Achieving Cooperative Sensing in Automotive Scenarios Through Complex Event Processing

Christoph Steup, Sebastian Zug, Jörg Kaiser
Department of Distributed Systems
Otto-von-Guericke-University
Magdeburg, Germany
Email: {steup, zug, kaiser}@ivs.cs.ovgu.de

***Abstract*—The actual trend towards more driver assisting systems in automotive scenarios induces a need for additional sensory data from the environment. Those perceptions can be generated by the inherent sensor systems or by external transducers. However, these data may be affected by faults and uncertainties, which may not be obvious to other cars receiving the data. Additionally, the management and dissemination of the data between the plethora of cars on the road is a big challenge. Therefore, this paper proposes an extension to existing complex event detection systems to support fault and uncertainty aware collaborative applications. An automotive scenario was chosen to illustrate the new approach. Consequently, the requirements of automotive scenarios were described and existing complex event systems were evaluated against these requirements. The extended system is called a complex event processing system, since it tries to enable sensor processing in an highly dynamic event system. To reach this goal additional attributes are introduced to the basic event schemes. Finally the processing steps are adopted to cope with the new quality attributes validity and uncertainty.**

***Keywords*—Autonomous vehicles; Command and control; Fault-tolerance; Sensor networks.**

I. Introduction

On of the important prerequisites for ubiquitous computing is the availability of environment data to allow context-aware computation and behaviour. Today, there is an enormous amount of this environment information potentially available, e.g., from traffic information infrastructure, floating car data, mobile devices and from instrumented smart spaces. Cooperatively perceiving environmental conditions and situations is a crucial component for improving the coordination of mobile entities like , e.g., team robots, autonomous transportation systems and cars. Recently, the use of autonomous air vehicles is considered that are partly controlled through collaborative sensing and coordination [1][2]. The locally perceived events of the environment need to be interpreted, related and combined to recognize more complex events and situations. The relations may be in the temporal domain , e.g., detecting that multiple subsequent events belong to a certain trajectory and in the spatial domain , e.g., predicting a jam or a collision situation of multiple vehicles. In this paper, we will describe our approach of complex event processing along a traffic scenario.

Actual approaches like PeerTIS by Rybicki et al. [3] or SOTIS by Wischoff et al. [4] try to enable collaboration between multiple cars. However, both only deal with the low level data dissemination between cars. To efficiently support collision warning systems, active collision avoidance and adaptive cruise control, a higher abstraction is needed since all cars will have different sensory equipment as well as different features regarding communication and automated control. An early example of a using floating car date to support an overtaking action is provided in [5]. In this example, Jini [6] is exploited to discover and use the front camera of a preceding car to see whether the road is free. Although using remote sensor information, this is a singular event establishing a unique channel between the cars. A promising approach for complex event detection systems emerged from the wireless sensor systems and the active database communities.

To evaluate the different approaches we will introduce an example scenario. It is shown in Figure 1. In this scenario some of the cars are able to communicate (C_{R1}, C_{R2} and C_{R2}) and others are not (C_{L0} and C_{R0}).

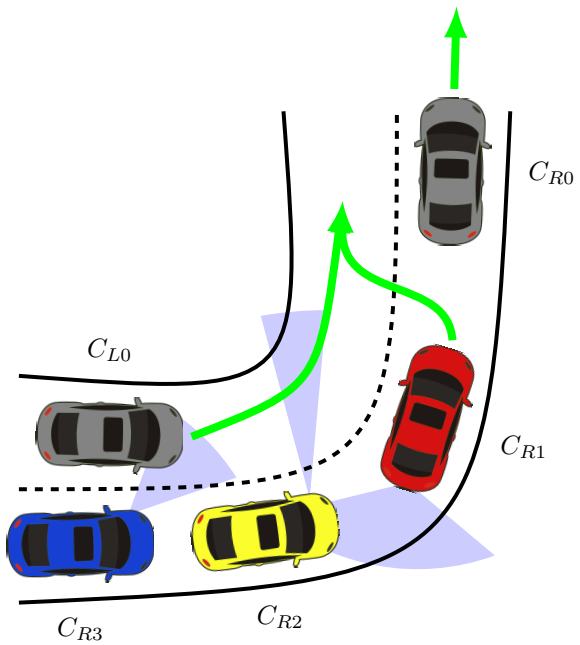

Fig. 1. Example of a dangerous situation in an automotive scenario. Car C_{R1} wants to change lane to overtake C_{R0} while car C_{L0} is driving with high speed on the left lane.

Car C_{R1} wants to overtake car C_{R0} and needs to change lane. However, this lane change is considered a dangerous manoeuvre, therefore C_{R1} wants to know if other cars may

 ISBN: 978-1-61208-289-9

collide during and after the manoeuvre. To detect possible collisions cars need to check their position against the position and speed of other cars. However, some cars cannot communicate and therefore their positions need to be observed and disseminated by the surrounding cars. In the example cars C_{R2} and C_{R3} might be able to detect the speed and the position of the endangered car C_{L0}. To achieve this they need to combine their individual position estimations of car C_{L0} to a speed estimation. The result can then be checked against the manoeuvre trajectory of car C_{R1}. Depending on the result a collision warning event may be issued.

This intuitive approach leads to an directed acyclic graph (DAG) of events and their combination. The event DAG of this example scenario is shown in Figure 2.

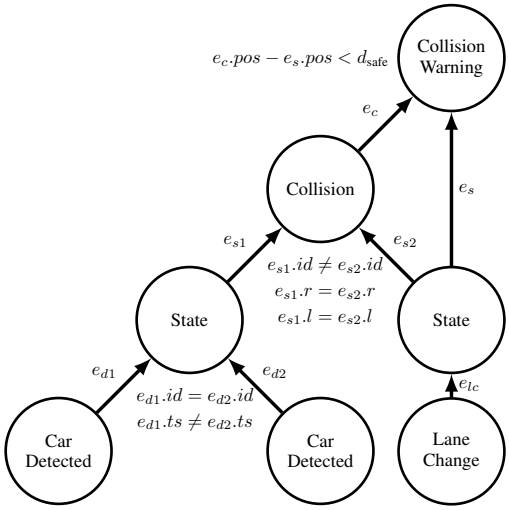

Fig. 2. The complex event DAG induced by the example scenario described in Section I. The nodes refer to types of events used in the systems. The links represent the individual named events travelling through the network.

Each event is a node in this graph. The nodes without any inputs are considered to be primitive events, which are generated by sensors. The top node represents the final event the application is interested in: the collision warning. To support the transition of the primitive events towards the final collision warning multiple combinations of intermediate events are needed. This leads to the definition of complex event detection systems (CEDS). These CEDS are researched mainly in the context of wireless sensor networks (WSN) and active databases.

The following Section II evaluates some typical system against this scenario. Afterwards we extend the CEDS through sensor processing descriptions to Complex Event Processing Systems(CEPS) in Section III . This Section is followed by an approach towards the distribution of the processing within the network in Section IV. The paper ends with a conclusion in Section V.

II. State of the Art

Generally, the existing complex event detection system can be categorized into query based systems and stream based system. The main difference is the considered use case. Query based system efficiently allow single queries to the network. Examples for these systems are Snoop by Chrakravarthy et al. [7] and Abstract Events by Katsiri et al. [8]. Snoop is limited to an active database context, which only supports centralized evaluation of the expressed complex events. However, the language imposed by Snoop is very rich. Abstract events on the other hand only support a very basic language based on temporal first order logic, but allow distribution within the network.

The stream based systems handle periodic events and therefore consider network issues like bandwidth and latency. Examples for these systems are Sensid by Krantz [9] and Solar by Chen et al. [10]. Sensid focus on wireless sensor networks with limited resources. Therefore, the expressiveness is limited and the evaluation of the complex event is centralized in one node. Solar on the other hand considers distributed detection of events and also specifies rules to drop specific events in case of overload. The description of the composition is XML-based, but limited to basic operations.

So far, complex event detection systems focus on the pure existence of events and neglect the contents or the specific attributes of the detected events. They introduce extensive languages to describe sets of events that generate the complex event. However, the generation of the attributes of the new event is almost always limited to basic operations like average, min or max. Considering the needed steps in the example scenario there are multiple steps that cannot be expressed efficiently with the described complex event detection systems. The state estimation of car C_{L0} for example needs to compute the current velocity based on two detection events. To do this, it needs to evaluate the position as well as the time between two events to approximate the velocity.

Another problem emerges from the fact that sensor readings are not perfect. This fact is usually not considered in the related work. Sensor data is affected by transient errors and uncertainties, which may have a varying impact on the sensor data. Some complex event detection systems like the one of O'Keeffe [11] tries to handle uncertainties by the definition of detection policies, which may influence the system decisions in uncertain situations. However, this system only considers timing uncertainties originating from network latencies and no value errors. Brade et al. [12] defined ways to express the trustworthiness of sensor data based on the error probabilities of the sensor. This can be exploited to extend the detection policies of O'Keeffe.

Liebig et al. [13] considered the problem of uncertainty in the timestamps of events. To cope with this uncertainty they expressed the time stamp as an interval and formulated appropriate algebraic operations to replace a simple time stamp in computations. However, the proposed mechanism can not be transferred to other data like positions or sensory values easily.

To overcome the described short comings of the existing systems, we propose the transition from a pure complex event detection system to a complex event processing system, which applies sensory processing steps onto the sensory inputs contained in the events. The following Section will describe our approaches towards a fault and uncertainty aware complex event processing for collaborative sensing.

 ISBN: 978-1-61208-289-9

III. Complex Events for Sensor Data Processing

As complex sensor events are used to represent arbitrary sensor data their structure needs to be appropriate and flexible. Sensor values are a representation of the current state of the environment and therefore are only valid for a limited time as described by Kopetz in [14]. The accuracy of these time-value entities drops over time. Therefore, the time stamp of the sensory information is essential in processing sensory inputs. As described by Liebig et al. [13] an interval representation of the time stamp is appropriate to convey this aging information over multiple nodes in the network. The resulting time stamp $[ts, \pm\alpha_{ts}]$ is defined through the approximate time stamp (ts) and the distance to the interval bounds (α_{ts}).

Most of the sensing equipment available today has a limited range as well as a limited observation area. These information is crucial for processing, because only sensors observing the correct area are useful. Therefore, representation of the observation area as well as the position and orientation of the sensor is needed to support remote processing of sensor data. Like the time representation, these values are subject to uncertainties since localization algorithms are limited in accuracy. The representation of position is dependant on the application. Sensor network applications may benefit from GPS-compatible 2D positions to directly convey their information to the internet. Whereas aeronautic applications need a position representation containing three dimensional coordinates and attitude.

In the case of the example scenario described in Section I, the position may be described as a triple of road segment (r), lane (l) and linear position within the segment (o). In this case, it is sufficient to describe only the offset as an interval with an uncertainty of α_o, because the road and lane can be deduced quite efficiently using maps and camera systems.

The data each sensor produces needs to be described as a general event scheme, since the set of available sensors is dynamic and not known on design-time. If we consider car C_{R3} to use a camera based system to identify overtaking vehicles, the resulting sensor data may be described through the unique identification number id of the vehicle. The identification needs to be deduced from the camera's picture in a variation of light conditions, which may impact the detection's effectiveness. Therefore, a validity value v representing the error probability based on the work of Brade et al. [12] needs to be attached to the value as well. For simplicity an uncertainty abstraction of car identifications is omitted at this point.

Putting together the descriptions of time, space and content we can formulate an event scheme of an identifying vehicle detector as:

$$E_{\mathrm{D}} = ([ts, \pm\alpha_{ts}], r, l, [o, \pm\alpha_o], id, v) \tag{1}$$

In consequence, a general event scheme for sensory data can be derived. The general scheme consists of a time stamp, which is described as an interval together with the position of the event. The position of an event can be derived from the sensor observation area and the current position of the sensor. Since localization mechanism are subject to a defined uncertainty, this value needs to be appended to the event scheme too. Additionally, the positions may be represented differently based on the application's scenario. The sensor's values are represented based on the sensor data sheet as described by IEEE standard 1451 [15] or the MOSAIC framework [16]. Finally, a validity value is attached representing the error probability within the sensor.

$$E_{\text{Sensor}} = ([ts, \pm\alpha_{ts}], [\text{pos}], [\text{sensor data}], v) \tag{2}$$

IV. Distributed Complex Event Processing

The combination of multiple input events towards a complex event needs a domain specific language as they have been introduced in Section II. However, as mentioned these languages lack the support for special sensor fusion operations. As an example, the collision warning system scenario described in Section I may be used. To detect a collision it is necessary to compute the intersection point of two cars extrapolating their current states (speed v_i and position o_i). The time to collision can be expressed as:

$$\Delta t_{\text{collision}} = \frac{o_1 - o_0}{v_1 - v_2} \tag{3}$$

All described systems are not able to compute such a new attribute value for the resulting event. Therefore, these schemes cannot express the concept of virtual sensors as proposed by the MOSAIC sensory middleware [16]. Additionally they consider the data of the events to be perfect, which may result in failures detecting an event.

A collision warning event needs to be issued whenever the time to collision is smaller then a defined safety time: $\Delta t_{\text{collision}} < t_{\text{safe}}$. However, variations of the speed of the two vehicles may have a huge impact on the space and time estimation of a detected collision. Therefore, these uncertainties may change the outcome of the detection.

In the described example, the important car C_{L0} had no mechanism to supply its current state to other cars, therefore the state of this car needed to be derived from other sensor readings. This deduction lowers the quality of the provided sensory data since the resulting speed (v_1 or v_2) may be susceptible to a higher uncertainty or in the case of unreliable detection, to lower validity. Through the provided validity information the impact of this deduction can be estimated and propagated from one processing step to another. Finally, our approach allows the application to describe its quality requirements in terms of validity or uncertainty predicates.

To supply the necessary validity and uncertainty values they need to be automatically determined during the processing of the individual events. To support this, we need to extend the processing description from a simple event set comparison to an applications specific function which handles uncertainties and fault probabilities. We consider the well-known Event-Condition-Action (ECA) rule mechanism with the action being an event generating function.

Such an ECA rule may be defined for the deduction of the speed of non-communicating cars as:

$$\{E_D, E_D\}, (e_0.id = e_1.id) \wedge (e_0.ts \neq e_1.ts) \rightarrow E_S : f_s(e_0, e_1) \tag{4}$$

This rule contains as input events two detection events E_D if both events detected a car with the same id $e_0.id = e_1.id$ and the timestamps of both detections is different $e_0.ts \neq$

$e_1.ts$. The result of this rule will be a state event of type E_S created through the application of function f_s to the input events (e_0, e_1).

The application of these rules within the network creates a directed acyclic graph of complex events processed to finally deliver the collision warning event. The resulting graph including the predicates of the combination operations is visible in Figure 2.

It is important to mention, that the resulting graph heavily depends on the current events available and their attributes. Especially if validity or uncertainty predicates are used, additional invocations of sensor fusion rules may be needed to increase the validity or decrease the uncertainty to pass the event filter.

A general ECA rule will accord the following pattern:

$$\{E_0, ..E_j, ..E_m\}, p_0[\vee|\wedge]p_i[\vee|\wedge]p_n \rightarrow E_k : f(e_0, .., e_j, .., e_m) \quad (5)$$

Therefore, the rule consist of a set of input events of type $\{E_0, ..E_j, ..E_m\}$ as well as some predicates p_i combined through either or ($\vee$) or and ($\wedge$). The resulting event will be of type E_k and is created through the function $f(e_0, .., e_j, .., e_m)$. The function itself is defined to be side-effect free, so that the repositioning of a processing step within the network only needs event dissemination.

There already exists some work on the effective positioning of processing nodes within a network. The CED system of O'Keeffe [11] tackles the problem for overlay networks using spring relaxation algorithms like the one of Pietzuch et al. [17]. However, these algorithms need to be tested in highly dynamic environments like automotive scenarios.

V. Conclusion

This paper proposes an extension of the complex event detection mechanism to support sensor driven collaborative applications. The requirements of automotive scenarios are described and existing complex event systems are evaluated against these requirements. To support fault and uncertainty awareness needed in these applications the event schemes and detection steps of classical complex event detection systems are extended, which lead to a new approach called Complex Event Processing.

Applications based on the new approach are able to express individual quality requirements against their input events. Contributing to this classical sensor fusion algorithms may be exploited in a distributed fashion. Related to the automotive scenario applications like CWS systems may enhance their performance through these quality attributes.

The algorithm of the classic complex event detection systems for WSN need to be evaluated against the requirements of the automotive domain. Especially the highly dynamic network topology and the short connection times between cars may impose additional challenges to the processing placement subsystem. The definition of a basic set of event schemes and combination operations for automotive scenarios lead to a domain specific language easing the development of applications of the automotive domain. Finally the described basic event scheme and ECA rule system needs to be evaluated against multiple scenarios to detect short comings in the expressiveness.

ACKNOWLEDGEMENTS

This work is (partly) funded and supported by the EU FP7-ICT program under the contract number 288195 "Kernel-based ARchitecture for safetY-critical cONtrol" (http://www.karyon-project.eu).

References

[1] D. Steitz, J. Anderson, and K. Rochon, May 2013, [retrieved: Jul, 2013]. [Online]. Available: http://www.nasa.gov/directorates/spacetech/centennial_challenges/uas/index.html

[2] A. Casimiro, "Karyon to improve safety and performance of smart vehicle coordination," Nov. 2011, [retrieved: Jul, 2013]. [Online]. Available: http://www.karyon-project.eu/wp-content/uploads/2011/11/KARYON_Press_Release_English.pdf

[3] J. Rybicki, B. Scheuermann, M. Koegel, and M. Mauve, "PeerTIS: a peer-to-peer traffic information system," in *Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking*, ser. VANET '09. New York, NY, USA: ACM, Sep. 2009, pp. 23–32.

[4] L. Wischoff, A. Ebner, H. Rohling, M. Lott, and R. Halfmann, "SOTIS - a self-organizing traffic information system," in *Vehicular Technology Conference 2003-Spring: Proceedings of the 57th IEEE Vehicular Technology Conference*, vol. 4, Apr. 2003, pp. 2442–2446.

[5] N. Nils Gura, A. Held, and J. Kaiser, "Proactive services in a distributed traffic telematics application," in *GI-Workshop: Mobile communication over wireless LAN:Research and applications*, Sep. 2001, pp. 585–592.

[6] K. Arnold, R. Scheifler, J. Waldo, B. O'Sullivan, and A. Wollrath, *Jini Specification*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.

[7] S. Chakravarthy and D. Mishra, "Snoop: An expressive event specification language for active databases," in *Data & Knowledge Engineering* , vol. 14, no. 1, Nov. 1994, pp. 1–26.

[8] E. Katsiri, J. Bacon, and A. Mycoft, "An extended Publish/Subscribe protocol for transparent subscriptions to distributed abstract state in sensor driven systems using abstract events," in *DEBS Proceedings of the International Workshop on Distributed Event-based Systems*, Edinburgh, Great Britain, May 2004, pp. 68–73.

[9] M. Kranz, "SENSID: a situation detector for sensor networks," Ph.D. dissertation, University of Western Australia, Jun. 2005.

[10] G. Chen, M. Li, and D. Kotz, "Design and implementation of a large-scale context fusion network," in *In First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous)*, 2004, pp. 246–255.

[11] D. O'Keeffe, "Distributed complex event detection for pervasive computing," University of Cambridge, Computer Laboratory, Tech. Rep. UCAM-CL-TR-783, Jul. 2010. [Online]. Available: http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-783.pdf

[12] T. Brade, J. Kaiser, and S. Zug, "Expressing validity estimates in smart sensor applications," in *ARCS International Conference on Architecture of Computing Systems 2013*, Jan. 2013.

[13] C. Liebig, M. Cilia, and A. Buchmann, "Event composition in time-dependent distributed systems," in *COOPIS Conference on Cooperative Information Systems: Proceedings*, Sep. 1999, pp. 70—78.

[14] H. Kopetz and K. Kim, "Temporal uncertainties in interactions among real-time objects," in *Reliable Distributed Systems, 1990. Proceedings., Ninth Symposium on*, 1990, pp. 165–174.

[15] Institute of Electrical and Electronics Engineers, "IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats," in *IEEE Std 1451.0-2007*, 2007, pp. 1–335.

[16] S. Zug, A. Dietrich, and J. Kaiser, "An architecture for a dependable distributed sensor system," in *IEEE Transactions on Instrumentation and Measurement*, vol. 60 Issue 2, Feb. 2011, pp. 408 – 419.

[17] P. Pietzuch, J. Ledlie, J. Shneidman, M. Roussopoulos, M. Welsh, and M. Seltzer, "Network-aware operator placement for stream-processing systems," in *ICDE Proceedings of the 22nd International Conference on Data Engineering*, 2006, p. 49.

 ISBN: 978-1-61208-289-9