \texttt{Parking\_Lane} \sqcup \texttt{Walkway}$
- $\texttt{Standing\_Dynamical\_Object} \equiv$ $\texttt{Dynamical\_Object} \sqcap \exists\texttt{has\_speed}.\{0.0\}$

We specifically highlight the power of the OWA to enable iterative specifications. For example, we specified a sufficient condition for parking vehicles based on dynamical objects, albeit there may be parking vehicles that are no dynamical objects. However, at the time of specification we can safely ignore this matter, as, per the OWA, a thing can not be inferred to not be a parking vehicle just because it can not be inferred that it is one. Axioms such as the inclusion of broken vehicles might be added in later development phases.

This complex background knowledge can now be used to query for the presence of certain situations in data. For such a situation recognition task, *conjunctive queries* (CQs) were found to strike a balance between applicability and performance. A CQ is a set of questions for which an answer has to be found for a given knowledge base. In our simple example, we can ask for all VRUs by $\texttt{VRU}(x)$, returning $x \mapsto \texttt{p}$ as an answer. CQs also allow quantification of variables, e.g., $\exists x.\ \texttt{VRU}(x)$, whose existence may only be implied by existential quantification $\mathcal{T}$.

3 Temporal Queries over Description Logic Ontologies

DLs do not only provide inference capabilities over 'static' observations but can also be leveraged on a temporal level, which is of specific relevance for automated systems interacting with their environment over time. One temporal reasoning mechanism are *temporal conjunctive queries* (TCQs) over a general TBox and a sequence of ABoxes. These queries are similar to Temporal Logic (TL) formulae, e.g., Linear TL (LTL), but allow for CQs in place of atomic propositions. For TCQs based on future-time LTL, we can use the modalities $\Diamond\varphi$ (φ holds *eventually*), $\Box\varphi$ (φ holds *globally*), $\varphi_1\,\mathcal{U}\varphi_2$ (φ_1 holds *until* φ_2), and $\bigcirc\varphi$ (φ holds in the *next* step). For example, $\Diamond\texttt{VRU}(x)$ asks for all things that are VRUs in some ABox. As a more intricate example, a temporal situation to recognize may be the passing of parking vehicles on two-lane roads due to possibly occluded pedestrians trying to pass the road, as the distance to the other road side is small. A TCQ asking for all things x that move past a parking vehicle y on some two-lane road is given by

$$\begin{gathered}\Box(\exists r.\texttt{Vehicle}(x) \wedge \texttt{2\_Lane\_Road}(r) \wedge \\ \texttt{intersects}(x,r)) \wedge \Diamond(\texttt{is\_in\_front\_of}(y,x) \wedge \\ \bigcirc((\texttt{Parking\_Vehicle}(y) \wedge \texttt{in\_proximity}(x,y) \wedge \\ \texttt{to\_the\_side\_of}(y,x))\,\mathcal{U}\texttt{is\_behind}(y,x))))\end{gathered}$$

For answering this query on a given observation, one has to rely on inferences based on the background knowledge, including axioms on two-lane roads and parking vehicles.

The community has recognized the potential of such queries: the theoretical setting of TCQs over expressive DLs and past- and future-time LTL was examined by Baader et al. [2]. It turns out that answering TCQs is of high computational complexity. Although there is tooling for TCQs over so-called lightweight DLs, i.e., fragments with rather low expressiveness, to the best of the authors' knowledge, no tooling currently exists for TCQ answering in expressive DLs. Here, we see ample opportunity for researching practical approaches, as such a tooling can enable the much-needed situation recognition capabilities for analyzing complex operational domains.

4 A Practical Temporal Query Language

Our goal is hence to move TCQ answering to practice. For this, we require a TCQ language with features providing benefit to users in said application. For finding a suitable formalism, we again turn to the example of automated driving. Here, situation recognition is needed both at design time, e.g., for data analysis, and at run time, e.g., for decision-making. We focus on the first, where engineers are tasked with designing a safe and efficient automation.

For ADSs, the so-called *scenario-based approach* has emerged as an engineering tool for designing, developing, and safeguarding [11,8]. Here, the complex domain is decomposed into a limited set of scenario classes, where a scenario is a finite temporal evolution of snapshots of the environment. Scenario classes are then instantiated to finite time traces using simulation or real-world data, on which situation recognition is performed. Such a situation recognition is a key tool during the design and development process as well as operation. For example, risk quantification [10] requires data analysis of safety-critical factors, such as potential triggering conditions of hazardous behaviors[9,5]. These factors include *occlusions* [15], violating the *safety distance* [16], and critical maneuvers such as *cut-ins*.

We arrive at the following required features for a TCQ language in our domain:

(1) The TL has to operate only on finite traces as the operational domain is decomposed into finite 'chunks'.
(2) Duration constraints are often required during specification, e.g., to distinguish actions of certain lengths. Therefore, the TL has to allow for metric operators.
(3) As we assume the time traces to be recorded and analyzed a-posteriori, we are not in a run-time verification setting and do not require both past and future time TL operators, in contrast to situation awareness at run time.
(4) The DL has to offer a strong expressiveness, as we benefit from strong inferences on the domain model and have loose computational constraints.

5 A Proposal for MTCQs

A TCQ language with these features is a much-needed advancement in bringing theoretical foundations on temporal queries over complex operational domains into practice. We plan to contribute to this advancement by lifting the work of Baader et al. on querying expressive DLs [2] from LTL to Mission-Time LTL (MLTL) [6] with an unconstrained until. MLTL is a suitable candidate for features (1) to (3) and is similar to LTL on finite traces but replaces until with an interval-constrained version, i.e., we allow for the typical future-time LTL operators until ($\varphi_1 \mathcal{U} \varphi_2$) and next ($\bigcirc\varphi$), but add an interval-constrained variant for until by $\varphi_1 \mathcal{U}_{[a,b]} \varphi_2$ for $a, b \in \mathbb{N}$. This variant means that somewhere in $[a, b)$ φ_2 has to hold, and for all time steps from a on is preceded by φ_1. We call the queries based on this TL *MTCQs*.

On the DL side, this approach satisfies feature (4) by working on all fragments for which consistency is decidable. However, we assume a tree-shape of the query graph, where nodes are the quantified variables and edges are the role atoms of the query. Otherwise, query answering may not be possible [4]. However, if the underlying DL has the finite- or tree-model-property, cycles can be broken by just replacing a quantified with a free variable. Moreover, the tree-shape violation has to occur in the quantified variables. Enforcing such cycles in unnamed individuals using TBox axioms is not a property particularly relevant for examining the complex, but still finite context of automated systems, which also justifies the acyclicity assumption. The assumption has thus a negligible practical impact.

Currently, developments on this proposed MTCQ language are ongoing, including an implementation of an answering engine for tree-shaped MTCQs in Openllet[3] as to close the identified gap of tooling availability. To showcase the performance of such tools, we aspire an evaluation of their efficacy and thus also develop a benchmark generator that allows an evaluation in as well as comparisons of our tool to future developments. We are looking forward to fruitful discussions on potential features, practical applications, and algorithmic performance improvements. We point out that specifically the latter is mandatory to bring the proposed approach from theory to practice.

References

1. Artale, A., Franconi, E.: A survey of temporal extensions of description logics. Annals of Mathematics and Artificial Intelligence **30**, 171–210 (2000)
2. Baader, F., Borgwardt, S., Lippmann, M.: Temporalizing ontology-based data access. In: Automated Deduction–CADE-24: 24th International Conference on Automated Deduction, Lake Placid, NY, USA, June 9-14, 2013. Proceedings 24. pp. 330–344. Springer (2013)
3. Baader, F., Calvanese, D., McGuinness, D., Patel-Schneider, P., Nardi, D.: The description logic handbook: Theory, implementation and applications. Cambridge university press (2003)

---

[3] https://github.com/Galigator/openllet/

4. Horrocks, I., Tessaris, S.: A conjunctive query language for description logic aboxes. In: AAAI/IAAI. pp. 399–404 (2000)
5. International Organization for Standardization: ISO 21448: Road vehicles – Safety of the intended functionality (2022)
6. Li, J., Vardi, M.Y., Rozier, K.Y.: Satisfiability checking for mission-time ltl. In: Computer Aided Verification: 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part II 31. pp. 3–22. Springer (2019)
7. Lutz, C., Wolter, F., Zakharyaschev, M.: Temporal description logics: A survey. In: 2008 15th International Symposium on Temporal Representation and Reasoning. pp. 3–14. IEEE (2008)
8. Neurohr, C., Westhofen, L., Henning, T., de Graaff, T., Möhlmann, E., Böde, E.: Fundamental Considerations around Scenario-Based Testing for Automated Driving. In: 2020 IEEE Intelligent Vehicles Symposium (IV). pp. 121–127 (2020). https://doi.org/10.1109/IV47402.2020.9304823
9. Neurohr, C., Westhofen, L., Butz, M., Bollmann, M.H., Eberle, U., Galbas, R.: Criticality Analysis for the Verification and Validation of Automated Vehicles. IEEE Access **9**, 18016–18041 (2021). https://doi.org/10.1109/ACCESS.2021.3053159
10. Putze, L., Westhofen, L., Koopmann, T., Böde, E., Neurohr, C.: On quantification for sotif validation of automated driving systems. In: Proceedings of the 2023 IEEE Intelligent Vehicles Symposium (IV). IEEE (2023)
11. Riedmaier, S., Ponn, T., Ludwig, D., Schick, B., Diermeyer, F.: Survey on scenario-based safety assessment of automated vehicles. IEEE access **8**, 87456–87477 (2020)
12. SAE International: J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles (2021)
13. Ulbrich, S., Menzel, T., Reschka, A., Schuldt, F., Maurer, M.: Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving. In: 2015 IEEE 18th international conference on intelligent transportation systems. pp. 982–988. IEEE (2015)
14. Wang, D., Hu, P., Wałega, P.A., Grau, B.C.: Meteor: practical reasoning in datalog with metric temporal operators. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 5906–5913 (2022)
15. Westhofen, L., Neurohr, C., Butz, M., Scholtes, M., Schuldes, M.: Using ontologies for the formalization and recognition of criticality for automated driving. IEEE Open Journal of Intelligent Transportation Systems **3**, 519–538 (2022)
16. Westhofen, L., Stierand, I., Becker, J.S., Möhlmann, E., Hagemann, W.: Towards a congruent interpretation of traffic rules for automated driving-experiences and challenges. In: Proceedings of the International Workshop on Methodologies for Translating Legal Norms into Formal Representations (LN2FR 2022) in association with the 35th International Conference on Legal Knowledge and Information Systems (JURIX 2022). pp. 8–21 (2022)