Process Mining on FHIR - An Open Standards-Based Process Analytics Approach for Healthcare

Emmanuel Helm[1,3][0000−0002−1323−3511], Oliver Krauss[1,3][0000−0002−8136−2606], Anna Lin[1][0000−0003−4840−8788], Andreas Pointner[1][0000−0001−8642−1161], Andreas Schuler[2,3][0000−0003−1074−3222], and Josef Küng[3]

[1] Research Department Advanced Information Systems and Technology, University of Applied Sciences Upper Austria, 4232 Hagenberg, Austria
[2] Department of Medical and Bioinformatics, University of Applied Sciences Upper Austria, 4232 Hagenberg, Austria
firstname.lastname@fh-hagenberg.at
https://aist.fh-hagenberg.at
[3] Johannes Kepler University, 4040 Linz, Austria
josef.kueng@jku.at

Abstract. Process mining has become its own research discipline over the last years, providing ways to analyze business processes based on event logs. In healthcare, the characteristics of organizational and treatment processes, especially regarding heterogeneous data sources, make it hard to apply process mining techniques. This work presents an approach to utilize established standards for accessing the audit trails of healthcare information systems and provides automated mapping to an event log format suitable for process mining. It also presents a way to simulate healthcare processes and uses it to validate the approach.

Keywords: Process Mining · Healthcare · HL7 FHIR.

1 Introduction

We provide a process analytics approach to enable the mining of standardized audit trails of healthcare information systems by transforming them into eXtensible Event Stream (XES) logs via an automated mapping approach. We tested it by simulating a radiology practice workflow, and analyzed the results with a process mining tool.

With diverse use cases and different approaches, techniques, and algorithms, process mining became its own scientific discipline over the last 20 years [1]. With the goal of understanding and improving the real-world processes, process mining provides an evidence-based (i.e., data-driven) view on the processes recorded by information systems. An increasing number of case studies also show the applicability of process mining in the healthcare domain (cf. the reviews in [4, 18]). Most of those case studies focus their analysis on single hospitals or even departments due to problems of data integration or data availability [4].

1.1 Problem Statement

Rebuge and Ferreira [17] conclude in their work that healthcare processes, both organizational and medical treatment, are *highly dynamic*, *highly complex*, *increasingly multi-disciplinary* and *generally ad-hoc*. All four characteristics make it hard to apply process mining techniques. In this work we focus on the aspect of high complexity, partly caused by the high number of participants, heterogeneous information systems, and the resulting lack of interoperability [4, 17].

Rojas et al. [18] found in their review that three implementation strategies for process mining projects in healthcare exist: (1) The majority of case studies work with *direct implementations*, where data is gathered directly from hospital information systems (HIS) for building an event log. Data extraction and building the correct event log poses major challenges here. (2) The second, *semi-automated*, strategy involves the integration and extraction of data from different sources via custom-made developments. The disadvantage here is the ad-hoc, proprietary nature of these developments, as they only work for specific data sources and environments. Both strategies, direct implementation and semi-automated, share the need to understand process mining tools and algorithms for conducting process analytics. (3) The third strategy is the implementation of an *integrated suite*. Specific data sources are connected and integrated, and specific process mining algorithms are executed in order to perform defined analytics tasks. Once implemented, these solutions are easily applicable, but like the semi-automated strategy, fail to integrate other data sources and environments.

We conclude, that a major problem with starting a process mining project in healthcare is that one has to choose between either complex manual data extraction and integration, or locking oneself in on specific data sources and environments (i.e., vendor lock-in).

1.2 Related Work

To overcome the problems of process mining on heterogenous data sources in healthcare, some studies tried to analyze standardized audit trails [3, 7, 16]. We will build on this work, using their concepts of audit events, mapping strategies, and multi-perspective process mining.

Cruz-Correia et al. [3] were the first to explicitly make the connection between standardized auditing in healthcare and process mining. They specifically looked at the Integrating the Healthcare Enterprise (IHE) integration profile Audit Trail and Node Authentication (ATNA). Being one of the core profiles dealing with IT infrastructure in healthcare, ATNA defines how to build up a secure domain that provides patient information confidentiality, data integrity, and user accountability. They analyzed ATNA audit trails from four different hospitals in Portugal and identified several data quality issues.

Later, Helm and Paster [7] investigated the suitability of event logs recorded by the means of IHE ATNA for process mining. They adopted a direct mapping approach, transforming IHE audit messages into XES event logs. They encountered issues regarding the determination of trace identifiers and semantics preserving mapping.

De Murillas et al. [16] took on the previous approach [7] and presented a method to overcome the problems of trace identification and incorrect mappings. By integrating the audit trail data into a generic meta model (OpenSLEX), they provided the means to query and analyze the data from different perspectives.

While these approaches try to solve the issue of heterogeneous data sources, they either lock the user in on a predefined mapping [7] or provide a non-standardized interface to the process data [16] – two shortcomings that can be avoided with our approach.

1.3 Proposed Solution

Supporting definition, instantiation, and execution of workflows is still a topic of vivid discussions in the respective standards development working groups. For the analysis part, first steps have been taken. Standardized Operational Log of Events (SOLE) is a recently developed IHE integration profile. It is a supplement for the radiology technical framework and currently in revision 1.2, published for trial implementation in mid 2018 [13]. SOLE describes the capture and retrieval of operational events in the radiology domain and utilizes transactions from the ATNA profile, including the new RESTful ATNA [12], based on the Health Level Seven (HL7) standard Fast Healthcare Interoperability Resources (FHIR). The profile authors' incentive for writing the SOLE integration profile was the strong desire of healthcare providers "to increase throughput and efficiency, both to improve the quality and timeliness of care and to control costs" [13]. They conclude, that workflow events must be captured in order to be able to apply *business intelligence tools* [13].

We propose an open standards-based process analytics approach for healthcare information systems to overcome the problems mentioned above. It enables the development of tools that combine the easy applicability of an integrated suite with the ability to integrate different data sources. This will make existing process mining tools the *business intelligence tools* the community wants.

To this end, this paper aims to show how existing concepts can be utilized and what changes in the standard are necessary to enable process mining based on HL7 FHIR. This paper also contributes to the field by presenting a novel approach to utilize a process simulation tool in a healthcare environment.

2 Background

This section provides a brief overview on the two major standards involved in building the open process analytics approach, HL7 FHIR and XES.

2.1 HL7 FHIR

FHIR[4] is the latest addition to the family of healthcare interoperability standards maintained and published by HL7 International [8]. FHIR provides a

[4] HL7, FHIR and the FHIR logo are the registered trademarks of Health Level Seven International and their use does not constitute endorsement by HL7.

comprehensive information model which is geared towards supporting semantic interoperability of clinical data. The fundamental building blocks for this information model are *resources*. A resource as described by Mandel et al. [15] is a coherent expression of clinical data and is based on a set of well-defined fields and data types. Every resource comprises the standard defined data content, a human-readable representation of respective content and has an identity. The FHIR specification defines resources for common clinical concepts, e.g., Patient, Medication, Observation, Condition. Besides that, FHIR leverages modern web technologies together with a strong foundation of web standards and offers support for RESTful architectures. Following the RESTful paradigm, FHIR allows to alter the state of a particular resource using a set of predefined actions for Create/Read/Update/Delete (CRUD). If required by a given use-case, it is also possible to apply a more Remote Procedure Call (RPC)-like interaction paradigm. This is achieved by defining operations that work on input and produce an output [9]. The operations can be executed on the server level, on the resource type level, or on the instance level of a specific resource and are typically invoked by a HTTP POST or can alternatively be invoked by a HTTP GET if no changes are caused on the server.

According to HL7 International [8], a central challenge for the FHIR specification is handling the wide variety and variability in diverse healthcare processes. This challenge is solved by offering a simple framework for extending the existing resources and describe use cases based on profiles. Profiling a resource allows to constrain and extend a resource specification for a given context [15]. By providing reference implementations for the specification, HL7 intends to reduce the entry barrier for developing FHIR conformant solutions. The development of the specification and the standard follows a developer first approach, which is reflected by the specification as a mixed standard comprising normative portions and parts still undergoing trial use [8].

2.2 XES

Log data is created from a variety of different systems with their own proprietary data models, formats, and semantics. Process mining techniques require their input data in a specific format. Some tools directly integrate data from (1) Enterprise Resource Planning (ERP) systems, (2) databases, or (3) Comma Separated Value (CSV) files, all three in a proprietary way. However, developed in 2010, XES became the IEEE standard for "achieving interoperability in event logs and event streams" [11]. Today, XES is supported by the majority of process mining tool vendors.

XES defines three basic objects: log, trace and event. Log (the process) contains a collection of traces (execution instances) and a trace contains a collection of events [20]. Each object can contain an arbitrary set of strongly typed attributes in the form of key-value pairs. Every attribute value has a data type, like string, boolean, or date. To add semantics to these data types, XES defines the concept of extensions. An extension defines a set of attributes, their types, and keys with a specific semantic meaning.

3 Materials and Methods

This section describes which standards and tools were used in building the analytics suite and how we utilized and extended them to enable process mining based on HL7 FHIR.

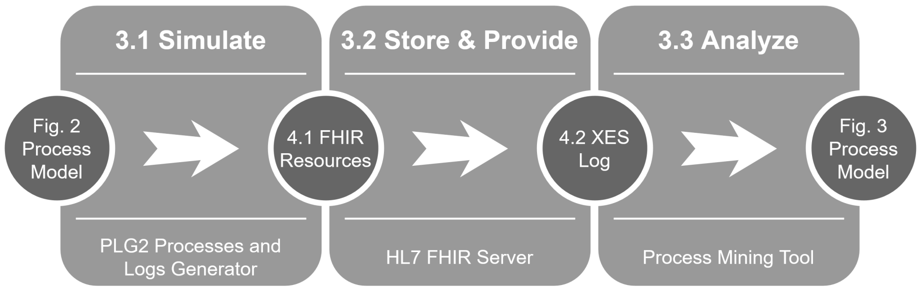

Fig. 1. The three steps of the interface test setting including the respective consumed and produced data. The numbers correspond to sections or figures in this paper.

Figure 1 depicts the three steps (1) simulate, (2) store&provide, and (3) analyze, that aim to show how the open standardized process analytics approach works. The circles represent data consumed and produced in those three steps.

To test the approach, a simple process was used. Figure 2 shows a simplified process model for an examination in a radiology practice using Business Process Model and Notation (BPMN). It shows the main steps from the appointment scheduling to the distribution of the diagnostic report. It is based on the work of Erickson et al. on business analytics in radiology [5] and on the process model used for evaluation in [7]. This is of course just an example and the approach is applicable to other healthcare domains as well.

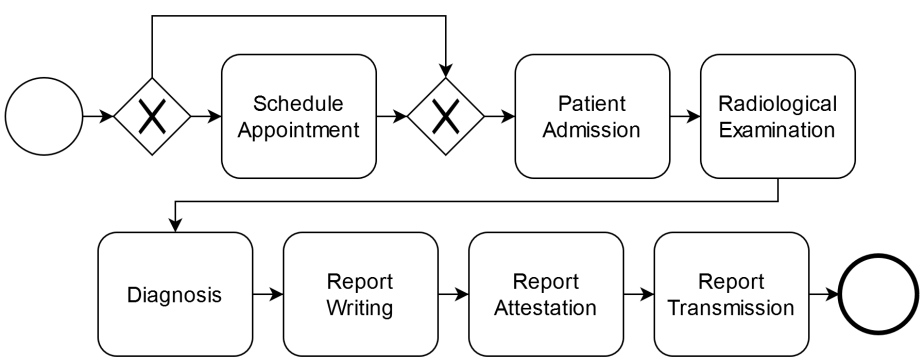

Fig. 2. BPMN process model of the radiology practice workflow based on [5, 7].

In the first step, a patient that, e.g., received a referral for a radiological examination, calls the practice to *schedule an appointment*. In some cases of our simulation, this step can be skipped and the patient arrives without a scheduled appointment. On the day of the examination, the patient arrives at the reception and is placed on the waiting list (*patient admission*). When called, the patient enters the procedure room and the *radiological examination* takes place. Afterwards, the radiologist makes a *diagnosis* and dictates the report. The *report writing* is done by trained specialists. The resulting report is *attested* by the radiologist. Finally, the report is sent to a requesting physician or handed out directly to the patient (*report transmission*).

3.1 Simulate

In order to be able to automatically generate process data, some sort of process engine or simulator is required. Burattin [2] developed a tool specifically designed to simulate processes and generate event logs for process mining, the Processes and Logs Generator (PLG2). The tool allows to generate and simulate random BPMN models, and to add randomized noise (e.g., double activity execution, skipping activities, etc.). The tool also allows to load an existing model, in our case the model from figure 2, and simulate it.

To use PLG2 for the simulation, we needed to make REST calls to our HL7 FHIR server. PLG2 allows to specify the execution time of different activities using Python scripts [2]. We adapted those scripts to execute REST calls using Client for URLs (cURL). By default PLG2 provides a single parameter, that is, the case identifier (caseId), to these python functions. We used this parameter to make the process instances distinguishable by deriving resource identifiers from it (i.e., patientId and encounterId).

Each activity in the process from figure 2 was extended with REST calls, creating, reading, or updating resources and executing operations on the FHIR server (according to the mapping described in the next section). The process was then simulated 10 times without randomized noise, each run resulting in one process instance recorded on the server.

3.2 Store & Provide

We set up a FHIR server including the required extensions and operations to automatically record audit trails, and to transform and provide this information in the XES format for process mining.

FHIR Server. We implemented our FHIR Server based on the open-source project "HAPI-FHIR Starter"[5]. This project provides a fully working FHIR server, including a database connection, based on the HAPI FHIR JPA project. Adjustable configuration files and the interceptor framework [19] create high

[5] https://github.com/hapifhir/hapi-fhir-jpaserver-starter

flexibility for custom changes and for adding extensions to the existing server implementation.

We utilized the Consent Interceptor, which amongst other functionalities has the ability to hook into the point of the server code, where a CRUD operation (e.g., creating an appointment or reading a patient record) has been finished. One of the Consent Interceptor's roles is to write audit trail records, creating an AuditEvent resource every time an operation has been finished successfully or with a failure.

In addition to the interceptor implementation, we provided the FHIR operation *$fhirToCDA* as part of our custom extensions to the server implementation. The operation can be executed on a specific instance of the DiagnosticReport resource and it returns an empty document to the client. An AuditEvent recording the execution of this operation in the context of a radiology workflow encounter will, for mapping purposes, be interpreted as a report transmission activity.

To query for an event log in the XES format, we extended our FHIR server by the *$xes* operation, which is defined to work on the AuditEvent resource type and is there to identify and transform all AuditEvents of the radiological workflow "rad-wf" into the XES format:

```
GET [fhirserver]/AuditEvent/$xes?plandefinition=PlanDefinition/rad-wf
```

Extending AuditEvent. We filled the AuditEvent resource with request details that are automatically provided for any standard CRUD operation. In order to be able to query for relevant AuditEvent resources, we needed to identify grouping elements. We decided to extend the AuditEvent resource by references to the Encounter and PlanDefinition resources (cf. section 5.1). Geared to the other resources containing the Encounter resource reference as part of their standard FHIR resource definition, we named the extended AuditEvent element "encounter". An additional extension "basedon" is used to reference the PlanDefinition resource "rad-wf", that defines the radiological workflow. This element can later be used to filter AuditEvent resources related to the executions of the radiological workflow process, while Encounter references are used to distinguish the single process instances (i.e., the traces).

Mapping FHIR AuditEvent to XES. For the test setting, we base our mapping on the assumption that Encounter identifier can be utilized as trace identifiers and that recorded events refer to a common process description, i.e., a medical guideline or pathway defined as a PlanDefinition. Of course, this is just one perspective, and different perspectives can be taken on the data (cf. section 5.3).

Let R be the set of all resources on the FHIR server. Let $A \subseteq R$ be the set of all AuditEvent resources, and $E \subseteq R$ be the set of all Encounter resources, and $P \subseteq R$ be the set of all PlanDefinition resources. All three subsets are disjoint, i.e., $A \cap P = \emptyset$, $A \cap E = \emptyset$, and $E \cap P = \emptyset$. Resources can refer to other resources

Table 1. Mapping table of operations on specific FHIR resources to activities of the radiology practice workflow, ordered by occurence in the simulated model in figure 2.

Operation	FHIR Resource	$\mapsto$	Activity
create	Appointment		Schedule Appointment
update	Appointment		Patient Admission
create	Procedure		Radiological Examination
create	Media		Diagnosis
create	DiagnosticReport		Report Writing
update	DiagnosticReport		Report Attestation
execute	*$fhirToCDA		Report Transmission

via the predicate refersTo$(r, r') :\Leftrightarrow (r, r') \in R$, where r' is referenced by r, i.e., r contains the identifier of r'.

Let $p_w \in P$ be the PlanDefinition resource "rad-wf" defining the radiology workflow. Then, $A_w = \{a \in R | \forall_{a \in A} \text{ refersTo}(a, p_w)\}$ is the set of all AuditEvent resources recorded during the execution of radiology workflows.

For our mapping, let A_w be a set of disjoint sets A_{wi}, where every A_{wi} represents a set of AuditEvents recorded during a specific radiology workflow encounter $\exists e \in E$ of one patient. Then, every A_{wi} will be mapped to a trace σ in an XES event log L.

For testing the approach, we only map to mandatory fields in L, e.g., concept:name of the event (providing the activity name) and time:timestamp of the event (for ordering). Table 1 describes which recorded combination of operation and resource is mapped to which activity name. The timestamp is mapped directly from the recording time AuditEvent.recorded.

3.3 Analyze

Querying the FHIR server for AuditEvent resources using the $xes operation returns an XES event log. Since the operation already utilizes XES standard extensions (i.e., Concept and Time), the semantics of the fields are clear for process mining tools. The next step is to analyze if the simulated process matches the one stored and provided by the HL7 FHIR server. Thus, we want to compare the input model with a model generated based on the retrieved XES event log. We use the process mining tool ProM 6.9 [20] with the Visual Inductive Miner plugin [14] to generate a model.

4 Results

This section shows three exemplary results of the implementation: (1) a FHIR resource generated by the simulator, (2) the corresponding event in the XES event log, and (3) the process model created based on the event log. All results and examples can also be found in our GitHub open-source project[6].

[6] https://github.com/fhooeaist/ProcessMiningOnFHIR/

4.1 FHIR Resources

As described in the mapping in table 1, the Report Writing activity is associated with creating a DiagnosticReport resource. The simulator thus executes the following cURL statement:

```
POST [fhirserver]/DiagnosticReport
{ "resourceType": "DiagnosticReport",
  "subject": { "reference": "Patient/[patientId]" },
  "encounter": { "reference": "Encounter/[encounterId]" },
  "status": "preliminary",
  "code": {
    "coding": [ {
        "system": "http://loinc.org",
        "code": "LP31534-8",
        "display": "Study report"
    } ]
  }
}
```

This triggers the creation of an AuditEvent resource. This one is shown in abbreviated form, focusing on the elements relevant for the mapping:

```
{ "resourceType": "AuditEvent",
  "extension": [
    { "url": "https://fhirserver.com/extensions/auditevent-encounter",
    "valueReference": { "reference": "Encounter/[encounterId]" }},
    { "url": "https://fhirserver.com/extensions/auditevent-basedon",
    "valueReference": { "reference": "PlanDefinition/rad-wf" }}
  ],
  "action": "C",
  "recorded": "2020-08-14T08:42:51.523+02:00",
  "entity": [ {
    "what": { "type": "DiagnosticReport" },
    "detail": [ {
      "type": "RequestedURL",
      "valueString": "[fhirserver]/DiagnosticReport/"
    } ]
  } ]
}
```

The created AuditEvent resource refers to the respective Encounter resource and to the PlanDefinition resource “rad-wf” that defines the radiology workflow. The *action* field indicates the type of operation (C=Create) and the *entity* element contains details about the manipulated resource, i.e., the DiagnosticReport.

4.2 XES Log

The query for AuditEvent resources with the $xes operation returns the following XES event log (only one trace with one event is shown, extensions left out):

```
<?xml version="1.0" encoding="UTF-8" ?>
<log xmlns="http://www.xes-standard.org/">
  <string key="concept:name" value="PlanDefinition/rad-wf"/>
  <trace>
    <string key="concept:name" value="Encounter/enccase55"/>
    <event>
      <string key="concept:name" value="Report Writing"/>
      <date key="time:timestamp" value="2020-08-14T08:42:51.523+02:00"/>
    </event>
  </trace>
</log>
```


This detail of the resulting XES log shows the concept:name attributes on log and trace level, derived from the referenced PlanDefinition and Encounter resources respectively. The event (*report writing*) was generated for the AuditEvent resource presented in the previous section 4.1, that recorded the creation of a DiagnosticReport.

4.3 Process Model

Figure 3 shows the resulting model after importing the XES event log in ProM and analyzing it with the Inductive Visual Miner [14]. It is split up in two parts and highlights the similarity to the input model in figure 2. All traces were identified based on their Encounter reference and all AuditEvents were correctly mapped according to table 1. All 10 recorded executions are visible, with 5 skipping the first (*schedule appointment*) activity.

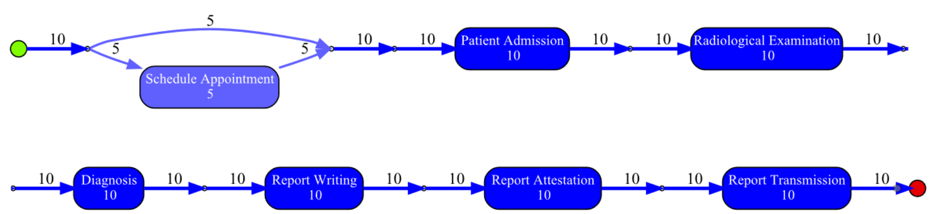

Fig. 3. Process model generated with the Inductive Visual Miner.

5 Discussion

The presented work is a proof of concept, making the case for a standards-based process analytics approach and making sure that the standard in development, HL7 FHIR, is aware of the capabilities and requirements of process mining. We were able to show how only minor extensions, namely the addition of Encounter and PlanDefinition references, and a simple mapping, enabled the analysis of the radiology practice workflow with process mining tools.

5.1 Impact on Standardization

In the FHIR Workflow project, the authors made a case for checking the usability of FHIR resources for process mining. Together, the working group members proposed the addition of a trace identifier to the AuditEvent and Provenance resources[7]: "We want to be able to search on all events (creates, updates, deletes, etc.) that happened during a given encounter, that happened based on a particular protocol or as a result of a particular order.". Based on the discussions in that group, we decided to use PlanDefinition and Encounter for the grouping and mapping approach. A proposal to extend AuditEvent to support this is currently under review for inclusion in the next FHIR release R5.

5.2 AuditEvent vs. Provenance

In this work we analyzed AuditEvent resources, building on existing approaches that aimed to analyze audit data [3, 7, 16]. However, HL7 FHIR also makes use of the concept of *provenance*, recording "information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness" [6]. A Provenance resource is created by the client (i.e., the person or system conducting the work) as opposed to the AuditEvent resource, which is created automatically by a server. The client should explain for what purpose a resource was edited (created, updated, deleted). In addition, a client can add information about the process (or policy) behind the edit, and provide reasoning why something was done (i.e., which path of a process model was taken). However, Provenance is (1) not widely used (yet), and (2) not documenting non-changing access to a resource (i.e., read). To summarize, Provenance can provide more detailed information on a process, but relies on the clients to record it and might thus be not present at all. Further research on the utilization of the Provenance resource for process mining is needed.

5.3 Considering Different Perspectives

In our example, A_w, the set of all AuditEvent resources recorded during the execution of a radiology workflow (as defined by the referenced PlanDefinition "rad-wf"), was split to traces based on the referenced Encounter resources. However, in fact, A_w represents a multiset of traces, that can be split based on the perspective you take on the data. A more generic approach should thus indicate the grouping behaviour in the query, based on the concepts developed in [12].

Another viable perspective would be, for example, to look at the active participants of the workflow. AuditEvent.agent is described as "an actor taking an active role in the event or activity that is logged" [10]. Mapping name and role to the corresponding fields of the XES Organizational extension allows for additional analysis, e.g., social networks or handover of work for medical or care personnel.

[7] https://jira.hl7.org/projects/FHIR/issues/FHIR-28100

References

1. van der Aalst, W.: Academic view: Development of the process mining discipline. In: Process Mining in Action: Principles, Use Cases and Outlook, pp. 181–196. Springer (2020)
2. Burattin, A.: Plg2: Multiperspective process randomization with online and offline simulations. In: BPM (Demos). pp. 1–6 (2016)
3. Cruz-Correia, R., Boldt, I., Lapão, L., Santos-Pereira, C., Rodrigues, P.P., Ferreira, A.M., Freitas, A.: Analysis of the quality of hospital information systems audit trails. BMC medical informatics and decision making **13**(1), 84 (2013)
4. Erdogan, T.G., Tarhan, A.: Systematic mapping of process mining studies in healthcare. IEEE Access **6**, 24543–24567 (2018)
5. Erickson, B.J., Meenan, C., Langer, S.: Standards for business analytics and departmental workflow. Journal of digital imaging **26**(1), 53–57 (2013)
6. Groth, P., Moreau, L.: W3C Working Group Note “PROV-Overview” (2013), https://w3.org/TR/2013/NOTE-prov-overview-20130430/, last access: 1.8.2020
7. Helm, E., Paster, F.: First steps towards process mining in distributed health information systems. Int. Journal of Electronics and Telecommunications **61**(2), 137–142 (2015)
8. HL7 International: FHIR Specification (v4.0.1: R4) Executive Summary (2019), https://hl7.org/fhir/summary.html, last access: 1.8.2020
9. HL7 International: FHIR Specification (v4.0.1: R4) Operations (2019), https://hl7.org/fhir/operations.html, last access: 1.8.2020
10. HL7 International: FHIR Specification (v4.0.1: R4) Resource AuditEvent - Detailed Desc. (2019), https://hl7.org/fhir/auditevent-definitions.html, last access: 1.8.2020
11. IEEE Task Force on Process Mining: IEEE Standard for eXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams. IEEE Std. **1849-2016**, 53–57 (2016). https://doi.org/10.1109/IEEESTD.2016.7740858
12. IHE IT Infrastructure Technical Committee: Add restful atna (query and feed). IHE ITI-TF Supplement, Rev. 3 (2019)
13. IHE Radiology Technical Committee: Standardized operational log of events (sole). IHE RAD-TF Supplement, Rev. 1.2 - Trial Implementation (2018)
14. Leemans, S.J., Fahland, D., van der Aalst, W.: Process and deviation exploration with inductive visual miner. BPM (Demos) **1295**(8) (2014)
15. Mandel, J.C., Kreda, D.A., Mandl, K.D., Kohane, I.S., Ramoni, R.B.: SMART on FHIR: a standards-based, interoperable apps platform for electronic health records. Journal of the American Medical Informatics Association **23**(5), 899–908 (2016)
16. de Murillas, E.G.L., Helm, E., Reijers, H.A., Küng, J.: Audit trails in openslex: paving the road for process mining in healthcare. In: Int. Conference on Information Technology in Bio- and Medical Informatics. pp. 82–91. Springer (2017)
17. Rebuge, Á., Ferreira, D.R.: Business process analysis in healthcare environments: A methodology based on process mining. Information systems **37**(2), 99–116 (2012)
18. Rojas, E., Munoz-Gama, J., Sepúlveda, M., Capurro, D.: Process mining in healthcare: A literature review. Journal of biomedical informatics **61**, 224–236 (2016)
19. Smile CDR Inc.: HAPI FHIR Interceptors: Overview (2020), https://hapifhir.io/hapi-fhir/docs/interceptors/interceptors.html, last access: 25.8.2020
20. Verbeek, H., Buijs, J.C., Van Dongen, B.F., van der Aalst, W., et al.: Xes, xesame, and prom 6. In: CAiSE Forum. vol. 72, pp. 60–75. Springer (2010)