Event Detection and Diagnosis for Intelligent Transport Systems *

Patrik Schneider[1 2]
Advisor: Thomas Eiter[2]

[1] Siemens AG Österreich, Vienna, Austria
[2] Vienna University of Technology, Vienna, Austria

1 Introduction

The development of (semi)-autonomous vehicles requires extensive communication between vehicles and the infrastructure called V2X communication. This should allow to increase road safety, which is a major objective of Cooperative Intelligent Transport Systems (C-ITS), and can be achieved by analyzing traffic scenes in real-time and detecting events that could lead to accidents, e.g., red light violations [2]. Roadside C-ITS stations will support V2X communication with cars and the infrastructure such as traffic lights, but also could be extended for more complex tasks as traffic scene analysis. We illustrate the necessity of analyzing traffic scenes by two real-world scenarios, which are known problems in the field of C-ITS regarding safety and optimization [2].

The first scenario, called *road intersection safety*, was identified in [2], where the authors consider "road intersection monitoring" as an important application to improve road safety. For this scenario, we assume a complex intersection with a sensor-based roadside C-ITS stations, where traffic accidents happen frequently. The second scenario, called *changing traffic situations*, concerns the deployment and maintenance of these stations. Currently, a roadside station is configured once at deployment, hence it cannot react dynamically on a changing environment such as road construction or traffic jams due to a misconfiguration of signal phases. Smart roadside stations could become autonomous by dynamically adapting to the changing environment and traffic situations. For instance, they could recognize unoptimized signal phases and adjust the phases according to the new situation. Enabling the analysis of traffic scenes as in the above scenarios, includes more general characteristics such as dealing with the complex C-ITS domain, as well as handling large quantities of message-based and spatio-temporal data. This characteristics are not only relevant fo the C-ITS domain, but also exits in other fields such as robotics or geospatial analysis. Primarily, we have identified two different abstract levels of understanding for the analysis, each of them poses its own challenges as:

1. How do we efficiently analyze C-ITS streams for detecting short-term problems as (complex) events, e.g., accident detection;

* This thesis research is conducted within the project *LocTraffLog* (http://www.kr.tuwien.ac.at/research/projects/loctrafflog/) funded by the industrial PhD program of the FFG, in cooperation with Siemens AG Austria.

2. How do we diagnose complex problems, e.g., traffic-jams, which need a long-term observation span.

Each level can be seen independently and poses the additional challenge of intervening event detection and diagnosis in an efficient way. Another challenge arises from the nature of roadside C-ITS stations, which are (a) designed with limited memory and processing resources; (b) deployed in a distributed manner as a mesh network with a particular (spatial) topology. For enabling traffic scenes analysis, we aim to investigate tractable (lightweight) Knowledge Representation & Reasoning (KRR) methods for event detection and model-based diagnosis. The methods, namely rule- and ontology-based reasoning, have to be extended to streaming and spatial data taking a C-ITS domain model into account.

Section 2 describes state-of-the-art. Section 3 outlines the problem description, which is addressed by the research question and goals of Section 4 using the methods described in Section 5. Section 6 concludes with results and future work.

2 State-of-the-Art

V2X Communication and Integration. The base communication technologies (i.e., the IEEE 802.11p standard) allow wireless access in vehicular environments, which enables messaging between vehicles themselves and the infrastructure, called V2X communication. Traffic participants and roadside C-ITS stations broadcast every 100ms messages for informing others about their current state such as position, speed, and traffic light signal phases [2]. The main types of V2X messages are *Cooperative Awareness Messages* that provide high frequency status updates of a vehicle's position, speed, vehicle type, etc.; *Map Data Messages* that describe the detailed topology of an intersection, including its lanes and their connections; *Signal Phase and Timing Messages* that give the projected signal phases (e.g., green) for a lane; and *Decentralized Environmental Notification Messages* that inform if specific events like road works occur in a designated area.

The *Local Dynamic Map* (LDM) is a comprehensive integration effort of V2X messages; the SAFESPOT project [2] introduced the concept of an LDM as an integration platform to combine static geographic information system (GIS) maps with data from dynamic environmental objects (e.g., vehicles, pedestrians). This was motivated by advanced safety applications (e.g. detect red light violation) that need an "overall" picture of the traffic environment.

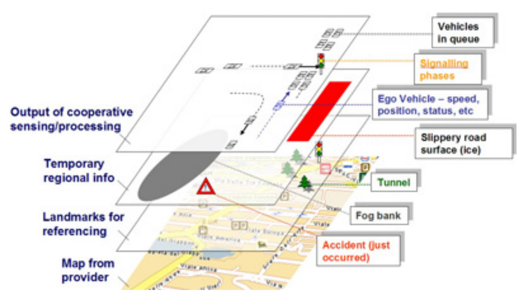

Fig. 1. The four layers of a LDM [2]

The LDM has the following four layers (see Fig. 1): *Permanent static* (static information from GIS maps); *Transient static* (detailed local information such as intersection features); *Transient dynamic* (temporary regional information like weather); *Highly dynamic* (dynamic information such as V2X messages). Current research, e.g., [20], on architectures of an LDM identified that it can be built on top of a spatial RDBMS enhanced with streaming capabilities. As recognized by [20], an LDM should be represented by a world model, world objects, and data sinks that allow the integration of the streamed V2X messages.

Stream-Processing, Complex Event Detection, and Stream-Reasoning. Stream reasoning studies how to introduce reasoning processes into scenarios that involve streams of continuously produced information. In that, domain models provide background knowledge for the reasoning and lift streams to a “semantic” level. Particular aspects of stream reasoning are incremental and repeated evaluation, either push-based, i.e. on data arrival, or pull-based at given points in time, and using data snapshots (called windows) to reduce the data volume. Windows can be obtained by selecting e.g. data based on temporal conditions (time-based windows), or data counts (tuple-based windows). Besides the seminal Continuous Query Language (CQL) [4], many formalisms and languages for stream reasoning exist. Among them are (1) extensions of the SPARQL web query language, e.g., Morph-streams [9], and CQELS [22]; (2) extensions of ontology languages to streams e.g. by Ren and Pan [25], and STARQL [21]; (3) rule-based formalisms e.g., ETALIS [3], Reactive ASP [16], Teymourian et al. [27], and LARS [6]; (4) usage of temporal operators such as LTL in [5], and [8].

Diagnosis. In A.I., diagnosis is one of the classical application areas and evolved around expert systems such as MYCIN. It can roughly be divided in the data-driven [19], [28], case-based reasoning [1], and model-based diagnosis approaches. Data-driven and case-based reasoning have the disadvantage of not guaranteeing sound and completeness for the diagnoses, which is crucial for traffic management because a predictable behavior is desired to ensure safety regulations, e.g., no conflicts in the signal phases. Model-based diagnosis can be further subdivided into the categories consistency-based diagnosis [26], [24] and abductive diagnosis [11], [23]. Both categories include observations O of a real system, a (fault) model of the real system S that simulates the predicted observation, and a list of system components C, which can be healthy or faulty. In contrast to consistency-based diagnosis, with abductive diagnosis the relation between causes and effects can be directly encoded in S. Since we focus on abductive diagnosis, a diagnose D is a subset of C, which is consistent with S and combined with S entails O. In the work of Lecue et al. [18], stream reasoning with description logics (namely $\mathcal{EL}$) has been applied to diagnose “quasi” real-time traffic congestions in Dublin using semantic matchmaking. In Khalastchi et al. [17], the authors developed new methods of model-based diagnosis for car accidents in an autonomous vehicle setting by reducing the problem into a SAT-based and a conflict-driven approach.

3 Problem Description

The motivating scenarios *road intersection safety* and *changing traffic situations* provide the setting, where we have identified theoretical and related technical problems. The problems are in the scope of lightweight systems with limited processor and memory resources. Hence, we limit ourselves to tractable KRR methods and techniques. We already have defined in [12] an integration layer with a DL-Lite [10] ontology representing the LDM and capturing its layers. We divide the problems into the topics listed below.

Event Detection. Event detection is crucial to filter safety relevant events like vehicle collisions, or traffic patterns like a traffic jam. It is closely tied to stream processing, since the events are filtered from different C-ITS data streams, where the integration layer allows us to map the streams to the ontology. Since, there is no query and ontology language yet suited for querying the streams that include spatial data, we face the following problems: Which data model and query language is suited for our streams and what "features"this language should include. For instance, what kind of window operators and aggregate functions could be applied. In case of DL-Lite, our extension might affect its computational properties as First-Order-Rewritability (FO-Rewritability). Further, how can we deal with missing and inconsistent data inside a window. Simple events might be detected by this new query language, but complex events (e.g., multiple-vehicle collisions), which are a chain of simple events satisfying specific temporal and spatial relations, are not captured. This might require more powerful methods related to stream reasoning.

Diagnosis. With event detection, we cannot find the cause for a traffic jam, merely the observations and aggregated facts are captured. Since we only consider model-based diagnosis, we aim to find a minimal set of multiple diagnosis models, where different combinations of faulty components might occur. Creating multiple diagnoses models could be computationally expensive, hence we need a techniques of iteratively calculating these models. Furthermore, we need to investigate how the quality of a model can be determined by allowing preferences. Another challenge arises from the complex domain model of C-ITS, which adds temporal, spatial, and ontological aspects to the diagnosis. For instance, one diagnosis model might be valid now, but already invalided at the next time point. For event detection a query answering based approach might suffice, but for diagnosis a rule-based language like ASP or Datalog might be better suited. However the mentioned languages have to be adapted according to the desired features, e.g., streaming data, but keeping desired computational properties as tractability.

Interaction between Event Detection and Diagnosis. Since the calculation of a diagnosis step is slower than updates in the C-ITS data streams, a two-level approach has to be applied, where the input data is first fed into event detection component for shallow reasoning to determine whether the changed data needs deeper reasoning on the second level, i.e. the full diagnosis. However, an efficient and simple interaction needs to been thoroughly investigated.

4 Research Questions and Goals

Research Questions. The overall research question is whether lightweight KRR methods and techniques are suitable for C-ITS applications such as traffic scene analysis. This question can be divided into the following sub-topics:

- Which tractable methods and techniques are suitable to extract (complex) events from C-ITS data streams having an elaborate domain model?

- Based on the detected events, how can model-based diagnosis be applied to find long-term problems in roadside C-ITS stations?
- How is an efficient interaction between both components feasible?

Following from the research questions, the goals of this thesis are as follows:

1. Goal: Event Detection for C-ITS. Based on the integration layer with an LDM ontology, we aim to work out an event detection component by extending DL-Lite for query answering over C-ITS streams. For this, we aim (a) to define a data model and query language suited for spatial data streams; (b) to extend the semantics for query answering with DL-Lite including window operators and aggregates over streams and spatial relations over spatial objects; (c) to provide a technique for query rewriting taking the above into account. The above results might not suffice for complex event detection, hence the language has to be extended to capture temporal relations as in LTL.

2. Goal: Model-based Diagnosis for C-ITS. We aim to develop a model-based diagnosis component that includes a clear definition and encoding of observations O, a (fault) model S, and a list of system components C applied to the C-ITS domain and in particular to roadside stations. The encoding has to capture the temporal, spatial, and ontological aspects of C-ITS. Based on the encoding, we aim to apply or extend standard rule-bases evaluation with an ASP or Datalog solver. We might encounter technical limitations, e.g., tractability, with standard techniques and might need an iterative approach for evaluation.

3. Goal: Component Integration. Since the event detection part must communicate changes in the requirements to the diagnosis component, a suitable interface and sharing of the domain is required. We aim to define the bidirectional interface between the event detection and diagnosis component, and design methods and techniques to realize the 2-level approach for shallow and deep reasoning, such that unnecessary diagnosis computations are avoided while necessary ones are correctly initiated.

5 Methodology

Since our overal goal regards lightweight KRR methods and techniques, our focus is on tractable rule- and ontology-based reasoning. As already mentioned, we start with DL-Lite$_A$ for the LDM ontology, which is the main language for ontology-based data access [10] (OBDA). First, we describe our research plan and its current progress, and give afterwards details on the steps:

1. We define a framework that includes all components such as the LDM ontology, the integration layer the query answering, and the diagnosis component. Further, we define the application scenarios and related benchmarks. This step is already finished and the results published in [12];
2. We work out the component for query answering over streams to allow event detection. This step is also finished and results are available in [14] and [15];
3. We develop the model-based diagnosis component and connect both components. This step has started and some outline of the ideas is given below;
4. We evaluate and implement the developed methods and techniques. This should lead to an initial prototype suited for C-ITS environments, where

the actual evaluation for each scenario will be conducted using the traffic simulation PTV Vissim.[3]

Query Answering over Streams. The query answering allows us a fast access to the streamed data in the LDM.

Example 1. The following query illustrates the component as it detects red-light violations on intersections by searching for vehicles y with speed above 30km/h on lanes x whose signals will turn red in 4s:

$$\begin{aligned}\mathsf{q(x,y)} : \ & \mathsf{LaneIn(x)} \land \mathsf{hasLocation(x,u)} \land \mathsf{intersects(u,r)} \land \ \mathsf{pos[line,4s](y,r)} \land \\ & \mathsf{Vehicle(y)} \land \mathsf{speed[avg,4s](y,v)} \land \ (\mathsf{v} > 30) \land \ \mathsf{isManaged(x,z)} \land \\ & \mathsf{SignalGroup(z)} \land \ \mathsf{hasState[first,-4s](z,s)} \land (\mathsf{s} = \mathsf{Stop})\end{aligned}$$

Query q exhibits the different dimensions that need to be combined: $\mathsf{Vehicle(y)}$ and $\mathsf{isManaged(x,z)}$ are ontology atoms that have to be unfolded with respect to the LDM ontology; $\mathsf{intersects(u,v)}$ and $\mathsf{hasLocation(x,u)}$ are spatial atoms, where the first checks spatial intersection and the second the assignment of geometries to objects; $\mathsf{speed[avg,4s](y,v)}$ resp. $\mathsf{pos[line,4s](y,r)}$ defines a window operator that aggregates the average speed resp. positions (as points) of the vehicles over the streams *speed* and *pos*; $\mathsf{hasState[first,-4s](z,Stop)}$ gives us the traffic lights that switch in 4s to the state "Stop".

For the evaluation of this query, we have to extend OBDA to handle spatial and streaming data, which is not considered in the standard approaches as [10]. In detail, we aim at answering pull-based queries at *a single* time point T_i with stream atoms that define *aggregate functions* on different windows sizes relative to T_i. For this, we consider a semantics based on *epistemic aggregate queries* (EAQ) over ontologies by dropping the order of time points inside a window and handle the streamed data items as *bags* (multi-sets). Roughly, we perform two steps, where we (1) calculate only "known" solutions, and (2) evaluate the rewritten query, which contains the rewritten TBox axioms, over these solutions. EAQs are evaluated over *temporal filtered and merged* sets of data items, called *windowed* ABoxes. The filtering and merging, relative to the window size and T_i, creates for each EAQ ϕ one *windowed* ABox $A_{\boxplus_\phi}$, which is the union of the static ABox A and the filtered streaming data items in the associated windows. The EAQ is then applied on $A_{\boxplus_\phi}$, which creates groups of aggregated normal objects, constant values, and spatial objects.

We introduce a *bag-based epistemic semantics* for the queries, so that we locally close our world for $A_{\boxplus_\phi}$ and avoid "wrong" aggregations due to the open world semantics of DL-Lite$_A$. For *normal objects* and *constant values*, we allow different aggregate functions such as *min*, *max*, *sum* on the data items of a stream. For *spatial objects*, geometric aggregate functions such as *point*, *line*, *poly* are applied, which create new geometries based on the aggregates.

Model-based Diagnosis. As already outlined, we will support model-based diagnosis as our method of choice. First, we need to define our roadside station and the related problems (e.g. the model of an accident) as a formal diagnosis problems with O, S and C. Observations O are taken from the detected events

[3] http://vision-traffic.ptvgroup.com/de/produkte/ptv-vissim/

via the bidirectional interface; C is defined as the list of system components that includes details on the actors involved in traffic scene such as traffic lights, vehicles, and the topological definition of an intersections. Further S is the model of a real system's behavior, which includes a definition of correct signal plans for traffic lights, driving behavior of cars, and traffic properties like accidents and traffic flow. A diagnosis D is then a subset of C, which is consistent with S and combined with S entails O. Then, we could compile the resulting diagnosis problem into a standard encoding for an ASP or Datalog solver. The results of Beck et al. [7] might be a good starting point for the encoding and compiling.

Combining Rules and Ontologies. Different approaches for combining rules and ontologies could be used for the interfacing. We focus on *loose coupling*, where the rule and ontology level are kept as separate, independent components, and an interface mechanism (guaranteed decidability on both sides) connects both components allowing the exchange of knowledge between them (e.g., [13]).

6 Conclusion

In this paper, we have presented a framework for allowing traffic scene analysis in C-ITS systems based on tractable KRR methods and techniques such as rule- and ontology-based reasoning. For the scene analysis, we have identified that we need a fast event detection and a model-based diagnosis component that have a common vocabulary based on an LDM ontology. Then, we have shown that event detection is feasible by extending OBDA with query answering over streams of spatial data. Further, the model-based diagnosis component was outlined, so we will be able to identify faulty components of a C-ITS system.

The first two steps are almost finished, however results regarding correctness and handling possible inconsistencies in query answering are desired. An initial prototype needs further development including optimizations and complex spatial aggregates. The third step is ongoing work that currently involves a clear definition and encoding of the model-based diagnosis problem including all involved components, which should allow us to encode and compute it with a ASP/Datalog solver. At the same time, we need to specify the bidirectional interface and work out the information flow between the components. Finally, we aim at evaluating and testing the integrated components using a traffic simulation under real conditions.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Commun. 7(1), 39–59 (1994)
2. Andreone, L., Brignolo, R., Damiani, S., Sommariva, F., Vivo, G., Marco, S.: Safespot final report. Tech. Rep. D8.1.1 (2010)
3. Anicic, D., Fodor, P., Rudolph, S., Stojanovic, N.: EP-SPARQL: a unified language for event processing and stream reasoning. In: WWW 2011. pp. 635–644 (2011)
4. Arasu, A., Babu, S., Widom, J.: The CQL continuous query language: semantic foundations and query execution. VLDB J. 15(2), 121–142 (2006)
5. Artale, A., Kontchakov, R., Kovtunova, A., Ryzhikov, V., Wolter, F., Zakharyaschev, M.: First-order rewritability of temporal ontology-mediated queries. In: Proc. of IJCAI 2015. pp. 2706–2712 (2015)

6. Beck, H., Dao-Tran, M., Eiter, T., Fink, M.: LARS: A logic-based framework for analyzing reasoning over streams. In: Proc. of AAAI 2015. pp. 1431–1438 (2015)
7. Beck, H., Eiter, T., Krennwallner, T.: Inconsistency management for traffic regulations: Formalization and complexity results. In: JELIA 2012. pp. 80–93 (2012)
8. Borgwardt, S., Lippmann, M., Thost, V.: Temporalizing rewritable query languages over knowledge bases. J. Web Sem. 33, 50–70 (2015)
9. Calbimonte, J., Mora, J., Corcho, Ó.: Query rewriting in RDF stream processing. In: Proc. of ESWC 2016. pp. 486–502 (2016)
10. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The *dl-lite* family. J. Autom. Reasoning 39(3), 385–429 (2007)
11. Console, L., Portinale, L., Dupré, D.T.: Focussing abductive diagnosis. AI Commun. 4(2/3), 88–97 (1991)
12. Eiter, T., Füreder, H., Kasslatter, F., Parreira, J.X., Schneider, P.: Towards a semantically enriched local dynamic map. In: In: Proc. of ITS World Congress on Intelligent 2016 (2016)
13. Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the semantic web. Artif. Intell. 172(12-13), 1495–1539 (2008)
14. Eiter, T., Parreira, J.X., Schneider, P.: Towards spatial ontology-mediated query answering over mobility streams. In: Proc. of SR Workshop 2016 (2016)
15. Eiter, T., Parreira, J.X., Schneider, P.: Spatial ontology-mediated query answering over mobility streams. In: Proc. of ESWC 2017. To appear (2017)
16. Gebser, M., Grote, T., Kaminski, R., Obermeier, P., Sabuncu, O., Schaub, T.: Stream reasoning with answer set programming: Preliminary report. In: Proc. of KR 2012 (2012)
17. Khalastchi, E., Kalech, M., Rokach, L.: A hybrid approach for fault detection in autonomous physical agents. In: Proc. of AAMAS 2014. pp. 941–948 (2014)
18. Lécué, F., Schumann, A., Sbodio, M.L.: Applying semantic web technologies for diagnosing road traffic congestions. In: Proc. of ISWC 2012. pp. 114–130 (2012)
19. Murray, J.F., Hughes, G.F., Kreutz-Delgado, K.: Machine learning methods for predicting failures in hard drives: A multiple-instance application. Journal of Machine Learning Research 6, 783–816 (2005)
20. Netten, B., Kester, L., Wedemeijer, H., Passchier, I., Driessen, B.: Dynamap: A dynamic map for road side its stations. In: Proc. of ITS World Congress 2013 (2013)
21. Özçep, Ö.L., Möller, R., Neuenstadt, C.: Stream-query compilation with ontologies. In: Proc. of AI 2015. pp. 457–463 (2015)
22. Phuoc, D.L., Dao-Tran, M., Tuán, A.L., Duc, M.N., Hauswirth, M.: RDF stream processing with CQELS framework for real-time analysis. In: Proc. of DEBS 2015. pp. 285–292 (2015)
23. Poole, D.: A methodology for using a default and abductive reasoning system. Int. J. Intell. Syst. 5(5), 521–548 (1990)
24. Reiter, R.: A theory of diagnosis from first principles. Artif. Intell. 32(1), 57–95 (1987)
25. Ren, Y., Pan, J.Z.: Optimising ontology stream reasoning with truth maintenance system. In: Proc. of ACM Conference on Information and Knowledge Management 2011. pp. 831–836 (2011)
26. Struss, P., Provan, G.M., de Kleer, J., Biswas, G.: Special issue on model-based diagnostics. IEEE Trans. Systems, Man, and Cybernetics (A) 40(5), 870–873 (2010)
27. Teymourian, K., Paschke, A.: Plan-based semantic enrichment of event streams. In: Proc. of ESWC 2014. pp. 21–35 (2014)
28. Weiss, G.M.: Mining with rare cases. In: Data Mining and Knowledge Discovery Handbook, 2nd ed., pp. 747–757 (2010)