

Ontology-Driven Enhancement of Process Mining With Domain Knowledge

Simon Eichele^{1,†}, Knut Hinkelmann^{1,2,†} and Maja Spahic-Bogdanovic^{1,2,*,†}

¹FHNW University of Applied Sciences and Arts Northwestern Switzerland, Riggenbachstrasse 16, Olten, 4600, Switzerland

²University of Camerino (UNICAM), Via Madonna delle Carceri 9, Camerino MC, 62032, Italy

Abstract

Process mining is a technique used to analyze and understand business processes. It uses as input the event log, a type of data used to represent the sequence of activities occurring within a business process. An event log typically contains information such as the case ID, the performed activity's name, the activity's timestamp, and other data associated with the activity. By analyzing event logs, organizations can gain a deeper understanding of their business processes, identify areas for improvement, and make data-driven decisions to optimize their operations. However, as the event logs contain data collected from different systems involved in the process, such as ERP, CRM, or WfMS systems, they often lack the necessary context and knowledge to analyze and fully comprehend business processes. By extending the event logs with domain knowledge, organizations can gain a more complete and accurate insight into their business processes and make more informed decisions about optimizing them. This paper presents an approach for enhancing process mining with domain knowledge preserved in domain-specific OWL ontologies. Event logs are typically stored in structured form in relational databases. This approach first converts the process data into an event log which is then mapped with ontology concepts. The ontology contains classes and individuals representing background knowledge of the domain, which supports the understanding of the data. A class for the specific activities forms the link between the event log and the ontology. In this manner, it is possible to map the domain knowledge to a particular case and activity. This allows to determine conditions that must be satisfied for executing tasks and to prune discovered process models if they are too complex. This approach is demonstrated using data from the student admission process at FHNW and has been implemented in Protégé.

Keywords

Process mining, Domain knowledge, Ontology, Enhancement of event log, Event data, Knowledge-augmented process mining

1. Introduction

Process mining is a technique used to analyze and understand business processes [1]. It uses as input the event log, a type of data used to represent the sequence of activities occurring within

In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2023 Spring Symposium on Challenges Requiring the Combination of Machine Learning and Knowledge Engineering (AAAI-MAKE 2023), Hyatt Regency, San Francisco Airport, California, USA, March 27-29, 2023.

*Corresponding author.

†These authors contributed equally.

✉ simon.eichele@students.fhnw.ch (S. Eichele); knut.hinkelmann@fhnw.ch (K. Hinkelmann); maja.spahic@fhnw.ch (M. Spahic-Bogdanovic)

ORCID 0000-0002-1746-6945 (K. Hinkelmann); 0000-0003-1625-0162 (M. Spahic-Bogdanovic)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

a business process. An event log typically contains information such as the case ID and for each the performed activity the name, the timestamp, and other data associated with the activity. By analyzing event logs, organizations can gain a deeper understanding of their business processes, identify areas for improvement, and make data-driven decisions to optimize their operations [2]. However, as the event logs contain data collected from different information systems involved in the process, such as ERP, CRM, or WfMS systems, they often lack the necessary context and knowledge to analyze and fully comprehend business processes. By extending the event logs with domain knowledge, organizations can gain a more complete and accurate insight into their business processes and make more informed decisions about optimizing them. Figure 1 gives an overview of the different applications of process mining presupposing event logs.

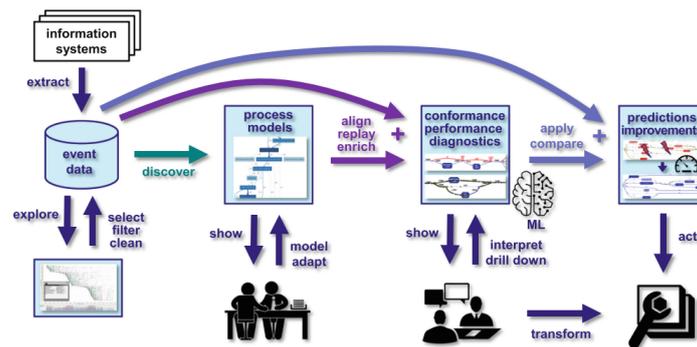


Figure 1: High-level view on process mining [1]

Since event logs are usually stored in structured form in relational databases or a sequential process event log [3], process mining techniques are designed to work with structured event logs [2]. In reality, many of a company's processes are neither digitized nor structured or are executed manually. Especially in such situations, business logic is required, so it is important to recognize how individual employees or applications react to the actual work environment and perform the tasks [4].

This paper presents an approach for incorporating domain knowledge stored in domain-specific OWL ontologies into process mining, enabling semantic process mining and reasoning behind activity execution. The approach was developed using the design science research methodology and applied to the partially digitized master's degree program application process at the FHNW University of Applied Sciences and Arts Northwestern Switzerland (FHNW). Many of the process steps are executed by process experts and require domain knowledge. The process data is recorded manually in an Excel file, where each line represents one application and contains both input and output data of the process activities. The developed approach maps the case ID, activities, and application data with the ontology concepts. The ontology contains classes and individuals representing background knowledge of the domain, which supports the understanding of the data. A class for the specific activities forms the link between the event log and the ontology. In this manner, it is possible to map the domain knowledge to a particular case and activity. This allows to determine conditions that must be satisfied for executing tasks and to prune discovered process models if they are too complex. Additionally, it enables the

use of machine learning, for example, to learn types of activities, to predict possible next tasks in the process flow, and to learn the conditions of gateways using knowledge represented in concepts and relations of the ontology. The developed artifact is implemented in Protégé, and data from the student admission process at FHNW was used for the evaluation.

The benefit of this approach is providing context and the ability to reason on data, which in return helps to make decisions on accurate information and to transform raw data into actionable knowledge.

The paper is structured as follows: Section 2 discusses the state-of-the-art. The research strategy used to develop the artifact is described in Section 3. The specific development steps are detailed in Section 4. Finally, Section 5 elaborates on the research contribution and discusses further research steps.

2. State of the Art on Semantic Process Mining

In the following subsections, relevant works in the area of semantic process mining are presented. At the end of this section, the research gap is discussed.

2.1. Process Mining and Knowledge-Intensive Processes

With regard to business processes, a distinction can be made between process and business logic [4, 5, 6]. Process logic deals with the flow and sequence of activities within a business process and is focused on the "how" of a business process, typically represented in a process model [4]. On the other hand, business logic refers to the rules and policies that govern operations and decision-making within an organization. Business logic represents the knowledge in the process [4]. It includes understanding how individual workers or applications respond to real-world situations of work and perform the tasks assigned to them [5, 6].

Further, a business process can be classified as structured, semi-structured, or unstructured (ad-hoc) [7]. Structured processes can be predefined and have rules that guide the process flow. Unstructured processes, conversely, are unforeseen and do not have a predefined sequence of activities [7]. These processes are often knowledge-intensive and carried out by knowledge workers or domain experts who rely on their expertise to determine the activities required to achieve the process goals [8]. Business logic is, therefore, critical to understanding how unstructured processes work [7]. Semi-structured processes combine elements of both structured and unstructured processes, with some activities having predefined sequences while others do not [7]. The correlation between business logic and process logic in semi-structured processes can vary depending on the specific process, and its level of standardization [9]. In general, the less structured a process is, the more determining its flow relies on business logic. The flow of semi-structured and especially unstructured processes depends heavily on the knowledge of the knowledge worker [10]. Unstructured and partially semi-structured processes are associated with the class of knowledge-intensive processes (KiP) [10]. According to Folino and Pontieri [2] full spectrum of process mining techniques can be exploited when the event log data are of good quality and the process is structured. As soon as the process is unstructured or less structured, or the event logs are incomplete, the existing process mining techniques reach their limits.

According to Beerepoot et al. [11], one of the nine biggest unsolved Business Process Management (BPM) problems is enhancing process mining through the integration of domain-specific and common sense knowledge. They argue that the event logs are prone to noise or incompleteness and that conventional process mining is insufficient to generate high-quality results. Enhancing process mining by integrating domain-specific and commonsense knowledge would lead to high-quality, trustworthy results (see Figure 2). Data alone is not enough to provide insight or make decisions. Data must be analyzed, interpreted, and contextualized to derive meaning and value. Applying reasoning techniques to data can identify trends, correlations, or anomalies that may not be apparent immediately. Reasoning on data provides the ability to derive new knowledge from existing data, uncover hidden patterns and relationships, and make informed decisions based on facts [11].

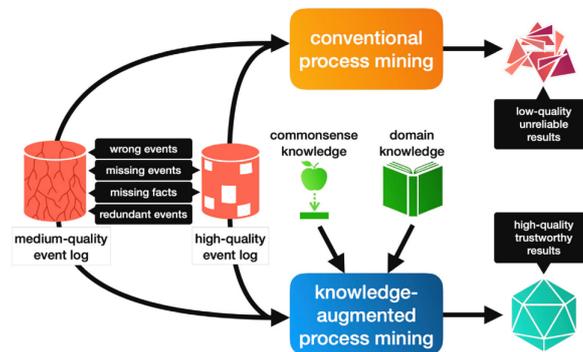


Figure 2: Enhancing process mining by integrating domain-specific and commonsense knowledge [11]

2.2. Semantic Process Mining and Enrichment of Event Logs

Semantic or knowledge-augmented [11] process mining aims to use the semantics (meaning or relationship of meanings) of data recorded in event logs to provide results at a conceptual level [12]. Various studies have already demonstrated the potential of enriching event logs with additional information. Ingvaldsen and Gulla [13] have shown, based on a real case, the application of process mining when event logs are enriched with data from several information management systems. Such integration type is also called the data warehouse approach. However, this approach does not consider activities performed outside the ERP system. In contrast, De Medeiros et al. [12] pointed out that event logs are simply strings without any meanings and provide purely synthetic analysis. For this reason, they have developed a semantic-based and context-aware framework based on Semantic Web Services technology, which supports semantic process mining. Figure 3 represents the main building blocks: ontology, mapping from the process model or event logs to ontology concepts, and ontology reasoner. As defined by Gruber [14], ontologies establish a collection of commonly understood concepts essential for analysis and formalize their interrelationships and attributes. Building on this framework, Tran et al. [15] have designed an approach that enables ontology-based data integration and knowledge discovery. Thereby, the event logs are mapped with concepts of the TOVE (TOronto

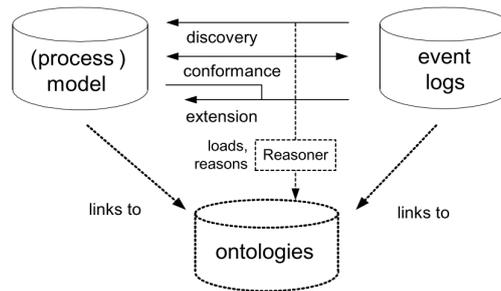


Figure 3: The fundamental components required for creating semantic process mining algorithms [12]

Virtual Enterprise) ontology. This approach assumes that event logs can be extracted from Process-Aware Information System (PAIS) or other database systems. In addition, the minimum requirements for the event logs are mandated, such as case ID, activity, and timestamp.

Nykänen et al. [16] introduced an approach that links ontology structures with event logs, primarily applicable to engineering and maintenance processes. A process and product ontology, shown in Figure 4, forms the foundation of this approach, which assumes that the process is divided into phases, each consisting of specific activities. The product ontology describes the documents involved in the process, and they are linked to activities recorded in the event log. Since activities are defined as classes and sub-classes in the ontology, there is a hierarchical relationship between them, allowing the examination of the process at various levels of detail, such as individual phases. The approach was developed based on a fictitious process and requires an event log that meets the minimum requirements, i.e., contains case ID, activity, and a timestamp.

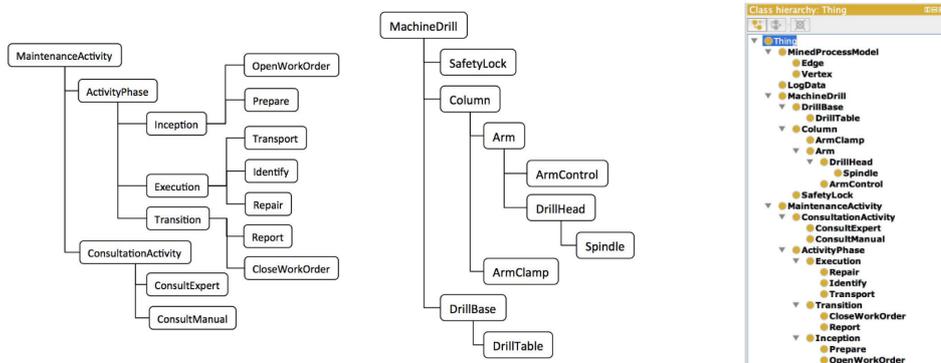


Figure 4: Process and product ontology [16]

Dixit et al. [17] argue that domain knowledge can improve process discovery by addressing the limitations of the data. The evolved approach specifies domain knowledge in the form of constraints and is applied at the post-processing stage. To form the background knowledge, a declarative process model encoded in DECLARE language is used. This approach modifies already existing event logs. Okoye et al. [18] and Okoye [19] also highlight the importance of machine-understandable systems that process information semantically annotated or formally

represented in an ontology. Therefore, a Semantic Process Mining and Model Analysis Framework (SPMaAF) approach was developed, which annotates semantically process instances with concepts from the real world and links them to a domain ontology. This improves the process mining results by providing relevant domain knowledge or information about the process instance. Figure 5 represent the framework's components. In the first step, process discovery is

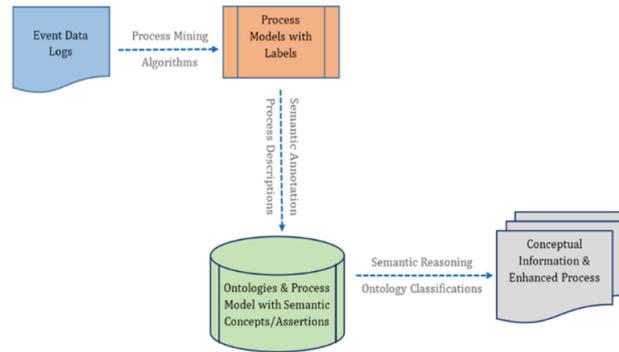


Figure 5: Semantic Process Mining and Model Analysis Framework (SPMaAF) [18]

performed, then cross-validation between the process model and traces from the event log is executed to check the trace fitness. Subsequently, information related to the different entities present in the event log and the process model is mapped to the concepts of the underlying ontology. The proposed approach assumes that an event log, which satisfies the minimum requirements, exists and the activities are executed sequentially.

Khan et al. [20] propose a knowledge-centric framework to address the semantic incompleteness of event traces. The approach identifies missing events by inferring potential relations between entity pairs based on existing knowledge, aiming to complement the event log. The approach generates a set of candidate event traces as output to improve the utility of mined models. The approach assumes the knowledge graph encodes event data, enterprise domain knowledge, and business rules. However, the designed approach primarily targets issues with process discovery when event logs are noisy.

In contrast to the previous approaches, Adamo et al. [21] have extended the Business Process Model and Notation (BPMN) standard to represent historically, causally, and rationally based co-occurrence dependencies and their rationales. Figure 6 represent the developed annotation. They argue that the relationship between activities has different types and motivations (e.g., a norm, goal, or an ontological law-of-nature). Process modelers and analysts have to invest time in learning the notation in this approach. While the visualization of relationships enhances the comprehension and outcome of the redesign, it is still not yet sufficient to incorporate domain knowledge and the underlying reasoning behind the decisions made during process execution.

Although various approaches already exist, integrating domain or common sense knowledge into process mining is one of the nine biggest unsolved BPM challenges [11]. The approaches mentioned previously assume that a process logic exists. Some approaches assume that most activities are performed sequentially. In contrast, processes for which event logs exist only partially or for which there is no event log and the flow should be learned only from process

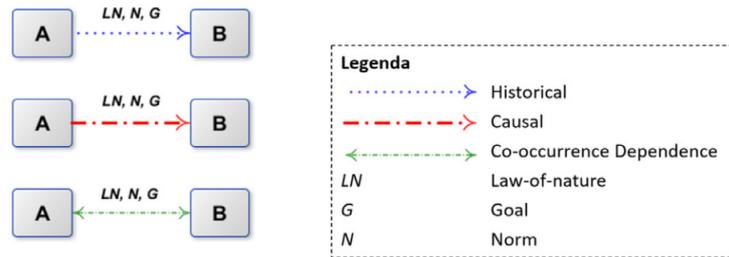


Figure 6: Proposed BPMN extension

data remain an ongoing challenge. This paper aims to contribute to this challenge.

3. Design Science Research

The approach presented in this paper focuses on developing an artifact that solves a real-life problem. Therefore, the Design Science Research (DSR) paradigm was applied [22].

The problem was identified through a literature review and a case study that refers to the student admission process for the master’s program Business Information Systems (BIS) at FHNW. A workshop and interview were conducted to collect data within the case study. The suggestion phase involved identifying appropriate methods for representing domain knowledge and ensuring machine interpretability. A representation of domain knowledge in the form of an ontology was identified as appropriate. The domain knowledge was captured in an OWL ontology, following the methodology for creating ontologies proposed by Noy and McGuinness [23].

In the evaluation phase, the effectiveness of the domain knowledge in improving the understanding of the data was assessed. A knowledge worker formulated several questions and tested whether they could be answered using domain knowledge. These queries were based on the competency question. In the last phase of the DSR, the conclusion, the contribution to the body of knowledge, and further research were discussed.

4. Design and Development of the Approach

This section provides a detailed discussion of the development process of the artifact. It includes an explanation of the main components of the proposed solution and an overview of the design and development steps.

4.1. Student Admission Process

The FHNW admission process is a knowledge-intensive and semi-structured process that involves multiple knowledge workers, such as administrators and program heads. The process begins when a candidate fills out an online form, after which all activities are performed manually, and outcomes are documented in an Excel file. Due to the manual execution of

activities, timestamps are absent. This makes it impossible to determine the order in which activities are executed for each process instance. The process has more than 30 activities and involves admission criteria such as university accreditation, language proficiency, and grades. The sequence of activities is not always significant, and knowledge workers have the flexibility to execute activities in various orders. If an activity outcome reveals that a candidate does not meet any of the mandatory requirements, the application is rejected, thus ending the process. Over the years, regulations have changed, and transforming grades, particularly for foreign universities, remains a significant challenge as multiple formulas can be used. It also happens that none of the formulas gives a correct result. Domain knowledge is critical to executing the process effectively and ensuring the applicants are treated equally. Table 1 shows an excerpt of the manually recorded process data.

Table 1
Process data

Last Name	First Name	Start	FF	FF	Gender	Bachelor	University	University Country	University (H, B, MSc)	Official Grade	Grade (+, -, 0)	Experience (y/m)	Experience (+, -, 0)	English	SDP	Deputy HoP	Interview	Salary	Condition	BPM	BA	IS	Justification	Citizenship	Location of residence	Visum ?	Remarks
Sep 22	X	M	B	Tech	Information	Sankhdeep Institute	India	H	7.22	5.30	-	31y	++	BA was already in EN	accept	accept	done	accepted in FHNW		X	X			India	India	Yes	Accepted
Sep 22	X	M	Business Administration	University of Applied	Germany	H	2.4	4.90	+	-3y	++			BA was already in EN	accept	accept	done	accepted in FHNW		X	X			Germany	BE	No	
Sep 22	X	M	Business Administration	HSLU	Switzerland	H	5.00	-	4y 5m	++			Basel/Basel HSLU	accept	accept		accepted in FHNW		X	X			CH	BE	No		
Sep 22	X	F	Computer Science	FHNW	Switzerland	H	5.10	+	8y 10m	++			FCE level C1, grade A	accept	accept		accepted in FHNW		X	X			CH	AD	No		
Sep 22	X	M	Engineering and Management	FHNW	Switzerland	H	4.80	-	10y 8m	++			good EN grades in	accept	accept		accepted in FHNW		X	X			CH	AD	No		
Sep 22	X	M	Business Information	FHNW	Switzerland	H	5.60	++	6y 6m	++				accept	accept		accepted in FHNW							CH	AD	No	
Sep 22	X	M	Business Information	BFH	Switzerland	H	5.00	-	5y 9m	++			FCE level B2, grade C	accept	accept		accepted in FHNW							CH	BE	No	
Sep 22	X	M	Wirtschaftsinformatik	FHNW	Switzerland	H	5.30	++	5y 6m	++			CAE C1, Grade C	accept	accept		accepted in FHNW							CH	AD	No	
Sep 22	X	F	Betriebswirtschaftslehre	University of Liechtenstein	Liechtenstein	H	5.80	-	3y 9m	++			1y high school USA	accept	accept		accepted in FHNW							Germany	ZH	No	
Sep 22	X	M	Bachelor of Arts, Major	University of Notre Dame	United States of	H	5.80	++	-12y	++			BA was already in EN	accept	accept		accepted in FHNW		X	X			USA	BE	No		
Sep 22	X	M	Business Engineering	FHNW	Switzerland	H	4.70	0	3y 7m	++			CAE C1, Grade C	accept	accept		accepted in FHNW		X	X			CH	ZH	No		
Sep 22	X	M	Business Information	ZHAW	Switzerland	H	4.80	0	-3y	++			CAE level B2	accept	accept		accepted in FHNW							CH	ZH	No	
Sep 22	X	M	Wirtschaftsinformatik	FHNW	Switzerland	H	5.00	-	4y 6m	++			good EN grades in	accept	accept		accepted in FHNW							CH	BL	No	Bachelor
Sep 22	X	M	Betriebsökonomie	FHNW	Switzerland	H	4.80	-	8y 8m	++			CAE level C1, Grade C	accept	accept		accepted in FHNW							CH	AD	No	Bachelor

An application corresponds to a single row and can be equated with a case. Whenever a new application is received, this is recorded in the next free row, and thus the row number can be equated with the case ID. Each column in the dataset represents an activity that can be carried out by one or multiple knowledge workers and may occur once or multiple times in one process instance. As there is no event data except for the process trigger, the process data and its business logic provide the foundation for the knowledge base. The quality of the data had to be ensured, therefore, preprocessing of process data was necessary.

4.2. Building a Knowledge Base from Process Data and Business Logic

The first step to building the ontology was to define the scope. For this purpose, several competence questions [24] were defined, which the ontology should be capable of answering:

1. Which task should be executed in the process model for a given application?
2. How can an overview be created to find any errors in the ontology?
3. What are the criteria for accepting a candidate at FHNW?
4. Who is conducting which specific task?

These questions were used to evaluate the ontology. A bottom-up approach was taken to determine the classes and their hierarchy [23].

Initially, the class "Person" and "Application" were introduced. The subclass "Candidate" and "Staff" were included under the "Person" class, and the "Role" class was created to identify the different roles that persons can have. An application involves several documents, and as a result, the "Documents" class was added, containing details about the candidate's country of origin

("Country" class), individual degree ("IndividualDegree" class), English skills ("EnglishSkills" class), university ("HigherEducationInstitution" class), grading system ("GradingSystem" class), and study program ("StudyProgram" class). Candidates specify the semester (Spring or Fall) and study mode (Part Time or Full Time) they wish to enroll in, leading to the development of the "Semester" and "StudyMode" classes. At the end of the process, a decision is made regarding whether the candidate must attend a pre-master program ("PreMaster" class). Lastly, the specific process activities were added as a class "Activity_BPMN".

Ontologies can establish connections between classes and instances by assigning properties to associations that allow inferences to be made about them [25]. Object properties establish links between individuals, classes, or both [25]. The relationships were defined based on the described current student admission process from the interviews, regulations, and process data. Figure 7 visualizes the different classes and object properties defined by using the tool Protégé.

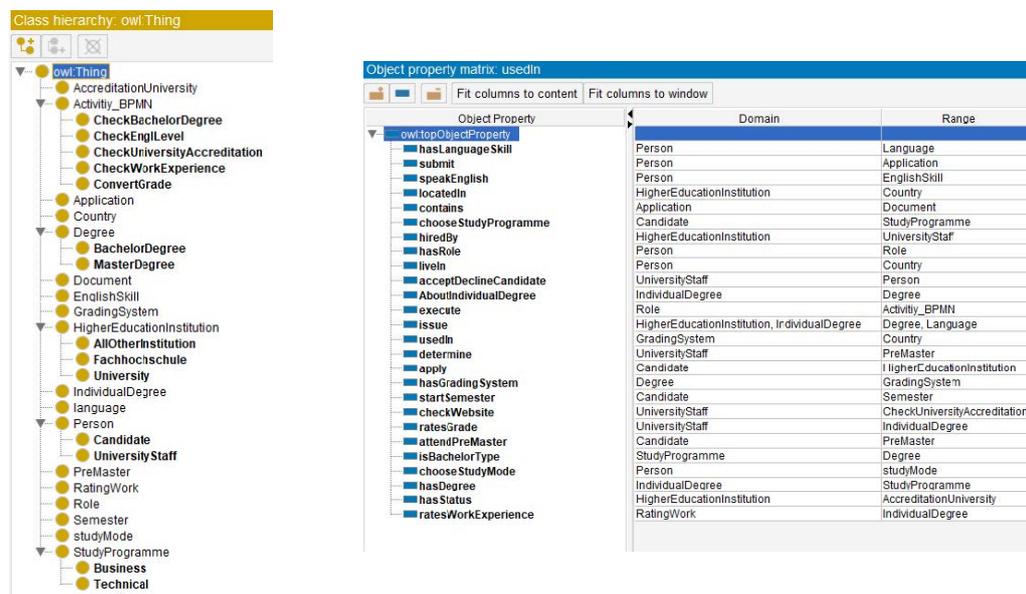


Figure 7: Classes and object properties

Data properties, on the other hand, establish connections between individuals and data values. Figure 8 illustrates the hierarchy of data properties, including both asserted and inferred properties. The "Range" column specifies the classes to which each data property is linked [25]. For instance, the data property "ratingGradeStatus", assigned to the class "IndividualDegree", has a data type of "xsd:string." Properties can have different facets that describe various characteristics of the values they hold, including value type, allowed values, and cardinality [25].

Instances, also called individuals, represent objects within a domain [25]. Each individual contains a unique value for each property of the related class. The combination of instances and classes forms the knowledge base. Figure 9 shows such an example of an instance within a class and individually with its own allocations to class, object, and data properties.

Data Property	Domain	Range
owl:topDataProperty		
ratingGradeStatus	IndividualDegree	xsd:string
startDate	Semester	xsd:string
firstName	Person	xsd:string
hasWorkExperience	Candidate, RatingWork	xsd:string
minGradeAtOf	GradingSystem	xsd:string
conductInterview	Candidate	xsd:string
hasGradeNumber	IndividualDegree, GradingSystem	xsd:double
passingGrade	GradingSystem	xsd:integer
PassEngl	EnglishSkill	xsd:string
recognitionStatus	AccreditationUniversity	xsd:string
hasGender	Person	xsd:string
hasAverageGrade	IndividualDegree	xsd:double
hasEnglLevel	EnglishSkill	xsd:string
maxGradeAtOf	GradingSystem	xsd:string
minGrade	GradingSystem	xsd:integer
GradeInEngl	IndividualDegree	xsd:double
passingGradeAtOf	GradingSystem	xsd:string
staffID	Person	xsd:integer
hasTitle	Degree	xsd:string
lastName	Person	xsd:string
candidateID	Person	xsd:integer
hasSemester	studyMode	xsd:double
hasCreditNumber	IndividualDegree	xsd:integer
hasCantonOfResidence	Candidate	xsd:string
ratingWorkExperience	RatingWork	
decision	IndividualDegree	xsd:string
hasEnglBA	EnglishSkill	xsd:string
HighEngl	EnglishSkill	xsd:string
LowEngl	EnglishSkill	xsd:string
maxGrade	GradingSystem	xsd:integer

Figure 8: Data property

The screenshot shows an ontology editor interface. On the left, a 'Data hierarchy' tree shows the class 'Country' with its instances: Afghanistan, Austria, Germany, Italy, Switzerland, and USA. The 'USA' instance is selected. On the right, the 'Direct instances: Simon' view shows the instance 'Simon' with its type 'Candidate'. Below this, the 'Property assertions: Simon' view lists various assertions for the instance, including 'hasName "Simon"', 'hasWorkExperience "10y3m"', and 'hasRole Candidate'.

Figure 9: Example of an instance in the class "Country" and with allocations to the class and property

4.3. SWRL Rules in the Ontology

SWRL, the Semantic Web Rule Language, enables the creation of rules expressed in OWL concepts that allow more powerful deductive reasoning than OWL alone [26]. The SWRL rules are relevant to answering competency question: "Which task should be executed in the process model for a given application?".

An OWL ontology consists of axioms and facts, and there are different types of axioms, e.g., SubClass, EquivalentClass, and Rule axioms. A rule axiom contains an antecedent (body) and a consequent (head), both of which have a set of atoms [27]. The SWRL rule has a "human-readable" form, with the antecedent and consequent conjunctions of atoms written as $a^1 \wedge \dots \wedge a^n$. Variables are specified by preceding them with a question mark $?x$.

Further considerations were required to interpret the loose values once the ontology was enriched with data. The activities of the admissions process, such as verification of bachelor's degree and work experience, were represented as instances in the ontology. Thus, when a new application arrives, the corresponding activity should be triggered based on the candidate's

information. To achieve this, the SWRL rules must be executed. For example, if the candidate has a Bachelor's degree from a Swiss university, then the knowledge worker must check the accreditation and, thus, perform the activity "Check Website Swiss universities". This can be represented as the following SWRL rule:

```
HigherEducationInstitution(?h) ^ locatedIn(?h, Switzerland) ->
execute(Assistant, CheckWebsiteSwissuniversities)
```

Furthermore, there is the need to map a negation. For example, if the university at which the candidate received the bachelor's degree is not in Switzerland. In such a case, the knowledge worker performs another activity and checks the accreditation on a different website. This negative can be represented as the following SWRL rule:

```
fhnw_admission:HigherEducationInstitution(?h) ^
fhnw_admission:locatedIn(?h, ?c) ^
differentFrom(?c, fhnw_admission:Switzerland) ->
fhnw_admission:execute(fhnw_admission:Assistant,
fhnw_admission:CheckWebsiteAnabin)
```

Hermit reasoner was used to detect whether the ontology is consistent and identify subsumption relations between classes.

4.4. Evaluation of the Ontology

The developed ontology was tested against the defined competency questions, and the knowledge worker verified the output. For this purpose, various SPARQL queries were specified. This query is relevant to answer the competency question "What are the criteria for accepting a candidate at FHNW?":

```
PREFIX sss: <http://www.simoneichele.ch/FHNW_Admission#>
SELECT ?Candidate ?Application ?Grade ?WorkExperience
?RatingWork ?UniAccreditation ?Accepted
WHERE
{
?Candidate sss:submit ?Application.
?Candidate sss>AboutIndividualDegree ?IndividualDegree.
?Candidate sss:hasWorkExperience ?WorkExperience.
?Application sss:ratesWorkExperience ?RatingWork.
?Application sss:decision ?Accepted.
?IndividualDegree sss:hasAverageGrade ?Grade.
?IndividualDegree sss:issue ?AccreditationUniversity .
?HigherEducationInstitution sss:hasStatus ?UniAccreditation.
FILTER (?Accepted = "accepted informed") }
```

The result is a table showing the criteria for a positive decision:

The next SPARQL query is relevant to answering competency question "Who is conducting which specific task?":

Table 2

Result of the SPARQL query

Candidate	Application	Grade	WorkExperience	RatingWork	AccreditationUniversity	UniAccreditation	Accepted	Bachelor
Simon	SimonApplication	4.7	10y3m	+	FHNW	StatusH+	accepted informed	SimonBachelor
1HansMusterr	1HansMustermann	4.7	10y3m	++	FHNW	StatusH+	accepted informed	1HansMustermann

```

PREFIX sss: <http://www.simoneichele.ch/FHNW_Admission#>
SELECT ?Task ?Role
WHERE { ?Role sss:execute ?Task. }

```

As a result, the individual roles and activities they perform are listed.

4.5. Automatic Mapping of Process Data to Ontology through Axioms

The process data is stored in a spreadsheet, so the Cellfie plugin was used for the import [28]. MappingMasterDSL, a domain-specific language (DSL) based on Manchester syntax, is utilized to construct transformation expressions that define mappings from spreadsheet content to OWL ontologies [29]. For this purpose, it was necessary to create axioms. First, the transformation rules were defined in the Transformation Rule Editor. The following is an example where a new candidate is imported where the Individual defines the instance name:

```

Individual: @B2
Types: Candidate
Facts: candidateID @A2(xsd:integer)
Facts: lastName @C2
Facts: firstName @D2
Facts: startSemester @E2
Facts: chooseStudyMode @F2
Facts: chooseStudyMode @G2
Facts: hasGender @H2
Facts: hasWorkExperience @N2
Facts: hasLanguageSkill @P2
Facts: conductInterview @S2(mm:DefaultLiteral="No")
Facts: liveIn @Z2
Facts: attendPreMaster @V2
Facts: attendPreMaster @W2
Facts: attendPreMaster @X2
Facts: AboutIndividualDegree @AE2
Facts: submit @AF2

```

After defining the rules, the axioms can be generated, and a preview is displayed. Based on the Cellfie plugin [28], a total of 14 axioms are generated, which can be added to either a new or an existing ontology. Figure 10 visualizes the generated axioms and the instance import. During the import process, it is verified if there is an existing instance of the connected object properties or data properties. A link between the values to the corresponding classes or data

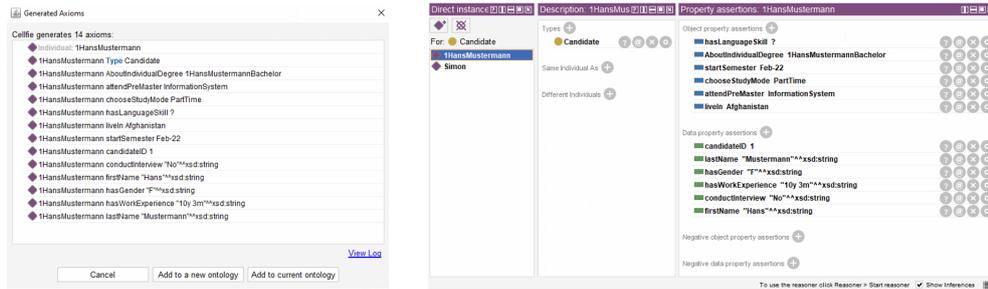


Figure 10: Preview of Generates Axioms and Instance of Imported Process Data

values is created if such an instance exists. However, if no instance is present, the link cannot be established, and an unlinked instance without assignment to any class is generated. In such cases, manual mapping of the instance to a class is necessary.

Currently, a complete event log for the Admission process does not exist. Nevertheless, the ontology has already been built in such a way that a subsequent import of event logs would be possible. The event log is assumed to meet the minimum requirements and has a case ID and the activity performed. A synthetic event log was generated for demonstration purposes. Figure 11 visualizes the mapping procedure.

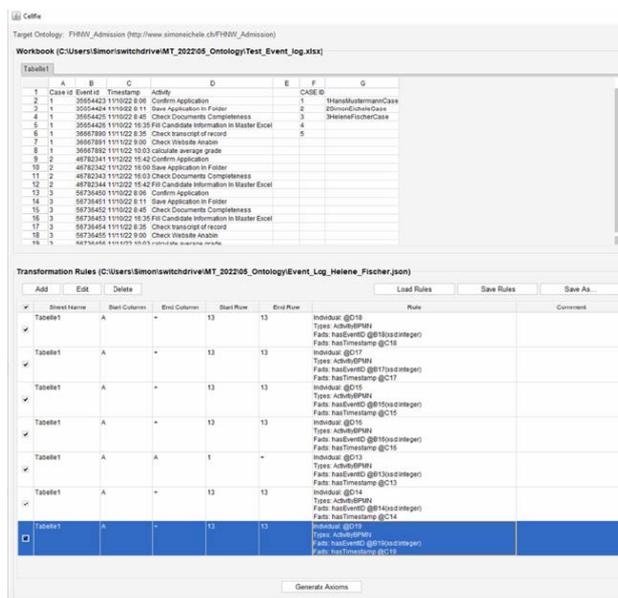


Figure 11: Event log mapping using Cellfie plugin [28]

Specific to the event log, transformation rules are defined, and then activities from the event log are mapped to the activities in the "Activity_BPMN" class. Event logs that contain one or more fully executed process instances, as well as event logs that have open process instances, could be loaded into the ontology. In this way, it is possible to make suggestions for ongoing instances, such as which activities still need to be carried out. In order not to generate redundant

data, the event logs are not stored in the ontology but are afterward deleted and thus remain in the original system, such as CRM, ERP, or WfMS.

5. Conclusion

The presented approach shows the potential of enriching process mining with domain knowledge even in the absence of explicit event logs. Traditional process mining approaches start with the analysis of event logs. However, the approach presented in this paper uses process data generated during process execution as a starting point. In addition, the approach can also use event logs as a starting point.

The approach has also shown that reasoning and justification can be provided regarding the execution of certain activities. The ontology provides information about which activities are necessary and which are not or identify criteria for the process flow depending on the knowledge about decision criteria. For example, the ontology relates information about applicant, university and applications. For example, if the accreditation status of a university, at which the applicant made the bachelor degree, is already included in the ontology, the activity "CheckUniversityAccreditation" can be skipped for this application. The approach presented is still a work in progress, and there are several opportunities for future research. One such area is enriching the ontology with additional concepts such as time. Furthermore, exploring the combination of ontology with generic ontologies such as TOVE or the BPMN 2.0 ontology [30] is another opportunity for future research.

Ultimately, the artifact can be applied to different types of process mining, such as process discovery, conformance checking, and action-oriented process mining.

References

- [1] W. M. P. van der Aalst, Process mining: A 360 degree overview, in: W. M. P. van der Aalst, J. Carmona (Eds.), *Process Mining Handbook*, Springer, 2022, pp. 3–34. doi:10.1007/978-3-031-08848-3_1.
- [2] F. Folino, L. Pontieri, AI-Empowered Process Mining for Complex Application Scenarios: Survey and Discussion, *Journal on Data Semantics* 10 (2021) 77–106. doi:10.1007/s13740-021-00121-2.
- [3] S. Esser, D. Fahland, Multi-Dimensional Event Data in Graph Databases, *Journal on Data Semantics* 10 (2021) 109–141. doi:10.1007/s13740-021-00122-1. arXiv:2005.14552.
- [4] K. Hinkelmann, Business Process Flexibility and Decision-Aware Modeling—The Knowledge Work Designer, in: D. Karagiannis, H. C. Mayr, J. Mylopoulos (Eds.), *Domain-Specific Conceptual Modeling*, Springer, 2016, pp. 397–414. doi:10.1007/978-3-319-39417-6_18.
- [5] B. von Halle, L. Goldberg, *The Decision Model: A Business Logic Framework Linking Business and Technology*, 1st ed., Auerbach Publications, 2009. doi:10.1201/9781420082821.
- [6] S.-M.-R. Beheshti, B. Benatallah, S. Sakr, D. Grigori, H. R. Motahari-Nezhad, M. C. Barukh,

- A. Gater, S. H. Ryu, *Process Analytics: Concepts and Techniques for Querying and Analyzing Process Data*, Springer, 2016. doi:10.1007/978-3-319-25037-3.
- [7] A. Gadatsch, *Grundkurs Geschäftsprozess-Management*, Springer, 2020. doi:10.1007/978-3-658-27812-0.
- [8] M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers, *Fundamentals of Business Process Management*, Springer, 2018. doi:10.1007/978-3-662-56509-4.
- [9] S. Dustdar, T. Hoffmann, W. van der Aalst, Mining of ad-hoc business processes with teamlog, *Data Knowledge Engineering* 55 (2005) 129--158. doi:10.1016/j.datak.2005.02.002.
- [10] C. Di Ciccio, A. Marrella, A. Russo, Knowledge-Intensive Processes: Characteristics, Requirements and Analysis of Contemporary Approaches, *Journal on Data Semantics* 4 (2015) 29--57. doi:10.1007/s13740-014-0038-4.
- [11] I. Beerepoot, C. Di Ciccio, H. A. Reijers, S. Rinderle-Ma, W. Bandara, A. Burattin, D. Calvanese, T. Chen, I. Cohen, B. Depaire, G. Di Federico, M. Dumas, C. van Dun, T. Fehrer, D. A. Fischer, A. Gal, M. Indulska, V. Isahagian, C. Klinkmüller, W. Kratsch, H. Leopold, A. Van Looy, H. Lopez, S. Lukumbuzya, J. Mendling, L. Meyers, L. Moder, M. Montali, V. Muthusamy, M. Reichert, Y. Rizk, M. Rosemann, M. Röglinger, S. Sadiq, R. Seiger, T. Slaats, M. Simkus, I. A. Someh, B. Weber, I. Weber, M. Weske, F. Zerbato, The biggest business process management problems to solve before we die, *Computers in Industry* 146 (2023) 103837. doi:10.1016/j.compind.2022.103837.
- [12] A. De Medeiros, A. Karla, W. Van Der Aalst, C. Pedrinaci, Semantic process mining tools: Core building blocks, in: *16th European Conference on Information Systems*, 2008. URL: <http://is2.lse.ac.uk/asp/aspecis/20080169.pdf>.
- [13] J. E. Ingvaldsen, J. A. Gulla, Model-Based Business Process Mining, *Information Systems Management* 23 (2006) 19--31. doi:10.1201/1078.10580530/45769.23.1.20061201/91769.3.
- [14] T. R. Gruber, A translation approach to portable ontology specifications, *Knowledge Acquisition* 5 (1993) 199--220. doi:10.1006/knac.1993.1008.
- [15] T. Tran, T. Kim, H. Werthner, An Ontology-based Framework for Enriching Event-log Data, in: P. Lorenz, E. Ammann (Eds.), *The Fifth International Conference on Advances in Semantic Processing SEMAPRO*, 2011, pp. 110--115. URL: https://publik.tuwien.ac.at/files/PubDat_201966.pdf.
- [16] O. Nykänen, A. Rivero-Rodriguez, P. Pileggi, P. A. Ranta, M. Kailanto, J. Koro, Associating event logs with ontologies for semantic process mining and analysis, in: *Academic-MindTrek '15: Proceedings of the 19th International Academic Mindtrek Conference*, Association for Computing Machinery, 2015, pp. 138--143. doi:10.1145/2818187.2818273.
- [17] P. M. Dixit, J. C. A. M. Buijs, W. M. P. van der Aalst, B. F. A. Hompes, J. Buurman, Using Domain Knowledge to Enhance Process Mining Results, in: P. Ceravolo, S. Rinderle-Ma (Eds.), *Data-Driven Process Discovery and Analysis. SIMPDA 2015*, volume 244 of *Lecture Notes in Business Information Processing*, Springer, 2017, pp. 76--104. doi:10.1007/978-3-319-53435-0_4.
- [18] K. Okoye, S. Islam, U. Naeem, M. S. Sharif, M. A. Azam, A. Karami, The Application of a Semantic-Based Process Mining Framework on a Learning Process Domain, in: K. Arai, S. Kapoor, R. Bhatia (Eds.), *Intelligent Systems and Applications. IntelliSys 2018*, volume

868 of *Advances in Intelligent Systems and Computing*, Springer, 2019, pp. 1381–1403. doi:10.1007/978-3-030-01054-6_96.

- [19] K. Okoye, Semantic-based process mining: A conceptual model analysis and framework, in: A. Abraham, M. Panda, S. Pradhan, L. Garcia-Hernandez, K. Ma (Eds.), *Innovations in Bio-Inspired Computing and Applications. IBICA 2019*, volume 1180 of *Advances in Intelligent Systems and Computing*, Springer, 2021, pp. 188–200. doi:10.1007/978-3-030-49339-4_20.
- [20] A. Khan, A. Huda, A. Ghose, H. K. Dam, Towards Knowledge-Centric Process Mining, arXiv, 2023. doi:10.48550/ARXIV.2301.10927.
- [21] G. Adamo, C. Di Francescomarino, C. Ghidini, F. M. Maggi, Beyond arrows in process models: A user study on activity dependences and their rationales, *Information Systems* 100 (2021) 101762. doi:10.1016/j.is.2021.101762.
- [22] V. K. Vaishnavi, W. Kuechler, *Design Science Research Methods and Patterns: Innovating Information and Communication Technology*, 2nd ed., CRC Press, 2015. doi:10.1201/b18448.
- [23] N. F. Noy, D. L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*, Technical Report, Knowledge Systems Laboratory, 2001. URL: http://protege.stanford.edu/publications/ontology_development/ontology101.pdf.
- [24] M. Grüninger, K. Atefi, M. S. Fox, Ontologies to Support Process Integration in Enterprise Engineering, *Computational Mathematical Organization Theory* 6 (2000) 381–394. doi:10.1023/A:1009610430261.
- [25] T. C. Jepsen, Just what is an ontology, anyway?, *IT Professional* 11 (2009) 22–27. doi:10.1109/MITP.2009.105.
- [26] M. O’Connor, M. Musen, A. Das, Using the semantic web rule language in the development of ontology-driven applications, in: A. Giurca, D. Gasevic, K. Taveter (Eds.), *Handbook of research on emerging rule-based languages and technologies: open solutions and approaches*, IGI Global, 2009, pp. 525–539. doi:10.4018/978-1-60566-402-6.ch022.
- [27] B. Motik, P. F. Patel-Schneider, B. Parsia, C. Bock, A. Fokoue, P. Haase, R. Hoekstra, I. Horrocks, A. Ruttenberg, U. Sattler, M. Smith, *OWL 2 Web Ontology Language - Structural Specification and Functional-Style Syntax*, Technical Report September, W3C, 2012. URL: <http://www.w3.org/2007/OWL/draft/ED-owl2-syntax-20090611/>.
- [28] M. O’Connor, Cellfie, GitHub, 2022. URL: <https://github.com/protegeproject/cellfie-plugin>.
- [29] M. O’Connor, Mappingmasterdsl, GitHub, 2020. URL: <https://github.com/protegeproject/mapping-master/wiki/MappingMasterDSL>.
- [30] C. Natschläger, Towards a bpmn 2.0 ontology, in: R. Dijkman, J. Hofstetter, J. Koehler (Eds.), *Business Process Model and Notation*, volume 95 of *Lecture Notes in Business Information Processing*, Springer, 2011, pp. 1–15. doi:10.1007/978-3-642-25160-3_1.