Event Correlation for Business Processes on the Basis of Ontologies

Tobias Metzke
Hasso-Plattner-Institute at the University of Potsdam, Germany
tobias.metzke@student.hpi.uni-potsdam.de

Abstract: Business process execution generates great amounts of information on process results and intermediate activities. Most of this information is represented by events linked to their related process execution. Process monitoring uses such events and links to correlate incoming events to their corresponding process instances. These links may however get lost in the process of event capturing or not even be present in distributed process environments like logistics. Nonetheless, monitoring depends on the correct correlation of events in such scenarios. Furthermore, the correlation of external data like weather and traffic information, which has no connection to process executions, can be useful in the planning, execution and monitoring of processes as well. The approach presented in this paper uses semantic technologies to automatically identify those process executions that are related to the data of an occurring event. It uses linked data principles and graph-based algorithms to detect relatedness of events and process instances. The approach allows for the inclusion of external data and the correlation of external events without relying on process specific queries.
Keywords: Event Correlation, Semantic Event Processing, Business Processes

1 Introduction

Process monitoring is an essential method for improving a company's processes and procedures. In fully automated process environments the monitoring can be managed by process engines that keep track of triggered events and their origin. As stated in [HMW13], such logging mechanisms are however not always available and the correlation of events needs to be done based on the processes' context data like associated transport plans, vehicle drivers, and transported goods.

Furthermore, external information (e.g. weather information that is provided by an external weather service) can be valuable for the monitoring and execution of a process. However, there is no native connection to their corresponding process instances. The necessary background data for the correlation of such information can be organized in a hierarchical manner as well as it can be changing over time, which makes the correlation of external information to processes even more complex.

In order to correlate incoming events to existing process instances, current approaches use specific queries or rules that are linked to process instances as shown

A. Baumgraß, N. Herzberg, G. Kappel, J. Mendling, A. Meyer, S. Rinderle-Ma (eds.): Proceedings on Inter-Organizational Process Modeling and Event Processing in Business Process Management, Vienna, Austria, 12-06-2014, published at http://ceur-ws.org

in [HMW13]. Incoming events are then queried against the designed rules and queries in order identify relevant events, extract information from them, and provide the extracted information to the instances. Query designers can thus be forced to gain profound background knowledge of the domain before writing complex queries and procedures to ensure that all relevant event data is connected to the relevant process executions. This can complicate the inclusion of arbitrary event sources and hinder the dynamic consideration of new sources without updating existing routines and queries. The approach presented in this paper thus includes the following contributions:

Graph-based correlation. The presented approach operates on semantic graphs in order to identify process executions that need to be informed about incoming event data. The search is thereby directed from event data to relevant process instances. It uses path detection in graphs to identify relatedness of events and instances. Semantic filters furthermore improve the precision of the approach compared to native path finding.

Independence from process-specific queries. The approach does not rely on process-specific correlation queries or routines and eliminates the need for query updates and extensions when new event sources are added or background knowledge changes. It rather employs path finding in knowledge graphs to identify processes related to event data. Furthermore, events from new sources will directly be considered in the correlation process. The approach can also be used as a complement to traditional query-per-process based event correlation.

This work is structured as follows: Section 2 provides introductory information on basic terms from the fields of semantic technologies and business processes. Afterwards, Section 3 introduces use cases that further outline the need for a new approach before Section 4 details the taken approach that helps identifying the relevant process instances for incoming data events. Section 5 then presents one prototypical implementation of the approach, after which Section 6 positions the approach in the field of event correlation. Section 7 then concludes the paper.

2 Background

The approach presented in this paper bases on the concepts of process models and process instances. Based on the definition provided in [Wes12], a process model can be described as a directed graph, containing a set of nodes and a set of edges, whereas the edges represent the control flow in the model. Based on this definition of a process model, an instance of a process in [Wes12] is defined as a partially ordered set of events which contains events for all node instances of the corresponding process model. The events are ordered according to the execution constraints defined in the model.

In this work, the first outline of a semantic-based correlation approach details how to use these concepts in combination with semantic technologies in the search for process instances that correlate to the information of occurring events. Especially, the basic principles of a *semantic knowledge base* and a *knowledge graph* play a major role in the approach.

A *semantic knowledge base* SKB is a set of statements *(subject, predicate, object)*, with subject and predicate being *Uniform Resource Identifiers* (URI)[1], and the object being a string expression or a URI. The subject and the object of a statement are also called **semantic entities**. Every semantic entity is of a certain *type* that is assigned to it by the property **rdf:type**[2]. This connection is a standard property that is declared as best practice when working with semantic knowledge.

The language that is used to describe an SKB is often based on Description Logic (DL). The knowledge described by DL can be divided into a *TBox* (terminological box) and an *ABox* (assertional box), where the TBox describes the concept hierarchies while the ABox states where individuals belong in this hierarchy. Furthermore, DL allows the creation of *restrictions* and *rules* on concepts and individuals that enable the deduction of new knowledge from the concepts described in an SKB with existing reasoning tools like Pellet[3]. Beyond that, the concepts and individuals of an SKB can be depicted in a *semantic knowledge graph*.

A *semantic knowledge graph* is *directed graph* representation of a semantic knowledge base. Subjects and object in the knowledge base build the set of vertices of the graph, connected by directed edges from subject to object.

In the context of event correlation, a semantic knowledge base and its corresponding graph provide a way to model concepts like *processes*, *process instances*, and *events* in a formal, explicit, and unambiguous way. The concepts then hold a defined semantic meaning and can be shared, used, and reused between people and software agents.

As described by Lopez et al. [LdCCVG+10], semantic correlation is not limited to syntactically exactly equal values of event attributes and process context data attributes but rather allows a consideration of the semantic meaning of attributes and their relationships between each other. As stated by Lopez et al. [LdCCVG+10], semantic correlation can thus be understood as an evolution of traditional correlation techniques.

[1] The definition of a URI can be found in http://www.ietf.org/rfc/rfc2396.txt, last accessed at 01/17/2014.

[2] The definition can be found in http://www.w3.org/TR/rdf-schema/#ch_type, last accessed at 01/17/2014.

[3] Visit http://clarkparsia.com/pellet/ for more information, last accessed at 03/05/2014

3 Scenarios

This section details three scenarios from the logistics domain. These use cases are examples of current demands and real-world scenarios. For the correlation of incoming events to existing business processes they outline the need for (1) the inclusion of external knowledge that can be hierarchically structured and changing over time (Section 3.1), (2) the consideration of location data (Section 3.2), and (3) the ability to add new event sources that should be automatically considered (Section 3.3).

3.1 Parcel Tracking – Changing Hierarchical Data

Track and trace as described in [VD02, SH11] is one of the most common use cases in logistics. Consider a logistics company, shipping goods worldwide and providing status websites for parcels that allow customers to track their goods. Behind every status website there is a process execution instance connected to the specific parcel ID. The parcels are transported in containers on ships or trucks. The company's information system receives events with information on ships and trucks like their current locations.

Incoming information needs to be checked whether it is relevant for the parcel or not. Either created manually or automatically, a query must include the specific container number the parcel is transported in as well as the ship or truck the container is transported on.

Such a query can work as long as this data does not change. However, in transportation processes, containers are often loaded from ships onto trucks and vice versa. In case of unloading, all queries that are connected to the truck or ship need to updated accordingly. Otherwise, the status website may display the parcel position as if it was still on a the ship, although it is transported by a truck now.

State of the art approaches either require the query designers to have knowledge of the background information (e.g. which parcel is transported in which container on which vehicle) or that queries are build and can operate based on it. Furthermore, they can imply update procedures of queries in case of changes in that background knowledge.

3.2 Weather Information – Location Based Correlation

Consider a logistics company that transports goods with a fleet of ships and trucks worldwide. All their transportation processes are instantiated with a specific transport plan comprising of the used vehicles, their routes, drivers, and estimated arrival times at specific points.

The company's information system receives events that hold information on weather conditions all over the world. In order to identify whether the incoming weather information is valuable for the company, it needs to identify the region affected by this weather condition and evaluate if any of the company's transport plans go through this region.

Current approaches require a check of the event's region against every region of a transport plan. The complexity of such a query depends on the information provided in the event as well as in the transport plan. The better the two data structures match, the simpler the query will be. If the event and the transport plan comprise of location data of the same granularity (e.g. GPS coordinates), the query will be simpler compared to the event holding region information like 'Northern Germany' and the transport plan comprising GPS coordinates only.

3.3 Critical Events – Adding Event Sources

Consider the scenario described in Section 3.2. In order to improve a company's ability to react to critical incidents that may occur near to their transport plans, the logistics company subscribes to a new event source. It publishes events with location information in case of incidents like pirate attacks, road blockages due to riots, forest fires and the like.

For current approaches, either a new query for every process instance needs to be created or existing queries need to be adjusted in the information system dealing with region checks against the new event's region. The rules of complexity are comparable to those mentioned in Section 3.2.

4 Approach

The approach presented in this section enables automatic identification of relevant process executions for occurring events. It is based on *data connectivity* which will be explained in detail in Section 4.1. It furthermore allows the automatic consideration of hierarchical data, changes in that data, and dynamically added event sources.

Throughout the presented work, the focus lies on ABox knowledge that represents individual processes, events, and instances of other public knowledge and their relationships between each other. More specifically, the presented approach focuses on the evaluation of the *semantic knowledge graph* that can be derived from ABox data.

This limitation to a specific part of the knowledge aids in keeping the search space for the identification of processes related to incoming events to a reasonable minimum and will be further supported by the means presented in Section 4.2 and

Section 4.3. These sections present different methods for data exclusion that will be explained in detail for use cases from the logistics domain. Other domains may require different exclusions. Finally, a mechanism for identifying all instances that have a location-based interest in an occurring event will be detailed in Section 4.4.

The approach will not be based on deduction rules or subsumption matchmaking at this point of research. These techniques will play an important role in future work on this topic and need to be compared to the results achieved with the work provided here.

4.1 Data Connectivity

The scenarios shown in Section 3 state a need for the use of external data for sophisticated event correlation to business process executions. This data can also be of a hierarchical nature and change over time.

The approach uses semantic ontology data as a central information database that captures public data, event data, and process data. Public data can comprise different ships and their characteristics, event data stores event attributes like the location of events and their connections to public data like specific ships, and process data holds existing processes, their activities, current running instances, and process context data like a specific ship, truck, and driver. Figure 1 illustrates a knowledge base that stores semantic information on two logistics processes, whose execution instances are connected to context data like ships or parcels. Furthermore, the connections between parcels, ships, and containers are visible.

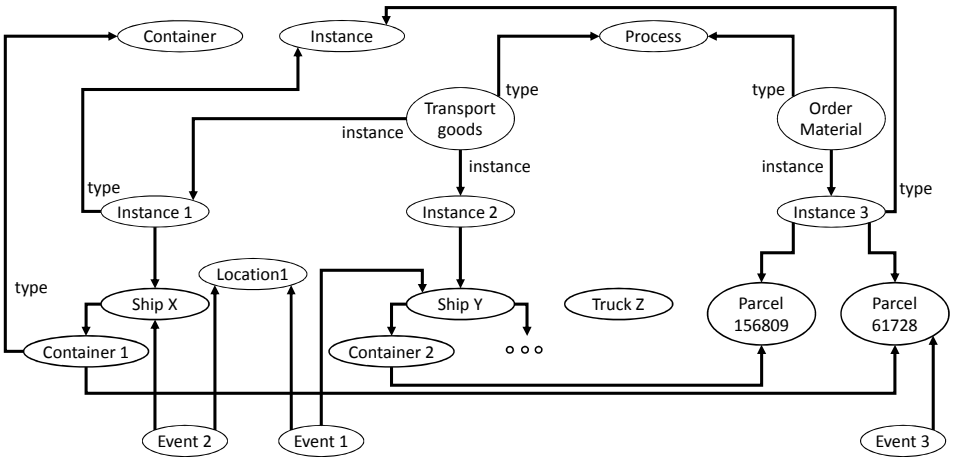

Figure 1: Exemplary semantic knowledge graph containing public data, process data, and event data related to the domain of logistics

Since semantic data is an integral part of this approach, semantic data integration and knowledge engineering as described in [Gar05, ES+07, BBR+11] are the preconditions for the approach to work. The precision of the approach presented in this work depends on up-to-date knowledge especially concerning process data and public data.

The identification of relevant process instances for an event is based on its *data connectivity* to those instances. Only those instances that have a data path to the event's information are considered as relevant. A *path* in graph theory is defined as a walk between two vertices where neither an edge nor a vertex is repeated [Fou92]. Based on this definition, a subject and an object are connected if there is a path between them in the undirected underlying graph of a knowledge graph.

The knowledge base shown in Figure 1 is enriched by an incoming event *Event1*. All process instances that are interested in the *Event1*'s data need to be identified. The correlation is based on the data connectivity between the event and the instances.

In order to identify instances that are related to the event's data, the approach:

1. Retrieves a list of current process instances from the knowledge base.
2. Checks for data connectivity between the event and every instance from the list.
3. Returns all instances that have a data connection to the event.

Regarding the knowledge base in Figure 1, the approach would return *Instance1*, *Instance2*, and *Instance3* as relevant instances that have a data connection to the event. Note that, if changes in the knowledge base occur (e.g. in Figure 1, the *Container 2* is unloaded and transported by *Truck Z*, not *Ship Y*), these updates are directly considered in this approach and therefore ensure an up-to-date result.

However, this approach always identifies all instances as interested in the event's data. Every semantic entity, e.g. the occurring event's entity itself, is of a certain type like *Instance*. These types are often hierarchically ordered in semantic ontologies and inherit from one global entity. Thus, all semantic entities in a knowledge base are connected. Therefore, a connection to all executions can be found if the searchable graph space is not limited.

4.2 Allowed Direction Changes

The problem of *over identifying* (i.e. identifying too many instances as related to the event's data) with the basic data connectivity approach can be tackled by a limitation of the search space. This can be achieved by restricting the allowed number of direction changes on the path. This is a simple method preventing the escalation of the search for instances to all parts of the graph.

In a semantic knowledge graph SKG $= (V, E)$, a direction change between two edges (u, v) and (w, x) can be found if $u \neq x$ *and* $v \neq w$, with $(u, v), (w, x) \in E$ and $u, v, w, x \in V$. The number of direction changes on a path between a subject and an object can thus be counted, with the minimum number of direction changes possible being decisive.

Given the example knowledge base shown in Figure 1, a limitation of the approach to only search for connections that have no direction changes on the path would yield no instance to be interested. A limitation to a maximum of one direction change would lead to the identification of *Instance2* and *Instance3* as relevant. *Instance1* would not be identified as related to the event's data. All allowed numbers greater than one would yield the result achieved by the basic data connectivity approach. Thus, this rather small knowledge base already highlights that it is of great importance how the number of changes is limited. It has not yet been evaluated if it is even possible to always find a suitable restriction for the whole knowledge base.

This task becomes even more complex with growing knowledge base sizes and complexities. Furthermore, this approach is very dependent on the modelling style of the knowledge base and the edge directions. Besides, although this restriction works for some parts of the graph, it might not be suitable for others that for example show a higher rate of direction changes between the data entities.

Beyond all, it does not take the semantics of the edges into account. Some edges are of more interest than others when it comes to searching for relevant instances. How this can be accomplished is shown in Section 4.3.

4.3 Graph Cutting

An alternative approach to limiting the search space of the data connectivity approach is the selective cutting of the knowledge graph. With this concept, the search is intentionally restricted to specific areas of the knowledge base. In the following, four exclusions will be detailed. Regarding the knowledge graph displayed in Figure 1, these restrictions lead to a search space as shown in Figure 2 and results equal to those of a limitation to one allowed direction change on the path.

4.3.1 Meta Level Exclusion

All semantic entities are of a specific *type*. By restricting the search space to exclude meta level entities, reaching other entities of the same *type* (e.g. all other instances or ships) can be avoided. If information is given on one individual (e.g. a specific ship), the other individuals of that type are not of interest just because they are of the same type. They may be of interest as well, but not through this connection. For the approach this implies: when looking at a semantic entity, ignore the outgoing edges named *rdf:type* when looking for a data connection to the event. In Figure 2,

all edges named *type* are therefore marked as *irrelevant* for the search of a data connection.

4.3.2 Process Exclusion

Business processes are connected to their instances. When searching for relevant instances, finding one instance directly leads to the discovery of all sibling instances due to their edge to the parent process. This sibling edge does not qualify an instance for being interested in an event's data, therefore this edge is excluded. For the approach this implies: when looking at an instance, ignore the edge that leads to the parent process[4] when looking for a data connection to the event. In Figure 2, all edges named *instance* are therefore marked as *irrelevant* for the search of a data connection.

4.3.3 Instance Context Exclusion

Instances are connected to context data like a vehicle, a driver, cities, loaded goods and more. When the edge to one entity of this data leads to the identification of an instance as interested in the event, all other context data will be used as a path to search for other instances. This sibling context data edge however does not qualify for the identification of relatedness to the event's data. For the approach this implies: when reaching an instance, ignore the edge that points to the instance's context data[4] when looking for a data connection to the event. In Figure 2, the edge of *Instance3* to *Parcel61728* is therefore marked as *irrelevant* for the search of a data connection.

4.3.4 Former Events Exclusion

The knowledge base is enriched by events and their information over time. These events point to other entities in the knowledge base and therefore insert links between entities. These links however do not qualify for the identification of relevant instances and are thus excluded. For the approach this implies: when looking at a semantic entity, ignore edges that link to former events[4] when looking for a data connection to the current event. In Figure 2, all edges to data from events other than the current *Event1* are therefore marked as *irrelevant* for the search of a data connection.

4.4 Location

As shown in Section 3, the location of an event can be an important factor in the search for relevant instances. In previous work [MRSB+14], the importance

[4]The name of this edge depends on the used ontology.

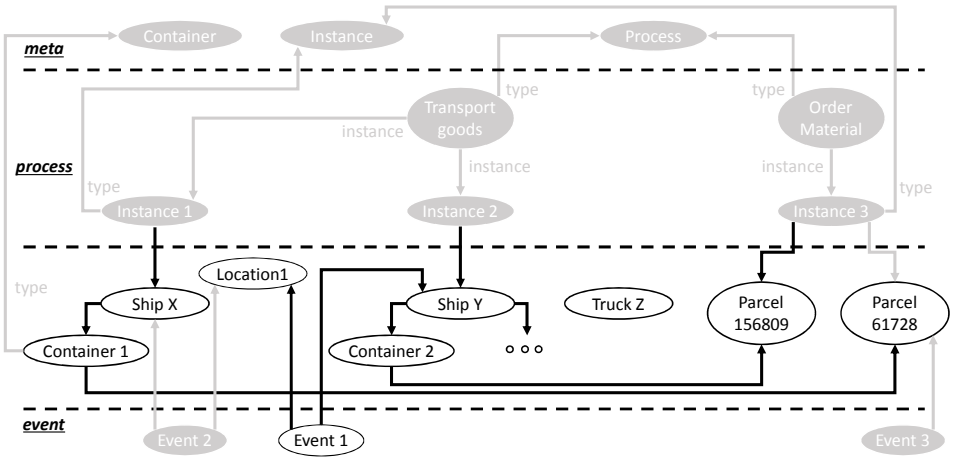

Figure 2: Data model used for the identification of interested process instances. It is restricted to public data only, excluding the meta, process and event level from the search in order to prevent the *over identifying* of the general data connectivity approach. Grey connections are not actively used for the search.

of locations for event processing in the domain of logistics has been shown. The technique detailed in that work will be reused in the automatic search for relevant instances. Thus, the approach is extended by a domain-specific filter mechanism that highlights the flexibility and extensibility of the overall approach.

In particular, process instances can be associated with transport plans as shown in use cases in the work done by Herzberg et. al [HMW13]. Among other things, these transport plans contain GPS coordinates that describe the route the associated vehicle takes. With the help of this data as well as public geographical data, a *nearby function* can determine, whether the location of an event is near to a transport plan.

If any instances are connected to that transport plan, they automatically become relevant for the occurring event and its data although they may not be connected to the event's data by a path through the knowledge graph.

5 Architecture/Implementation

The presented approach can be implemented in various ways. In a proof-of-concept prototype, it has been realized in a single SPARQL query that is executed on various datasets via the Apache JENA framework[5] in Java. The query language *SPARQL*[6] can be used for querying knowledge bases that are written in RDF. It is

[5]To be found at http://jena.apache.org/, last accessed at 01/13/2014.
[6]Recursive acronym for *SPARQL Protocol And RDF Query Language*

a graph-based query language that allows to retrieve data from and manipulate the data of an SKB.

The query in Listing 1 returns (line 1) all current process instances (line 2) that can be reached by at least one path starting from the event. The used method of *property paths*[7] (lines 3 to 10) translates the *graph cutting* into the SPARQL query. In the search for a path from event to instance, the defined edges are excluded by wrapping the disjunction of all irrelevant predicates in a negation.

The resulting set is merged (line 11) with those instances that have a location-based interest in the event due to their transport routes (lines 12 to 16). For the latter, it uses the *nearby function* defined in [MRSB+14].

Listing 1: The basic search algorithm for unlimited direction changes and the incoming event *Event1* in a SPARQL query. Prefix definitions are omitted for brevity.

```
1  SELECT DISTINCT ?instance WHERE {
2    ?instance rdf:type/rdfs:subClassOf* bp:Instance .
3    {
4      event:Event1 !(rdf:type|
5      bp:hasInstance|
6      bp:hasAttribute|
7      ^bp:hasInstance|
8      ^event:hasEventData|
9      ^event:hasEventInfo) ?instance .
10   }
11   UNION
12   {
13     ?instance bp:hasAttribute ?transportPlan .
14     ?transportPlan a dbo:transportation_route .
15     FILTER ( %nearby(event:Event1, ?transportPlan, 30) )
16   }
17 }
```

6 Related Work

Event correlation has been a prominent research topic for several years.
Lopez et al. [LdCCVG+10] examine a variety of methods and implementations for event correlation and compare them regarding their strengths, weaknesses, and possible fields of use. They also detail how semantic event correlation can overcome some of the limitations of basic approaches. Based on the principles explained in their work, the approach presented here uses semantic technologies to create links between events, external knowledge, and process data. However, the approach rather focuses on the correlation of events to specific process instances than to other events.

In [ZSP12], Zhou et al. examine the use of semantic technologies in complex event processing. They detail an architecture with a state of the art CEP engine extended

[7] To be found at http://www.w3.org/TR/sparql11-property-paths/, last accessed at 01/17/2014.

by semantic event queries that enable the querying of past, present and future event data. Teymourian et al. [TP10] outline an architecture that uses a rule-based engine to allow comparable queries to those of Zhou et al. The approach presented in this paper focuses on the use of already correlated and aggregated events and the detection of process instances that are related to the insights provided by such complex events. It uses similar technologies to those presented by Zhou et al. and Teymourian et al.

Rozsnyai et al. [RSL11] detail an algorithm that allows for the detection of correlation rules from an arbitrary list of data sources that provide information stored in inhomogeneous data structures. They process the incoming events, establish relations between them and are thus able to build aggregate groups of events that can be used in further analyses of the executed processes. The approach presented in this paper also strives to detect correlations in an automatic manner, but rather focusing on connecting events to specific and well-defined process instances and not to each other. Rozsnyai et al. try to (semi-)automatically build correlations from the events' data without pre-defined rules, an approach that aligns with the mechanism presented in this paper.

Herzberg et al. [HMW13] present, in a straight-forward approach, how data of manual process executions can be matched to relevant process instances using basic value-based methods. The presented approach in this paper goes beyond the value-based matching approach and operates on linked data that allows for a broader search for relevant instances that is not limited to the equality of defined variables but can take semantics of values into account.

7 Conclusion

The monitoring of business processes relies on sophisticated event handling mechanisms. The inclusion of external knowledge in the event correlation process can improve these capabilities. Adding new event sources dynamically in the process of correlation is a common task that needs to integrate seamlessly in the whole event handling procedure and should not require complex updates and reconfigurations of the event handling system. The approach introduced in this paper provides a semantic based solution that moves the correlation task away from the instances towards the event itself, thus enabling an always up-to-date correlation that is more flexible than current approaches. It can be used as a standalone method of identifying related instances to incoming event data or as a complement to an existing correlation infrastructure.

References

[BBR+11] Zohra Bellahsene, Angela Bonifati, Erhard Rahm, et al. *Schema matching and mapping*, volume 20. Springer, 2011.

[ES+07] Jérôme Euzenat, Pavel Shvaiko, et al. *Ontology matching*, volume 18. Springer, 2007.

[Fou92] Leslie R Foulds. *Graph theory applications*. Springer, 1992.

[Gar05] Stephen P Gardner. Ontologies and semantic data integration. *Drug discovery today*, 10(14):1001–1007, 2005.

[HMW13] Nico Herzberg, Andreas Meyer, and Mathias Weske. An Event Processing Platform for Business Process Management. In *Proceedings of the 17th IEEE International Enterprise Distributed Object Computing Conference*, pages 107–116, 2013.

[LdCCVG+10] Sergio Lopez, Maria del Carmen Calle Villanueva, Emitza Guzman, Tobias Röhm, Benoit Gaudin, and Newres Al Haider. State-of-the-art of event correlation and event processing, 2010.

[MRSB+14] Tobias Metzke, Andreas Rogge-Solti, Anne Baumgrass, Jan Mendling, and Mathias Weske. Enabling Semantic Complex Event Processing in the Domain of Logistics. In *ICSOC 2013 Workshops*, 2014.

[RSL11] Szabolcs Rozsnyai, Aleksander Slominski, and Geetika T Lakshmanan. Discovering event correlation rules for semi-structured business processes. In *Proceedings of the 5th ACM international conference on Distributed event-based system*, pages 75–86. ACM, 2011.

[SH11] A Shamsuzzoha and Petri T Helo. Real-time tracking and tracing system: Potentials for the logistics network. In *Proceedings of the 2011 international conference on industrial engineering and operations management*, pages 22–24, 2011.

[TP10] Kia Teymourian and Adrian Paschke. Enabling knowledge-based complex event processing. In *Proceedings of the 2010 EDBT/ICDT Workshops*, EDBT '10, pages 37:1–37:7. ACM, 2010.

[VD02] Kees-Jan Van Dorp. Tracking and tracing: a structure for development and contemporary practices. *Logistics Information Management*, 15(1):24–33, 2002.

[Wes12] Mathias Weske. *Business process management: concepts, languages, architectures*. Springer, 2012.

[ZSP12] Qunzhi Zhou, Yogesh Simmhan, and Viktor Prasanna. SCEPter: Semantic complex event processing over end-to-end data flows. Technical report, Technical Report 12-926, Computer Science Department, University of Southern California, 2012.