Dealing with Artifact-Centric Systems: a Process Mining Approach

Guangming Li[1] and Renata Medeiros de Carvalho[2]

Abstract: Process mining provides a series of techniques to analyze business processes based on execution data in enterprises. It has been successfully applied to classical processes on WFM/BPM systems, in which one process execution consists of events attached with the same case id. However, existing process mining techniques suffer from problems when dealing with artifact-centric systems, such as ERP and CRM, in which a business process involves a set of interacting artifacts and a case notion for the whole process is missing. Some typical problems are convergence and divergence in XES logs, and lost interactions between multiple instances in process models. Existing artifact-centric approaches try to address these problems, but have not yet solved them satisfactorily. For instance, one has to pick an instance notion in each artifact, the description of the end-to-end behavior is distributed over multiple diagrams, and the interactions between the data perspective and the behavioral perspective are not explicitly presented. This paper proposes a set of new techniques, such as a novel log format and a novel modeling language, to enable process mining for artifact-centric systems.

Keywords: Artifact-Centric Systems, Process Mining, Object-Centric, Event Logs, Process Models

1 Introduction

Nowadays, information systems are widely used in enterprises to support their daily business process executions. Such an information system is called *Process-Aware Information System (PAIS)*, since it needs to be aware of business processes [Du05]. A typical class of PAISs is formed by generic systems that are process-centric and driven by explicit process models, i.e., one process execution on these systems is constituted by a single case with a unique case identifier. Examples are Workflow Management (WFM) systems [vdAvH04] and Business Process Management (BPM) systems [We07]. Another class of PAISs consists of artifact-centric systems that do not have a unique case notion, which could be used to trace and isolate its executions. The entire process on these systems is seen as a set of interacting business entities called artifacts. Examples are Enterprise Resource Planning (ERP) systems (SAP, Oracle, etc.) and Customer Relationship Management (CRM) systems [O'00].

Process executions on PAISs generate various data, e.g., relational database tables and event logs, which can be analyzed to discover insights to reflect the "health"condition of

[1] Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands g.li.3@tue.nl
[2] Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands r.carvalho@tue.nl

enterprises. Process mining provides a set of techniques to analyze business processes from different angles. For instance, process discovery is a technique to automatically discover a process model from recorded process executions. Conformance checking detects the disagreement between the actual process executions and a reference process model. Performance analysis identifies the bottlenecks in business processes.

Process mining has been successfully applied to classical WFM/BPM systems, which assume a case notion for the whole process. However, one can easily see that this assumption is violated by real-life processes supported by ERP/CRM systems. These systems are artifact-centric and often have one-to-many and many-to-many relationships between data objects (e.g., customers, orders, invoices and payments). Existing process mining techniques need a case notion to correlate events both in event logs (MXML and XES) and process models (Petri nets, EPCs and BPMN). Therefore, they are doomed to fail on artifact-centric systems. In this paper, we propose new process mining techniques, which do not depend on case notions and are suited for artifact-centric systems.

2 Challenges

As mentioned above, artifact-centric systems do not assume case notions in their business processes. Therefore, existing process mining techniques suffer from the following problems when they are applied to these systems.

The XES log format harms the quality of original data. There often exist one-to-many and many-to-many relationships in the data generated by artifact-centric systems. Therefore, a case notion for the whole process is difficult to be identified. If we straightjacket such data into XES logs, it flattens multi-dimension data as separate traces, which leads to convergence and divergence problems. Besides, the XES format focuses more on the behavioral perspective (i.e., only considering events and information related to events), which may not present useful information on the data perspective.

Existing modeling languages are difficult to model interactions (i) between process instances and (ii) between the data perspective and the control-flow perspective. Existing process modeling languages (e.g., Petri nets, EPCs and BPMN) consider process instances in isolation. The interactions between instances cannot be described properly. Besides, they mainly focus on the control-flow perspective. Powerful constructs present in ER models [Ch88] and UML class models, which can easily deal with one-to-many and many-to-many relationships are not employed at all. Moreover, constraints on the data perspective must influence behavior, but this interaction is not described by existing languages .

Deviations are not totally detected. Some deviations on the behavioral perspective can only be detected by considering multiple instances and constraints in the class model. In this situation, the weak data perspective in existing models makes such deviations undetectable.

Performance analysis may not be reliable. Due to convergence and divergence problems, the performance analysis result may be imprecise (e.g., inaccurate frequencies). Besides, because of missing information on data perspective, useful insights for users on this perspective may not be provided by performance analysis.

3 Approaches

The problems discussed in Section 2 prevent the employment of "classic" process mining techniques on artifact-centric systems. In this section, we propose new process mining techniques to solve these problems, as shown in Figure 1. In general, the spectrum of our approaches are consistent with the lifecycle of "classic" process mining research, i.e., our new process mining techniques try to reach the same goals on artifact-centric systems, as the "classic" process mining approaches reach on WFM/BPM systems. More precisely, based on a novel log format and a new type of models, we propose new process mining approaches covering log extraction, model discovery, conformance checking and performance analysis, to enable process mining on artifact-centric systems.

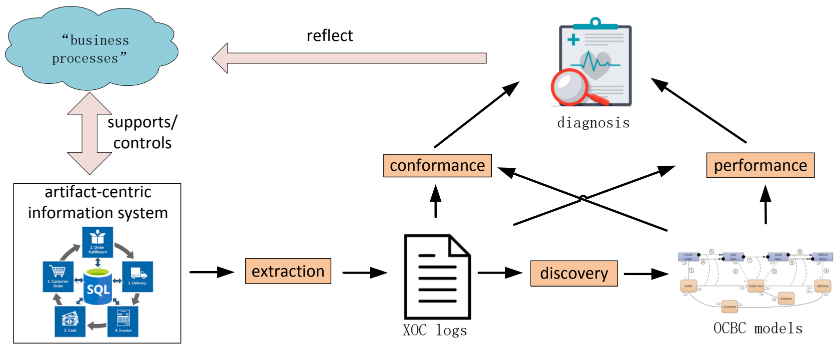

Fig. 1: The framework of our approaches.

eXtensible Object-Centric event logs. We propose a novel log format named eXtensible Object-Centric (XOC) to organize the data generated by artifact-centric systems [vdALM17]. Artifact-centric systems do not have a clear case notion for the whole process, but they follow an intuitive principle that each occurred event on the system changes the state of the system (i.e., adding, updating or deleting records in the underlying database). Triggered by this idea, a XOC log consists of a set of ordered events and each event corresponds to an object model representing the database, which provides an evolutionary view of the system. Note that an object model may represent only the tables involved in the target process when the database covers multiple processes.

Object-Centric Behavioral Constraint models. We propose a novel modeling language [vdALM17], that combines data modeling languages (ER, UML, or ORM) and declarative languages (Declare, CMMN, or GSM), resulting in Object-Centric Behavioral Constraint (OCBC) models. More precisely, as shown in Figure 2, an OCBC model consists of a class model (presenting cardinality constraints between classes on the data perspective), a behavioral model (presenting declarative constraints between activities on the control-flow perspective), and relationships (① ∼ ④) which connect these two models by relating activities to classes. Unlike existing declarative languages, the scope of each behavioral constraint (e.g., ⑦) is identified by classes (e.g., “order line") rather than case notions.

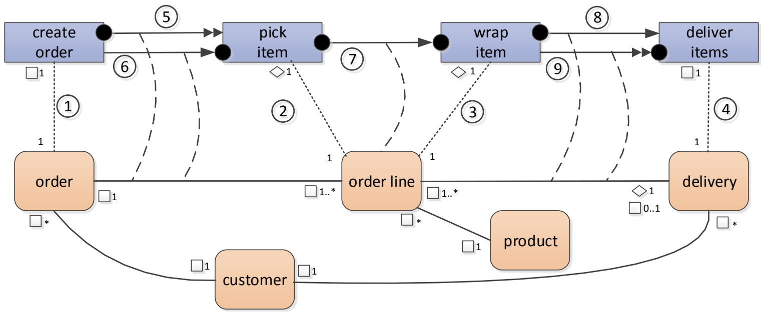

Fig. 2: A small Object-Centric Behavioral Constraint (OCBC) model.

Besides, we propose approaches to automatically extract XOC logs from relational databases of artifact-centric systems and discover OCBC models from XOC logs [LdCvdA]. Based on a XOC log and a reference OCBC model, a set of rules are defined to check the conformance between them [vdALM17]. In future, we also plan to analyze the performance based on the statistics of frequencies and time, which can be obtained by replaying a XOC log on an OCBC model. More precisely, various metrics can be defined to analyze the performance of business processes from different angles.

4 Related Work

The artifact-centric approaches [CH09] (including the earlier work on proclets [Aa01]) consider the entire process as a set of interacting artifacts. Each of these artifacts can be described by an information schema (called an artifact schema) and a non-trivial lifecycle (indicating how the artifact evolves through a process execution). However, these approaches suffer from the following problems: (i) within an artifact (proclet, or subprocess), one is forced to pick a single instance notion (although a case notion for the whole process is not required); (ii) the description of the end-to-end behavior needs to be distributed over multiple diagrams (e.g., one process model per artifact); (iii) the control-flow cannot be related to an overall data model (i.e., there is no explicit data model or it is separated from the control-flow); (iv) interactions between different entities are not visible or separated (because artifacts are distributed over multiple diagrams); and (v) cardinality constraints in the data model cannot be exploited while specifying the intended dynamic behavior.

Besides, colored (data-aware) Petri nets [Je96] add "color" on tokens to attach a data perspective on the behavioral perspective. BPMN [Gr10], Data flow chart and UML activity diagram [EP00] can describe behavioral perspective and its communication with data perspective by data objects and data stores. Concepts like lanes, pools, and message flows in conventional languages like BPMN can model interactions between process instances. In summary, these models mentioned above can describe the data perspective and interactions to some extent, but more powerful constructs present in ER models and UML class models are not employed at all in these models.

References

[Aa01] van der Aalst, W.M.P. et al.: Proclets: A framework for lightweight interacting workflow processes. International Journal of Cooperative Information Systems, 10(04):443–481, 2001.

[Ch88] Chen, P.P.S.: The Entity-Relationship Model-Toward a Unified View of Data. In: Readings in artificial intelligence and databases, S. 98–111. Elsevier, 1988.

[CH09] Cohn, D.; Hull, R.: Business Artifacts: A Data-Centric Approach to Modeling Business Operations and Processes. IEEE Data Engineering Bulletin, 32(3):3–9, 2009.

[Du05] Dumas, M. et al.: Process-aware information systems: bridging people and software through process technology. John Wiley & Sons, 2005.

[EP00] Eriksson, H.E.; Penker, M.: Business Modeling with UML. New York, S. 1–12, 2000.

[Gr10] Group, Object Management: Business Process Model and Notation. OMG, 2010.

[Je96] Jensen, K.: Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Springer-Verlag, Berlin, 1996.

[LdCvdA] Li, G.; de Carvalho, R.M.; van der Aalst, W.M.P.: Automatic Discovery of Object-Centric Behavioral Constraint Models. In: Proceedings of BIS 2017. S. 43–58.

[O'00] O'Leary, Daniel E: Enterprise resource planning systems: systems, life cycle, electronic commerce, and risk. Cambridge university press, 2000.

[vdALM17] van der Aalst, W.M.P.; Li, G.; Marco, M.: Object-Centric Behavioral Constraints. Corr technical report, arXiv.org e-Print archive, 2017. Available at https://arxiv.org/abs/1703.05740.

[vdAvH04] van der Aalst, W.M.P.; van Hee, K.M.: Workflow management: models, methods, and systems. MIT press, 2004.

[We07] Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.