Towards the Enhancement of Business Process Monitoring for Complex Logistics Chains*

Cristina Cabanillas[1], Anne Baumgrass[2], Jan Mendling[1], Patricia Rogetzer[3], and Bruno Bellovoda[3]

[1]Institute for Information Business at Vienna University of Economics and Business, Austria {cristina.cabanillas,jan.mendling}@wu.ac.at
[2]Hasso Plattner Institute at the University of Potsdam, Germany Anne.Baumgrass@hpi.uni-potsdam.de
[3]Institute for Production Management at Vienna University of Economics and Business, Austria {patricia.rogetzer,bruno.bellovoda}@wu.ac.at

Abstract. Logistics processes have some characteristics which are fundamentally challenging from a business process management perspective. Their execution usually involves multiple parties and information exchanges and has to ensure a certain level of flexibility in order to respond to unexpected events. On the level of monitoring, potential disruptions have to be detected and reactive measures be taken in order to avoid delays and contract penalties. However, current business process management systems do not exactly address these general requirements which call for the integration of techniques from event processing. Unfortunately, activity-based and event-based execution paradigms are not thoroughly in line. In this paper, we untangle conceptual issues in aligning both. We present a set of three challenges in the monitoring of process-oriented complex logistics chains identified based on a real-world scenario consisting of a three-leg intermodal logistics chain for the transportation of goods. Required features that such a monitoring system should provide, as well as related literature referring to these challenges, are also described.

Key words: business process management system, process monitoring, complex event processing, information flow in international logistics, logistics process

1 Introduction

The processes related to logistics chains have considerable differences with respect to processes realized in other domains. On the one hand, these processes are flexible, esspecially due to the fact that unexpected events can occur at any moment in the transportation process, e.g., due to accidents or to unfavorable

* The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement 318275 (GET Service).

weather conditions. On the other hand, logistics processes are often complex because of different means of transportation and/or various parties being involved, and a large amount of information being exchanged among the different parties. The resources involved are not only human resources (i.e., people), but also a variety of non-human resources are needed to assist in the transportation chain or in the exchange of information (e.g., cranes, Global Positioning System (GPS) devices, transponders). Specifically, the information exchange between dependent resources has to be ensured.

These special characteristics have an impact on the completion of all the phases of the business process lifecycle in the logistics domain. Grounding on the process lifecycle described by Dumas et al. [1], discovering and modeling such processes can be cumbersome: (i) expressive modeling notations supporting exception handling functionalities would be required; and (ii) the resulting process models could be large and difficult to read. Therefore, in order to implement and monitor the execution of logistics processes, special features are also required by the process engine and monitoring systems used. These features are mainly related to monitoring the collaboration among resources and handling of complex and unexpected events originating from different resources, so integrating functionality of Complex Event Processing (CEP) engines [2] into Business Process Management Systems (BPMSs) is required [3].

In this paper, we focus on the monitoring of complex logistics chains using process models as the mechanism for process execution, assuming that the previous lifecycle phases have already been addressed. Furthermore, we assume there is a system capable of capturing and processing events that occur during the transportation chain to monitor the information exchange among parties. Based on real processes discovered in the context of the EU-FP7 GET Service project[1], we describe a set of challenges to be faced for the monitoring of complex logistics processes. In particular, we look into monitoring from three perspectives, namely the monitoring of the status of the process, the monitoring of activities based on events, and the monitoring of the cargo being transported in the logistics chain. For each of these challenges, we provide an illustration and a conceptual description of the problem, a functional description referred to the monitoring system functionalities, and related work describing a similar problem in different application domains and solutions suggested.

In Section 2, we introduce the basic concepts handled in complex logistics chains by describing a real-world scenario. Section 3 describes the peculiarities of the monitoring of complex logistics processes. Section 4 presents the challenges identified, including the aforementioned information for each of them. Finally, conclusions and future work are summarized in Section 5.

2 Introduction to Complex Logistics Chains

In the following, we describe a real scenario that allows us to define common terms in complex logistics chains and to study the requirements for monitoring

[1] `http://getservice-project.eu/`

the related processes. It consists of a logistics chain for the transportation of goods from the *client*'s warehouse in Austria to one of its distribution warehouses in Romania [4]. There is a *Third Party Logistics Provider (3PL)* in charge of organizing and controlling the whole transportation according to the *Service Level Objectives (SLOs)* [5] defined by the client, which involve information such as the type of goods, amount, departure and delivery locations, and due time of arrival. Other *planners* participate in such a transportation too which is usually performed as depicted in Fig. 1.

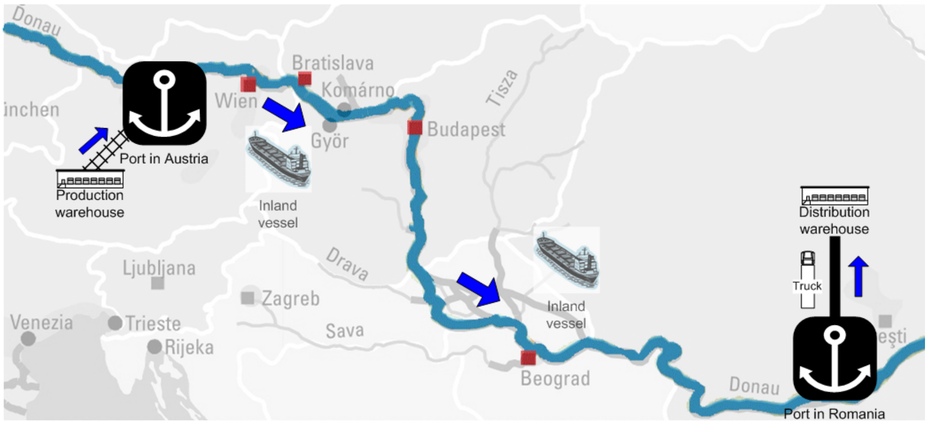


Fig. 1: Inland waterway transportation from Austria to Romania

A train picks up the goods in the production warehouse and takes them to the port. At the *terminal* of the harbor, the *cargo* is loaded onto a vessel and transported to the port in Romania via the Danube river, traversing several countries (some of them not belonging to the EU customs, e.g., Serbia) and a total of ten locks. In each lock, the vessel must wait until the water level is regulated and the captain of the vessel receives permission to continue the transportation. Information about water level, traffic, opening and closing hours for locks, and updates on the *Estimated Time of Arrival (ETA)* can be received at any moment from an information and management technology service called River Information Services (RIS) [2]. Once in the port in Romania, a truck picks up the goods from the *terminal* and drives them to the distribution warehouse. In every *transportation leg*, the drivers of the vehicles and the captain of the vessel, must carry a set of documents called *waybills* [5] that contain the requirements and transportation information according to the contract for that specific transportation leg, also known as *forwarding instructions* [5]. These documents have to be presented upon request, e.g., in case of an inspection. The transportation chain finishes

[2] RIS is a framework of compatible systems across Europe focusing on safety and traffic aspects of inland waterway transportation. Further information can be found here: `http://www.ris.eu/`.

when an operator at the distribution warehouse reviews the freight and accepts the shipment. The process ends when payment and invoices are handled.

This logistics chain is complex and requires strong collaboration among all the parties involved due to several reasons. Firstly, it involves three transportation legs, that is, several well-limited steps in the transportation chain[3]. Secondly, there are different *transportation modes* (also called *means of transportation*) used throughout the process, namely railway, inland waterway, and road transportation. In particular, it is an *intermodal* logistics chain because there are several transportation modes involved but the *transportation unit* is not changed in the process, being it always units of goods. Lastly, more than thirty activities are performed by at least twelve different *stakeholders*, including the client, four *planners*, and seven *operators* belonging to different *shipping companies*, according to the process discovered [1].

Complex logistics chains involving many parties require much information exchange between different participants as well as punctual delivery for each transportation leg. The goal is to achieve the so-called Complete and On-Time Delivery (COTD) [6], that is, to transport the entire cargo from origin to destination, meeting the conditions agreed between *client* and *planner* in the SLOs. For that purpose, a number of systems must be used when required, e.g., Advanced Planning Systems (APSs), Transportation Management Systems (TMSs), and Intelligent Transport Systems (ITSs). In the case at hand, RIS is one of them. GPSs, transponders, and similar devices are also necessary. As a consequent, achieving reliable information exchange depends on the connection and collaboration between the parties and the systems involved. Thus, the events (e.g., positioning information) that are produced by the systems used in a logistics chain must be processed and appropriately distributed among the participants. Furthermore, proper reaction mechanisms to *disruptions* caused by unexpected events must be put into place and triggered when disruptions are detected. Altogether, reliable communication and reaction to disruptions in logistics chains depend on the identification and distribution of events as well as the correlation and aggregation of events to activities in the corresponding logistics process.

The business process for our logistics chain has been modeled in Business Process Model and Notation (BPMN) 2.0 [7], giving rise to the collaboration between participants represented in twelve pools, with more than thirty activities in total and plenty of messages exchanged. Fig. 2 shows an abstract representation of such a BPMN model. We will use some of the activities there represented as example throughout this paper.

3 Process Monitoring in Logistics

Traditional BPMSs allow for modeling, executing, and analyzing business processes [1]. Each system requires an explicit business process model, e.g., modeled via BPMN, to enforce the execution of tasks by the right person at the right

[3] Please note that *logistics chain* refers to technical and organizational activities, whereas *transportation chain* disregards the latter.

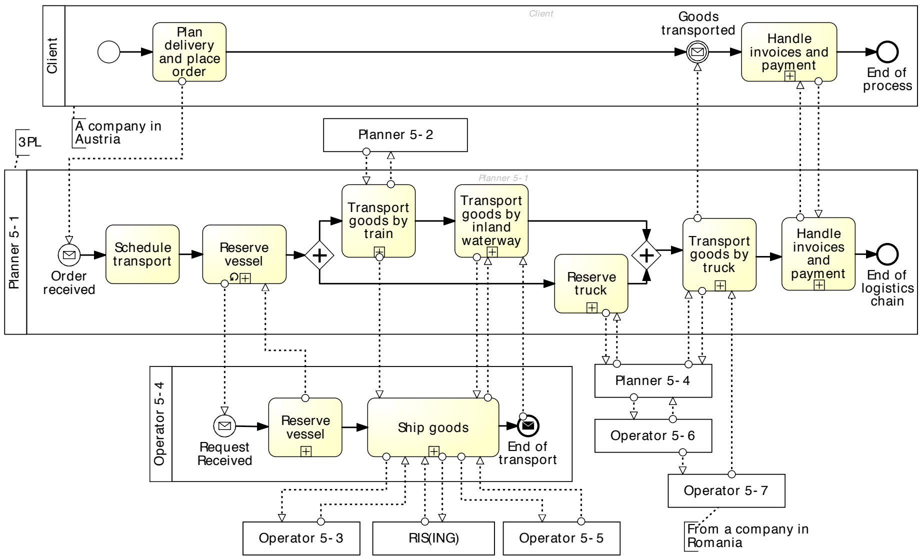


Fig. 2: Excerpt of the process model for inland waterway transportation

time using all necessary non-human resources. Our example in Section 2 shows that logistics chains demand information exchange among many parties since activities are executed across enterprise boundaries which in turn involve the need of having several different systems directly connected to a BPMS for controlling and monitoring the complete logistics chain. BPMSs are strong in coordinating and tracing discrete state changes of a business process. However, several logistics activities unfold in a continuous way, such as transportation which involves a continuous change of positions.

Recently, approaches have emerged that allow processing the great amount of events (e.g., positioning information) and at the same time permit using events as a basis for information exchange in inter-organizational processes (see, e.g., [3, 8–10]). These approaches inspired our work to use events and derive information to manage, control, and monitor inter-organizational logistics chains. The detection and processing of events originating from different systems can be handled by so-called CEP engines [11, 12]. While BPMS operate on the basis of business process models, CEP engines process events based on event patterns. An event pattern describes the structure, causal dependencies, timing, data parameters, and context of events formalized via an Event Processing Language (EPL), e.g., Esper [13]. In the context of an Event-Driven Architecture (EDA), a CEP engine can consume, process, and publish/emit events from and to different systems, however, without controlling or monitoring the execution of a complete logistics chain. For logistics, a BPMS must be extended with CEP capabilities to allow the execution of logistics steps while managing the collaboration among different parties by integrating information coming from different systems. In other words, BPMS and CEP engines must be integrated to aggregate and correlate events to business process activities in order to enable the control and monitoring of a complete logistics chain.

In the context of CEP, the technical consumption, processing, and distribution of events in event-based systems has been discussed and is covered by existing approaches (see, e.g, [3, 11, 12]); however, less focus has been put on automated event handling in business processes in real-world scenarios. Especially in the context of logistics, we see three major challenges on the conceptual level for processing events to capture and monitor logistics chains, derived from the following three questions:

1. How can streaming events be mapped to state changes of activities?
2. How can events be aggregated to different activities?
3. How can events be correlated to different units of observation?

These challenges are discussed in detail in the following section.

4 Challenges in Logistics Chain Monitoring

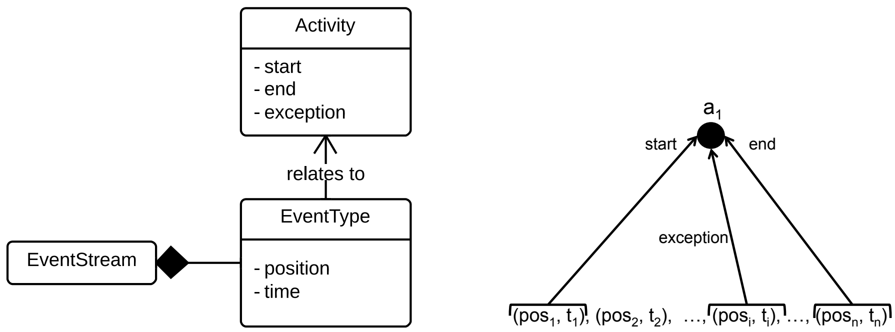
For the specification of a monitoring service for inter-organizational logistics processes, we have identified three major challenges based on the study of the scenario outlined in Section 2. In the following, for each challenge we describe and illustrate the underlying problem with reference to our scenario, define its conceptual problem, and summarize related research that provides a basis for tackling it.

4.1 Discretization for Monitoring Status based on Streaming Events

The first challenge relates to a gap between how transportation operations can be observed and how state changes are typically represented in business process models. Transportation operations unfold as a continuous movement of physical objects. In contrast to this, state transitions in a business process are discrete. For example, in subprocess "Railway transportation" in Fig. 2 there is an activity named "Take goods to port" in which the train driver performs the same activity during the entire activity execution, breaks to rest or put petrol apart. When we trace this activity, for instance using GPS sensor information, we receive a continuous stream of events related to geographic positions. This event stream per se does not inform us about the start and end of this activity, nor about exceptions related to potential problems occurred during the shipment.

The challenge is here to appropriately align continuous event streams with discretionary state changes. Specifically for monitoring, this entails the following problems of identifying start and end, as well as exceptions, as depicted in Fig. 3. First, additional information is required in order to measure the progress of a transportation activity. This requires traceability of where assets are at which moment in time (pos_i, t_i). Such information can be obtained from the

GPS coordinates of the train (truck, or vessel) that are sent from some device attached to the vehicle or carried by the driver. Based on this type of data, a clear and explicit definition of start and end conditions have to be provided. In case such a definition is not possible, a human agent has to be involved in order to confirm start and end. Second, the event stream has to be continuously analyzed in order to notice exceptional behavioral or potential problems. Again, this requires the identification of exceptional events and corresponding conditions or patterns. If not all exceptional events can be defined in such a way, a human agent has to be involved.



(a) Metamodel (b) Problem illustration

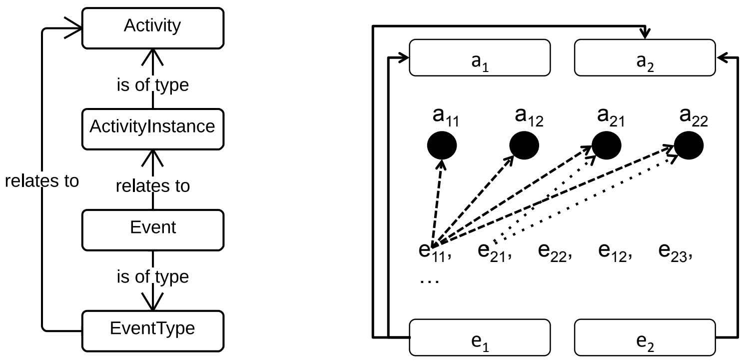
Fig. 3: Monitoring status based on event streams

In order to avoid such problems, a discretization of the transportation chain is required. That means that criteria have to be established regarding the types of events that need to be received during continuous activities. Ideally, those types of events would refer to information of interest for the track and trace of the process execution, e.g., when a vehicle has achieved a percentage of the total distance, when the ETA has exceeded the due arrival time, or when the vehicle is stopped for an unexpected period of time. Therefore, the system should contain knowledge about the events to be taken into consideration and their potential consequences in the transportation process.

As aforementioned, the described challenge can be related to the problem of discretizing the transportation chain. In [14], Zaharia et al. introduce a new programming model called *Discretized streams (D-Streams)* that treats a streaming computation as a series of deterministic batch computation on small time intervals, thus lowering the event computation frequency of typical *record-at-a-time* processing models. The proposal by Appel et al. introduces event stream processing units as a conceptual frame for integrating complex event processing into BPMS [15]. In this concept, event streaming is a subordinate concept that can be started and completed within a classical workflow paradigm. Further modeling concepts are presented in [16, 17].

4.2 Aggregation for Monitoring Activities based on Fine-Granular Events

The second challenge relates to the fact that logistics operations provide an extensive amount of low-level event data. Therefore, activity monitoring requires the ability to automatically aggregate events to the activity instances of a business[4], in order to track and trace the process execution. Some logistics operations share part of the actions that are necessary to complete them, hence, there are events that can correspond to different process activities. For instance, several activities of the business process in Fig. 1 might share the action of creating a new order, e.g., activities contained in the "Reserve vessel" and "Reserve truck" subprocesses of pool "Planner 5-1". As a consequence, it is not only necessary to associate event types with activities (cf. Section 4.1) but also to associate specific events with activity instances, as illustrated in Fig. 4a.



(a) Metamodel (b) Problem illustration

Fig. 4: Monitoring coarse-granular activities based on fine-granular events

The challenge is here to deal with the different granularity levels between activities and events, i.e., activities provide less details than events. On the one hand, an event type can be associated with several activities. On the other hand, an event can be associated with running instances of all the activities potentially related to this event type. Fig. 4b contains an explicit representation of the problem. Events of type e_1 (e_{11}, e_{12}) can occur during the execution of activities a_1 and a_2, and events of type e_2 (e_{21}, e_{22}, e_{23}) are always related to instances of activity a_2. Thus, for each appearance in the event stream there are several aggregation options. In particular, e_{11} could refer to the four activity instances represented in the figure, e_{21} could be associated to the two instances of activity a_2, and so forth.

[4] In this paper, we define *aggregation* as a mapping of one or more events to one or more activity instances.

In order to deal with this issue, the system should support the cumbersome task of automatically aggregating events to activity instances during process execution. Sometimes, some events can be directly left aside for aggregation to an activity instance because of referring to a past state in the execution of the activity. For example, let us assume that we have three event types: e_1 indicates the start of an activity, e_2 the execution process, and e_3 its completion. Then, if we find an event e_{1i} that may be related to several activity instances, the activity instances a_{1i} for which an event of type e_1 has already been identified, can be disregarded, as event type e_2 is expected for them. Nonetheless, the problem persists for any other activity instance of type a_1 that has not yet started. In those occasions, decisions have to be made on how to associate events to activity instances. Appropriate heuristics are required in order to reduce the error margin in the aggregation.

Baier and Mendling state that an event to activity mapping is always a combination of both a mapping on type and a mapping on instance level [18]. They look at this challenge from the perspective of event logs. In particular, they assume the event logs contain specific information about each event, namely its name, the time when it occurred, and its transaction type in terms of whether the event has been completed or not; and they provide insights of all the possible mappings at type and instance level. For event to activity instance aggregation, they propose several heuristics for the definition of the *instance border conditions* such as the maximal distance between two events that belong to one activity instance. Clustering of events to activities is also discussed in other work. Günther, Rozinat and van der Aalst cluster events to activities based on time and position distance [19]. This approach is enhanced by considering co-occurrence of terms in [20].

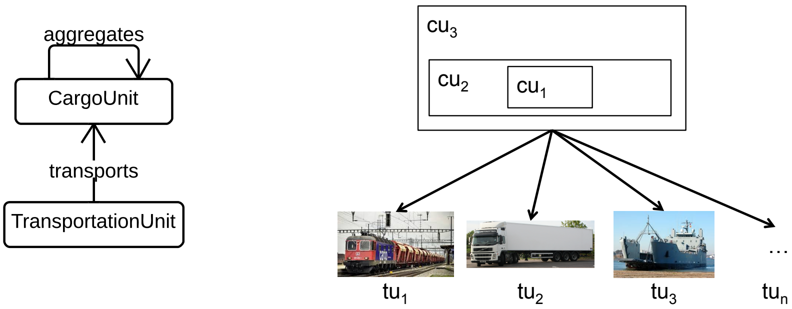
4.3 Correlation for Monitoring Cargo based on Events of Different Focus

The third challenge relates to the fact that cargo might be bundled, unbundled and rebundled during a *multimodal* transportation[5] activity, i.e., the so-called *focus shift* [21]. Goods are usually grouped and bundled into pallets, which are in turn distributed among containers, so the cargo cannot be considered as a single entity. Such bundling and unbundling tasks can be performed at different stages of the transportation chain. It does not apply to our application scenario because it uses intermodal transportation.

The challenge in this case is to keep track of which *cargo unit* (*cu*) is loaded on which vehicle or vessel, which is outside the scope of process models, as they show an abstract representation of the general behavior of the process (i.e., for several executions) but cannot deal with such variability. Technically, this means that a multi-level containment hierarchy has to be stored for all legs of the transportation, as depicted in Fig. 5a. This also implies that there is potentially a 1:n relationship between process instance and transportation

[5] Use of two or more transportation modes changing the transportation unit.

operations depending on the unit of observation (i.e., the *transportation unit* (tu)). If the vessel is the unit of observation, the transportation of the whole cargo from A to B relates to a single process instance. If each container is considered as a unit of observation, the same transportation relates to the process of each container (cf. Fig. 5b).



(a) Metamodel (b) Problem illustration

Fig. 5: Monitoring cargo based on events of different focus

In order to address this problem, the different correlation levels at which cargo will be monitored must be defined, and the track and trace information must be updated after every bundling, unbundling or rebuilding operation. For instance, if bottle crates were the goods being transported in the scenario described in Section 2, in the production warehouse each bottle crate would be tracked and traced separately, i.e., the cargo unit would be a single bottle crate. Once several bottle crates were collected to be delivered as part of a transportation order, they would have to be considered as a whole, i.e., the cargo unit would be a group of bottle crates. Therefore, COTD would only be achieved if every single bottle crate belonging to that cargo unit, arrived at the destination point on time. Regarding the transportation unit to carry the freight in the as-is application scenario, it is likely to be a single train in the railway transportation leg. Similarly, the transportation unit in the inland waterway transportation is a unique vessel. However, if a large amount of cargo is being delivered, several trucks might be required to take the goods from the port to the destination warehouse. Thus, keeping track of every truck involved, is required.

As stated by Werner and Schill [5], modern Radio Frequency Identification (RFID) and sensor technologies enable the automatic identification of tagged items by eligible readers in combination with environmental information and thus can be used to monitor SLOs. Making use of this, they identify seven requirements for the monitoring of individual quality objectives for goods transportation using distributed event data, and describe the architecture of a corresponding monitoring system. However, the problem of focus shifts on cargo described above is not explicitly considered. Gerke et al. [21] investigate how the

EPCglobal standard[6] for processing RFID can be utilized to construct supply chain case information from event logs, pointing out focus shift as key challenge in their approach. Besides such cargo monitoring mechanisms, some software must be implemented dealing with the monitoring of the transportation unit with which cargo is being transported. In [22], Patroumpas and Sellis define concepts for real-time traffic surveillance over densely congested road networks in large metropolitan areas. Their monitoring dashboards can also show current weather conditions, which may negatively affect the ETA. Further software solutions for the monitoring of events in logistics chains are, for instance, ProModel[7], TIBCO[8] and APAMA[9].

5 Conclusions and Future Work

Complex logistics processes require dedicated support of BPMS in order to monitor the expected execution and the occurrence of unexpected events. However, there are conceptual challenges for integrating event processing and business process management. This paper provides a basis for the conceptual enhancement of BPMS with CEP functionality for supporting logistics processes. Based on the case of a real-world logistics chain, we identified three major challenges that make the monitoring of such processes difficult. These include the discretization of streaming events, the aggregation of fine-granular event sets to activities, and the correlation of events that relate to the same cargo unit. For each of these challenges, we discussed related research contributions. This discussion reveals that there is currently no approach available that deals with these challenges in an integrated manner. In future work, we aim to address this research gap. More specifically, we will build a system that helps in discretizing, aggregating, and correlating events such that the overall business process can be traced.

References

1. M. Dumas, M. L. Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management.* Springer, 2013.
2. G. Mühl, L. Fiege, and P. Pietzuch, *Distributed Event-Based Systems.* Springer, 2006.
3. M. Daum, M. Götz, and J. Domaschka, “Integrating CEP and BPM: how CEP realizes functional requirements of BPM applications (industry article),” in *Proc. of the 6th ACM International Conference on Distributed Event-Based Systems (DEBS)*, 2012.
4. S. Treitl, P. Rogetzer, M. Hrusovsky, C. Burkart, B. Bellovoda, W. Jammernegg, J. Mendling, E. Demir, T. van Woensel, R. Dijkman, M. van der Velde, and A.-C. Ernst, “GET Service Project Deliverable 1.1: Use Cases, Success Criteria and Usage Scenarios,” 2013.

[6] http://www.gs1.org/epcglobal

[7] http://www.promodel.com/solutions/logistics/

[8] http://www.tibco.com/industries/logistics/default.jsp

[9] http://www.progress.com/en/apama/apama-solutions.html

5. K. Werner and A. Schill, “Automatic Monitoring of Logistics Processes using Distributed RFID based Event Data,” in *Proc. of the Third International Workshop on RFID Technology (IWRT)*, 2009.
6. APQC, *Blueprint for Success: Logistics.* 2 ed., 2011.
7. “BPMN 2.0,” Recommendation, OMG, 2011. Available at: `http://www.omg.org/cgi-bin/doc?dtc/09-08-14.pdf`.
8. S. Rozsnyai, G. T. Lakshmanan, V. Muthusamy, R. Khalaf, and M. J. Duftler, “Business Process Insight: An Approach and Platform for the Discovery and Analysis of End-to-End Business Processes,” in *SRII Global Conference (SRII), 2012 Annual*, IEEE, 2012.
9. N. Herzberg and M. Weske, “Enriching Raw Events to Enable Process Intelligence - Research Challenges,” Tech. Rep. 73, HPI at the University of Potsdam, 2013.
10. M. Roth and S. Donath, “Applying Complex Event Processing towards Monitoring of Multi-party Contracts and Services for Logistics - A Discussion,” in *BPM 2011 Workshops*, vol. 99 of *LNBIP*, 2012.
11. O. Etzion and P. Niblett, *Event Processing in Action.* Manning Publications Co., 2011.
12. D. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems.* Addison-Wesley, 2002.
13. EsperTech, “Esper - Complex Event Processing,” as of May 2013. Available at: `http://esper.codehaus.org`.
14. M. Zaharia, T. Das, H. Li, S. Shenker, and I. Stoica, “Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters,” in *Proc. of the 4th USENIX conference on Hot Topics in Cloud Computing (HotCloud)*, (Berkeley, CA, USA), 2012.
15. S. Appel, S. Frischbier, T. Freudenreich, and A. P. Buchmann, “Event Stream Processing Units in Business Processes,” in *Proc. of the 11th International Conference of Business Process Management (BPM)*, Lecture Notes in Computer Science, Springer, 2013.
16. S. Kunz, T. Fickinger, J. Prescher, and K. Spengler, “Managing Complex Event Processes with Business Process Modeling Notation,” in *BPMN* (J. Mendling, M. Weidlich, and M. Weske, eds.), vol. 67 of *LNBIP*, Springer, 2010.
17. A. Caracas and T. Kramp, “On the Expressiveness of BPMN for Modeling Wireless Sensor Networks Applications,” in *BPMN* (R. M. Dijkman, J. Hofstetter, and J. Koehler, eds.), vol. 95 of *LNBIP*, Springer, 2011.
18. T. Baier and J. Mendling, “Bridging Abstraction Layers in Process Mining by automated Matching of Events and Activities,” in *Proc. of the 11th International Conference of Business Process Management (BPM)*, Lecture Notes in Computer Science (LNCS), Springer, 2013.
19. C. W. Günther and W. M. P. van der Aalst, “Mining Activity Clusters from Low-Level Event Logs,” in *BETA Working Paper Series*, vol. WP 165, Eindhoven University of Technology, 2006.
20. C. W. Günther, A. Rozinat, and W. M. P. van der Aalst, “Activity Mining by Global Trace Segmentation,” in *BPM Workshops*, 2009.
21. K. Gerke, A. Claus, and J. Mendling, “Process Mining of RFID-Based Supply Chains,” in *Proc. of the 2009 IEEE Conference on Commerce and Enterprise Computing (CEC)*, 2009.
22. K. Patroumpas and T. Sellis, “Event processing and real-time monitoring over streaming traffic data,” in *Web and Wireless GIS*, 2012.