Discovering Interacting Artifacts from ERP Systems (Extended Abstract)[3]

D. Fahland[1], X. Lu[1], Marijn Nagelkerke[2], Dennis van de Wiel[2]

Abstract: *Enterprise Resource Planning* (ERP) systems are widely used to manage business documents along a business processes and allow very detailed recording of event data of past process executions and involved documents. This recorded event data is the basis for auditing and detecting *unusual flows*. *Process mining* techniques can analyze event data of processes stored in linear event logs to discover a process model that reveals unusual executions. Existing techniques assume a linear event log that use a single case identifier to which all behavior can be related. However, in ERP systems processes such as *Order to Cash* operate on multiple interrelated business objects, each having their own case identifier, their own behavior, and interact with each other. Forcing these into a single case creates ambiguous dependencies caused by *data convergence* and *divergence* which obscures unusual flows in the resulting process model. We present a new semi-automatic, end-to-end approach for analyzing event data in a plain database of an ERP system for unusual executions. We identify an *artifact-centric process model* describing the business objects, their life-cycles, and how the various objects *interact* along their life-cycles. The technique was validated in two case studies and reliably revealed unusual flows later confirmed by domain experts. The work summarized in this extended abstract has been published in [Lu15].

Keywords: Process Mining, ERP-System, Artifact-Centric Model, Object Life-Cycle, Interaction Discovery

1 Introduction and Problem Description

Information systems (IS) not only store and process data in an organization but also record *event data* about how and when information changed. This "historical event data" can be used to analyze, for instance, whether information processing in the past conformed to the prescribed processes or to compliance requirements. *Process mining* [Aa11] offers automated techniques for this task. In particular exploring visual models *discovered from event data* allows to identify unusual flows and their circumstances; based on which concrete measures for process improvement can be devised [Ec15]. Prerequisite to this analysis is a *process event log* that holds events about information changes with the assumption that each event belong to one specific execution of a specific process.

In general, information access is not tied to a particular process execution; rather the same information can be accessed and changed from various processes and applications. For

[1] d.fahland@tue.nl, Eindhoven University of Technology, The Netherlands
[2] KPMG IT Advisory N.V., Eindhoven, The Netherlands
[3] This article summarizes problem, approach, and selected findings of a study published as *Xixi Lu, Marijn Nagelkerke, Dennis van de Wiel, and Dirk Fahland. Discovering Interacting Artifacts from ERP Systems. Services Computing, IEEE Transactions on, 8(6), 2015* doi:10.1109/TSC.2015.2474358 [Lu15].

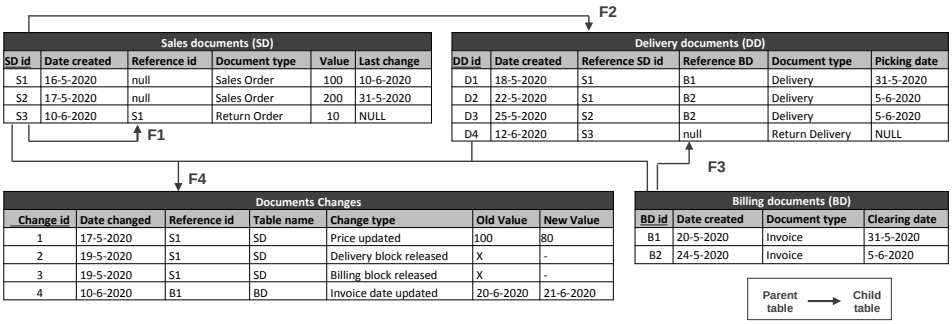

Fig. 1: The tables of the simplified OTC example

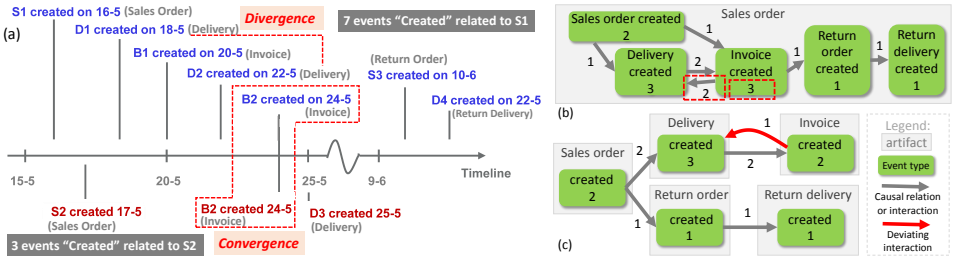

Fig. 2: Creation of documents of Fig. 1 along time (a), a classic process model based on a single case identifier (b), and an artifact-centric model based on 5 case identifiers (c).

instance, in *Enterprise Resource Planning* (ERP) systems (such as SAP and Oracle Enterprise), information is stored in *business objects* (or *documents*) which are linked via one-to-many and many-to-many relations, typically in the relational database. The objects themselves are encapsulated in services [AMZ00] which are invoked by high-level end-to-end business processes; each invocation is called a *transactions* which is logged in the data object itself.

Fig. 1 shows a simplified example of the transactional data of an *Order to Cash* (OTC) process supported by SAP systems; Fig. 2(a) visualizes the events of Fig. 1 that are related to document creation. There are two sales orders S1 and S2; creation of S1 is followed by creation of a delivery document D1, an invoice B1, another delivery document D2, and another invoice B2 which also contains billing information about S2. Creation of S2 is also followed by creation of another delivery document D3. Further, there is a return order S3 related to S1 with its own return delivery document D4. The many-to-many relations between documents surface in the transactional data of Fig. 1: a sales document can be related to multiple billing documents (S1 is related to B1 and B2) and a billing document can be related to multiple sales document (B2 is related to S1 and S2). This behavior already contains an *unusual flow*: delivery documents were created twice *before* the billing document (main flow), but once the order was reversed (B2 before D3).

When applying classical process mining techniques, one first has to extract an event log based on a *single case identifier* to which all event data can be related. Choosing *SD id* in Fig. 1 leads to the two sequences of events shown in Fig. 2(a). Process discovery on

this log yields the model of Fig. 2(b) which is wrong: two invoices are created before their deliveries instead of one, and three invoices are created instead of two (known as *divergence* and *convergence*, respectively) [Pi11].

2 Approach: Discovering Artifact-Centric Models

We propose to approach the problem under the "conceptual lens" of *artifact-centric models* [CH09]. An *artifact* is a data object over an information model; each artifact instance exposes *services* that allow changing its informational contents; a *life-cycle model* governs when which service of the artifact can be invoked; the invocation of a service in one artifact may trigger the invocation of another service in another artifact. Information models of different artifacts can be in one-to-many and many-to-many relations allowing to describe behavior over complex data in terms of multiple objects interacting via service invocations. Under this lens, each document of an ERP system can be seen as an artifact; a transaction on a document is a service call on the artifact; behavioral dependencies between transactions of documents can be seen as life-cycle behavior and dependencies of service calls. Describing the transactional data of Fig. 1 with artifact-centric concepts yields the model of Fig. 2(c); it visualizes the order in which objects are created and also highlights the unusual flow of invoice B2 being created before delivery D2.

The problem of discovering an artifact-centric process model from relational ERP data decomposes into two sub-problems. (1) Given a relational data source, identify a set of artifacts, extract for each artifact an event log, and discover a model of its life-cycle. (2) Given a set of artifacts and their data source, identify interactions between the artifacts, between their instances, between their event types and between their events. Figure 3 shows the overview of our approach. (1.1) We use the data schema of the data source to discover *artifact types* which detail all timestamped columns related to a particular business object. (1.2) For each artifact we then extract a classical *event log* [Aa11], each *case* describes all events related to one *instance* of the artifact. (1.3) Existing process discovery algorithms allow discovering a life-cycle model of the artifact. In parallel, (2.1) we discover interactions between artifacts from foreign key relations in the data source; (2.2) during log extraction, each case of an artifact is annotated with references to cases of other artifacts this case interacts with. (2.3) The case references are refined into interactions between activities of different artifact life-cycles.

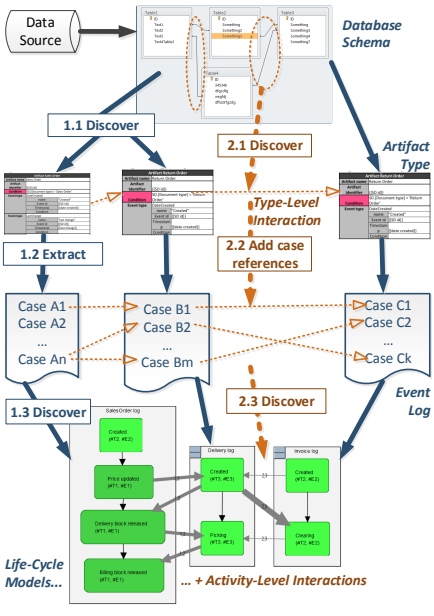

Fig. 3: An overview on our approach.

3 Results

We implemented our approach based on [NvDF12] and conducted two case studies. By separating data into artifacts along one-to-many relations, we eliminated divergence and convergence, the *interaction flows* discovered from one-to-many relations were meaningful to business users, and unusual flows were detected.

Fig. 4 shows models obtained from 2 months data of an SAP Order-to-Cash process (11 document header tables, 134,826 records of 5-49 attributes); the model at the top was obtained with a classical approach (only 29 of 77 edges are correct); the model at the bottom was obtained using our approach. In both case studies the discovered process models were assessed as accurate graphical representations of the source data by domain experts; all edges including outlier edges were assessed as correct and traced back to the source data together with domain experts. These insights could be obtained exploratively and much faster than with existing best practices.

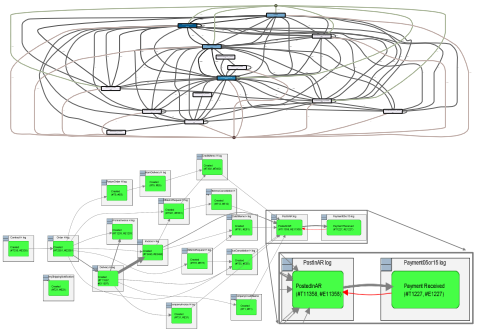

Fig. 4: SAP OTC process, classical process model obtained from single event log (top) and artifact-centric model highlighting outliers (bottom)

References

[Aa11] Aalst, W.M.P. van der: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, 2011.

[AMZ00] Al-Mashari, Majed; Zairi, Mohamed: Supply-chain re-engineering using enterprise resource planning (ERP) systems: an analysis of a SAP R/3 implementation case. IJPDLM, 30(3/4):296–313, 2000.

[CH09] Cohn, D.; Hull, R.: Business artifacts: A data-centric approach to modeling business operations and processes. Bulletin of the IEEE Computer Society TCDE, 32(3):3–9, 2009.

[Ec15] van Eck, Maikel L.; Lu, Xixi; Leemans, Sander J. J.; van der Aalst, Wil M. P.: PM ^2 : A Process Mining Project Methodology. In: CAiSE 2015. volume 9097 of LNCS. Springer, pp. 297–313, 2015.

[Lu15] Lu, Xixi; Nagelkerke, Marijn; van de Wiel, Dennis; Fahland, Dirk: Discovering Interacting Artifacts from ERP Systems. IEEE Trans. Services Computing, 8(6):861–873, 2015.

[NvDF12] Nooijen, Erik H. J.; van Dongen, Boudewijn F.; Fahland, Dirk: Automatic Discovery of Data-Centric and Artifact-Centric Processes. In: DAB'12. volume 132 of LNBIP. Springer, pp. 316–327, 2012.

[Pi11] Piessens, D.A.M.: Event Log Extraction from SAP ECC 6.0. Master's thesis, Eindhoven University of Technology, 2011.