

# Case Construction for Mining Supply Chain Processes

Kerstin Gerke<sup>1</sup>, Jan Mendling<sup>2</sup>, and Konstantin Tarmyshov<sup>1</sup>

<sup>1</sup> SAP Research, CEC Dresden  
Chemnitzstr. 48, 01187 Dresden, Germany  
kerstin.gerke@sap.com, konstantin.tarmyshov@sap.com

<sup>2</sup> Humboldt-Universität zu Berlin  
Unter den Linden 6, 10099 Berlin, Germany  
jan.mendling@wiwi.hu-berlin.de

**Abstract.** Process mining algorithms aim at the automatic extraction of business process models from logs. Most of these algorithms perform well on single-system event traces that explicitly refer to a process instance or case. However, in many operational environments such case identifiers are not directly recorded for events. In supply chain processes there are even further challenges, since different identification numbers, vertical integration and numerous aggregation steps prevent individual work steps to become traceable as a case. As a result, there are little experiences with the use of process mining in supply chains. To address this problem, we consider Radio Frequency Identification (RFID) for identifying the movements of the business objects. Based on an example process from the Supply Chain Operations Reference Model (SCOR), we highlight the two key challenges of making RFID events available for process mining: case identification and focus shifts. We demonstrate how RFID events that conform to the EPCglobal standard can be used to construct cases such that process mining can be applied. A respective algorithm is sketched that we implemented in a tool which generates MXML process mining data from EPCglobal event traces. In this way, we provide a contribution towards applying process mining techniques for supply chain analysis.

**Keywords:** Process Mining, Supply Chain Management, RFID, EPCglobal, Business Process Modeling.

## 1 Introduction

Emerging technologies for interorganizational information systems provide new opportunities to analyze and optimize supply chain processes globally as in the past this was only possible in a local, intraorganizational setting. A prominent example that highlights the importance of global optimization is the so-called bullwhip effect that can be observed in supply chains [1]: due to delay of information processing between different parties and information aggregation, it can happen that small demand variation with retailers can build up to excessive demand variation for early suppliers at the beginning of the supply chain.

Information sharing is one of the strategies to circumvent such problems as the bullwhip effect, and Radio-Frequency Identification (RFID) is discussed as a technology to achieve this information sharing [2].

While there is consensus about the need for global supply chain analysis, there has been little work on suitable techniques that would provide the basis for analyzing business processes that span a whole supply chain. In particular, we observe a need to present transparent analysis results in a visual and intuitive manner. Process mining is a technique that meets these requirements by generating a graphical process model from a set of event logs of a process [3,4]. Process mining has been successfully applied to various intraorganizational problems [5], but the challenges of using event logs that span different companies is little understood so far.

At the current stage, most RFID scenarios have been implemented to support operational requirements of companies while strategic potentials are mainly neglected (see e.g. [6]). In this paper we contribute to the domain of RFID and process mining research by highlighting in how far the recent EPCglobal standard for exchanging event data can facilitate bridging the gap between data availability and process analysis capabilities. In particular, we focus on a supply chain scenario and illustrate how EPCglobal event types can be used to group event data into cases on a dynamic basis. To be more concise, we sketch an algorithm that assigns events to cases based on the record of EPC events following the EPCglobal standard.

Against this background, the paper is structured as follows. In Section 2 we use an example taken for the SCOR reference model for supply chain operations to illustrate the requirements of handling RFID-event data on an interorganizational level. In Section 3 we present the data available for events according to the EPCglobal standard. Furthermore, we describe an algorithm to assign events to cases based on the event data available in EPCglobal data. In Section 4 we discuss related work. Finally, in Section 5 we conclude the paper and give an outline of future research.

## 2 Challenges of Mining RFID-Based Supply Chain Processes

In this section we investigate challenges of mining supply chain processes. In Section 2.1 we give a brief overview of RFID capabilities before Section 2.2 highlights the mining challenges by using an example from the Supply Chain Operations Reference Model (SCOR) as an illustration.

### 2.1 RFID Technology in the Supply Chain

Radio Frequency Identification (RFID) is an automatic identification technology that is used to track location and movements of physical objects. A major advantage of RFID technology in comparison to traditional identification techniques like barcode is that data can be read contactless and as a non-line-of-sight

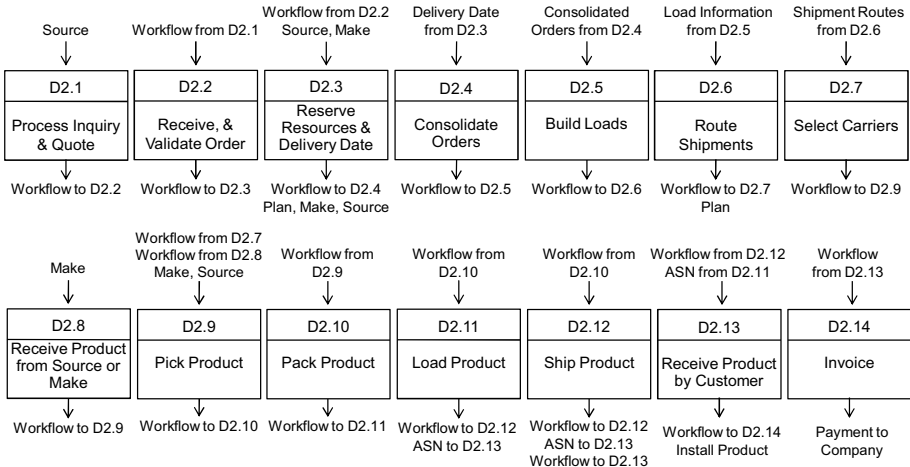
transfer via radio signals [7]. Beyond that, RFID tags enable bulk scanning and rewriting which is not possible with barcodes [8]. The usage of RFID technology requires different components such as tags, readers, middleware, and applications. *RFID tags* are attached to physical objects and store at least a unique identifier of the object that they are attached to. *RFID readers* are the hardware devices that directly interact with the RFID tags. An antenna on the tag emits radio waves generating voltage in the inductor of the passive transponder or triggering the active transponder to send data. The transponder chip starts working with this voltage, uses the inductor as antenna, and sends its ID to the reader antenna in bit-serial form. The major role of the *middleware* is to coordinate a number of RFID readers that are located, for example, at a production line within a single plant or distributed across a whole supply chain. The tasks of the middleware include, among others, buffering, aggregation, filtering of reader data and providing it in a suitable format to *RFID applications* such as include material tracking and tracing, and the identification of work in progress (WIP) materials [7]. Currently, international standards are emerging for the provisioning of RFID data across the internet. The most significant is the EPCglobal standard [9]. It builds on electronic product codes (EPCs) that are made available within an “Internet of Things”. This information network, also called the EPCglobal Network, allows producers, suppliers, and consumers to easily access and exchange information about product locations and movements [7]. We get back to details of EPCglobal in Section 3.

## 2.2 Challenges of Mining Supply Chain Processes

Process mining is a technique for analyzing how processes are executed in reality. The input for process mining is a set of event logs that relate to different instances of a process. These event logs typically stem from process-aware information systems such as workflow applications or ERP systems (see [10,11]). Based on process mining algorithms, it is possible to construct process models from these event logs automatically [3,4]. In contrast to many manually created process models, the models generated in process mining give an accurate account of what is really happening in the company [5]. This would be valuable information also in a supply chain setting. But there are different reasons why the events that are generated by RFID applications cannot directly be used for process mining.

We use the Supply Chain Operations Reference Model (SCOR) [12] to illustrate some of the issues. The SCOR model serves as a reference for designing operations in a supply chain. It is organized in three levels of decomposition. It starts with five primary management processes on the top level (Plan, Source, Make, Deliver, and Return), which are refined on level two according to three different strategies Stock, Order, and Engineer. Level three defines processes for each strategy of level two including inputs and outputs. In particular, we use the SCOR process Deliver-to-order, denoted as D2, as an example (see Figure 1). The inputs and outputs in the figure refer to other SCOR processes (e.g. Source-to-order S2 etc.) [12].

## D2 Deliver Make-to-Order Product



**Fig. 1.** SCOR Process Category D2

The processing in D2 starts after the receipt and response to general customer inquiries and requests for quotes (D2.1 Process Inquiry & Quote). After the receipt of a customer order the order is entered into an order processing system for validation (D2.2 Receive, and Validate Order) which triggers the reservation of required inventory and planned capacity for specific orders. Furthermore, a delivery date is scheduled (D2.3 Reserve Resources & Determine Delivery Date). If procurable, orders are grouped to ensure least cost or best service fulfillment (D2.4 Consolidate Orders). Afterwards, transportation modes are selected and efficient loads are built (D2.5 Build Loads). In the following step (D2.6 Route Shipments) the loads are consolidated and routed by mode, lane, and location. Afterwards a suitable carrier is selected (D2.7 Select Carriers). After receiving the products, they are verified and put away (D2.8 Receive Product from Source or Make). The pick process contains a series of activities including retrieving orders to pick, verifying inventory availability, building the pick wave, picking the product, recording the pick, and delivering product to packing area in response to an order that belongs to the process (D2.9 Pick Product). Then, after sorting or packing, pasting tags etc., the picked products are delivered to the shipping area for loading (D2.10 Pack Product), where they are placed onto modes of transportation (D2.11 Load Product). The loading process includes the generation of the documentation necessary to meet internal, customer, carrier and government needs (e.g. the advanced shipping notification (ASN)) and triggers the process of shipping the product to the customer’s site (D2.12 Ship Product). Finally, the customer receives the shipment and verifies it for completeness and delivery terms. (D2.13 Receive & Verify Product by Customer). The process ends with a signal which is sent to the financial organization that the order has been shipped and that the billing process should begin (D2.14 Invoice).

If we now assume that this whole scenario would be supported with RFID, there are essentially two major problems that hinder a direct evaluation of the process with process mining techniques: case identification and focus shifts.

- The problem with *case identification* stems from the fact that EPC IDs are processed. This means that every event that can be logged only relates to an EPC - there is no explicit notion of a case identifier that groups events belonging to the same process instance. In order to make process mining applicable, we first have to relate every event to a process instance. Using an EPC as a process ID does not work in the general case because of the second problem.
- The problem of *focus shifts* is illustrated in Figure 2. Due to various packing and unpacking, assembly and disassembly operations, it is difficult to follow the object flow of a single process instance. In the example of Figure 2, there is first a packaging step at the shipper's site that includes packing cartons on a pallet. In this process step an aggregation takes place by which the Serialized Global Trade Item Number (SGTIN) is associated with the Serial Shipping Container Code (SSCC) of the pallet on which they are packed. SGTIN serves as an EPC associated with items. The SSCC code is characteristic to containers. Aggregation allows to track which exact products are packed into a container or pallet. Still, the focus on the business object shifts from the single product to the pallet. In the second packing step the pallets are stuffed into a container identified by the Global Returnable Asset Identifier (GRAI). The GRAI contains all SSCC providing the capability to identify and verify individual pallets. The monitoring of each stage of containerization of the product provides the supply chain partners between the manufacturer and the point of final distribution with the ability to understand precisely which products, at a unique pack level, are shipped. This allows e.g. for accurate recall management. Now, the focus has shifted from the single pallet to the container.

There are different ways of dealing with these problems. In the following section, we investigate how cases can be constructed from event logs that follow the EPCglobal standard.

### 3 Case Construction Based on EPCglobal

In this section, we analyze how the EPCglobal standard can help to make RFID event data available for process mining. In Section 3.1 we introduce the data that is attached to EPCglobal events. Section 3.2 then presents our approach to construct case information from these events.

#### 3.1 Logistic Event Types of EPCglobal

EPCglobal is a nonprofit organization founded in 2003 by the global standardization consortium GS1. Its goal is to define global standards for RFID technology.

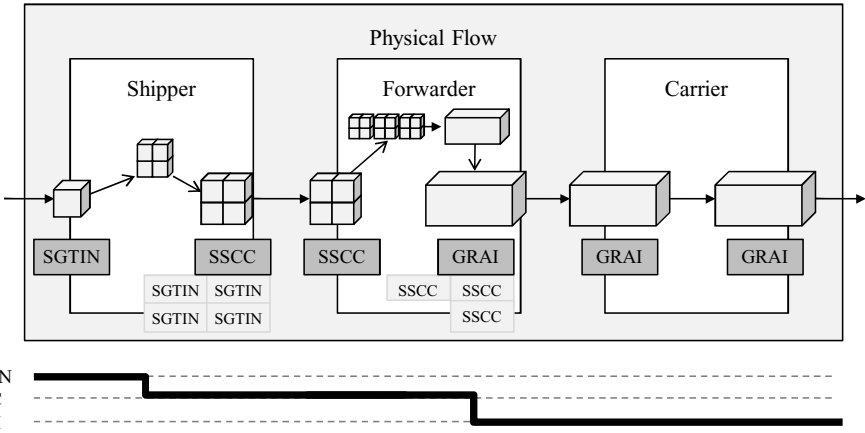


Fig. 2. Packaging Process

The EPCglobal Architecture Framework describes how the different components and standards suggested by EPCglobal will fit together to form the EPCglobal Network. The EPCglobal Network is an information network that will allow producers, suppliers, shipper, and consumers to easily access and exchange information about products. The information access is based on Electronic Product Codes (EPC), these are globally unique identifiers that unambiguously identify each product and can be used as a reference to information stored in the EPCglobal Network. In contrast to barcodes, the EPC identifies each product instance rather than only product categories. One key component is the EPC Information Service (EPCIS). The EPCIS stores the information that can be referenced through the EPCs and is accessible via the Internet [7]. Moreover, the EPCglobal Architecture Framework covers all aspects of reading EPC tag data and participating in the EPCglobal Network including the Object Name Service and the Product Markup Language. The Object Name Service (ONS) is a global registry of EPC information and the Product Markup Language (PML) is a language specific to EPC data storage and retrieval [8]. For detailed information we relate to [9].

In the context of the EPCs' reading processes the term event is commonly used. EPCglobal defines four *event types* including object, aggregation, quantity, and transaction events. ObjectEvent encompasses information that relates to one or more EPCs. For example, a reader at a warehouse entry door could always record an event when products are moved from the goods entry area to the warehouse. An AggregationEvent is used when objects have been physically aggregated. In this case, numerous objects (e.g. boxes) are put together in a higher-level unit (e.g. pallet). AggregationEvents are created that assign the child EPCs to the parent EPC of the container. AggregationEvents are created recursively if containers contain other containers. For entering inventories a QuantityEvent is used. In this case, the amount is recorded and not the

serialized number section. A *TransactionEvent* describes the assignment of physical objects with one or more transactions. For example, in such an event, the reference to a certain delivery notification could be linked to the objects declared to be inside it [9].

The event types contain *attributes*. These attributes can be divided into four dimensions of information about the object (what = EPC, *epcList*, *transactionList*, EPC class, and quantity), the time (when = *eventTime*), location (where = read point, business location), and the business context (why = process, disposition). The *eventTime* stores the date when the event is captured. The *epcList* is a list of EPCs naming the physical objects to which the event pertained. In the case of a transaction event, these objects are linked to the transaction. The *parentID* represents an identifier of the parent of the EPCs given in the EPC list. The attribute *action* relates the event to the lifecycle of the related EPCs. The attribute can accept the values ADD, OBSERVE, or DELETE. They capture the semantics of the event. Consider the case of a pallet. When it is created by bundling different items, the respective event is an *AggregationEvent* with action attribute ADD. Once it is disassembled again, e.g. having reached its shipment destination, an *AggregationEvent* with action attribute DELETE must be reported. The attribute *bizStep* reflects the business step of which the event is part of. The *disposition* shows the business condition of the objects associated with the EPCs. The read point at which the event took place is stored in the attribute *readPoint*. The attribute *bizLocation* references the business location where the objects associated with the containing and contained EPCs may be found. The attribute *type* is an identifier that indicates what kind of business transactions this business transaction denotes. The identifier *bizTransaction* denotes a specific business transaction. The D2 process works as follows if it is supported by RFID technology (see Figure 3). All captured tag data is categorized in terms of the event types *ObjectEvent*, *TransactionEvent*, *QuantityEvent* or *AggregationEvent*. After the validation of the order the process begins with the production of a product, which is immediately tagged with an EPC tag. The related object event contains information about the EPC, location and time. The subsequent storage of goods is monitored by means of quantity events. The receipt of the single product and the pallet is documented by a RFID reader of the packing site door by means of object events. The following packaging step is linked to an aggregation event and a transaction event, which in turn link the pallet, the cartons on the pallet and the related order. At the packaging site doors and at the shipping site entry door the pallets' move will be captured by means of object events. Prior to passing through the shipping dock door, the pallets are linked to a purchase order and a transport order via a transaction event. Finally, the shipping is monitored by recording an object event. Prior to passing through the shipping dock door, the cartons' movement will be monitored and recorded by means of object events at significant places, such as doorways or shelves. At the shipping dock doors, the EPCs of cartons and pallets are linked to the corresponding advanced shipping notification (ASN) via a transaction event.

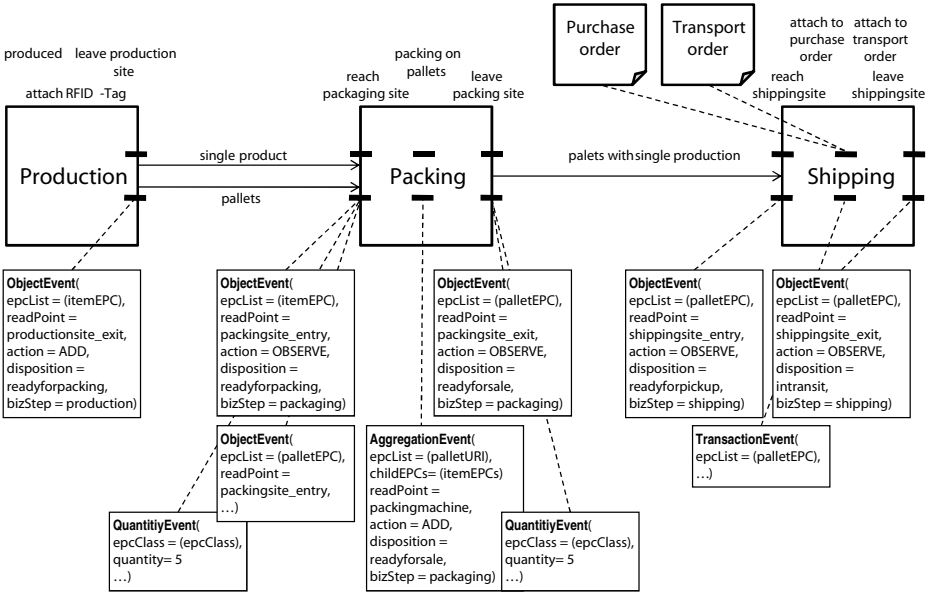


Fig. 3. Internal Process of a Supplier and corresponding Events [13]

### 3.2 Case Construction Based on EPCglobal Event Types

In order to make RFID event logs accessible to process mining, we have to assign each event at least to a process activity, a time stamp, and a process instance [14]. There are different pieces of information in an EPCglobal event that can be directly mapped to these fields: the *bizStep* attribute translates to the activity that is executed in the process and the time stamp is stored in the *eventTime* attribute. The main challenge is assigning different events and the related EPCs to a process instance in such a way that the shifting focus between different assembled and disassembled business objects of varying granularity is handled appropriately.

Below we sketch an algorithm to construct cases based on assembly (Aggregation add) and disassembly events (Aggregation delete). The key idea of this algorithm is that there are two distinct types of aggregation: production aggregation and shipment aggregation. *Production aggregation* is of a persistent nature: once different pieces are assembled they become a new object of its own that is meant to be persistent. That is different to *shipment aggregation*: for this case the aggregation is of a temporary nature. In a way, it is transparent to the physical production process. Both types of aggregation can be distinguished based on EPCglobal event logs: shipment aggregations occur with both an assembly and a disassembly event while production aggregations only cover an assembly. We use this distinction for our algorithm. Its focus is to construct cases based on physical movement and physical assembly. Therefore, all events that belong to the same production aggregation line are grouped into the same



process instance. As a consequence, shipment aggregations are treated to become transparent.

For our algorithm, we can make certain assumptions about the EPCglobal event log. In particular, we assume that the events are in chronological order. This implies that EPCs are in hierarchical order at any point in time due to the fact that the containment relationship (childEPCs) is none overlapping. The goal of the algorithm is to construct a tree of EPCs that belong to the same physical production process. The structure of this tree is built up by links between EPCs. As leaves, there can be different events assigned to an EPC node. We assume objects of type process, of type EPC and of type event to be available. The process object contains the different trees of EPCs which group different events. The objects provide the following methods for constructing trees:

- event.getEPC() to get the EPC of an event,
- event.getChildEPCs() to get the child EPCs,
- EPC.add(event) to add an event,
- process.getEPC(EPC) to retrieve an EPC,
- process.mergeCases(NewRootEPC,List of OldRootEPCs) to group different cases under a new root event,
- process.splitCases(rootEPC) to copy the root EPC to all its child EPCs and delete it.

As input we assume a list of EPCglobal events which we can process in chronological order using an iterator EventChronology. Based on these notations, we can define the algorithm:

1. Set currentEvent to EventChronology.next(),
2. If process.getEPC(currentEvent).exists() then
  - (a) If currentEvent.isNotAggregation() then  
process.getEPC(currentEvent).add(currentEvent),
  - (b) If currentEvent.isObserveAggregation() then  
process.getEPC(currentEvent).add(currentEvent),
  - (c) If currentEvent.isAddAggregation() then  
Set currentEPC to currentEvent.getEPC(),  
process.mergeCases(currentEPC,currentEPC.getChildEPCs())  
currentEPC.add(currentEvent)
  - (d) If currentEvent.isDeleteAggregation() then  
Set currentEPC to currentEvent.getEPC(),  
currentEPC.add(currentEvent),  
process.splitCases(currentEPC)
3. Repeat if EventChronology.hasNext(),

For evaluation purposes we have prototypically implemented the algorithm in the context of the ProM process mining framework [15] using the MXML interchange format for log files [14]. This implementation takes EPCglobal event logs as input to generate an MXML file according to the presented algorithm. The challenge for a thorough real-life evaluation is that companies that use the EPCglobal in

their logistic processes are still rather scarce. Therefore, we simulated supply chain processes to generate log files using a simulator that builds on top of the EPCIS implementation Fosstrak.<sup>1</sup> By feeding the generated MXML files into ProM and applying a process mining plug-in, we could verify that the approach yields the expected model, also when aggregation and disaggregation operations are included.

## 4 Related Work

Our work can be related to different streams of research in the business process management domain. Complex event processing [16] is recently receiving increased attention to support process execution in an open and distributed web environment. The challenge there is to assign incoming messages to the correct process instance at runtime. Several executable process modeling languages provide support for this problem by offering so-called correlation sets. A correlation set is essentially a query that retrieves identifiers from messages that are unique for a particular process instance. The correlation set concept is included in BPEL [17] and BPMN [18]. The correlation problem has been discussed theoretically in [19,20]. The idea of a correlation set is somewhat related to our approach. Our algorithm also builds upon the observation that certain identifiers, namely EPCs, belong to a unique process instance. There are some works that relate to mining correlation information including [21] where the authors use an algorithm to identify correlation relevant fields that can be used to construct case identifiers. In [22] the authors define an algorithm to construct a finite state machine of business protocols from logs without case identifiers. In our approach, we reconstruct the correlation relationships that are implicitly given in the EPCglobal event log data.

Our contribution can also be related to practical challenges of process mining. There is a growing body of knowledge that reports on case studies in different application domains. While there are considerable achievements [5], several problems still need to be solved [23]. A particular problem relates to the reformatting and enrichment of log data that stems from information systems that are not directly process-aware. The complexity of this problem becomes apparent in the work on mapping SAP log data to MXML [11]. Our work is unique in this context, as it identifies how RFID event data (available in EPCglobal format) can be used to construct cases such that process mining can be applied.

## 5 Conclusion

In this paper, we have addressed the current research challenge of making RFID event logs accessible for process mining analysis. We have used an example from the SCOR Supply Chain Operations Reference Model for identifying case identification and focus shifts as major obstacles. Furthermore, we have demonstrated

---

<sup>1</sup> [www.fosstrak.org](http://www.fosstrak.org)

how information that is available in EPCglobal compliant RFID event data can be used to construct cases. To be more concise, we have sketched an algorithm to group events to cases. This algorithm handles different types of events including assembly and disassembly events.

We have implemented the algorithm as part of a tool that converts EPCglobal event logs to MXML files. These can be processed with the process mining workbench ProM. From both an industry and a research perspective there are open questions on how the shifting focus can be appropriately supported. Currently, we construct a tree of EPCs belonging to the same process instance in such a way that the most coarse-grained physical object is on top of the tree and all child EPCs are included. In a supply chain, different parties might be interested in following the event stream on different levels of granularity. This challenge requires future research both in terms of mining algorithms but also in terms of interactive tool support. Furthermore, we aim to apply our approach to a real-world scenario. Currently, we are searching for potential partners who already use the EPCglobal standard.

## References

1. Lee, H., Padmanabhan, V., Whang, S.: The Bullwhip Effect in Supply Chains. *Sloan Management Review* 38, 93–102 (1997)
2. Sharma, A., Citurs, A., Konsynski, B.: Strategic and Institutional Perspectives in the Adoption and Early Integration of Radio Frequency Identification (RFID). In: *Hawaii International Conference on Systems Sciences*, vol. 40, pp. 37–55. IEEE, Los Alamitos (2007)
3. van der Aalst, W.M.P., Weijters, T., Maruster, L.: Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering* 16(9), 1128–1142 (2004)
4. Greco, G., Guzzo, A., Manco, G., Saccà, D.: Mining and reasoning on workflows. *IEEE Transactions on Knowledge and Data Engineering* 17(4), 519–534 (2005)
5. van der Aalst, W., Reijers, H., Weijters, A., van Dongen, B., Alves de Medeiros, A., Song, M., Verbeek, H.: Business process mining: An industrial application. *Information Systems* 32(5), 713–732 (2007)
6. Ivantysynova, L.: RFID in Manufacturing: Mapping the Shop Floor to IT-Enabled Business Processes. PhD thesis, *Wirtschaftswissenschaftliche Fakultät, Humboldt-Universität zu Berlin* (2008)
7. Günther, O., Kletti, W., Kubach, U.: RFID in Manufacturing. Springer, Heidelberg (2008)
8. Niederman, F., Mathieu, R.G., Morley, R., Kwon, I.W.: Examining RFID Applications in Supply Chain Management. *ACM Transactions on Software Engineering and Methodology* 50(7), 92–101 (2007)
9. EPCglobal Inc., EPC Information Services (EPCIS) Version 1.0.1 Specification (2007), [http://www.epcglobalinc.org/standards/epcis/epcis\\_1\\_0\\_1-standard-20070921.pdf](http://www.epcglobalinc.org/standards/epcis/epcis_1_0_1-standard-20070921.pdf)
10. Dumas, M., ter Hofstede, A., van der Aalst, W. (eds.): *Process Aware Information Systems: Bridging People and Software Through Process Technology*. Wiley Publishing, Chichester (2005)

11. Ingvaldsen, J., Gulla, J.: Preprocessing support for large scale process mining of sap transactions. In: ter Hofstede, A.H.M., Benatallah, B., Paik, H.-Y. (eds.) *BPM Workshops 2007*. LNCS, vol. 4928, pp. 30–41. Springer, Heidelberg (2008)
12. Supply-Chain Council: *Supply Chain Operations Reference Model, SCOR, Version 8.0* (2006)
13. Fleisch, E.: *Das Internet der Dinge: Ubiquitous Computing und RFID in der Praxis. Visionen, Technologien, Anwendungen, Handlungsanleitungen*. Springer, Heidelberg (2005)
14. van Dongen, B.F., van der Aalst, W.M.P.: A Meta Model for Process Mining Data. In: Pastor, Ó., Falcão e Cunha, J. (eds.) *CAiSE 2005*. LNCS, vol. 3520, pp. 309–320. Springer, Heidelberg (2005)
15. van Dongen, B.F., de Medeiros, A.K.A., Verbeek, H., Weijters, A., van der Aalst, W.M.P.: The ProM Framework: A New Era in Process Mining Tool Support. In: Ciardo, G., Darondeau, P. (eds.) *ICATPN 2005*. LNCS, vol. 3536, pp. 444–454. Springer, Heidelberg (2005)
16. Luckham, D.: *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley, Reading (2001)
17. Alves, A., Arkin, A., Askary, S., Barreto, C., Bloch, B., Curbera, F., Ford, M., Golland, Y., Guizar, A., Kartha, N., Liu, C., Khalaf, R., Koenig, D., Marin, M., Mehta, V., Thatte, S., van der Rijn, D., Yendluri, P., Yiu, A.: *Web services business process execution language version 2.0. Committee specification 31 January 2007*, OASIS (2007)
18. Object Management Group: *Business Process Modeling Notation (BPMN) Specification. Final Adopted Specification, dtc/06-02-01*, Object Management Group (February 2006)
19. Barros, A., Decker, G., Dumas, M., Weber, F.: Correlation patterns in service-oriented architectures, pp. 245–259 (2007)
20. Decker, G., Mendling, J.: Instantiation semantics for process models. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) *BPM 2008*. LNCS, vol. 5240. Springer, Heidelberg (2008)
21. Pauw, W.D., Hoch, R., Huang, Y.: Discovering conversations in web services using semantic correlation analysis. In: *2007 IEEE International Conference on Web Services (ICWS 2007)*, Salt Lake City, Utah, USA, pp. 639–646. IEEE Computer Society, Los Alamitos (2007)
22. Serrou, B., Gasparotto, D.P., Kheddouci, H., Benatallah, B.: Message correlation and business protocol discovery in service interaction logs. In: Bellahsene, Z., Léonard, M. (eds.) *CAiSE 2008*. LNCS, vol. 5074, pp. 405–419. Springer, Heidelberg (2008)
23. Genrich, M., Kokkonen, A., Moormann, J., zur Muehlen, M., Tregear, R., Mendling, J., Weber, B.: Challenges for business process intelligence: Discussions at the BPI workshop 2007. In: ter Hofstede, A.H.M., Benatallah, B., Paik, H.-Y. (eds.) *BPM Workshops 2007*. LNCS, vol. 4928, pp. 5–10. Springer, Heidelberg (2008)