Artifact Lifecycle Extension

B.F.A. Hompes

[1] Department of Mathematics and Computer Science,
Eindhoven University of Technology, Eindhoven, The Netherlands
[2] Philips Research, Eindhoven, The Netherlands
b.f.a.hompes@tue.nl

August 20th, 2018 Version: 1

Abstract. This document introduces the XES artifact lifecycle extension. It allows for events to relate to transitions in the lifecycles of multiple business artifacts, which are key conceptual entities that behave according to state-based transactional lifecycle models. We present a definition of the concepts of the extension, discuss how it relates to the standard lifecycle extension, and provide an example event log.

1 Introduction

An emerging family of approaches for event analysis uses (business) artifacts that combine both data aspects and process aspects into a holistic unit. Artifacts have been defined as key conceptual entities that are central to the operation of part of a business and who's lifecycles define the overall business process [2, 5]. A typical example artifact-centric process is a build-to-order process where several customer orders are collected, and based on the order, materials need to be ordered from suppliers. Customer orders can relate to one or more supplier orders which contain goods related to multiple customer orders [6]. Other domains where business artifacts play an important role are healthcare [7], order-to-cash [4], and logistics and transportation. Furthermore, business artifacts may relate to multiple lifecycle models, to enable analysis on multiple levels of abstraction.

Using the artifact lifecycle extension presented here, events may signal transitions in the lifecycles of multiple artifact instances, and multiple lifecycle models can be identified for the different artifact types. As such, this extension allows for the simultaneous analysis of multiple artifacts on different levels. For example, given process execution data in terms of an event log or stream, the extension enables artifact-centric performance analysis [3]. Several different artifact lifecycle models have been used in literature, tools, and *Process-Aware Information Systems (PAIS)* such as ERP, CRM, and BPM systems. The IEEE standard for eXtensible Event Streams (XES) [1] already contains two distinct lifecycle models for activities in the standard lifecycle extension. However, only activity lifecycle models are considered and the lifecycles of cases are expressed using process models, as will be discussed in Section 4.

2 Definition

In the artifact-centric setting, multiple artifact types can be defined. Typical examples are documents, orders, deliveries, etc. Traditional concepts such as activities, cases, and resources can be considered artifacts as well, each following their own lifecycle models. For every given artifact type, multiple instances usually exist. For example, 'Bob', 'John', and 'Mary' are instances of the resource artifact type, and 'document 11', 'order 19', and 'delivery 31' are instances of the artifact types document, order, and delivery respectively. Typically, some form of hierarchy can be identified as well. For example, activity instances are generally grouped by their activity name, i.e. activity A, B, C, or 'prepare order', 'send invoice', 'notify customer', etc. Each artifact's behavior is described in one or more lifecycle models. Note that the modeling language does not matter here. The principle is that events may record multiple lifecycle transitions.

The artifact lifecycle extension provides one top-level event attribute: moves. The moves attribute contains zero or more lifecycle move entries. A lifecycle move is represented by an attribute: model. Each model entry should contain two meta-attributes, namely the lifecycle transition and the artifact instance, i.e. the instance which the transition in the lifecycle model was recorded. The artifact lifecycle extension is defined as shown in Table 1, and the attributes it defines are listed in Table 2.

Table 1. Definition of the artifact lifecycle extension.

Extension definition	
Name	ArtifactLifecycle
Prefix	artifactlifecycle
Extension URI	`http://xes-standard.org/artifactlifecycle.xesext`

XML representation	`<extension name="ArtifactLifecycle"`
	`prefix="artifactlifecycle"`
	`uri="http://xes-standard.org/artifactlifecycle.xesext"/>`

Table 2. Attributes of the artifact lifecycle extension.

Level	**Key**	**Type**	**Occurrences**	**Remarks**
Event	moves	list	0-1	One per event
Meta	model	string	$0\text{-}\infty$	Entries of *moves* list
Meta	instance	string	$0\text{-}\infty$	One per *model* list entry
Meta	transition	string	$0\text{-}\infty$	One per *model* list entry

3 XES Extension

Listing 1.1 shows the Artifact Lifecycle extension in XML notation, following the XML Schema Definition (XSD) in [1]. Note that in this definition, the meta attributes model, instance, and transition are defined generically, i.e. it is not explicitly specified that the instance and transition attributes are meta-attributes of the model attribute.

Listing 1.1. The artifact lifecycle extension in XML format.

```
<xesextension
  name="ArtifactLifecycle"
  prefix="artifactlifecycle"
  uri="http://xes-standard.org/artifactlifecycle.xesext">
  <event>
    <list key="moves">
      <alias mapping="EN" name="Lifecycle Moves"/>
    </list>
  </event>
  <meta>
    <string key="model">
      <alias mapping="EN" name="Lifecycle Model"/>
    </string>
    <string key="instance">
      <alias mapping="EN" name="Artifact Instance"/>
    </string>
    <string key="transition">
      <alias mapping="EN" name="Lifecycle Transition"/>
    </string>
  </meta>
</xesextension>
```

4 Relation to the Standard Lifecycle Extension

XES has been structured to record activity flows for cases (process instances). The standard lifecycle extension[3] is limited in several ways and therefor not suitable for the recording and subsequent analysis of lifecycle transitions in artifact-centric processes. First of all, using the standard lifecycle extension, only activity lifecycles are considered. Other artifacts are not taken into account. Secondly, each event can only record a single lifecycle transition. Lastly, only one activity lifecycle model can be considered for the entire event log. The artifact lifecycle extension solves this problem by allowing events to record multiple lifecycle transitions for different lifecycle models and artifact instances. Additionally, the artifact lifecycle extension removes the need for an explicit case notion, thereby simplifying event stream analysis.

[3] See `http://xes-standard.org/lifecycle.xesext`

5 Example

To illustrate the use of the artifact lifecycle extension, we provide an example event log in Listing 1.2. Here, an excerpt of a synthetic event log is shown where the events relate to the lifecycle transitions of two artifact types: *order* and *delivery*. Orders are created and items are added until the order is finalized and sent, after which the order needs to be paid. A delivery is created as soon as an order is sent, after which it is put in a warehouse and picked up for delivery.

Listing 1.2. Example event log. The second event signals a transition in the lifecycle of both ‘order 142’ and ‘delivery 381’.

```
<log>
  ...
  <extension
    name="ArtifactLifecycle"
    prefix="artifactlifecycle"
    uri="http://xes-standard.org/artifactlifecycle.xesext"/>
  <trace>
    ...
    <event>
      <date key="time:timestamp"
            value="2018-01-01T15:04:12.000+02:00"/>
      <list key="artifactlifecycle:moves">
        <values>
          <string key="artifactlifecycle:model"
                  value="Order">
            <string key="artifactlifecycle:instance"
                    value="order 142"/>
            <string key="artifactlifecycle:transition"
                    value="finalize order"/>
          </string>
        </values>
      </list>
      ...
    </event>
    <event>
      <date key="time:timestamp"
            value="2018-01-01T15:04:58.000+02:00"/>
      <list key="artifactlifecycle:moves">
        <values>
          <string key="artifactlifecycle:model"
                  value="Order">
            <string key="artifactlifecycle:instance"
                    value="order 142"/>
            <string key="artifactlifecycle:transition"
                    value="send order"/>
          </string>
          <string key="artifactlifecycle:model"
                  value="Delivery">
      <string key="artifactlifecycle:instance"
          value="delivery 381"/>
      <string key="artifactlifecycle:transition"
          value="create delivery"/>
     </string>
    </values>
   </list>
   ...
  </event>
  <event>
   <date key="time:timestamp"
      value="2018-01-01T16:33:06.000+02:00"/>
   <list key="artifactlifecycle:moves">
    <values>
     <string key="artifactlifecycle:model"
         value="Delivery">
      <string key="artifactlifecycle:instance"
          value="delivery 381"/>
      <string key="artifactlifecycle:transition"
          value="put in warehouse"/>
     </string>
    </values>
   </list>
   ...
  </event>
  ...
 </trace>
 ...
</log>
```

Bibliography

[1] IEEE Standard for extensible event stream (XES) for achieving interoperability in event logs and event streams. *IEEE Std 1849-2016*, pages 1–50, Nov 2016.

[2] D. Cohn and R. Hull. Business artifacts: A data-centric approach to modeling business operations and processes. *IEEE Data Eng. Bull.*, 32(3):3–9, 2009.

[3] B.F.A. Hompes and W.M.P. van der Aalst. Lifecycle-based process performance analysis. Submitted to CoopIS 2018.

[4] X. Lu, M. Nagelkerke, D. van de Wiel, and D. Fahland. Discovering interacting artifacts from ERP systems. *IEEE Transactions on Services Computing*, 8(6):861–873, 2015.

[5] A. Nigam and N.S. Caswell. Business artifacts: an approach to operational specification. *IBM Systems Journal*, 42(3):428–445, 2003.

[6] V. Popova, D. Fahland, and M. Dumas. Artifact lifecycle discovery. *International Journal of Cooperative Information Systems*, 24(1), 2015.

[7] M.L. van Eck, N. Sidorova, and W.M.P. van der Aalst. Discovering and exploring state-based models for multi-perspective processes. In *Business Process Management - 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, September 18-22, 2016. Proceedings*, pages 142–157, 2016.