EURO J Transp Logist (2018) 7:25–56
https://doi.org/10.1007/s13676-017-0108-5

RESEARCH PAPER

A disruption management system for automotive inbound networks: concepts and challenges

Anne Meyer[1] · Suad Sejdovic[1] · Katharina Glock[1] · Matthias Bender[1] · Natalja Kleiner[1] · Dominik Riemer[1]

Received: 25 January 2016 / Accepted: 8 April 2017 / Published online: 15 May 2017

Abstract Production processes in the automotive industry are highly dependent on reliable inbound logistics processes, because in lean production systems delays or mistakes often result in expensive interruptions of production processes. However, transport processes are always subject to unavoidable disturbances, delays, or mistakes. The goal of the research project ProveIT is to provide an IT system improving the transparency by monitoring transport processes in real-time: deviations from the transport plans are identified predictively, and classified dynamically as disruptions if they have negative impacts on the subsequent processes. If a disruption occurs, the operations managers are provided with mitigation actions automatically generated by escalation-based online optimization algorithms. In this contribution, we introduce the use cases, the architecture and main concepts of the ProveIT disruption management system, and report on challenges faced during field experiments with our application partners, Bosch, ZF, and Geis.

✉ Anne Meyer
meyer@fzi.de

Suad Sejdovic
sejdovic@fzi.de

Katharina Glock
kglock@fzi.de

Matthias Bender
mbender@fzi.de

Natalja Kleiner
nkleiner@fzi.de

Dominik Riemer
riemer@fzi.de

[1] FZI Forschungszentrum Informatik, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany

Keywords Real-time analytics · Complex event processing · Online optimization · Transport and logistics · Automotive inbound networks

1 Introduction

Logistical networks are always subject to disturbances, for example due to traffic jams or delays and mistakes in logistical processes. At the same time, production processes—especially in lean production networks of the automotive industry—are highly vulnerable and dependent on reliable inbound logistics processes, because otherwise deviations or errors extend to the production and result in expensive interruptions (see Wagner and Neshat 2010).

The key goal of the operations managers of all participants of the inbound networks—suppliers, logistics service providers and all involved departments of the receiving plant—is to avoid these interruptions. However, the involved parties only have a restricted view on the state of the network. More specifically, none of the parties is informed about buffers in terms of time, inventory or capacity by the other partners of the network. This results in expensive reactions to deviations taken by one partner in an isolated manner: Fig. 1 shows the details of a planned route of vehicle A collecting parts from three suppliers for a receiving plant. The arrival time at supplier 3 is—due to several traffic incidents—delayed and the estimated time of

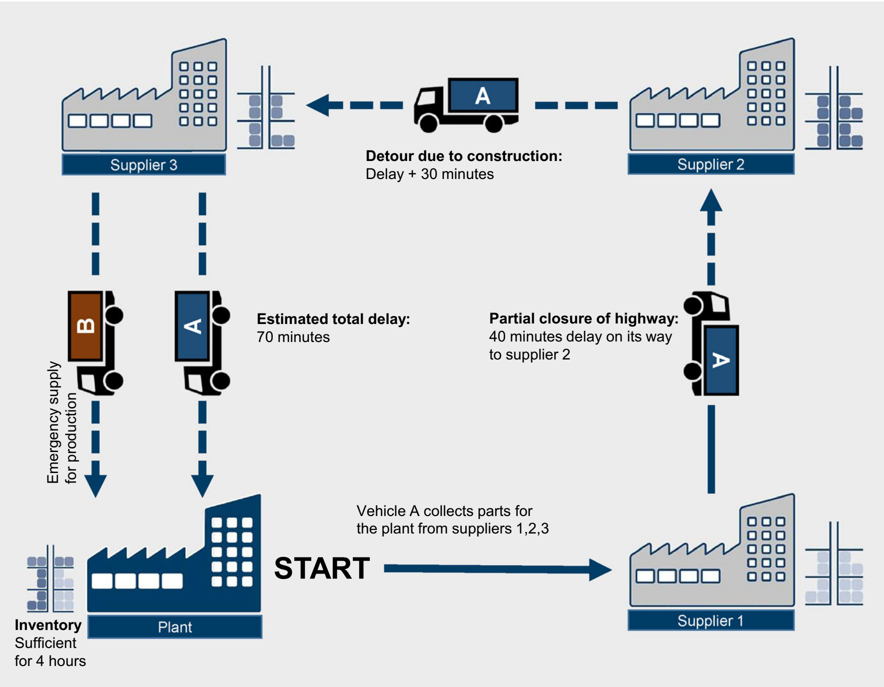

Fig. 1 Example for an expensive reaction of an operations manager of a logistics service provider due to missing transparency on buffers in terms of time or inventory

arrival (ETA) at the receiving plant is later than the agreed upper limit of the time window. The continuous calculation of the ETA at all planned, open (not yet visited) stops of the tour allows the early detection of delay and, hence, is the prerequisite for timely reactions. Without knowing any details about the freight, an operations manager of the logistics service provider decides to deploy vehicle B taking over the transport from supplier 3 to the receiving plant. This expensive special trip could be avoided if the operations manager knew that the estimated delay would not affect the production of the receiving plant since enough material is on stock.

The tracking and tracing applications of parcel services, such as DHL,[1] or for full container load shipments (see Baumgrass et al. 2014) show that there exist technologies to provide the recipients with more and more reliable ETAs in spite of geographically scattered processes. However, to profitably use these technologies in an inbound network of the automotive industry, continuously updated ETAs and other events describing the progress of the processes must be the basis for better informed disruption management decisions: that means, events must be transformed and aggregated to information for the operations manager of the different involved parties to achieve an end-to-end visibility (Boschian and Paganelli 2016). This is only possible if real-time and plan data considering the transports are shared across the involved companies so that deviations from the planned processes can be detected and assessed. If a deviation has a negative impact on subsequent processes, it is classified as a disruption, and the operations managers should be provided with automated recommendations on how to react efficiently. In Boschian and Paganelli (2016) such an IT-based disruption management service used commonly by the logistics service providers and shippers—in the automotive industry usually the receiving plant—is referred to as “cross-chain collaboration”.

From our experience, the main challenges for a disruption management system in practice are the following: (1) the involved companies—different suppliers, logistics service providers and their changing subcontractors, as well as the recipients—have very heterogeneous IT systems from which different types of data such as event data, plan data, and master data are relevant. (2) The business environment for the real-time analysis and the automated generation of recommendations is very complex and must be tailored to different, heterogeneous logistics concepts and to contractual agreements. As an example, the individually negotiated agreements with different logistics service providers often result in different responsibilities for mitigation actions in case of disruptions.

In order to tackle these challenges, the whole decision process must be addressed incorporating the following tasks: (1) gathering and integrating the relevant plan, master and event data from different parties; (2) analysing the event streams and classifying the deviations from the planned processes in real-time; (3) deciding on mitigation actions based on available information, and (4) in the end, distributing the relevant information to the point of execution. From a scientific point of view, this combines technologies from different communities: knowledge management, real-time stream processing, data analytics, and mathematical optimization.

[1] https://www.dhl.de/en/paket/information/sendungsverfolgung.html.

The goal of our research project, ProveIT, is to develop a (semi-)automated disruption management system for transport processes, and to evaluate its functionality with our application partners, two of the world's largest automotive suppliers Robert Bosch GmbH (Bosch) and ZF Friedrichshafen AG (ZF), as well as, the logistics service provider Geis Transport und Logistik GmbH (Geis). We start this contribution by introducing the two use cases of our application partners. Both use cases address automotive inbound networks, but they differ in the underlying logistics concepts. We give a short overview of the scientific work in this field before introducing the architecture and concepts of our disruption management system. Besides the overview on the overall system, we give a more in-depth introduction to the decision models addressed in the system and shortly evaluate the proposed heuristic on real-world instances. Following this, we give an impression of the technical and organizational challenges one faces in the practical application of the system. We conclude the paper with a short summary and an outlook to further research.

2 Use cases

In the German automotive industry three standard transport concepts exist for inbound traffic (see Projektgruppe 2008): point-to-point transports, area forwarding services, and milk runs. In a point-to-point transport, a large shipment is brought by a freight forwarder directly without further consolidation from a supplier to a plant. A subcontractor executes these transports. In case of area forwarding services a producing company outsources all smaller transport orders from a predefined geographical area—for example all suppliers of a ZIP code area—to a certain plant to a logistics provider. The recipient leaves it to this area forwarder to plan and execute these transports considering third party orders and consolidation centres of its own or of its network partners. The transports are split into pre-leg, main-leg and—if necessary sub-leg. Usually, a contractual run time of 2 days from pickup until delivery is arranged within Germany. The area forwarder uses its own fleet and the fleet of its network partners and subcontractors. In contrast, milk runs are regularly—for example daily—scheduled tours connecting a set of suppliers with a plant for guaranteeing reliable and transparent transport processes. In this case the planning responsibility lies with the recipient and the logistics service provider only executes the tour and informs the recipient about delays or other problems during the execution. The pickups and deliveries are often on the same day.

For evaluating the ProveIT disruption management system two use cases were selected. One use case covers the execution of a regional supplier milk run of a production site of Bosch in Homburg (UC1). The second use case addresses the area forwarding services of Geis for several production sites of ZF in Friedrichshafen (UC2). Point-to-point transports can be considered as a less complex, special case of area forwarding and, hence, are not further considered.

The operative planning steps before the execution of the tours are different for the two considered use cases. In a simplified way the steps can be described as follows (for more details see Stadtler 2015):

UC1: Milk runs (Bosch)

- Bosch: Electronic KANBAN cards sent to the respective suppliers implement a pull-based replenishment policy for the supplier plant relations served by the considered milk run. In this milk run, Bosch sends parts to the suppliers for refinement and receives parts for production. The milk run schedule is generated within a tactical planning step and is valid for several months.
- Logistics service provider (LSP): a milk run driver executes the daily milk run following the fixed schedule. The driver picks up and delivers all parts provided by Bosch and the respective suppliers.

UC2: Area forwarding (ZF/Geis)

- ZF: The operational material requirements are determined and transport lot sizes are built for the area forwarding services by the planning systems of ZF. The transport orders are communicated to Geis one day before the goods are picked up at the supplier sites.
- Geis: The transport orders of ZF and other customers are consolidated to pre- and main-leg tours in the morning before the scheduled pickups. The tours are communicated to a mobile app used by the driver of Geis or of a subcontractor.

Without the application of the ProveIT disruption system in UC1, Bosch has no information about the progress of the milk run tour except the driver or the responsible manager of the LSP calls the respective contact person at the Bosch site in Homburg. Only the GPS track of the vehicle is available and recorded at the LSP in order to be able to prove that the arrivals were on time. That means, a near real-time information of Bosch about mistakes and delays—especially on potential delays at forthcoming stops—fully depends on the skills and the experience of the driver.

In UC2 ZF has no information about the execution of area forwarding transports. The processes are a black box. The drivers of Geis or the respective subcontractor receive the tour information by their mobile applications and approve some execution information such as the arrivals, departures, and start and end of services. The GPS tracks are also communicated. However, estimated time of arrivals are not calculated automatically. In case of problems or severe delays, the operations manager at Geis is called, who tries to fix the problems manually for example by prompting an emergency supply with an extra vehicle. The area forwarder Geis calls the contact person at ZF only in case of problems which cannot be fixed.

The first step of the research project was dedicated to elaborating the situations of interest (SOI) along the defined processes for the different involved parties and for the different transport concepts. It was possible to categorize all situations of interest as deviations in time, quantity, and quality: deviations in time are all positive and negative deviations from the schedule of a tour and the corresponding planned pickups or deliveries. Deviations in quantity describe all positive and negative deviations from the ordered quantities, and deviations in quality such as damages of the loading aid or the goods themselves.

For historical records all deviations are of interest in order to calculate performance indicators, for example average standing times at a supplier, and improve the input data quality for future planning. However, for the operational disruption management addressed in this contribution, especially disruptions—or deviations with a negative impact for subsequent processes—are of interest. The way how deviations are classified as disruptions based on the state of the inbound network is described later in this work. However, it can be stated generally: the earlier the involved parties Bosch, ZF, and Geis are aware of a disruption, the greater is the available reaction scope—both for reacting within the transport network and the production. Furthermore, the aggregated information of smoothly running processes allows the operational manager to dedicate their attention to other tasks than trouble shooting.

3 Related work

The following subsections give an overview of existing works in the three primarily related fields. Section 3.1 examines the usage of real-time event processing in transport use cases; Sect. 3.2 addresses the topic from the supply chain management (SCM) perspective, whereas in Sect. 3.3 relevant contributions from the field of mathematical optimization are examined.

3.1 Real-time processing in transportation

The benefits of event processing for the real-time monitoring of transport processes were already perceived in previous works, e.g. Baumgrass et al. (2014), Cabanillas et al. (2014), Feldman et al. (2013) and Metzger et al. (2012). In all four works the authors describe extensive use cases where event processing plays a major role for the monitoring of the presented processes.

In Baumgrass et al. (2014) and Cabanillas et al. (2014) the authors describe the transportation control tower of the GET Service Project which is aimed to support transport planners for managing and monitoring their orders. The authors present an architecture consisting of a monitoring component for the subscription to related events, a component for modelling a transport process consisting of the tasks to be performed, and a component to dynamically subscribe to relevant events depending on the current task. The main advantage of the solution is the possibility to describe the transportation plan as a business process workflow and keep track of the current execution state by means of business process management systems enhanced with complex event processing.

In Feldman et al. (2013) and Metzger et al. (2012) the authors pursue another goal with the help of event processing: the prediction of situations of interest. The idea is to use real-time monitoring and proactive alerting to anticipate relevant parameters like the weight of a transport order. The use cases address air freight and the authors tackle the challenge of over-/underloaded cargo flights. In both works the authors describe the FInest platform, which contains an extended event processing engine to enable the deployment of probabilistic expressions. Predictions are based on the standardized data that are available through the IATA Cargo 2000 System.

The presented approaches describe promising systems, but rely on some prerequisites which are not given in our context: with respect to Feldman et al. (2013) and Metzger et al. (2012), our use cases from automotive industry have less standardized transport processes and might differentiate significantly at runtime. Thus, we do not have milestones valid for each kind of process and require much more detailed data about the tour execution. Regarding Baumgrass et al. (2014), we take up the idea of a transportation control tower and enrich it by more tour execution details. Also we add an optimization component to react to occurred situations. Therefore, we add another data source, namely the users involved in the execution of the transport process (e.g. the driver). This leads to a comprehensive end-to-end view on the process and stakeholders are enabled to choose the most appropriate view.

3.2 Supply chain event management

Disruptions are events that fall outside the (planned) norm. Supply chain event management (SCEM) describes new approaches and tools to detect such events by real-time monitoring of the supply chain processes and comparison of the observed events (e.g. RFID/Barcode scans during the loading process, GPS coordinates of the moving vehicle, etc.) with predefined events. If an event does not occur when expected or occurs when not expected, it is recorded and the impact of the deviation is evaluated. Events that characterize critical plan deviations are filtered out and reported to the user or any other affected supply chain participant in real-time (Baader and Montanus 2008).

One of the better known supply chain event management systems is SAP EM as part of the SAP SCM tool. With SAP EM supply chain processes can be monitored for exceptions (Diessner and Rosemann 2008). Each relevant process is represented by an object that includes all relevant milestones and expected events. The observed events are categorized, and a rule engine determines whether the affected supply chain participants are notified (e.g. by e-mail). If a logistics partner knows early enough whether to take action, the management of the logistics processes has a decidedly pro-active flavour.

Pro-active process management is also the objective of arvato services (Becker 2008), which is achieved by proactively sending messages (by e-mail or SMS) to best suited supply chain process participants in case of the appearance or absence of events. The services include a clear trend towards automatic actions like the generation of an electronic order if the original delivery time is by far exceeded. For the generation one may draw on suggestions from a knowledge base.

Zimmermann et al. (2006) use an agent-based approach for SCEM in order to achieve pro-activity and to maintain the autonomy of the supply chain partners. It defines monitoring criteria, gathers information on the criteria and interprets it. In case of an unexpected event the system generates alerts and directs them to the actors. These tasks as well as the particular supply chain participants are realized by different agents, e.g. the communication with supply chain partners is managed by discourse agents, while the gathering of information is realized by a surveillance agent.

These three SCEM tools are only a small selection of the existing approaches. Heinecke (2013) provides an overview and a brief comparison of further SCEM systems, while Behdani et al. (2012) review different solutions for disruption management, including several SCEM approaches. Both works also introduce frameworks for disruption management in supply chains with a focus on production processes.

While some of the existing SCEM systems use a set of predefined response plans for disruption reaction, we aim to use optimization algorithms together with the event data to generate recommendations for mitigation actions in real-time. There are a few SCEM systems reviewed by Behdani et al. (2012) and Heinecke (2013), which already include modules for disruption recovery by online rescheduling, but these approaches focus on production instead of transportation. Besides, as mentioned by Heinecke (2013), common rescheduling mechanisms in production as well as logistics always consider the whole transportation plan aiming to achieve optimality, which can result in multiple plan changes even for resources not affected by the disruption. In our approach we aim to generate best possible mitigating actions with fewest changes to the existing transportation plan by introducing an escalation model for the disruption recovery.

3.3 Decision models of intelligent freight-transportation systems

The operative planning process before the execution of the tour described in the preceding section—also referred to as offline planning problems—contains in both use cases vehicle routing decisions: in (UC2) the area forwarder decides on how a fleet of vehicles fulfils a set of transport requests cost efficiently considering a great variety of constraints. In literature these types of problems are known as vehicle routing problem (VRP) (see for example Laporte 2009). The milk run schedule in (UC1) is determined on a tactical level by the recipient Bosch considering additional supply chain management aspects such as inventory levels and operational complexity (see for example Meyer 2015, Schmid 2013). Mitigation actions determined automatically by so-called intelligent freight-transportation systems (ITS) (the term is introduced in Crainic et al. 2009) are generated by online planning approaches or, as referred to in literature, by solving dynamic vehicle routing problems: in contrast to the offline VRP the dynamic variant considers evolving information becoming available to the planner during the tour execution—for example, changing travel times due to changing traffic conditions and changing transport requests (orders). A recent, comprehensive review on dynamic vehicle routing problems along with a classification and a review on time-dependent vehicle routing problems is given in Gendreau et al. (2015) and Pillac et al. (2013), respectively. Contributions focussing on optimization models considering historical, real-time, and predicted travel time information, as well as papers describing the architecture of intelligent freight transportation system are provided. The authors of Wang et al. (2012) propose a recovery model for the vehicle routing problem with time windows (VRPTW), which follows the idea of disruption management as it is applied in other domains, e.g. in the airline industry. They take into account the different perspectives of the involved parties and—among other objectives—try to minimize the deviation of the driving paths from the original plan to keep disturbances on drivers at a minimum. Furthermore, they consider different types of disruptions and model them as so-called "new-adding customer disruption events". Their model is able to consider the simultaneous occurrence of several disruptions.

However, none of the introduced approaches addresses a disruption management system for inbound networks of a producing company: the ProveIT disruption management system not only considers a dynamic rescheduling of tours, but also incorporates mitigation actions such as shifting orders to the next day or to the next scheduled milk run if the inventory levels at the receiving plants allow for it.

The ProveIT disruption management system links approaches from supply chain disruption management and from the intelligent freight-transportation systems. That means, it combines the two perspectives on the transport processes: the perspective of the producing recipients focussing on the in time supply of material and of the logistics service providers taking care of an undisturbed tour execution. It provides the automatic generation of mitigation actions for all involved parties based on state-of-the-art real-time processing technologies.

4 ProveIT disruption management: architecture and concepts

Abstracting from the concrete application, we define four phases typically necessary to derive good data-driven decisions: gathering and integrating data, analysing the data, taking decisions and delivering the relevant information. A similar classification of phases of the information and decision-making process can be found in Heinecke (2013). After giving an overview of the architecture, we introduce the core functions of the ProveIT disruption management system following these four phases.

4.1 Overview of architecture

As Fig. 2 shows, the ProveIT disruption management system is structured into four layers to implement the above described phases. This structure enables us to be very flexible regarding the different transport processes in our use cases.

The layered architecture consists of the following components:

1. *Front end layer* The front end layer consists of two components responsible for the user interaction: the ProveIT cockpit and the ProveIT app, a smartphone application for Android smartphones. The cockpit is a web application for the monitoring of transport processes. For example it is possible to see the status for all currently active tours which includes all basic information of the tours, as well as the incoming real-time data. Additionally, the cockpit has a management view for the user management where users can be registered with different roles. The smartphone app is intended to be used by the driver during the tour execution. By means of the app users are able to track the

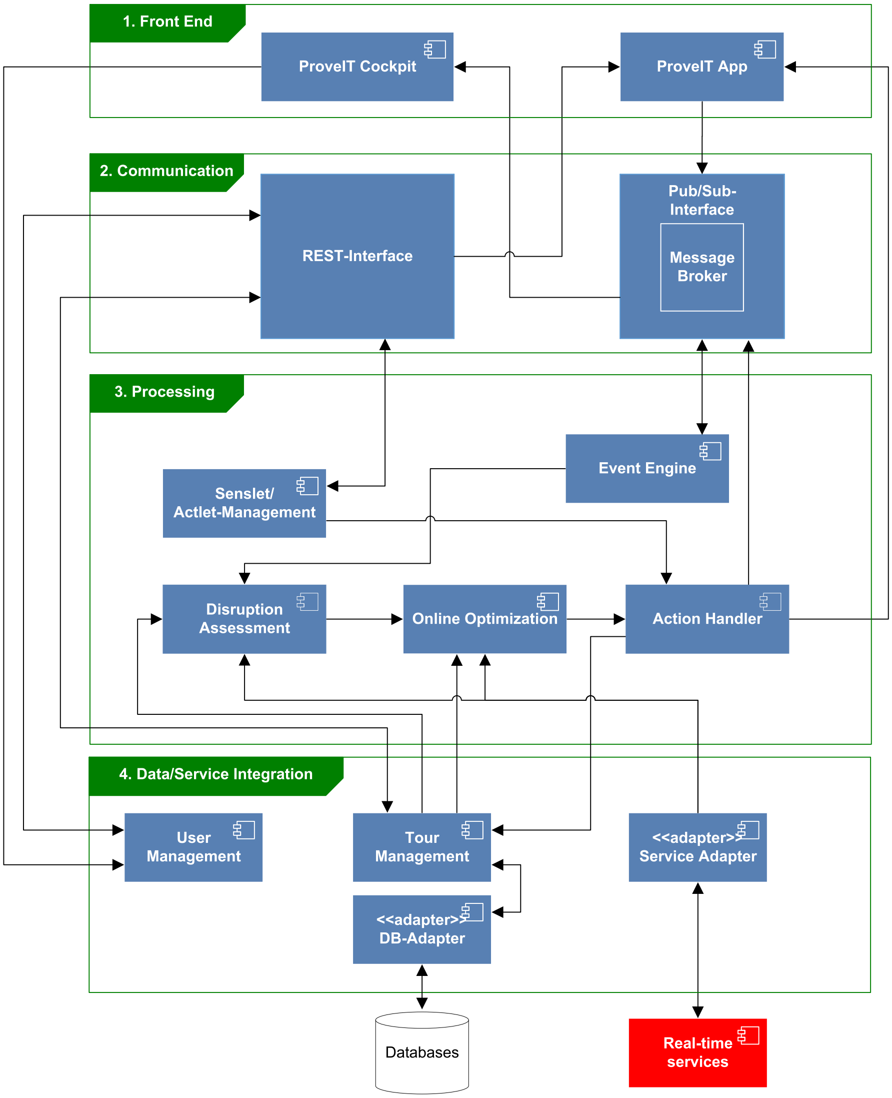

Fig. 2 Simplified overview of the layered architecture of the ProveIT disruption management system

transport process in any desired level of detail due to the flexibility of our senslet concept (details are described in Sect. 4.2).

2. *Communication layer* The communication layer contains two components handling most of the communication tasks. On the one hand, a REST interface serves as the central access point towards tour-related data like routes or user data. This interface is for example used by the smartphone app to pull tour-related information. On the other hand, a publish/subscribe message broker serves as the central access point towards real-time data. It is used by several

components to publish and consume real-time data like GPS updates, tour execution updates or analysis results.

3. *Processing layer* The processing layer consists of several components handling the data processing part of the disruption management system. Within the real-time processing part, an event-processing engine analyses the incoming data stream and identifies so-called situations of interest. The disruption assessment component assesses and classifies the occurred situations and triggers the online optimization component for generating mitigation actions in case of disruptions. The action handler sends instructions to the app instructing the user how to react to a identified situation. The senslet/actlet management component defines the abstract model of a senslet/actlet. A senslet represents a kind of software sensor that is highly configurable by business users to gather relevant data from users involved in the execution of the transport process, as well as, to gather automatically data like GPS updates. An actlet is based on the same principle, but is used to give users instructions with regard to the currently active tour.
4. *Data/service management layer* The fourth layer consists of components for the data management of different objects, as well as, adapters for externally allocated services and databases. The tour management component serves as the access point to tour data with a possible external data source, whereas the logistics services component gives access to external services, such as a tour planning service or a estimated time of arrival service. The role of the components within the four phases for data-driven decision-making is described in the following subsections.

4.2 Gather and integrate data

The ProveIT disruption management system gathers and integrates tour-related data from several different sources. The first part of this subsection is dedicated to describing relevant data categories and their sources, while the second part addresses a general concept for gathering event data by a mobile application.
Data categories and sources The relevant data for the ProveIT disruption management system can be divided into the three categories: plan and master data, event data, and data obtained from real-time services.

1. *Plan and master data* In this category, we subsume all information describing the *planned* execution of tours. Depending on the use case, the offline planning of tours is different: in (UC2) the logistics service provider Geis uses the tour planning solution xTour[2] of our project partner PTV Group (PTV) to plan the daily tours in the morning before the scheduled pickups. The resulting tours are communicated to the ProveIT disruption management system through a shared database. In (UC1) no tour planning is necessary. The milk run schedule and the transport orders for the day are merged to a concrete tour instance and these instances are communicated to the ProveIT disruption management system on

[2] http://xserver.ptvgroup.com/en-uk/home/ptv-xserver-en/.

the same way as in (UC2). Hence, in both cases, the results are available for the ProveIT disruption management system and comprise the same objects described in the following: a *tour* object contains highly aggregated information about a planned tour, e.g. the planned start and end time of the tour and its total distance. It contains several *stop* objects, each describing, amongst others, the planned sequence of the stop within the tour, its geographical coordinates, and the planned arrival and departure time. For each stop, in turn, there exist one or multiple *action point* objects. An action point object defines either the pickup or the delivery of goods at the stop and contains information about the quantity, weight, and volume of the goods to be transported. If data is provided, information on the level of *handling units*—referring in this case for example to concrete, identifiable pallets or other loading aids drivers are dealing with—is also available. Note that, in Sect. 4.3, this hierarchical structure of *tour*, *stop*, *action point*, and—if available—*handling unit* objects will be adopted to aggregate status information at different levels of granularity. *Resource* objects describe the drivers and vehicles that are scheduled for the tour.

2. *Event data* The ProveIT app delivers real-time information about the *actual* execution of a tour, so-called events. Event data serves to document the progress of the tour execution. For example, an event is sent when the driver confirms the arrival at a stop or the start of service at an action point. Events provide information about the quantity and quality of transported goods, e.g. the actual number of delivered pallets, details about damaged goods or continuous automatic updates (GPS). Also additional information, such as the driver's remaining driving time due to the driving time regulations of the European Union, is provided by events.
3. *Data from external real-time services* Services, which are provided by project partners, are used to query additional data during the execution of a tour. PTV Drive&Arrive[3] is a cloud service for the calculation of estimated times of arrivals based on real-time and historic traffic information. The service is queried periodically in short time intervals to obtain the ETAs for all stops that have not yet been reached during the execution of the tour. ETAs are used to anticipate late arrivals at stops and, thus, play an important role in the early detection of situations of interest. Additionally, the PTV Navigator Truck[4] is integrated in the ProveIT app, providing a navigation service also based on the current traffic situation. The last service is a web interface provided by the project partner LOCOM Software GmbH (LOCOM). For a given tour, the service provides information about the feasibility of a delayed delivery with respect to the inventory levels at the recipients of the transported goods. The service is queried when a disruption occurs that necessitates the modification of a tour which is currently being executed.

A generic concept for gathering event data With regard to the diversity of transport processes and use cases a generic concept was required for the data

[3] http://driveandarrive.ptvgroup.com.

[4] http://navigator.ptvgroup.com/en-uk/ptv-navigator-truck/features-for-android/.

gathering phase. We extended a concept called senslets—which was introduced in Abecker et al. (2012)—in the ProveIT disruption management system. Senslets can be considered as highly configurable software sensors, which can be flexibly configured by composing different basic building blocks, like text inputs or GPS updates. Two types of senslets can be distinguished: manual and automatic senslets. Whereas manual senslets require the input of a user (e.g. text or photo input), automatic senslets gather the data automatically and send it to the server (e.g. GPS updates). The senslet management component takes over the management of defined senslets and provides access to the basic building blocks. Senslets are mainly used in the ProveIT app during the execution of a transportation process. Once the senslets are requested by the smartphone app, an instance of the senslet model is generated based on the previously described tour plan objects. The app is able to dynamically render all basic components within the senslet model (see Fig. 3).

4.3 Analyse in real-time

Real-time analysis of data represents a powerful possibility to gain insights and is a prerequisite for a timely reaction to reduce the effects of unintended situations. Complex event processing is built on a signature-based detection paradigm (Hossbach and Seeger 2013) and can be used for the timely identification of situations. Patterns can be seen as the representation of situations of interest, describing the temporal causality within a set of events (Luckham 2001). Event processing engines are used to match predefined patterns on event streams and—in case a pattern is matched—materialize a complex event, representing a SOI. Patterns are described by means of rules in an event processing language (EPL). Within such rules, it is possible to correlate incoming real-time (e.g. tour execution data) data with non real-time data (e.g. data for a planned tour) stored in databases. The ProveIT disruption management system uses an open source complex event processing (CEP) engine called Esper.[5] Listing 1 shows an example of a rule written in the EPL of ESPER and is used to aggregate incoming information and correlate it with stored tour data to detect a delayed tour start.

```
INSERT INTO DelayTourstart
SELECT System.currentTimeMillis() AS timestamp,
nt.tour_exph_reference AS tourReference
FROM PATTERN[every nt=newTour -> timer:interval(
(nt.tour_startDatetime - System.currentTimeMillis()) /1000)
AND NOT (st=StartTour(tourReference = nt.tour_exph_reference)
OR OverplanTour(tourReferenceOld = nt.tour_exph_reference)
OR DeleteTour(tourReference = nt.tour_exph_reference))]
```

Listing 1: Example of an ESPER Rule detecting a delayed tour start

As described in Sect. 2, the operations managers of Bosch, ZF, and Geis are interested in aggregated information of smoothly running processes and critical deviations from the current plan in terms of time, quantity, and quality. In this

5 http://www.espertech.com/products/esper.php.

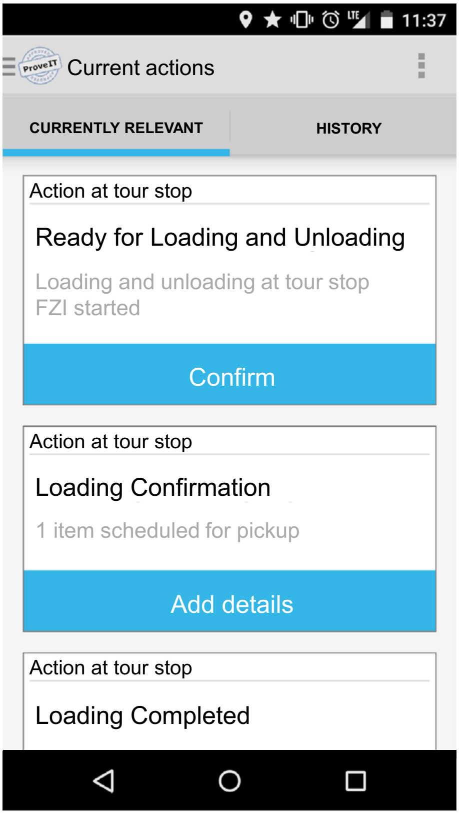

Fig. 3 Screenshot of the ProveIT app, showing three senslets during the tour execution: ready for loading and unloading, loading confirmation, loading completed

context the event engine and the disruption assessment components take over two tasks:

1. *Status calculation and recording* Incoming events are observed in order to change the status of the objects they are related to. If for example, a departure confirmation at a stop comes in, the status of the related stop is set to finished. Related deviations—in this case a positive or negative deviation from the planned departure time—are also recorded. Apart from that, aggregated current status are calculated from the lowest level in the object hierarchy to the highest level following the hierarchy—handling unit, action point, stop, tour—introduced in Sect. 4.2: if for example, all finished stops of a tour are executed so far without deviations in quantity and quality the aggregated status of this tour is classified with status ‘ok’ with respect to all tasks fulfilled in the past. Otherwise, the deviations are aggregated. The forward-looking status of a tour depends on the estimated arrival times of planned, not yet visited (open) stops. If all ETAs at these stops are in line with the schedule, the status is considered as ‘ok’, otherwise the maximum delay is calculated for an aggregation on the level of the tour. For aggregating the information from the level of handling units to action points and from action points to stops more logic is provided. A

complete picture of the current status of the inbound network is provided to the different parties of the supply chain in the ProveIT cockpit. Since all status changes are recorded with time stamps, the complete information is furthermore available for ex-post analysis. In Sect. 4.5, we introduce examples for ex-post analysis and for different views of the ProveIT cockpit.

2. *Classification of deviations* Besides the detection of deviations, which is also necessary for status calculation, the core function of the real-time analysis is the assessment of deviations. All deviations in quantity or quality, i.e. each missing or defect part, are classified as a disruption, which makes the call of the online optimization component necessary. The evaluation of a deviation from planned mile stones, such as a pickup or a delivery at a certain step is more complex. Depending on the current vehicle position, the current traffic situation and possibly updated service times or estimated delays, the estimated time of arrival at all open stops is calculated using the real time service PTV Drive&Arrive. If a hard time window at one stop is violated, the online optimization is automatically triggered. Furthermore, all disruptions are immediately reported to the operations manager of the affected companies by e-mail.

4.4 Decide on mitigation actions

When a deviation is classified as a disruption, the online optimization component is triggered to provide the operations manager with mitigation actions. In this subsection, we first describe the considered mitigation actions and then give a short overview of the related optimization models and the implementation.

4.4.1 Modelling mitigating actions

Depending on the transport concept and the concrete contractual situation, the scope of reactions and the responsibilities for confirming and executing an action are different, and, hence, the optimization models are different as well. However, for obtaining—to the greatest possible extent—reusable approaches, we derived general layers of mitigation actions based on the ideas of Pulter et al. (2010). Table 1 gives an overview of these layers adapted to the milk run use case of Bosch.

In general the layers are ordered in a way that the impact of a mitigation action on the inbound network is as small as possible and, hence, the propagation of disturbances through the network is minimal. Based on this assumption, the layers can be used as an escalation model: if a mitigation on a lower layer is not successful, the system steps to the next layer until an action is successful or, at last, until the production is informed that a delayed delivery cannot be avoided.

Please note that all types of disruptions are considered as delays and are addressed by the same mitigation actions: if parts are damaged or missing, we assume that a new date of availability at the supplier site is communicated.

Corresponding to the mitigation layers of Table 1 we give an overview of the decision models in Table 2: in the second column the most important inputs differing from the ones used for the respective offline planning task are given, while

Table 1 Mitigation layers, scope of reaction, and responsibility of confirming and executing an action tailored to UC1

Layer	Scope of reaction	Responsibility
1. Vehicle	Change routing of disrupted tour	Driver
2. Vehicle	Change stop sequence of disrupted tour	LSP
3. Tours	Shift of transport order(s) of disrupted tour to other nearby tours	LSP
4. Order	Leave out transport order(s) of disrupted tours in favour of: (a) shifting transport order to the next scheduled milk run tour, (b) shifting transport order to area forwarding network	LSP/Bosch
5. Fleet LSP	Shift of transport order(s) of disrupted tour to free emergency vehicle of LSP	LSP
6. Fleet Bosch	Shift of transport order(s) of disrupted tour to free emergency vehicle of Bosch	Bosch
7. Production	Inform production about the disruption	Bosch

in the last column the decision models along with special characteristics are described.

Dynamic routing based on current traffic information, which corresponds to the first layer, is already incorporated in many recent navigators. One example is the PTV Navigator Truck, which is integrated in the ProveIT app. Furthermore, especially in case of regular milk runs, the drivers have a very good knowledge of the area and, usually, take good routing decisions in case of traffic disruptions.

The decision models of the layers 2, 3, 5, and 6 all comprise vehicle routing problems with time windows (VRP-TW) or travelling salesman problems with time windows (TSP-TW) considering dynamic time-dependent travel times. That means, the travel times depend on the current traffic situation and change over the course of the day.

Within the ProveIT online optimization component, the mitigation actions for layers 2, 3, 5, and 6 are calculated based on the same approaches used for the offline planning. This ensures that all requirements from the offline planning task, such as driving time regulations and vehicle qualifications, are respected and that a certain stability in terms of solution characteristics is achieved. However, since the planning tool of our project partner PTV Group xTour is not yet able to consider time-dependent travel times, we propose a general approach for enhancing VRP heuristics to the TD-VRP case which is detailed below. This approach enables us to use the same planning component for both offline and online planning, which is powerful in terms of real world requirements such as break and rest rules and driver or vehicle qualifications.

At the fourth layer, it is checked if transport orders can be shifted to the next milk run or to the area forwarding service. This is done by predicting the range of inventory for the affected parts based on the current inventory level and on the planned or historical consumption. The shift of a transport order is assumed to be feasible if the new delivery date lies within the range of the inventory. This functionality is implemented within a real-time service of our project partner LOCOM.

Table 2 Overview of most important inputs different from the offline planning tasks, the decision models and special characteristics by layer for UC1

Layer	Most important inputs different from offline planning	Decision models and special characteristics
1.	Current vehicle position	Dynamic routing by navigator or driver
	Time-dependent driving time estimations based on current traffic situation	
2.	Ordered list of processed transport orders and transport orders in process of late tour	Dynamic time-dependent TSP-TW:
	Set of not yet processed (in the following referred to as "open") transport orders of late tour	Real-time travel times
	Current vehicle position of late tour	Fixed stops of processed orders
	Time-dependent driving time estimations based on current traffic situation	Virtual stop for the current position of the vehicle
3.	Ordered list of processed transport orders and transport orders in process of late tour and nearby tours	Dynamic time-dependent VRP-TW:
	Set of open transport orders of late tour and nearby tours	Real-time travel times
	Current vehicle position of late tour and nearby tours	Fixed stops of processed orders
	Time-dependent driving time estimations based on current traffic situation	Virtual stops for the current positions of the vehicles
4.	Schedule of next relevant milk run	Prediction of inventory range by real-time service: returns if a delivery at a later point in time is feasible with respect to inventory levels of the parts related to the transport order
	Lead times of area forwarding services	
5.	Same input as layer 1	Dynamic time-dependent VRP-TW:
	Availability, capacity restrictions and current position of emergency vehicle of LSP	Real-time travel times
		Fixed stops of processed orders
		Virtual stops for the current positions of the vehicles
6.	Same input as layer 1	Dynamic time-dependent VRP-TW:
	Availability, capacity restrictions and current position of emergency vehicle of Bosch	Real-time travel times
		Fixed stops of processed orders
		Virtual stops for the current positions of the vehicles

The services for layers 1 and 4 are provided by our project partners and are considered as black box. In this contribution we focus on models for the TD-VRP layers 2, 3, 5, and 6: the model corresponding to layer 3 is the most general out of the four, and it can be easily simplified to the models corresponding to other layers. Hence in the following we introduce the most general description for layer 3.

Please note that for evaluation purposes (see Sect. 4.4.4), we use a simplified model compared to the range of functions considered in ProveIT and provided by the powerful tool xTour: as three examples, we do not consider break and rest time regulations, we restrict the model to pure pickup tours, and we do not consider vehicle and driver qualifications.

4.4.2 A mixed integer model for TD-VRP layers

Offline model For the offline decision, i.e. the planning decision taken before any of the vehicles has started its tour in the morning, we consider time-dependent travel time estimations. To formally express this decision, we introduce the sets, parameters, and variables in Table 3. Please note that we use the term customer instead of supplier as it is common in VRP literature.

For keeping the network as sparse as possible, the arc sets $\mathcal{A}_k$ depending on the vehicle k are defined as follows:

$$\begin{aligned}\mathcal{A}_k = &\{(i,j,m)\,|\,i \in \mathcal{C}, \quad j \in \mathcal{C}, i \neq j, m \in \mathcal{M}_{ij}\}\\ &\cup \{(i,j,m)\,|\,i = \delta^{\mathcal{KS}}(k), \quad j \in \mathcal{C}, m \in \mathcal{M}_{ij}\}\\ &\cup \{(i,j,m)\,|\,i \in \mathcal{C}, \quad j = \delta^{\mathcal{KE}}(k), m \in \mathcal{M}_{ij}\}\\ &\cup \{(i,j,m)\,|\,i = \delta^{\mathcal{KS}}(k), \quad j = \delta^{\mathcal{KE}}(k), m \in \mathcal{M}_{ij}\}\end{aligned}$$

Using the introduced notation, the model corresponding to a TD-VRP as introduced for example in Chen et al. (2006) can be expressed as follows:

$$\min \sum_{k \in \mathcal{K}} \sum_{(i,j,m) \in \mathcal{A}_k} x_{ijmk} \cdot D_{ij} \cdot C_k^{DIST} \tag{1}$$

$$s.t. \quad \sum_{k \in \mathcal{K}} \sum_{j,m:(i,j,m) \in \mathcal{A}_k} x_{ijm\delta^{\mathcal{SK}}(i)} = 1 \quad i \in \mathcal{S} \tag{2}$$

$$\sum_{k \in \mathcal{K}} \sum_{i,m:(i,j,m) \in \mathcal{A}_k} x_{ijm\delta^{\mathcal{EK}}(j)} = 1 \quad j \in \mathcal{E} \tag{3}$$

$$\sum_{k \in \mathcal{K}} \sum_{i,m:(i,j,m) \in \mathcal{A}_k} x_{ijmk} = 1 \quad j \in \mathcal{C} \tag{4}$$

$$\sum_{i,m:(i,j,m) \in \mathcal{A}_k} x_{ijmk} - \sum_{i,m:(j,i,m) \in \mathcal{A}_k} x_{jimk} = 0 \quad j \in \mathcal{C}, k \in \mathcal{K} \tag{5}$$

$$t_i - T_i^{SERVICE} = T^{START} \quad i \in \mathcal{S} \tag{6}$$

$$t_j - t_i + B^1 \cdot (1 - x_{ijmk}) \geq T_{ijm}^{TRAVEL} + T_j^{SERVICE} \quad k \in \mathcal{K}, (i,j,m) \in \mathcal{A}_k \tag{7}$$

Table 3 Overview of sets, parameters, and variables

	Description
Set	
$\mathcal{C}$	Set of customer nodes
$\mathcal{K}$	Set of vehicles
$\mathcal{S}$	Set of start nodes, one for each vehicle
$\mathcal{E}$	Set of end nodes, one for each vehicle
$\mathcal{V}$	Set of all nodes $\mathcal{V} = \mathcal{C} \cup \mathcal{S} \cup \mathcal{E}$
$\mathcal{M}_{ij}$	Set of time intervals existing for the edge $i, j \in \mathcal{V}$
$\mathcal{A}_k$	Arc set defined for each vehicle $k \in \mathcal{K}$
Parameter	
	Variable cost of vehicle $k \in \mathcal{K}$ per unit distance
W_i	Demand at node $i \in \mathcal{C}$ given for example in kg
Q_k	Capacity of vehicle $k \in \mathcal{K}$ given in the same unit as demand
D_{ij}	Travel distance of edge $i, j \in \mathcal{V}$
	Service time at node $i \in \mathcal{V}$
	Estimated travel time of edge $i, j \in \quad \mathcal{V}$ during time interval $m \in \mathcal{M}_{ij}$
τ_{ijm}	Upper bound of time interval $m \in \mathcal{M}$ of edge $i, j \in \mathcal{V}$
L_i	Lower bound of time window at node i
U_i	Upper bound of time window at node i
	Start time of all vehicles
$\delta^{\mathcal{KS}}(k)$	Function mapping a vehicle $k \in \mathcal{K}$ to its start node $i \in \mathcal{S}$
$\delta^{\mathcal{KE}}(k)$	Function mapping a vehicle $k \in \mathcal{K}$ to its end node $i \in \mathcal{E}$
$\delta^{\mathcal{SK}}(i)$	Function mapping a start node $i \in \mathcal{S}$ to its vehicle $k \in \mathcal{K}$
$\delta^{\mathcal{EK}}(i)$	Function mapping an end node $i \in \mathcal{E}$ to its vehicle $i \in \mathcal{K}$
B^1	Big number: $\max\limits_{k \in \mathcal{K}, (i,j,m) \in \mathcal{A}^k} (\tau_{ijm})$
B^2	Big number: $\max\limits_{k \in \mathcal{K}} (Q_k)$
Variable	
x_{ijmk}	Binary decision variable, 1 if vehicle $k \in \mathcal{K}$ travels from node i to node j during time interval m with $(i, j, m) \in \mathcal{A}^k$, otherwise 0
t_i	Departure time at node $i \in \mathcal{V}$
w_i	Load of vehicle after node $i \in \mathcal{V}$ is visited

$$t_i - B^1 \cdot (1 - x_{ijmk}) \leq \tau_{ijm} \quad k \in \mathcal{K}, (i, j, m) \in \mathcal{A}_k \tag{8}$$

$$t_i \geq \tau_{ij(m-1)} \cdot x_{ijmk} \quad k \in \mathcal{K}, (i, j, m) \in \mathcal{A}_k \tag{9}$$

$$t_i - T_i^{SERVICE} \geq L_i \quad i \in \mathcal{V} \tag{10}$$

$$t_i - T_i^{SERVICE} \leq U_i \quad i \in \mathcal{V} \tag{11}$$

$$w_i = 0 \quad i \in \mathcal{S} \tag{12}$$

$$w_j - w_i + B^2 \cdot \left(1 - \sum_{k \in \mathcal{K}} \sum_{m:(i,j,m) \in \mathcal{A}^k} x_{ijmk}\right) \geq W_j \quad i \in \mathcal{S} \cup \mathcal{C}, \quad j \in \mathcal{C} \cup \mathcal{E} \tag{13}$$

$$w_{\delta^{\mathcal{KE}}(k)} \leq Q_k \quad k \in \mathcal{K} \tag{14}$$

$$x_{ijmk} \in \{0,1\} \quad k \in \mathcal{K}, (i,j,m) \in \mathcal{A}_k \tag{15}$$

$$t_i \in \mathbb{N}_0 \quad i \in \mathcal{V} \tag{16}$$

$$w_i \in \mathbb{R}_0^+ \quad i \in \mathcal{V} \tag{17}$$

Objective function (1) minimizes the variable cost incurred by the travelled distance. If the variable cost for vehicles in the fleet are the same, the cost parameter vehicle cost can be omitted. As an alternative the arrival times in the depot expressed as $\sum_{i \in \mathcal{E}} t_i$—possibly weighted with a vehicle specific factor—can be minimized.

The first set of constraints (2) assures that every vehicle k leaves its own start node $i = \delta^{\mathcal{KS}}(k)$ either heading to a customer node or directly to its end node. Constraint set (3) analogously makes sure that the route of every vehicle k ends in its own end node. Furthermore, it is assured that every customer $j \in \mathcal{C}$ is visited by a vehicle in constraint set (4), while the next set of constraints assures flow conservation. The tour start time is set to in constraint set (6). If necessary, it can also be set to a vehicle specific start time. Constraint set (7) assures that the departure time at node j is greater than or equal to the departure time at node i plus the service time of node j and the travel time of edge (i, j) if edge (i, j) is used by vehicle k during time interval m. The following two sets of constraints (8) and (9) make sure that the time interval m—or the corresponding arc—is selected in which the departure time of vehicle k at node i falls into. Constraints (10) and (11) enforce the start of service time into the time window of every customer. Constraint set (12) sets the load at the start nodes to zero assuming a pick-up tour scenario, while constraint set (14) assures that the vehicle capacity just before the depot is not exceeded. Constraints (13) assure that the load after node j at least sums up to the load of its predecessor i plus the demand of node j. Finally, the domains of the decision variables are set in constraints (15)–(17).

Since considering time-dependent travel times makes the vehicle routing problem more complex to solve, the operative planning step is usually conducted based on travel times not varying over time. In this case, for all edges the number of time intervals m is one and the travel times correspond for example to the 75% quantile of the historic travel times.

Online model Online decisions are taken from the moment, when the first vehicle has started its tour. In this case, parts of the decisions of the offline decision step are no longer reversible, and the decision model must consider the current state of the system incorporating the following aspects:

– The current position of each vehicle.
– Distance and current travel times from these positions.
– The current load of each vehicle.
– Customers already visited by a vehicle.
– Customers to be visited.

To this end, we introduce virtual customer nodes representing the current position of each vehicle $k \in \mathcal{K}$ at the current point in time $\mathcal{T}^{CURRENT}$ (see Table 4). If the vehicles are driving at $\mathcal{T}^{CURRENT}$ or are waiting at a customer, the service time $T_i^{SERVICE}$ of these virtual customers $i \in \mathcal{C}^{VIRTUAL}$ is 0. If a vehicle is executing service at a customer i, the service time at the customer node corresponds to the time of the service already processed, while the service time at the virtual vehicle node corresponds to the time left for finishing the service. The demand W_i of virtual nodes is 0.

All customers already visited and all virtual vehicle nodes are incorporated in the set $\mathcal{C}^{VISITED}$, while all customers not yet serviced are contained in the set $\mathcal{C}^{OPEN}$.

Furthermore, the distance and travel time matrices must be updated in a way that entries from the virtual nodes to all open customer nodes $i \in \mathcal{C}^{OPEN}$ and the end nodes $i \in \mathcal{E}$ are available and that the estimated travel times consider the current traffic situation. Therefore, a geocoding component is necessary, binding the current

Table 4 Sets and parameters complementing Table 3 for formulating the online model

	Description
Set	
$\mathcal{C}^{OPEN}$	Set of (open) customers not yet visited
$\mathcal{C}^{VISITED}$	Set of customers already visited
$\mathcal{C}^{VIRTUAL}$	Set of virtual customer visits representing the current position of the vehicle with $\mathcal{C}^{VIRTUAL} \subset \mathcal{C}^{VISITED}$
$\mathcal{C}$	Set of customer nodes with $\mathcal{C} = \mathcal{C}^{OPEN} \cup \mathcal{C}^{VISITED}$
$\bar{\mathcal{A}}_k$	Arc set for vehicle $k \in \mathcal{K}$
Parameter	
$\mathcal{T}^{CURRENT}$	Point in time at which a replanning is triggered
$\overline{D}_{ij}$	Travel distance of edge $i,j \in \mathcal{V}$
$\overline{T}_{ijm}^{TRAVEL}$	Updated travel time estimation of edge $i,j \in \mathcal{V}$ during time interval $m \in \mathcal{M}_{ij}$ containing virtual vehicle node $i \in \mathcal{V}^{\text{virtual}}$
$\overline{\tau}_{ijm}$	Upper bound of time interval $m \in \mathcal{M}$ of edge $i,j \in \mathcal{V}$
$\bar{x}_{ijk}$	Parameter indicating if arc (i, j) was used by vehicle k before $\mathcal{T}^{CURRENT}$
$\overline{T}_i^{ACTUAL}$	Actual departure time at visited node $i \in \mathcal{C}^{VISITED}$
B^1	Big number: $\max\limits_{k \in \mathcal{K},(i,j,m) \in \mathcal{A}^k} (\overline{\tau}_{ijm})$

geo position of the vehicles to a point in the street network. Furthermore, a component for predicting the time-dependent travel times on the respective edges based on current and historic traffic data is needed. The result of these two components, considered as black box services, is given in the updated travel time matrix represented by T_{ijm}^{TRAVEL}. The lower bound of the first interval $m = 1$ contained in this travel time matrix is $T^{CURRENT}$.

Parameter $\bar{x}_{ijk} = 1$ indicates for all edges from $i \in \mathcal{C}^{VISITED} \cup \mathcal{S}$ to $j \in \mathcal{C}^{VISITED}$ if it was actually used by a vehicle k during the tour execution until $T^{CURRENT}$.

Based upon these inputs the arc set for the online problem depending on vehicle $k \in \mathcal{K}$ can be formulated. For a better readability of the model, we differentiate between an arc set $\bar{\mathcal{A}}_k^{VISITED}$ containing arcs adjacent to nodes which both have been visited until $T^{CURRENT}$, and a set of arcs $\bar{\mathcal{A}}_k^{OPEN}$ incorporating arcs adjacent to nodes of which at least one node is contained in the set $\mathcal{C}^{OPEN}$, i.e. at least one node has not been visited yet.

$$\begin{aligned}\bar{\mathcal{A}}_k^{VISITED} &= \{(i,j,m) \,|\, i \in \delta^{\mathcal{KS}}(k), j \in \mathcal{C}^{VISITED}, m = 1, \bar{x}_{ijk} = 1\} \\ &\cup \{(i,j,m) \,|\, i \in \mathcal{C}^{VISITED}, j \in \mathcal{C}^{VISITED}, m = 1, \bar{x}_{ijk} = 1\}\end{aligned}$$

$$\begin{aligned}\bar{\mathcal{A}}_k^{OPEN} = \{(i,j,m) \,|\, & i \in \mathcal{C}^{VISITED}, j \in \mathcal{C}^{OPEN} \cup \delta^{\mathcal{KE}}(k), m \in \mathcal{M}_{ij}, \\ & \sum_{j:(i,j,m,k) \in \mathcal{A}^k} \bar{x}_{ijk} = 0\} \cup \{(i,j,m) \,|\, i \in \mathcal{C}^{OPEN} \cup \mathcal{C}^{VIRTUAL}, \\ & j \in \mathcal{C}^{OPEN} \cup \mathcal{E}, m \in \mathcal{M}_{ij}\}\end{aligned}$$

$$\bar{\mathcal{A}}_k = \bar{\mathcal{A}}_k^{VISITED} \cup \bar{\mathcal{A}}_k^{OPEN}$$

Please note that this arc set represents the case in which no vehicle has returned to its end node yet. If a vehicle is already back at the depot in $T^{CURRENT}$, it needs to be decided if it is allowed to leave the depot again for taking over tasks.

Based on additional inputs described above the online decision model is expressed as follows: constraint sets (18) and (19) are added to the base model, while sets (20)–(24) replace constraints (7)–(11) of the base model.

$$x_{ijmk} = \bar{x}_{ijk} \quad k \in \mathcal{K}, (i,j,m) \in \bar{\mathcal{A}}_k^{VISITED} \tag{18}$$

$$t_i = \overline{T}_i^{actual} \quad i \in \mathcal{C}^{VISISTED} \tag{19}$$

$$t_j - t_i + B_1 \cdot (1 - x_{ijmk}) \geq \overline{T}_{ijm}^{travel} + T_j^{service} \quad k \in \mathcal{K}, (i,j,m) \in \bar{\mathcal{A}}_k^{OPEN} \tag{20}$$

$$t_i - B_1 \cdot (1 - x_{ijmk}) \leq \overline{\tau}_{ijm} \quad k \in \mathcal{K}, (i,j,m) \in \bar{\mathcal{A}}_k^{OPEN} \tag{21}$$

$$t_i \geq \overline{\tau}_{ij(m-1)} \cdot x_{ijmk} \quad k \in \mathcal{K}, (i,j,m) \in \bar{\mathcal{A}}_k^{OPEN} \tag{22}$$

$$t_i - T_i^{service} \geq L_i \quad i \in \mathcal{C}^{OPEN} \cup \mathcal{E} \tag{23}$$

$$t_i - T_i^{service} \leq U_i \quad i \in \mathcal{C}^{OPEN} \cup \mathcal{E} \tag{24}$$

Constraints (18) along with (14) assure that the load w_i at the virtual customer node $i \in \mathcal{V}^{VIRTUAL} \subset \mathcal{C}^{OPEN}$ corresponds to the current load. In constraint set (19) the departure time variables of all nodes already visited are set to the actual departure time. For virtual nodes the actual departure time corresponds to the current point in time $\mathcal{T}^{CURRENT}$. Constraint sets (20)–(24) take—analogously to the offline model—care of the time-related variables for the open customer nodes and the end nodes.

4.4.3 IterTD: iterative approach for enhancing VRP solvers to the TD-VRP

As mentioned before, the xTour planning tool of our project partner PTV is not yet able to consider time-dependent travel times. This component therefore has to be adapted to the online optimization case in a way that current travel time estimations can be considered during the re-planning of disrupted tours. IterTD is a general approach for enhancing VRP planning tools to the TD-VRP case. It is visualized as *online optimization component* in the lower box of Fig. 4: while a tour is executed, the *disruption assessment component* (upper box) periodically receives status updates and vehicle coordinates. This information is required in order to verify whether the planned tour sequence is feasible with regard to real-time data. In the ProveIT system, this check is performed with the PTV Drive&Arrive service which covers the dynamic behaviour of travel times.

If a disruption is identified, the online optimization component implementing IterTD for mitigation layer 3 seeks to shift transport orders and change stop sequences such that tours become feasible again. To do so, we use—within an iterative approach for considering time-dependent travel times—the same planning tool as for the offline planning step. This avoids deviations from the original plan that originate solely from changing the underlying algorithms.

As described in the preceding subsection, irreversible decisions such as already completed pickups are fixed. In order to enable the VRP planning tool to access up-to-date travel time estimations, it is supplied with a 'snapshot' matrix ($\overline{T}_{ij1}^{TRAVEL}$) representing the state of the traffic situation at the point in time $\mathcal{T}^{CURRENT}$. Since this snapshot matrix does not cover the dynamic evolvement of the travel times, two types of errors might occur: the driving time of a link might be (1) overestimated, or the driving time is (2) underestimated. Due to the first error the solution space might be restricted in a way that actually feasible tours cannot be found. In this case, the online optimization escalates to the next layer. The second error is addressed by checking the resulting tour with the Drive&Arrive component. If this check succeeds, the proposed plan is feasible with respect to all available real-time information. If the PTV Drive&Arrive component detects an error even though the planning component was able to find a solution, the snapshot matrix is iteratively modified by increasing the travel time information for those links that are responsible for the failure of the time-dependent check. IterTD ensures that the

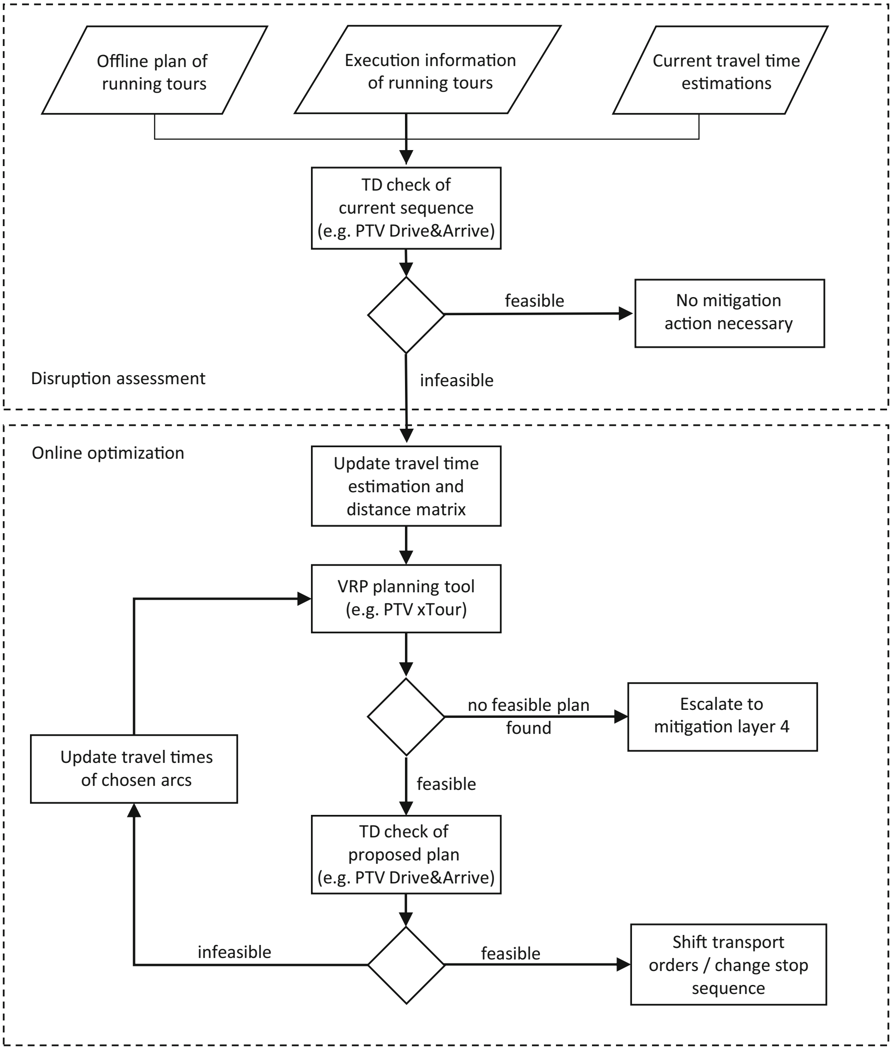

Fig. 4 Schematic view of the heuristic approach for mitigation layer 3

online planning component terminates and guarantees that all plans resulting from the online optimization phase are feasible with respect to time-dependent travel times. However, it is possible that the snapshot matrix is modified such that the VRP solver mistakenly does not find a feasible solution. In this case, IterTD leads to an early, unnecessary escalation.

4.4.4 Evaluation of IterTD

In order to evaluate the loss of optimality of IterTD without a bias possibly introduced by the black box heuristics of the PTV xTour, we substitute the PTV components by exact approaches: the VRP planning task of Fig. 4 is done by solving the mixed integer model introduced in the preceding subsection to optimality reducing the time expanded travel time matrix to the snapshot matrix. As modelling tool and solver we use the CPLEX Optimization Studio 12 of IBM[6] embedded in a Python 2.7 script realising the IterTD functions depicted in Fig. 4. The full TD-VRP model of the preceding section is used as a benchmark also solved by CPLEX. All experiments have been conducted on a Windows 7 machine with an Intel Core i7 processor (2.6–3.2 GHz) and 8 GB RAM using CPLEX default configurations.

A first evaluation of IterTD was based on the milk run tours included in the field tests (see also Sect. 5 below). However, these tours have been proven particularly robust: in most scenarios, only an increase in average travel time of more than 100% qualifies as a disruption and actually triggers the online optimization component. This is due to long opening hours for each tour stop as well as the particular design in which travel times are low compared to the service time required at each customer.

As a consequence, we additionally evaluate the approach based on a set of test instances created randomly from a data set of time-dependent travel time data of the city of Lyon provided in Melgarejo et al. (2015). In this contribution travel time estimations based on long-term observations are given in six-minute intervals, resulting in a high-resolution matrix. For creating the instances, we randomly chose up to 15 customer nodes each with a demand of 1 unit, we assigned time windows, and selected either one (TSP), two or three (VRP) vehicles each with a capacity between 5 and 10 units. For creating disruptions, all travel times are simultaneously increased by five different percentage values, modelling for example weather phenomena such as black ice or snow affecting all tours within a geographical region. For each instance, this is done at three different points in time.

Sites visited prior to the time of disruption are considered to have been performed in the sequence indicated by the offline plan. The initial offline solutions are provided using the TD-VRP model. As objective the maximum arrival time at the depot is chosen in order to reduce the average route duration.

In this experimental setup, the TSP instances correspond to mitigation layer 2, while the VRP instances represent the planning tasks of layer 3. The size of the instances with up to 3 tours is relevant in practical applications, as only a limited number of milk runs is usually being executed within the same area. Within the field tests, we considered two milk runs in the area of the Bosch site in Homburg: one with seven sites and one with two.

Figure 5 shows, for different increases of travel times, the number of instances for which a feasible solution could be found by our heuristic and by solving the TD-VRP model, respectively. The heuristic approach yields good results for minor disruptions of up to 25% travel time increase: it identifies almost always a feasible solution compatible to current travel time estimations within an average of around 2—at most 6—iterations. As can be seen in Fig. 6, runtime scales well for the heuristic approach, while the exact approach is prohibitively expensive even for

[6] http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud.

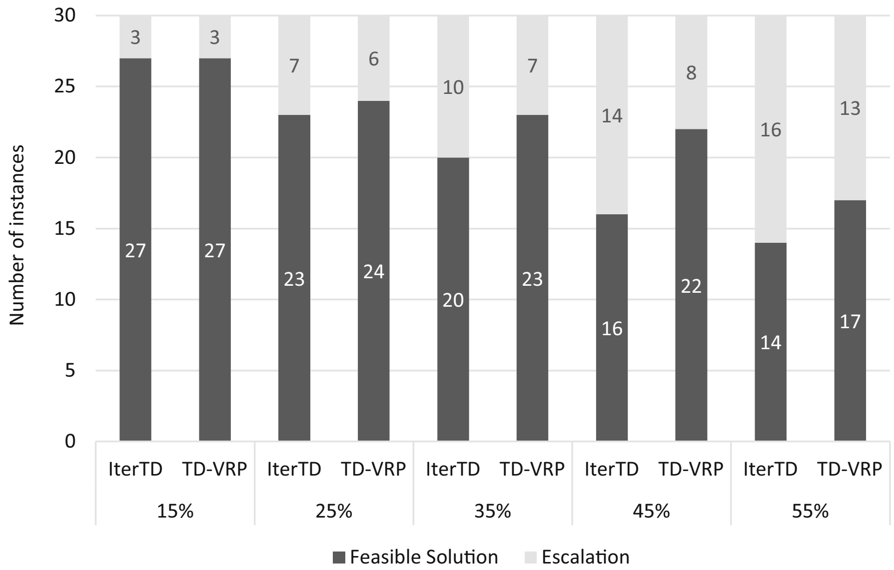

Fig. 5 Number of instances resulting in feasible solutions and escalations by algorithm and extent of travel time increase

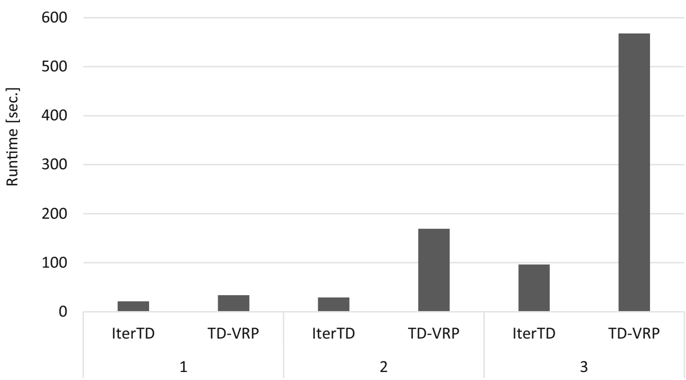

Fig. 6 Average runtime in seconds by algorithm and number of vehicles

small instances. However, in case of major disruptions, the heuristic frequently leads to an early escalation. Thus, it requires for example the use of an emergency vehicle even though the problem could be resolved within the lower mitigation layer when taking time-dependency into account.

In practical applications, the runtime of the online algorithm component is crucial. Considering this, the loss in quality incurred when using a heuristic approach can be considered acceptable, particularly taking into account the stability of solutions that comes from using the same VRP planning tool for both offline and online planning.

Apart from evaluating the IterTD algorithm by means of the exact mixed integer model, we derived an extension of the model taking into account all layers simultaneously. This model is dedicated to assessing the optimality loss resulting from the escalation-based approach. However, due to the limited scope of the field experiments (see Sect. 5), a conclusive evaluation of the overall approach needs to be based on extensive simulation experiments. This will be covered in a forthcoming publication.

4.5 Deliver results

The last step in the data-driven decision process is the delivery of relevant results and decisions to interested stakeholders. Recipients in this context can be IT-systems, as well as human beings, whereas the motivation of the recipients differs.

Human beings When it comes to informing human beings about results and decisions the graphical user interface (GUI) plays a major role. The ProveIT disruption management system offers two possibilities to visualize relevant results. The major user interface is represented by a web application (see Fig. 7). Here all users involved in the tour planning and monitoring have a role-dependent view on execution-related data, especially on the incoming real-time data and analysis results. In addition to the web application we use the previously mentioned smartphone app to inform users involved in the execution of the transport process about planned tours, identified disruptions, as well as, instructions on how to react. We provide e-mail notifications on relevant updates and analysis. Additionally, it is intended to integrate a news-ticker in the cockpit which allows the user to subscribe to planned tours so that no potentially important update is missed.

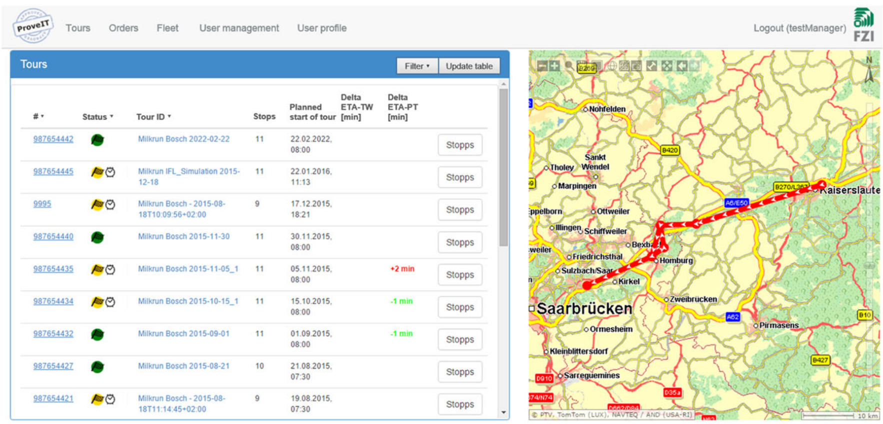

Fig. 7 Screenshot of the ProveIT cockpit, showing an overview with basic tour information, the aggregated status addressing the finished tasks in column two, and the deviation of the ETA from time windows and from planned arrival times in columns 6 and 7, respectively

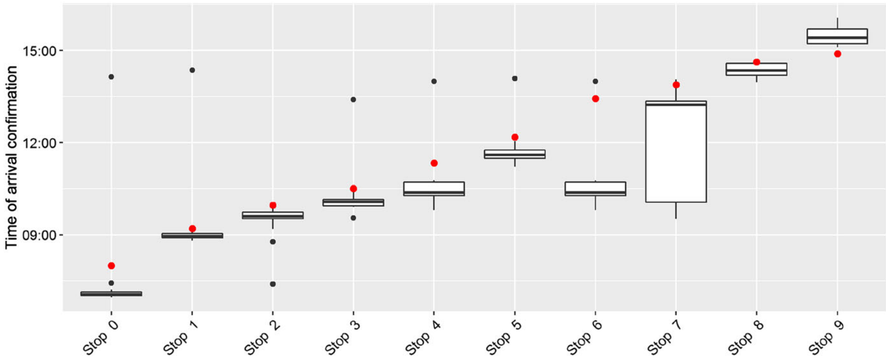

Fig. 8 *Box plot* diagram of arrival times confirmed by the driver for a daily milk run in (UC1). Planned arrival times are indicated in *red*

The information gathered during the execution of transports furthermore enables the analysis of planned tours and schedules on a tactical level. To this end, we determine performance indicators and provide visual information on key statistics such as average travel times or quantity deviations that can be interpreted by human decision-makers. Figure 8 provides an example of such an analysis, indicating the confirmed arrival times along a tour over the course of several weeks as well as the planned arrival times.

Systems The intention of supplying derived data back to systems is to achieve an automatic reaction of the system. To implement a reaction it is for example necessary to cancel transport orders and create new transport orders in the planning system of the receiving plant or to update tour plans in the transport management system of the logistics service providers. However, due to the diversity of the affected systems of the different parties, it is a major challenge to automatically assure a consistent state. Therefore, the ProveIT disruption management system is currently only giving recommendations to the users of the system, and leaves it to the users to implement the changes in the different systems.

5 Experiences and challenges in field experiments

Due to the restricted scope of the field experiments,[7] a final evaluation of the ProveIT disruption management system is not yet possible. In the following section, we therefore describe the experiences and discuss the main challenges of a turn into production of the ProveIT system.

Scope of the field experiments The scope of the field experiments was restricted for two reasons: the integration complexity of the IT-systems of the different companies for a large set of supplier plant relations, area forwarding regions or different transport and logistics concepts is significant. Additionally, the extra effort for an extensive test by the operations managers and by the respective drivers is—in

[7] Field tests are ongoing in ProveIT and follow-up projects.

addition to their daily tasks and the operation of the productive IT-systems—enormous.

The field tests addressed two daily milk runs of an automotive plant of Bosch in Homburg, Germany, (UC1) and was conducted daily during several weeks. The respective drivers were equipped with a smartphone with the ProveIT app installed, the regular tour was deposed on the system and communicated to the drivers by the ProveIT app. For (UC2) the ProveIT app and the cockpit had to be adapted to the tour structure of area forwarding networks usually consisting of a pre-leg and a main-leg. During several weeks the tours were planned by the responsible person of the area forwarder Geis and were transferred to the ProveIT disruption management system and communicated through the ProveIT app to the drivers.

Throughout both use cases, the background processing systems were running and all agreed manual and automatic events along the transport process were sent to the system. The backward- and forward-looking status could be followed on the ProveIT cockpit, and the persisted events were analysed ex-post. The persons in charge at our project partners received alerting e-mails if a certain threshold for delays was exceeded or if quantity or quality deviations occurred.

Key insights Due to the robustness of the considered tours with respect to delays in both use cases, a re-routing of the truck (mitigation layer 1) was enough in case of delays. However, Bosch used this insight to re-discuss the milk run schedules with the logistics service providers. Furthermore, our industrial partners came to the conclusion that the ProveIT disruption management system is especially helpful in case of long running tours, for example trans-European tours, transporting important bottleneck parts and, hence, having tight time windows. This type of tours was, due to organizational reasons, not in the scope of the field tests. However, an important insight is that a careful selection of considered tours for a ramp-up of a disruption management system is important for reaching an early acceptance for the involved parties.

As expected, the integration effort for the IT-systems of the different involved parties is high. However, it was surprising that for similar tasks within a tour the requirements of the partners differentiated. The involved partners expected individual contents displayed in the ProveIT app and different outputs from the system. Thus, the system had to be highly adaptive, especially with respect to the ProveIT app and the real-time processing components.

Regarding the usability of the smartphone app, we had several iterations in cooperation with the drivers. Thus, we were able to improve the app and its usability constantly. This included functionalities for supporting a more interactive use as well as the information that is given to the driver at a specific point during the tour execution. As an example, the drivers requested a much tighter navigation through the transport process, which resulted in a context sensitive display of relevant senslets. That means that only senslets are shown that are relevant for the current situation. The relevance can be for example assessed on the progress of the tour or based on the current position of the vehicle. The content of senslets was adapted, for example by giving more detailed information on the type and quantity of load carriers to be picked up at a specific location—a information which was only important in (UC2). In terms of usability, we provided a shortcut to the truck navigator app installed on the same device, making it possible for the drivers to request navigation to the next stop without having to enter the address by hand. We adapted the design and positioning of buttons and menus. Information requested from the driver, for example arrival confirmations or quality deviations, are now confirmed with haptic feedback.

Since the gathered data could be used for timekeeping or performance measurement of the drivers, we had to assure—in cooperation with Bosch—that the data is not used for that purpose. After several days of testing, the drivers accepted the tracking. In (UC1) one of the drivers started recognizing the advantages for themselves: he appreciates the possibility to easily prove that he arrived on time and to be able to track the standing times at tour stops by himself.

The motivation of the different involved parties has to be examined in detail. Depending on the current contractual conditions the benefit of using the system may be unequally distributed between the logistics service provider and recipients. Coming back to the introducing example: in the past, the supplier would have deployed another vehicle in case of a delay of the primary vehicle. With the ProveIT disruption management system the extra trip would not be deployed due to sufficient stock in the receiving plant. However, the benefit of this decision lies only at the logistics service provider. Thus, the incentive of the recipient to integrate and share relevant data might be missing. An interesting starting point for this path of research is a discussion of new business models for advanced ICT given in Boschian and Paganelli (2016).

As already mentioned, an extensive evaluation of disruption management systems for inbound networks in field experiments is rather difficult since a critical amount of tours needs to be achieved to fully prove the benefit. For further evaluation the field experiments need to be extended by means of tours and orders and, ideally, a use case containing both transport concepts simultaneously is added. Furthermore, for the evaluation of the benefits in a large scale network an extensive simulation study is necessary relying on the real data from the field experiments and additional data collected from the partners.

6 Conclusion and outlook

Within this paper we gave an overview of the architecture and the key functions of the ProveIT disruption management system addressing the standard transport concepts in the automotive industry. The system is aimed to cover two perspectives: the perspective of the receiving plant requesting and paying the transports, and the perspective of the logistics service provider executing and subcontracting the necessary transport tasks. We covered the whole data-driven decision process from gathering and integrating data, over analysing them in real-time, taking decisions on mitigation actions, and delivering the results to the relevant sites. To this end, we combined methods, technologies and principles from several communities: knowledge management, real-time stream processing, data analytics, and mathematical optimization.

To tackle the challenge of heterogeneous IT-environments and different requirements imposed by the different roles and companies, we designed a very flexible component-based architecture. On the level of components, we aimed for a high flexibility by relying on complex event processing for the real-time processing part, by designing reusable escalating disruption management layers for the online optimization part, and by developing the senslet concept for flexibly gathering data by a mobile application. However, adaptions at one component still make changes of the other components necessary and all involved experts need to contribute.

Hence, as one direction of further research we aim at improving the flexibility on the level of the whole system stepwise: at the end of this process, we want to empower the business users to configure the components—especially the real-time processing and online optimization part—during runtime. Thus, the adaptability to new logistical concepts, processes, tariff structures and industries will be ensured.

Furthermore, improvements on the level of components are planned: regarding the interaction with users we intend to expand the senslet model and increase the powerfulness by introducing a rule language for the description of business processes. By means of this rule language we will be able to generate more automatic senslet reports, which again will improve the disruption management by giving more insights during tour execution.

As already mentioned in the preceding section, the field experiments need to be broadened and complemented by an extensive simulation study based on the filed data and additional data collected from the partners. This is especially important for a broad evaluation of the proposed decision models and heuristics.

Acknowledgements This work is part of the ProveIT project, funded by the German Federal Ministry for Economic Affairs and Energy (19G13009D). We would like to thank our project partners Geis Transport und Logistik GmbH, Institute for Material Handling and Logistics at KIT, LOCOM Software GmbH, PTV Group, Robert Bosch GmbH, and ZF Friedrichshafen AG. Furthermore, we thank the anonymous referees for their valuable comments.

References

Abecker A, Braun S, Valikov A, Zacharias V (2012) Towards a technology for participatory sensing applications. P. & M. Cunningham (Hrsg.): eChallenges

Baader A, Montanus S (2008) Transparency in global supply chain networks—methods and tools for integrated supply chain event management. Strategies and tactics in supply chain event management. Springer, Berlin, pp 3–11

Baumgrass A, Dijkman R, Grefen P, Pourmirza S, Völzer H, Weske M (2014) A software architecture for a transportation control tower. BETA Working Paper 461, BETA Research School, Eindhoven

Becker T (2008) Supply chain event management: innovation in logistics services. Strategies and tactics in supply chain event management. Springer, Berlin, pp 3–11

Behdani B, Adhitya A, Lukszo Z, Srinivasan R (2012) How to handle disruptions in supply chains – an integrated framework and a review of literature. SSRN Electron J

Boschian V, Paganelli P (2016) Business models for advanced ICT in logistics. In: Lu M, De Bock J (eds) Sustainable logistics and supply chains, contributions to management science. Springer International Publishing, pp 15–51

Cabanillas C, Baumgrass A, Mendling J, Rogetzer P, Bellovoda B (2014) Towards the enhancement of business process monitoring for complex logistics chains. In: Business Process Management Workshops. Springer, pp 305–317

Chen HK, Hsueh CF, Chang MS (2006) The real-time time-dependent vehicle routing problem. Transp Res Part E Logist Transp Rev 42(5):383–408

Crainic TG, Gendreau M, Potvin JY (2009) Intelligent freight-transportation systems: assessment and the contribution of operations research. Transp Res Part C Emerg Technol 17(6):541–557

Diessner P, Rosemann M (2008) Supply chain event management: managing risk by creating visibility. Strategies and tactics in supply chain event management. Springer, Berlin, pp 3–11

Feldman Z, Fournier F, Franklin R, Metzger A (2013) Proactive event processing in action: a case study on the proactive management of transport processes (industry article). In: Proceedings of the 7th ACM international conference on distributed event-based systems, DEBS '13. ACM, New York, pp 97–106

Gendreau M, Ghiani G, Guerriero E (2015) Time-dependent routing problems: a review. Comput Oper Res 64:189–197

Heinecke G (2013) Resilient automotive production in vulnerable supply networks: a supply chain event management system. Ph.D. thesis

Hossbach B, Seeger B (2013) Anomaly management using complex event processing: extending data base technology paper. In: Proceedings of the 16th international conference on extending database technology, EDBT '13. ACM, New York, pp 149–154

Laporte G (2009) Fifty years of vehicle routing. Transp Sci 43(4):408–416

Luckham DC (2001) The power of events: an introduction to complex event processing in distributed enterprise systems. Addison-Wesley Longman Publishing Co., Inc, Boston

Melgarejo PA, Laborie P, Solnon C (2015) A time-dependent no-overlap constraint: application to urban delivery problems. In: 12th international conference on integration of AI and OR techniques in constraint programming (CPAIOR 2015). Springer, pp 1–17

Metzger A, Franklin R, Engel Y (2012) Predictive monitoring of heterogeneous service-oriented business networks: the transport and logistics case. In: SRII global conference (SRII), 2012 Annual, pp 313–322

Meyer A (2015) Milk run design: definitions, concepts and solution approaches. Ph.D. thesis, Karlsruher Institut für Technologie

Pillac V, Gendreau M, GuTret C, Medaglia AL (2013) A review of dynamic vehicle routing problems. Eur J Oper Res 225(1):1–11

Projektgruppe Standardbelieferungsformen (2008) Standardbelieferungsformen der Logisitik in der Automobilindustrie. In: Recommondation 5051. VDA—German Association of the Automotive Industry

Pulter (Kleiner) N, Nimis J, Lockemann PC (2010) Managing contingencies in timed transportation networks by agent technology. In: Proceedings of the workshop on artificial intelligence and logistics (AILog-2010)

Schmid V, Doerner KF, Laporte G (2013) Rich routing problems arising in supply chain management. Eur J Oper Res 224(3):435–448

Stadtler H (2015) Supply chain management and advanced planning : concepts, models, software, and case studies. Springer texts in business and economics, 5th edn. Springer, Berlin

Wagner SM, Neshat N (2010) Assessing the vulnerability of supply chains using graph theory. Int J Prod Econ 126(1):121–129

Wang X, Ruan J, Shi Y (2012) A recovery model for combinational disruptions in logistics delivery: considering the real-world participators. Int J Prod Econ 140(1):508–520

Zimmermann R, Winkler S, Bodendorf F (2006) Agent-based supply chain event management—concept and assessment. In: Proceedings of the 39th Hawaii international conference on system sciences