

On the Selection of Process Mining Tools

Panagiotis Drakoulogkonas * and Dimitris Apostolou

Department of Informatics, School of Information and Communication Technologies, University of Piraeus, 80, M. Karaoli & A. Dimitriou St., 18534 Piraeus, Greece; dapost@unipi.gr

* Correspondence: drakoulogkonas@unipi.gr

Abstract: Process mining is a research discipline that applies data analysis and computational intelligence techniques to extract knowledge from event logs of information systems. It aims to provide new means to discover, monitor, and improve processes. Process mining has gained particular attention over recent years and new process mining software tools, both academic and commercial, have been developed. This paper provides a survey of process mining software tools. It identifies and describes criteria that can be useful for comparing the tools. Furthermore, it introduces a multi-criteria methodology that can be used for the comparative analysis of process mining software tools. The methodology is based on three methods, namely ontology, decision tree, and Analytic Hierarchy Process (AHP), that can be used to help users decide which software tool best suits their needs.

Keywords: process mining; software tools; comparative analysis methodology; comparison criteria; ontology; decision tree; analytic hierarchy process (AHP)

Citation: Drakoulogkonas, P.; Apostolou, D. On the Selection of Process Mining Tools. *Electronics* **2021**, *10*, 451. <https://doi.org/10.3390/electronics10040451>

Academic Editor: Juan M. Corchado
Received: 25 December 2020
Accepted: 2 February 2021
Published: 11 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Today, many enterprise information systems store events generated during the system operation in structured logs. For example, Enterprise Resource Planning (ERP) systems log various transactions, e.g., users changing documents, filling out forms, etc. Customer Relationship Management (CRM) systems log many interactions with customers. Business-to-business (B2B) systems log exchange of messages with other parties. Workflow Management Systems (WfMSs) typically log the start and completion of activities [1].

System-generated event logs are typically the units of analyses of process mining. Process mining includes process discovery, i.e., extracting process models from event logs, conformance checking, i.e., monitoring deviations by comparing log and model, model repair, model extension, construction of simulation models, social network/organizational mining, case prediction, and history-based recommendations [2].

Researchers investigated and developed new process mining algorithms, several case studies proved their value in a number of sectors, and new process mining software tools, both academic and commercial, arose. Several works have surveyed process mining software tools. Agarwal and Singh [3] made a comparative analysis of process mining tools. Dakic, Sladojevic, Lolic, and Stefanovic [4] presented a comparison of two process mining tools. Claes and Poels [5] performed another survey of software tools. Turner, Tiwari, Olaiya, and Xu [6] presented a comparison of process mining tools. They provided an analysis of main techniques developed by academia and commercial entities and an outline of the practice of business process mining. Celik and Akçetin [7] compared process mining tools. Additional references on software tools can be found in the works of da Silva [8], van der Aalst [9], Van Dongen, de Medeiros, Verbeek, Weijters, and Van Der Aalst [10], and Van Der Aalst et al. [2].

Although many comparisons of process mining tools are available, there exists no rigorous methodology that can be used by practitioners to analyze available tools ac-

ording to their application needs and ultimately select the most appropriate tool. Our work is motivated by the apparent lack of a rigorous methodology and addresses the following questions:

Question 1: Which criteria could be used for the comparative analysis of process mining software tools?

Question 2: How can practitioners select process mining software according to their needs?

This paper provides an up-to-date list of process mining tools and identifies and describes criteria that can be used for the comparison of tools. Furthermore, it proposes a comparative, multi-criteria analysis methodology. To illustrate the methodology, it performs a comparative analysis of five prominent process mining software tools, namely Apromore Community Edition, Celonis, Disco, myInvenio, and ProM.

Section 2 illustrates prominent process mining perspectives, types, and tools. Section 3 describes some comparative analysis criteria that can be used for the comparison of process mining software tools. Section 4 introduces a new comparative analysis methodology that can be used for the comparison of any number of process mining software tools using any number of comparative analysis criteria. Section 5 describes ontology-based selection of software tools. Section 6 illustrates how the software tool(s) can be selected using a decision tree. Section 7 describes the selection of software tool(s) using the Analytic Hierarchy Process (AHP). Section 8 makes a comparative analysis of the five process mining software tools mentioned above, using the new comparative analysis methodology proposed in this paper. Section 9 discusses the findings of our analysis. Section 10 concludes the paper.

2. Process Mining

Process mining aims to exploit event data in a meaningful way, e.g., to improve processes, provide insights, recommend actions, find bottlenecks, record policy violations, and prevent problems. Process mining techniques can extract knowledge from event logs of information systems. They assume that events can be recorded sequentially, such that each event refers to an activity and is related to a specific case. Process mining techniques can use additional information stored in the event logs such as the resource (person or device) initiating or executing the event, the timestamp of the event, and/or data elements recorded with the event [2]. The advancement of technologies and the use of the Internet of Things (IoT) for the collection and transmission of data resulted in large volumes of data and an increasing variety of data types [11]. Process mining can adapt to the nature of high-variate data and extract knowledge [12].

2.1. Process Mining Perspectives

Process mining can cover different perspectives. The control-flow perspective is concerned with the ordering of activities. The aim is to find a characterization of all possible paths. Typically, the result is expressed in the form of a process notation, e.g., Petri net, Event-driven Process Chain (EPC), Business Process Model and Notation (BPMN), and Unified Modeling Language (UML) activity diagrams.

The organizational perspective focuses on information about resources, i.e., the actors (e.g., people, departments, roles, and systems) involved and their relation. The aim is to either display the social network or to structure the organization by classifying people according to their roles and organizational units.

The case perspective is concerned with the properties of the cases. A case can be characterized by its path in the process, by its actors, or by the values of the corresponding data elements. The time perspective focuses on the timing and frequency of events. When the events have timestamps, it is possible to find bottlenecks, monitor the utilization of resources, predict the remaining processing time of running cases, and measure service levels [2].

2.2. Process Mining Types

Event logs can be used to perform three types of process mining: discovery, conformance, and enhancement (Figure 1) [2].

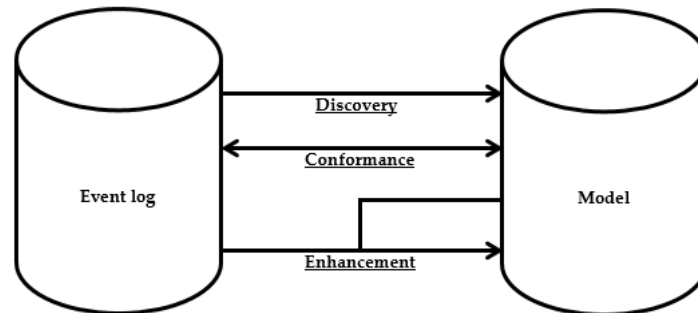


Figure 1. Process mining types.

2.2.1. Discovery

Discovery entails taking an event log and producing a model without using any other a priori information. The discovered model is, typically, a process model such as a Petri net, EPC, BPMN, or UML activity diagram. Besides, discovery can also describe other perspectives such as a social network [2]. An example of a discovery technique is the α -algorithm [13,14]. Using an event log, the α -algorithm can construct a Petri net explaining the behavior stored in the log. For example, given an event log containing enough example executions of a process, the α -algorithm can automatically produce a Petri net, without using additional knowledge. If an event log contains data about resources, discovery algorithms can be used to produce resource-related models, e.g., a social network [13].

2.2.2. Conformance

Conformance-checking techniques use, as input, both an event log and a model. The output is composed of diagnostic information that demonstrates commonalities and differences between the event log and the model. Conformance checking can be used to show if the reality, as registered in the event log, conforms to the model and vice versa [2]. In particular, conformance checking can be useful in order to locate, detect, and explain deviations and to evaluate the severity of the deviations detected [13]. An example is a conformance-checking algorithm, which is described by Rozinat and van der Aalst [15]. Taking an event log and the corresponding model as input, this algorithm can diagnose and quantify deviations [13].

2.2.3. Enhancement

Model enhancement techniques use, as input, both an event log and a model. The output is an extended or improved model. Enhancement uses information about processes, as registered in the event log, to improve or extend the existing process model [2]. One type of enhancement is repair, i.e., to modify the process model in order to reflect reality in a better way. If, for example, two activities are shown in a model to happen sequentially, while in reality they may happen in any order, then the process model can be modified to show this. Another type of enhancement is extension, i.e., to cross-correlate the event log with the process model in order to add new perspectives to the process model. For example, a process model can be extended with performance data. Using timestamps in events, a process model can be extended to show, for example, bottlenecks, frequencies, throughput times, information about resources, quality metrics, service levels, and decision rules [13].

2.3. Process Mining Tools

Table 1 outlines software tools that can be used for the execution of operations related to process mining. The descriptions in the following table are mainly based on information provided on the websites of the tools.

Table 1. Software tools that can be used for the execution of operations related to process mining.

Name	Description
ABBYT Timeline	ABBYT Timeline is a process intelligence platform that provides process mining technology and advanced tasks. Some of its features are discovery, monitoring, analysis and prediction of process behavior, etc. (https://www.abby.com/timeline/ accessed on 5 February 2021).
Apromore	Apromore is described in Section 8.1. (https://apromore.org/ accessed on 5 February 2021).
ARIS Process Mining	Some of the features of ARIS Process Mining are discovery, process analysis, visualization, process improvement, use of one integrated process lifecycle tool, etc. (https://www.softwareag.com/en_corporate/platform/aris/process-mining.html accessed on 5 February 2021).
Celonis	Celonis is described in Section 8.1. (https://www.celonis.com/solutions/ accessed on 5 February 2021).
CoBeFra	CoBeFra is a comprehensive benchmarking suite that can be used in order to set up large-scale conformance-checking experiments. (http://processmining.be/cobefra/ accessed on 5 February 2021).
Dbminer	Dbminer is a tool that can be used for the mining of Petri nets from a behavior described as the union of several transition systems. This tool is based on the theory of (generalized) regions. (http://people.ac.upc.edu/msole/homepage/dbminer.html accessed on 5 February 2021).
Disco	Disco is described in Section 8.1. (https://fluxicon.com/disco/ accessed on 5 February 2021).
EverFlow	Some of the features of EverFlow are discovery, monitoring of process executions, visual identification of bottlenecks, inconsistencies and inefficiencies, embedded analytics, process activities, and teams optimization, etc. (https://www.icarotech.com/en/everflowen/ accessed on 5 February 2021).
Explora Process	Explora Process can be used for the identification of process inefficiencies affecting process costs, quality, time, and risk. It can provide tools that may improve efficiency and productivity and solve issues. Moreover, it can verify the compliance of company processes with current legislation and internal procedures and it can identify where discrepancies occur, etc. (https://www.integris.it/explora-en/#explora-process accessed on 5 February 2021).

LANA Process Mining	Some of the features of LANA Process Mining are process visualization, conformance checking, monitoring, root cause analysis that allows the automatic identification of the root causes of process problems, etc. (https://lanalabs.com/en/lana-process-mining/ accessed on 5 February 2021).
Logpickr Process Explorer 360	Some of the features of Logpickr Process Explorer 360 are highlights on business processes, conformance, root cause analysis, end-to-end view, prediction, analysis, etc. (https://www.logpickr.com/en/product.html accessed on 5 February 2021).
MEHRWERK Process-Mining (MPM)	Some of the features of MEHRWERK ProcessMining (MPM) are visualization, conformance checking, root cause analysis, action and workflow management, process monitoring, and process prediction, etc. (https://mpm-processmining.com/en/mpm-processmining-digital-transformation-operational-excellence/ accessed on 5 February 2021, https://mpm-processmining.com/en/augmented-process-mining/ accessed on 5 February 2021).
Minit	Some of the features of Minit are root cause analysis, check process compliance, process simulation, use of custom metrics, hierarchical process mining, identification of repeating activities, process comparison, etc. (https://www.minit.io/software accessed on 5 February 2021).
MonkeyMiner	Some of the features of MonkeyMiner are collection, filtering, sorting, merging, and anonymization of data, discovery of process flows, discovery of new patterns and bottlenecks, reporting, etc. (https://www.monkeymining.com/home-english/ accessed on 5 February 2021).
myInvenio	myInvenio is described in Section 8.1. (https://www.my-invenio.com/ accessed on 5 February 2021).
PAFnow	Some of the features of PAFnow are visualization, process analysis, finding of root causes of abnormalities automatically, receiving notifications or automatically starting workflows in applications, etc. (https://pafnow.com/product/ accessed on 5 February 2021).
Process Diamond Intelligence	Some of the features of Process Diamond Intelligence are process discovery, compliance checking, finding differences in the way process instances are executed, rule specifications, dashboard, finding bottlenecks, simulation, predictive monitoring, analyses, filtering, etc. (https://processdiamond.com/product/ accessed on 5 February 2021).
ProM	ProM is described in Section 8.1. (http://www.promtools.org/doku.php accessed on 5 February 2021).

ProDiscovery	Some of the features of ProDiscovery are process map and regeneration, analysis of process pattern, various analyses, filtering and dashboard, etc. (https://www.puzzledata.com/process-mining_eng/ accessed on 5 February 2021, https://www.puzzledata.com/prodiscovery_eng/ accessed on 5 February 2021).
QPR ProcessAnalyzer	Some of the features of QPR ProcessAnalyzer are discovery, investigation of root causes, conformance, and long cases, building of dynamic dashboards, actions based on apps and business alerts, reporting, visualization, auditing, and compliance, etc. (http://www.qpr.com/products/qpr-processanalyzer accessed on 5 February 2021).
Scheer Process Mining	Some of the features of Scheer Process Mining are discovery, conformance, enhancement, etc. (https://www.scheer-group.com/en/process-mining/ accessed on 5 February 2021).
Signavio Process Intelligence	Some of the features of Signavio Process Intelligence are process discovery, conformance checking, task mining, process overview, etc. (https://www.signavio.com/products/process-intelligence/ accessed on 5 February 2021).
Skan	Some of the features of Skan are discovery, simulation, predictions, conformance, analysis, improvement, simulation, etc. (https://skan.ai/ accessed on 5 February 2021).
StereoLOGIC Discovery Analyst	Some of the features of StereoLOGIC Discovery Analyst are automatic extraction of business processes from business applications in real time, visualization, comparison capabilities, validation and extension of models, integration with other BPMN-enabled tools, etc. (https://www.iag.biz/services/rdm-alm-software/stereologic-discovery/ accessed on 5 February 2021).
UiPath Process Mining	Some of the features of UiPath Process Mining are visualization, animation, finding of bottlenecks, recommendations, monitoring, analysis, alerting, etc. (https://www.uipath.com/product/process-mining accessed on 5 February 2021).
Worksoft Analyze	Some of the features of Worksoft Analyze are discovery, visualization, analytics, identification of risks and inefficiencies, etc. (https://www.worksoft.com/products/worksoft-analyze accessed on 5 February 2021).
XMAnalyzer	Some of the features of XMAnalyzer are insight into the current operating business processes, ability to analyze sequence flow of processes based on transactions, events, or activities, graphical illustration of all process paths in one diagram, with the ability to see individual process paths, etc. (http://xmpro.com/xmpro-releases-new-ibos-process-mining-module-xmanalyzer/ accessed on 5 February 2021).

3. Comparative Analysis Criteria

In this section, we classify and describe criteria that can be used for the comparative analysis of any number of software tools. We selected criteria that cover a wide range of features and can help stakeholders to distinguish the process mining software tool that best suits their needs. We classified the criteria into four categories. In this way, it can be easier for users to find the criteria they want. The four categories are as follows (Figure 2):

1. General. Includes criteria that provide general information about the software tools. In the “general” category, we classify the criteria that cannot be classified in any of the other three categories.
2. Process Mining Types. Contains the three process mining types that were described in Section 2.2. These types are of great importance in the field of process mining.
3. Operational Support Activities. Includes the activities used for online operational support of running cases [13].
4. Discovery Problems Addressed. Contains criteria that can be used to check if the software tools can address specific discovery problems.

In Table 2, we describe each of the comparative analysis criteria.

Table 2. Comparative analysis criteria.

Criterion	Description
License	Type of license of the software tool.
Filtering	Check if the software tool can provide data filtering [16,17].
Process Animation	Check if the software tool can provide process animation [9,18,19].
Browser-based	Check if the software tool can run in a browser.
No Installation Required	Check if no local installation is required in order to use the software tool.
Social Network Mining	Check if the software tool can use the information recorded in the event log about the users that execute the activities in order to perform social network mining [20].
Statistics	Check if the software tool can provide statistics.
No Registration Required	Check if no registration is required in order to use the software tool without restrictions.
Delta Analysis	Check if the software tool supports delta analysis. Delta analysis compares the reference model with the generated model in order to provide answers to problems related to business alignment [7].
Algorithm(s)	Supported algorithm(s) [10,13].
Import Type(s)	Supported import type(s) (e.g., csv, xls, xes) [13,18].
Output Model(s)	Supported output model(s) notation (e.g., Petri nets, BPMN, Transition Systems, Fuzzy Model) [2,10,21].
Discovery	Check if the software tool can provide discovery. Discovery is described in Section 2.2.1.
Conformance	Check if the software tool can provide conformance. Conformance is described in Section 2.2.2.
Enhancement	Check if the software tool can provide enhancement. Enhancement is described in Section 2.2.3.
Detection	Check if the software tool can detect deviations at runtime. In detection, a model is compared with a partial trace, and if a violation is detected, then an alert can be generated [13].

Prediction	Check if prediction is supported. In prediction, the current case is compared to similar cases that occurred in the past. Based on this information, predictions about the events that will follow can be made [13].
Recommendation	Check if the software tool supports recommendation. In recommendation, based on historic information, recommendations about the selection of the next activity can be made [13].
Noise	Check if the software tool can deal with noise. Noisy, i.e., infrequent/exceptional, behavior should not be displayed in the discovered model. Stakeholders are typically interested about the main behavior. Furthermore, it is difficult to extract meaningful information by very rare activities or patterns [13].
Concurrent Processes	Check if the software tool has the ability to discover and represent a model that contains concurrent processes.
Duplicate Tasks	Check if the software tool can address the “Duplicate Tasks” problem. “Duplicate tasks” refers to situations where multiple tasks in a process have the same label. In situations such as this, algorithms may need extra effort to find out which log events belong to which transition [22].
Mining Loops	Check if the software tool can accurately discover a model that contains loops [13].

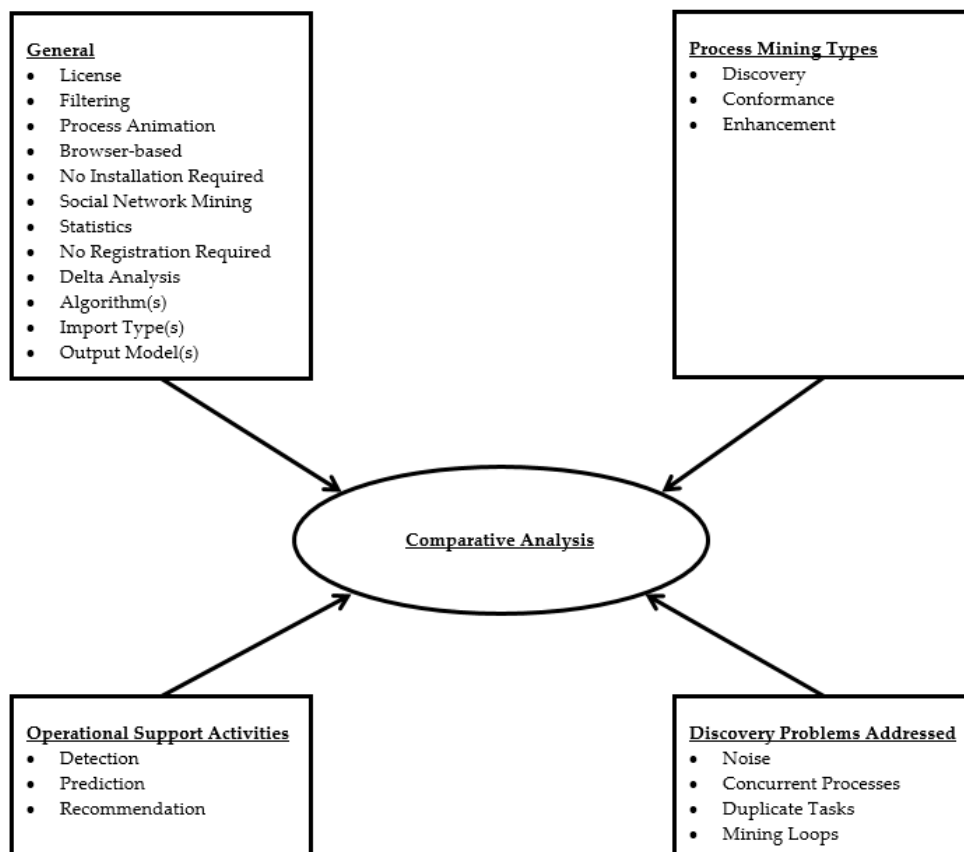


Figure 2. Comparative analysis criteria.

4. Comparative Analysis Methodology

In this section, we outline a new methodology that can be used for the comparative analysis of any number of process mining software tools using any number of criteria. The methodology is composed of four phases (Figure 3):

- Phase 1: Listing of Process Mining Software Tools to Be Compared. The aim of Phase 1 is to list the process mining software tools that we want to compare. For example, we could list some of the software tools mentioned in Section 2.3.
- Phase 2: Listing of Comparative Analysis Criteria. The aim of Phase 2 is to list the criteria that we want to use for the comparative analysis of the process mining software tools listed in Phase 1. For example, we could list some of the comparative analysis criteria described in Section 3.
- Phase 3: Listing of Comparative Analysis Criteria Values per Process Mining Software Tool. The aim of Phase 3 is to list the values of each of the comparative analysis criteria listed in Phase 2 per process mining software tool listed in Phase 1. We create a double-entry table. In the header row of the table, we enter the names of the process mining software tools listed in Phase 1. In the header column of the table, we enter the comparative analysis criteria listed in Phase 2. In the remaining table cells, we enter the comparative analysis criteria values per process mining software tool. An example can be seen in Section 8.3.
- Phase 4: Selection of Software Tool(s). The aim of Phase 4 is the selection of the process mining software tool that best suits user needs. Following the completion of Phase 3, one or more of the following three methods can be used for the selection of the software tool. The methods and the reasons for selecting each method for the comparative analysis of the software tools are illustrated in Table 3.
 - Ontology-based selection. The aim of this method is to select the software tool that best suits user needs from the list of the process mining software tools listed in Phase 1 by using an ontology, the comparative analysis criteria listed in Phase 2, and the values listed in Phase 3.
 - Selection of Software Tool(s) Using Decision Tree. The aim of this method is to select the software tool that best suits user needs from the list of the process mining software tool(s) listed in Phase 1 by using a decision tree, the comparative analysis criteria listed in Phase 2, and the values listed in Phase 3.
 - Selection of Software Tool(s) Using AHP. The aim of this method is to select the software tool that best suits user needs from the list of the process mining software tool(s) listed in Phase 1 using AHP, the comparative analysis criteria listed in Phase 2, and the values listed in Phase 3.

The four phases of the comparative analysis methodology are described in more detail in the following sections.

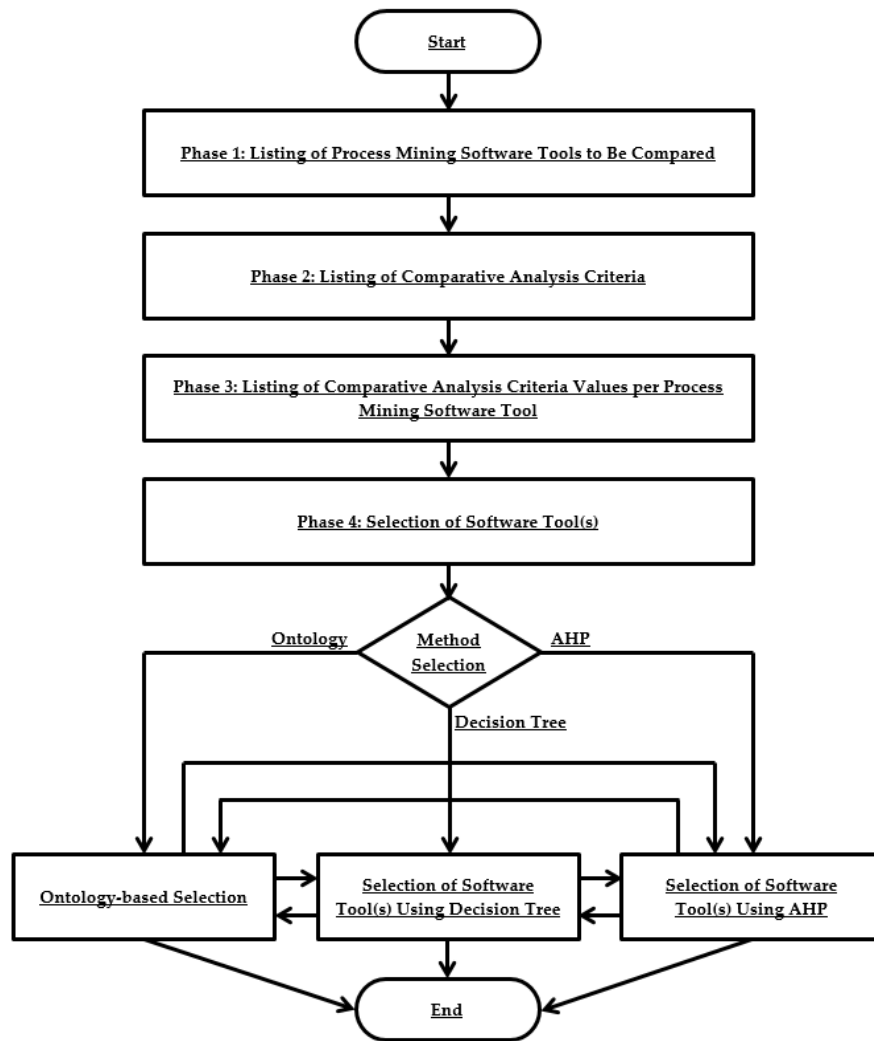


Figure 3. Comparative analysis methodology.

Table 3. Methods and reasons for selecting each method for the comparison of the software tools.

Method.	Reasons for Selecting the Method for the Comparative Analysis of the Software tools
Ontology	This method supports the creation of an ontology containing the software tools, the comparative analysis criteria, and their values for each of the process mining software tools to be compared. The created ontology can then be inserted into a tool such as Protégé. In this way, the methodology supports the execution of complex queries in order to find the software tool that is most suitable for the stakeholders. For example, the execution of a query searching for the software tool(s) that provide discovery, conformance, filtering, and simulation is supported. Ontology and Protégé are described in Section 5.
Decision Tree	This method supports the creation of a decision tree using an algorithm such as C4.5 and the Weka Workbench. In this way, the break-down of a complex decision-making process into a number of simpler decisions is supported, providing a solution which can be easier to interpret [23]. Furthermore, the proposed methodology allows stakeholders to see in a tree-like model which software tool is most suitable for them, depending on the values of the criteria. Decision tree and the C4.5 algorithm are described in Section 6.
AHP	This method supports the decomposition of the decision problem into a hierarchy of more easily understood sub-problems, each of which can then be analyzed independently, using AHP. After the hierarchy is built, the various elements can be evaluated by pairwise comparing them concerning their impact on an element that exists above them in the hierarchy. For the comparisons, the judgments of the stakeholders about the relative meaning and importance of the elements can be used. Therefore, in AHP, human judgments, and not just the underlying information, can be used to perform the evaluations. The AHP converts the evaluations to numerical values. The numerical values can then be processed and compared over the entire range of the decision problem. A numerical weight or priority is generated for all of the elements in the hierarchy, allowing them to be compared to one another consistently and rationally. Afterwards, numerical priorities are calculated for all of the decision alternatives. The numerical priorities indicate the relative ability of the alternatives to achieve the goal of the decision [24] and allow users to select the software tool that is most suitable for them. AHP is described in Section 7.

5. Ontology-Based Selection

In this method, we create an ontology containing all the process mining software tools listed in Phase 1, all the criteria listed in Phase 2, and all the values listed in the table created in Phase 3.

Ontologies were developed in Artificial Intelligence (AI) in order to facilitate the reuse and sharing of knowledge. The notion of ontology is popular in fields such as knowledge management, information retrieval, intelligent information integration, electronic commerce, and cooperative information systems. Ontologies aim to provide a shared and common understanding of domains, which can be communicated between people and application systems [25].

For the creation of the ontology, we may use Protégé. Protégé is an open-source tool that can be used to assist users in the construction of electronic knowledge bases. Its user

interface can be used for the creation and editing of domain ontologies that represent the concepts and relationships of application areas. Several plugins allow the management of multiple ontologies, enable the use of engines and problem solvers with Protégé ontologies, and provide alternative mechanisms of visualization and other functions. Protégé is based in Java and can run under a variety of operating systems. It can assist users to construct large electronic knowledge bases. Using the ontology, the system automatically constructs a graphical knowledge-acquisition system, which allows users to enter the content knowledge required for the applications [26] (<https://protege.stanford.edu/> accessed on 5 February 2021).

Different facilities can be provided by different ontology languages. Web Ontology Language (OWL), from the World Wide Web Consortium (W3C), is a development in standard ontology languages. OWL ontologies have components similar to the components of the Protégé-based ontologies. However, the terminology used to describe the OWL components is slightly different from the terminology used in Protégé. OWL ontologies can consist of Individuals, Properties, and Classes, which correspond to Protégé Instances, Slots, and Classes [27].

According to the new comparative analysis methodology proposed in this paper, we can follow the steps listed below in order to implement and use an ontology for the selection of suitable process mining software tool(s):

1. Determine the purpose of the ontology. In our case, the purpose of the ontology is the selection of the process mining software tool that best suits stakeholders' needs by comparing any number of tools and using any number of comparative analysis criteria.
2. List important terms in the ontology. Some important terms of our ontology are the software tools, the comparative analysis criteria, and their values.
3. Define the classes and their hierarchy. The terms listed in Step 2 can be used to define the classes of the ontology. We create a class for the software tools and another class for the comparative analysis criteria. We then develop the hierarchy of the classes:
 - We create a class for each one of the software tools that we want to compare (e.g., we create "Disco", "ProM", etc., classes). We define these classes as subclasses of the software tools class.
 - We create a class for each one of the comparative analysis criteria that we want to use for the comparison of the software tools (e.g., we create "Discovery", "Conformance", "Filtering", "Statistics", etc., classes). We define these classes as subclasses of the comparative analysis criteria class.
 - We create a class for each one of the values of each one of the comparative analysis criteria (e.g., we create "Yes" and "No" classes for the "Filtering" criterion, etc.). We define these classes as subclasses of the respective comparative analysis criterion classes (e.g., we define "Yes" and "No" as subclasses of the "Filtering" criterion class, etc.).
4. Define the properties. In this step, we define the properties of classes (e.g., "Provides_Discovery", "Provides_Conformance", "Provides_Filtering", "Provides_Statistics", etc.).
5. Assign values to all the properties of all the software tools. In this step, we assign values to all the properties defined in Step 4 of all the software tool subclasses defined in Step 3 (e.g., we assign the value "Yes" to the "Provides_Filtering" property of the "ProM" subclass of the software tools class, etc.).
6. Execute queries. If we create an ontology as described above and use a tool such as Protégé, we will be able to execute complex queries in order to find the suitable process mining software tool(s) (e.g., we could execute a query searching for browser-based open-source software tool(s) that provide discovery, conformance, filtering, and statistics, etc.).

In Section 8.4.1, we provide an example of an ontology-based selection of process mining software tool(s), using Protégé.

6. Selection of Software Tool(s) Using Decision Tree

In this method, we create a decision tree that uses all the process mining software tools listed in Phase 1, all the criteria listed in Phase 2, and all the values listed in the table created in Phase 3.

A decision tree is a tree in which the branch nodes represent choices between several alternatives and the leaf nodes represent decisions. Decision trees are commonly used to gain decision-making information. Starting from a root node, users can split each one of the nodes recursively, according to the decision tree learning algorithm. The result is a decision tree, where each branch illustrates a possible decision scenario and its outcome.

To classify instances, decision trees traverse from the root node to the leaf node. They start from the root node, test the attribute of this node, and then move down the tree branch, depending on the value of the attribute in the given set. The same process is then repeated at the sub-tree level [28].

According to the new comparative analysis methodology proposed in this paper, we can follow the steps listed below to implement a decision tree for the selection of suitable process mining software tool(s):

1. Determine the purpose of the decision tree. We define a relation describing the purpose. In our case, the purpose of the decision tree is to select the software tool that best suits stakeholders' needs by comparing any number of tools and using any number of comparative analysis criteria.
2. Define the attributes. We define one attribute for each one of the comparative analysis criteria. Each attribute includes the name of the respective criterion and all its possible values. For example, we can define an attribute "Filtering {Yes, No}", another attribute "License {Open_Source, Evaluation_Academic_Commercial}", etc. Furthermore, the last attribute that we define describes the result. The possible values of the result attribute are all the software tools that we want to compare. For example, we can define an attribute "Result {Celonis, myInvenio, ProM}". Software tools will be displayed as leaf nodes in the decision tree.
3. Define the data. We define the different combinations of the values of all the attributes (i.e., all the different combinations of the values of all the comparative analysis criteria and the software tools). In this way, we define the resulting software tool for the different combinations of comparative analysis criteria values.

For example, if, in Step 2, we have defined:

- Three attributes to describe comparative analysis criteria (e.g., "License {Open_Source, Evaluation_Academic_Commercial}", "Filtering {Yes, No}", and "Discovery {Yes, No}");
- The last attribute to describe the resulting software tools (e.g., "Result {Celonis, myInvenio, ProM}"),

then the data could be:

- Evaluation_Academic_Commercial, Yes, Yes, Celonis;
- Evaluation_Academic_Commercial, Yes, Yes, myInvenio;
- Open_Source, Yes, Yes, ProM.

4. Create the decision tree. After the completion of Steps 1, 2, and 3, we can use an algorithm such as C4.5 and a tool such as Weka (see below) to create the decision tree. In the resulting decision tree, the root and the internal nodes represent comparative analysis criteria, the lines represent different values of the criteria, and the leaf nodes represent software tools. Using the resulting decision tree, stakeholders will be able to easily see, in a tree-like model, the software tool that best suits their needs, depending on the values of the selected criteria.

For the creation of the decision tree, we can use the C4.5 algorithm. C4.5 is a statistical classifier, because it can generate decision trees that can be used for classification. It is possible to accept data with numerical or categorical values and it uses information gain as the splitting criterion. For the handling of continuous values, it generates a threshold. Then, it divides attributes with values below or equal to the threshold and values above the threshold. Missing values can easily be handled by the C4.5 algorithm, as it does not utilize missing attribute values in gain calculations [29]. To create the decision tree for the selection of the software tool, we can use the Weka open-source tool (<https://www.cs.waikato.ac.nz/ml/weka/> accessed on 5 February 2021).

In Section 8.4.2, we provide an example of selection of software tool(s) using a decision tree, the C4.5 algorithm, and Weka.

7. Selection of Software Tool(s) Using AHP

In this method, we use AHP [24,30,31], all the process mining software tools listed in Phase 1, all the criteria listed in Phase 2, and all the values listed in the table created in Phase 3. The AHP method is implemented in four steps: (a) the hierarchical analysis of the decision problem into decision elements, (b) the collection of preferences from the decision maker regarding the decision elements, (c) the calculation of individual priorities for the elements, and (d) the synthesis of the individual priorities into general priorities of the alternatives. The first two steps are carried out with the participation of the decision maker while the last two are purely computational.

1. Hierarchical analysis of the decision problem: In the first step, the ultimate goal pursued in the decision problem under study is broken down into sub-goals, which are then increasingly analyzed in the patterns of a hierarchical structure. At the top of this hierarchical structure is the ultimate goal, which, in our case, is the selection of the software tool that best suits our needs. The criteria are the comparative analysis criteria (e.g., discovery, conformance, filtering, statistics, etc.) that we want to use for the comparative analysis of the software tools. The alternatives are the leaves of the tree, which, in our case, are the software tools that we want to compare (e.g., Celonis, Disco, ProM, etc.).
2. Collection of preferences: At each level of the hierarchical structure, its elements, i.e., the criteria, are compared in pairs in terms of the degree of preference of one over the other in relation to the criterion of the immediately higher level, i.e., the parent element. This creates an array of pairs of comparisons, the number of which is the same as the number of nodes in the tree, excluding the leaves (alternatives). Therefore, in this step, we make pairwise comparisons of the comparative analysis criteria (e.g., discovery with conformance, then discovery with filtering, then discovery with statistics, etc.) concerning their importance in reaching the goal to select the software tool(s). The consistency of the collected preferences is evaluated with the Consistency Ratio (CR) [30].
3. Calculation of individual priorities: In the third step, which is purely computational, the relative priorities (weights) of the comparable decision elements are calculated for each comparison table in relation to the parent element. Hence, in this step, we pairwise compare the software tools (e.g., Celonis with Disco, then Celonis with ProM, etc.) with respect to their importance for each criterion separately (e.g., discovery, etc.).
4. Synthesis of the individual priorities: In the last step, which is also purely computational, the local weights of the data are synthesized, as they emerge from the individual comparison tables, into general priorities of the alternatives (leaves of the tree structure) with respect to the ultimate goal (root). Weight synthesis is performed with multiplication between bottom-up weight tables, that is, from the lowest to the highest hierarchical level. Thus, in this step, we find the software tool(s) having the highest overall priority.

In Section 8.4.3, we provide an example of applying AHP using the AHP Online System—Business Performance Management Singapore (BPMSG) [32,33].

8. Example

In this section, we use the proposed comparative analysis methodology to analyze five process mining software tools using eleven criteria. Our goal is to find the software tool that is more suitable for mining supply chain processes of a small/medium-sized enterprise (SME). In this case, supply chain processes represent the steps required to get the product from its original state to the customer. These steps include the procurement of raw materials and components as well as transportation and distribution of the products to the customers. Entities involved in the supply chain are producers, vendors, retailers, distribution centers, warehouses, and transportation companies.

8.1. Phase 1: Listing of Process Mining Software Tools to Be Compared

In our example, we compare the following process mining software tools:

- **Apromore Community Edition:** Apromore is an open-source collaborative business process analytics platform. Some of the advantages of Apromore are that it (i) has an easily extensible framework, where new plugins can be added to a system of advanced business process analytics capabilities [34]; (ii) provides a shared workspace of logs and models; (iii) includes a multi-log animation and flow comparison (<https://apromore.org> accessed on 5 February 2021).
- **Celonis:** Some of the advantages of Celonis are its (i) AI-driven learning, i.e., algorithms can learn from the outcomes of each recommended action in order to improve future recommendations—and, ultimately, execution capacity—over time; (ii) capability to identify process execution gaps and assess which of them have the greatest impact; (iii) capability to automate real-time interventions across systems and recommend next best actions (<https://www.celonis.com/solutions> accessed on 5 February 2021).
- **Disco:** Some of the advantages of Disco are the (i) project view, providing the ability to manage datasets and add notes for each of them; (ii) advanced mapping feature that makes configuration efficient and sorting of data fast; (iii) ability to choose between various process metric visualizations projected on a map (<https://fluxicon.com/disco/> accessed on 5 February 2021).
- **myInvenio:** Some of the advantages of myInvenio are its (i) ability to automatically discover processes from many company data and stakeholders (e.g., CRM, ERP, etc.) by providing end-to-end process streamlining; (ii) ability to identify best performers and critical activities and resources; (iii) ability to identify process improvements and simulate process savings (<https://www.my-invenio.com/> accessed on 5 February 2021).
- **ProM:** It is an open-source framework. Some of the advantages of ProM are that (i) it supports the development of plug-ins [17], which can be used for implementing process mining algorithms [10]; (ii) it supports a wide variety of process mining techniques. ProM is aimed largely at academic and research communities. (<http://www.promtools.org/doku.php> accessed on 5 February 2021).

Additional advantages of the five software tools can be seen in Section 8.3.

8.2. Phase 2: Listing of Comparative Analysis Criteria

In our example, we compare the five process mining software tools listed in Phase 1 in terms of the following criteria (described in Section 3): License, Filtering, Browser-based, Process Animation, No Installation Required, Social Network Mining, Statistics, No Registration Required, Discovery, Conformance, and Enhancement.

8.3. Phase 3: Listing of Comparative Analysis Criteria Values per Process Mining Software Tool

We created the double-entry Table 4. In the header row of the table, we entered the names of the five process mining software tools listed in Phase 1. In the header column of the table, we entered the eleven comparative analysis criteria listed in Phase 2. In the remaining table cells, we entered the comparative analysis criteria values per process mining software tool.

Table 4. Overview of software tools (✓ = yes, X = no).

Comparative Analysis Criteria	Apromore Community Edition	Celonis	Disco	myInvenio	ProM
License	Open Source	Evaluation/Academic/Commercial	Evaluation/Academic/Commercial	Evaluation/Academic/Commercial	Open Source
Filtering	✓	✓	✓	✓	✓
Browser-based	✓	✓	X	✓	X
Process Animation	✓	✓	✓	✓	✓
No Installation Required	X	✓	X	✓	X
Social Network Mining	✓	✓	X	✓	✓
Statistics	✓	✓	✓	✓	✓
No Registration Required	✓	X	X	X	✓
Discovery	✓	✓	✓	✓	✓
Conformance	✓	✓	X	✓	✓
Enhancement	✓	✓	✓	✓	✓

8.4. Phase 4: Selection of Software Tool(s)

According to the proposed methodology, after the completion of Phase 3, one or more of the three methods mentioned above, namely ontology, decision tree, and AHP, can be used for the selection of the suitable software tool. In our example, as we can see in the following lines, we decided to use all three methods.

8.4.1. Ontology-Based Selection

In our example, we used Protégé 5.5.0 for the creation of the ontology. Ontology and Protégé can be useful for the selection of a suitable process mining software tool. In particular, a user can create a class hierarchy, containing information about all the process mining software tools listed in Phase 1, all the criteria listed in Phase 2, and all the values listed in the table created in Phase 3.

A class hierarchy can be created in Protégé by selecting: Tools|Create class hierarchy. In our case, we created the class hierarchy displayed in Figure 4a.

Afterward, a user can create an object property hierarchy, containing an object property for each of the criteria listed in Phase 2. An object property hierarchy can be created in Protégé, by selecting: Tools|Create object property hierarchy. In our case, we created the object property hierarchy illustrated in Figure 4b.

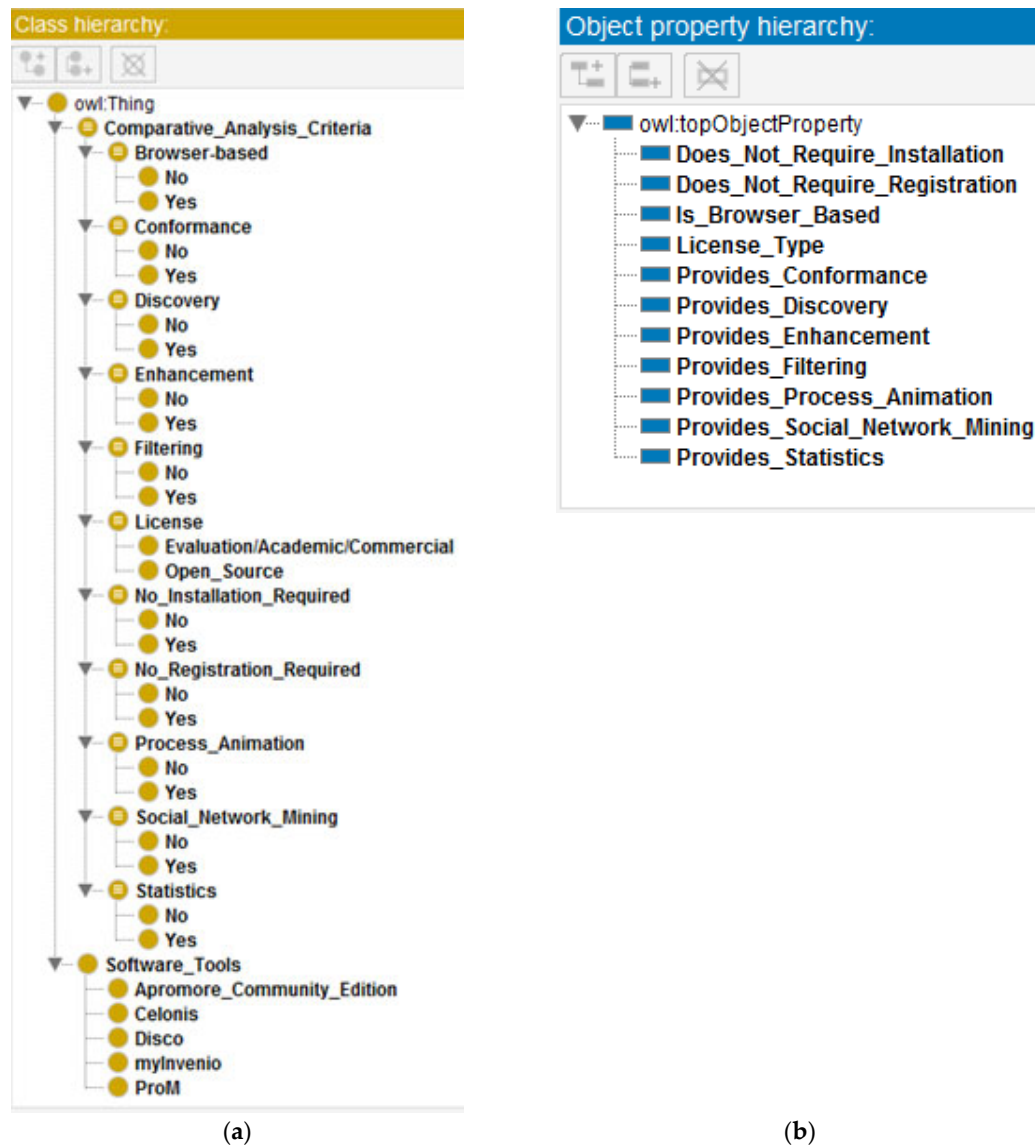


Figure 4. (a) Class hierarchy (Protégé 5.5.0); (b) object property hierarchy (Protégé 5.5.0).

Then, users can set the values of all the criteria listed in Phase 2 for each one of the software tools listed in Phase 1. For example, the description of the class describing ProM in Protégé is illustrated in Figure 5a.

Using ontology and Protégé, users can execute complex queries in order to find the software tool that best suits their needs. For example, in Figure 5b we can see the results of the execution of a query searching for browser-based open-source software tool(s) that provide filtering, process animation, statistics, and discovery. Description Logics (DL) query has been used, and as we can see in the query results, all the aforementioned properties are provided by Apromore Community Edition.

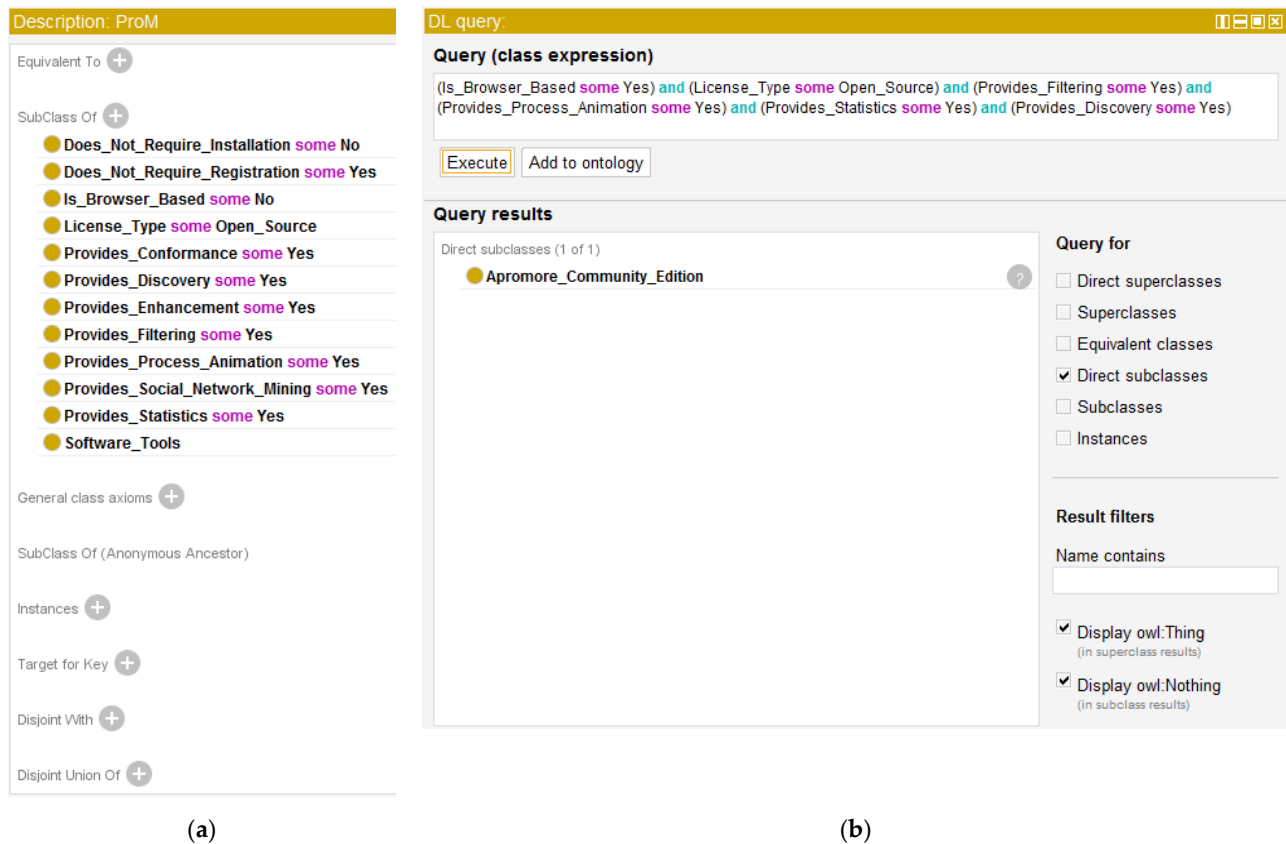


Figure 5. (a) Description of ProM class (Protégé 5.5.0); (b) DL Query (Protégé 5.5.0).

8.4.2. Selection of Software Tool(s) Using Decision Tree

In our example, we created a decision tree using the C4.5 algorithm and Weka 3.8.4. We used the standard Graphical User Interface (GUI) of Weka. In the Weka GUI Chooser window, we selected “Workbench” to open the Weka Workbench window and then we opened the file containing our data (Figure 6).

```

@RELATION Select_Software_Tool{s}

@ATTRIBUTE License {Open_Source, Evaluation_Academic_Commercial}
@ATTRIBUTE Filtering {Yes, No}
@ATTRIBUTE Browser-based {Yes, No}
@ATTRIBUTE Process_Animation {Yes, No}
@ATTRIBUTE No_Installation_Required {Yes, No}
@ATTRIBUTE Social_Network_Mining {Yes, No}
@ATTRIBUTE Statistics {Yes, No}
@ATTRIBUTE No_Registration_Required {Yes, No}
@ATTRIBUTE Discovery {Yes, No}
@ATTRIBUTE Conformance {Yes, No}
@ATTRIBUTE Enhancement {Yes, No}
@ATTRIBUTE Result {Apromore_Community_Edition, Celonis, Disco, myInvenio, ProM}

@DATA
Open_Source, Yes, Yes, Yes, No, Yes, Yes, Yes, Yes, Yes, Yes, Yes, Apromore_Community_Edition
Evaluation_Academic_Commercial, Yes, Yes, Yes, Yes, Yes, Yes, No, Yes, Yes, Yes, Celonis
Evaluation_Academic_Commercial, Yes, No, Yes, No, No, Yes, No, Yes, No, Yes, Disco
Evaluation_Academic_Commercial, Yes, Yes, Yes, Yes, Yes, Yes, No, Yes, Yes, Yes, myInvenio
Open_Source, Yes, No, Yes, No, Yes, Yes, Yes, Yes, Yes, Yes, ProM
    
```

Figure 6. Data used for the generation of the decision tree.

Then, we chose the J48 tree classifier, which can be used for generating decision trees using the C4.5 algorithm.

Afterward, we changed the minNumObj to 1 and the cross-validation folds to 3. Then, we pressed the “Start” button and we selected the “Visualize tree” option. The generated decision tree is illustrated in Figure 7. Decision trees can help stakeholders to easily see, in a tree-like model, the software tool that best suits their needs, depending on the values of selected criteria. For example, using the generated decision tree illustrated in the following figure, we can easily see that ProM is an open-source software tool that is not browser-based.

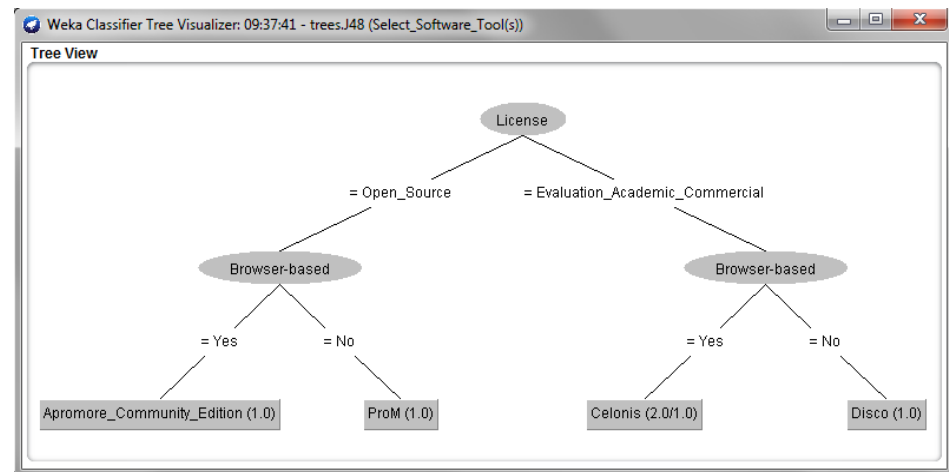


Figure 7. Generated decision tree (Weka 3.8.4).

8.4.3. Selection of Software Tool(s) Using AHP

In our example, we used the AHP Online System—BPMSG [32,33]. We created an AHP hierarchy, consisting of:

- A goal: Select software tool(s).
- Eleven Criteria: License; Filtering; Browser-based; Process Animation; No Installation Required; Social Network Mining; Statistics; No Registration Required; Discovery; Conformance; Enhancement.
- Five Alternatives (the software tools): Aprimore Community Edition; Celonis; Disco; myInvenio; ProM.

Next, we assigned greater important to criteria such as the licensing followed by each tool, since the SME prefers an open-source or economical solution. Then, we compared the alternatives two at a time, with respect to their importance for each of the eleven criteria separately. For the pairwise comparisons, AHP uses a scale ranging from 1 to 9. For example, the pairwise comparisons of all the alternatives with respect to License are illustrated in Figure 8a [32,35].

Then, we checked the CR; in all cases of our example the CR was acceptable. Therefore, we did not have to adjust any of our judgments in order to improve consistency [32,35].

The resulting priorities of the alternatives with respect to License can be seen in Figure 8b [32,35].

The overall priorities and ranking of the five alternatives are displayed in Figure 9a [32].

In Figure 9b [32,33,36], we can see the decision hierarchy that illustrates the derived priorities of the eleven criteria and the five alternatives.

With respect to License, which alternative fits better or is more preferable, and how much more on a scale 1 to 9?

A - wrt License - or B?		Equal	How much more?							
1	<input checked="" type="radio"/> Apromore Community Edition <input type="radio"/> Celonis	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input checked="" type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
2	<input checked="" type="radio"/> Apromore Community Edition <input type="radio"/> Disco	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input checked="" type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
3	<input checked="" type="radio"/> Apromore Community Edition <input type="radio"/> myInvenio	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input checked="" type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
4	<input checked="" type="radio"/> Apromore Community Edition <input type="radio"/> ProM	<input checked="" type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
5	<input checked="" type="radio"/> Celonis <input type="radio"/> Disco	<input checked="" type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
6	<input checked="" type="radio"/> Celonis <input type="radio"/> myInvenio	<input checked="" type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
7	<input type="radio"/> Celonis <input checked="" type="radio"/> ProM	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input checked="" type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
8	<input checked="" type="radio"/> Disco <input type="radio"/> myInvenio	<input checked="" type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
9	<input type="radio"/> Disco <input checked="" type="radio"/> ProM	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input checked="" type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
10	<input type="radio"/> myInvenio <input checked="" type="radio"/> ProM	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6	<input checked="" type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9

CR = 0% OK

Calculate Submit

Resulting Priorities

Cat	Priority	Rank
1	Apromore Community Edition 41.2%	1
2	Celonis 5.9%	3
3	Disco 5.9%	3
4	myInvenio 5.9%	3
5	ProM 41.2%	1

(a)

(b)

Figure 8. (a) Pairwise comparisons of all the alternatives for License [32,35]; (b) resulting priorities of the alternatives [32,35].

Cat	Priority	Rank
1	Apromore Community Edition 22.0%	1
2	Celonis 20.3%	3
3	Disco 15.7%	5
4	myInvenio 20.3%	3
5	ProM 21.7%	2

(a)

Decision Hierarchy							
Level 0	Level 1	Glb Prio.	Apromore Community Edition	Celonis	Disco	myInvenio	ProM
Select software tool(s)	License 0.057	5.7%	0.412	0.059	0.059	0.059	0.412
	Filtering 0.111	11.1%	0.200	0.200	0.200	0.200	0.200
	Browser-based 0.016	1.6%	0.273	0.273	0.091	0.273	0.091
	Process Animation 0.041	4.1%	0.200	0.200	0.200	0.200	0.200
	No Installation Required 0.021	2.1%	0.091	0.364	0.091	0.364	0.091
	Social Network Mining 0.029	2.9%	0.238	0.238	0.048	0.238	0.238
	Statistics 0.079	7.9%	0.200	0.200	0.200	0.200	0.200
	No Registration Required 0.013	1.3%	0.286	0.143	0.143	0.143	0.286
	Discovery 0.278	27.8%	0.200	0.200	0.200	0.200	0.200
	Conformance 0.149	14.9%	0.243	0.243	0.027	0.243	0.243
	Enhancement 0.206	20.6%	0.200	0.200	0.200	0.200	0.200
		1.0	22.0%	20.3%	15.7%	20.3%	21.7%

(b)

Figure 9. (a) Overall priorities and ranking of the five alternatives [32]; (b) decision hierarchy [32,33,36].

In Figure 9b, we can see the global priorities of the eleven criteria with regards to the goal of our example, i.e., to find the software tool that is more suitable for supply chain processes of a medium-sized company. As illustrated in this figure, the ranking of the eleven criteria depending on their global priorities is:

1. Discovery (27.8%). Discovery can be used to produce the process model of the company, using the event log [2]. The model is a prerequisite for enhancement and conformance.
2. Enhancement (20.6%). Enhancement can be used to modify the process model of the company to reflect reality in a better way. Moreover, enhancement can be used to add new perspectives to the process model of the company and show bottlenecks in

- company processes, information about resources, service levels, throughput times, frequencies, decision rules, and quality metrics [13].
3. Conformance (14.9%). Conformance can be used to detect, locate, and explain deviations in the supply chain processes of the company and to evaluate the severity of these deviations [13].
 4. Filtering (11.1%). Filtering can be used for displaying only specific information of the supply chain process.
 5. Statistics (7.9%). Statistics can be useful for providing an overview of company processes.
 6. License (5.7%). In our example, an open-source software tool is preferred.
 7. Process Animation (4.1%). Process Animation can be useful for displaying company processes and identifying bottlenecks.
 8. Social Network Mining (2.9%). Social Network Mining can be used for showing interactions among people, during supply chain processes.
 9. No Installation Required (2.1%). In our example, this criterion is not very important.
 10. Browser-based (1.6%). In our example, this criterion is not very important.
 11. No Registration Required (1.3%). In our example, this criterion is not very important.

According to Figure 9a,b, the ranking of the five process mining software tools is:

1. Apromore Community Edition (22.0%)
2. ProM (21.7%)
3. Celonis (20.3%)
3. myInvenio (20.3%)
5. Disco (15.7%)

Hence, the process mining software tool that best suits our needs is Apromore Community Edition. It is important to point out that the result is based on the specific comparative analysis criteria and our judgments. If someone else had selected different comparative analysis criteria and/or had made different judgments, then the software tool that best suits his/her needs could be Celonis, Disco, myInvenio, ProM, or Apromore Community Edition.

9. Discussion

The comparative analysis methodology proposed in this paper consists of four phases. In Phase 1, we list the process mining software tools that we want to compare. In Phase 2, we list the comparative analysis criteria that we want to use for the comparative analysis of the process mining software tools listed in Phase 1. In Phase 3, we list the values of each of the comparative analysis criteria listed in Phase 2 per process mining software tool listed in Phase 1. In Phase 4, we select the process mining software tool that best suits user needs, using one or more of the following three methods:

- Ontology-based selection. In this method, we select the software tool that best suits user needs, from the list of the process mining software tools listed in Phase 1, using ontology, the comparative analysis criteria listed in Phase 2, and the values listed in Phase 3.
- Selection of Software Tool(s) Using Decision Tree. In this method, we select the software tool that best suits user needs, from the list of the process mining software tools listed in Phase 1, using a decision tree, the comparative analysis criteria listed in Phase 2, and the values listed in Phase 3.
- Selection of Software Tool(s) Using AHP. In this method, we select the software tool that best suits user needs, from the list of the process mining software tools listed in Phase 1, using AHP, the comparative analysis criteria listed in Phase 2, and the values listed in Phase 3.

Using ontology and Protégé, users can create an ontology containing the process mining software tools and the criteria and their values for each of the software tools that

they want to compare. In this way, they will be able to execute complex queries in order to find the software tool that best suits their needs.

Furthermore, users can create a decision tree using an algorithm such as C4.5 and the Weka Workbench. Thus, they can break down a complex decision-making process into a number of simpler decisions and provide a solution that can be easier to interpret [23]. In this way, users will be able to see, in a tree-like model, which software tool best suits their needs, depending on the values of selected comparative analysis criteria.

Moreover, users can perform the decomposition of a decision problem into a hierarchy of more easily understood sub-problems using AHP. Each of the sub-problems can then be analyzed independently. After the hierarchy is built, the various elements can be evaluated by comparing them in pairs with respect to their impact on an element that exists above them in the hierarchy. For the comparisons, the judgments of the users about the relative importance and meaning of the elements can be used. Hence, in AHP, human judgements, and not just the underlying information, can be used for performing the evaluation. For example, a practitioner can use AHP and his/her own judgments in order to find the process mining software tool that best suits his/her own needs. This capability distinguishes AHP from other decision-making techniques.

The multi-criteria methodology introduced in this paper can be applied for the selection of a suitable software tool in many different areas. For example, if we want to select the software tool that is suitable for a specific production process, in Phase 2 of the methodology, we can select criteria that are of great importance for this process. Moreover, in Phase 4 of the methodology, we can select the software tool that is more suitable for this process. For example, in the case of AHP, when we pairwise compare the criteria with respect to their importance for the selection of the software tool, we can assign greater weights to criteria that are more important for the specific production process.

A limitation of our work is that we cannot guarantee full reliability of the information about the software tools provided in Tables 1 and 4 and in Sections 8.1 and 8.4.1–8.4.3. This information is based on our own research, on our own review of the five software tools mentioned above, and/or on information provided on the websites of the tools. We did not cross-check this information with the tool vendors.

This paper can be useful to practitioners because it describes prominent process mining software tools. Furthermore, the description of the comparative analysis criteria and the new comparative analysis methodology introduced in this paper can be very useful to practitioners for finding the process mining software tool which is more suitable for them.

The new methodology presented in this paper could be extended by researchers in the future to include more comparative analysis methods. Another possible extension to our work is the collection of feedback from the actual use of the process mining software tools by practitioners. This feedback could provide information about (i) the importance of each one of the features of the process mining software tools; (ii) possible problems of the tools; (iii) new features that may be useful to practitioners. Furthermore, this work can be extended in the future by focusing on the theoretical underpinnings of the methodology and by suggesting extensions as well as new research directions with regards to the adopted decision science methods.

10. Conclusions

This paper describes process mining, lists existing process mining software tools, identifies and describes many criteria that can be used for the comparison of the software tools, and proposes a new comparative analysis methodology. The proposed methodology can be very useful, since it can help users to make comparative analyses of process mining software tools and decide which tool best suits their own needs. The new methodology describes three different methods that can be used for the comparative analysis, namely ontology, decision tree, and AHP. Furthermore, this methodology provides a framework that allows users to compare any number of process mining software tools

using any number of comparative analysis criteria. More tools and/or more criteria can be added or removed, and the results of the comparisons can be updated easily. Compared to other related works, this paper provides a more extensive list of process mining software tools and identifies and describes more comparative analysis criteria. Furthermore, to the best of our knowledge, there is no related work providing a detailed comparative analysis methodology of process mining software tools such as the one described in this paper.

Author Contributions: Conceptualization, P.D. and D.A.; methodology, P.D. and D.A.; investigation, P.D. and D.A.; resources, P.D. and D.A.; writing—original draft preparation, P.D. and D.A.; writing—review and editing, P.D. and D.A.; visualization, P.D. and D.A.; supervision, P.D. and D.A.; project administration, P.D. and D.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Van Der Aalst, W.M.; Reijers, H.A.; Weijters, A.J.M.M.; Van Dongen, B.F.; De Medeiros, A.K.A.; Song, M.; Verbeek, H.M.W. Business process mining: An industrial application. *Inf. Syst.* **2007**, *32*, 713–732.
2. Van Der Aalst, W.; Adriansyah, A.; De Medeiros, A.K.A.; Arcieri, F.; Baier, T.; Blickle, T.; Burattin, A. Process mining manifesto. In *International Conference on Business Process Management*; Springer: Berlin/Heidelberg, Germany, August 2011; pp. 169–194.
3. Agarwal, N.; Singh, L. Process mining tools: A comparative analysis and review. *Adv. Comput. Sci. Inf. Technol. (ACSIT)* **2014**, *1*, 26–29.
4. Dakic, D.; Sladojevic, S.; Lolic, T.; Stefanovic, D. Process Mining Possibilities and Challenges: A Case Study. In Proceedings of the 2019 IEEE 17th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 12–14 September 2019; pp. 000161–000166.
5. Claes, J.; Poels, G. Process mining and the ProM framework: An exploratory survey. In *International Conference on Business Process Management*; Springer: Berlin/Heidelberg, Germany, September 2012; pp. 187–198.
6. Turner, C.; Tiwari, A.; Olaiya, R.; Xu, Y. Process mining: From theory to practice. *Bus. Process. Manag. J.* **2012**, *18*, 493–512.
7. Celik, U.; Akçetin, E. Process mining tools comparison. *Online Acad. J. Inf. Technol.* **2018**, *9*, 97–104.
8. Da Silva, L.F.N. Process Mining: Application to a Case Study. 2014. Available online: <https://core.ac.uk/download/pdf/143395465.pdf> (accessed on 5 February 2021).
9. Van der Aalst, W.M. Business process simulation revisited. In *Workshop on Enterprise and Organizational Modeling and Simulation*; Springer: Berlin/Heidelberg, Germany, June 2010; pp. 1–14.
10. Van Dongen, B.F.; de Medeiros, A.K.A.; Verbeek, H.M.W.; Weijters, A.J.M.M.; van Der Aalst, W.M. The ProM framework: A new era in process mining tool support. In *International Conference on Application and Theory of Petri Nets*; Springer: Berlin/Heidelberg, Germany, June 2005; pp. 444–454.
11. Aldwairi, T.; Perera, D.; Novotny, M.A. Measuring the Impact of Accurate Feature Selection on the Performance of RBM in Comparison to State of the Art Machine Learning Algorithms. *Electronics* **2020**, *9*, 1167.
12. Dogan, O.; Martinez-Millana, A.; Rojas, E.; Sepúlveda, M.; Munoz-Gama, J.; Traver, V.; Fernandez-Llatas, C. Individual Behavior Modeling with Sensors Using Process Mining. *Electronics* **2019**, *8*, 766.
13. Van Der Aalst, W. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*; Springer Science+Business Media: Berlin/Heidelberg, Germany, 2011.
14. Van der Aalst, W.; Weijters, T.; Maruster, L. Workflow mining: Discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 1128–1142.
15. Rozinat, A.; Van der Aalst, W.M. Conformance checking of processes based on monitoring real behavior. *Inf. Syst.* **2008**, *33*, 64–95.
16. Leemans, S.J.; Fahland, D.; van der Aalst, W.M. Exploring processes and deviations. In *International Conference on Business Process Management*; Springer: Cham, Switzerland, September 2014; pp. 304–316.
17. Van der Aalst, W.M.; van Dongen, B.F.; Günther, C.W.; Rozinat, A.; Verbeek, E.; Weijters, T. ProM: The process mining toolkit. *BPM (Demos)* **2009**, 489, 2.
18. Günther, C.W.; Rozinat, A. Disco: Discover Your Processes. *BPM (Demos)* **2012**, *940*, 40–44.
19. Van der Aalst, W. Spreadsheets for business process management: Using process mining to deal with “events” rather than “numbers”? *Bus. Process. Manag. J.* **2018**, *24*, pp. 105–127.
20. Van der Aalst, W.M.; Song, M. Mining social networks: Uncovering interaction patterns in business processes. In *International Conference on Business Process Management*; Springer: Berlin/Heidelberg, Germany, June 2004; pp. 244–260.
21. Augusto, A.; Conforti, R.; Dumas, M.; La Rosa, M.; Maggi, F.M.; Marrella, A.; Mecella, M.; Soo, A. Automated Discovery of Process Models from Event Logs: Review and Benchmark. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 686–705.

22. Vázquez-Barreiros, B.; Mucientes, M.; Lama, M. Mining Duplicate Tasks from Discovered Processes. In *ATAED@ Petri Nets/ACSD*; CEUR-WS.org: Brussels, Belgium, 2015; pp. 78–82.
23. Safavian, S.R.; Landgrebe, D. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern.* **1991**, *21*, 660–674.
24. Taticchi, P.; Tonelli, F.; Cagnazzo, L. A decomposition and hierarchical approach for business performance measurement and management. *Meas. Bus. Excell.* **2009**, *13*, pp. 47–57.
25. Fensel, D. *Ontologies*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 11–18.
26. Noy, N.F.; Crubézy, M.; Ferguson, R.W.; Knublauch, H.; Tu, S.W.; Vendetti, J.; Musen, M.A. Protégé-2000: An open-source ontology-development and knowledge-acquisition environment. In *AMIA Annual Symposium Proceedings*; American Medical Informatics Association: Washington, DC, USA, 2003; pp. 953–953.
27. Horridge, M.; Knublauch, H.; Rector, A.; Stevens, R.; Wroe, C. *A Practical Guide to Building OWL Ontologies Using the Protégé-OWL Plugin and CO-ODE Tools Edition 1.0*; University of Manchester: Manchester, UK, 2004.
28. Peng, W.; Chen, J.; Zhou, H. An Implementation of ID3-Decision Tree Learning Algorithm. 2009. Available online: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.471.5158&rep=rep1&type=pdf> (accessed on 5 February 2021)
29. Brijain, M.; Patel, R.; Kushik, M.; Rana, K. A Survey on Decision Tree Algorithm for Classification. 2014. Available online: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.673.2797&rep=rep1&type=pdf> (accessed on 5 February 2021).
30. Mu, E.; Pereyra-Rojas, M. Understanding the analytic hierarchy process. In *Practical Decision Making*; Springer: Cham, Switzerland, 2017; pp. 7–22.
31. Saaty, T.L. Decision making with the analytic hierarchy process. *Int. J. Serv. Sci.* **2008**, *1*, 83.
32. Goepel, K.D. Implementation of an Online Software Tool for the Analytic Hierarchy Process (AHP-OS). *Int. J. Anal. Hierarchy Process.* **2018**, *10*, doi:10.13033/ijahp.v10i3.590.
33. Goepel, K.D. AHP Online System—AHP-OS. 19 September 2019. Available online: <https://bpmsg.com/academic/ahp.php> (accessed on 24 December 2020).
34. Fornari, F.; La Rosa, M.; Polini, A.; Re, B.; Tiezzi, F. Checking Business Process Correctness in Apromore. In *International Conference on Advanced Information Systems Engineering*; Springer: Cham, Switzerland, June 2018; pp. 114–123.
35. Bpmsg. Available online: [https://bpmsg.com/ahp/ahp-altcalc.php?n=5&t=License&c\[0\]=Apromore+Community+Edition&c\[1\]=Celonis&c\[2\]=Disco&c\[3\]=myInvenio&c\[4\]=ProM](https://bpmsg.com/ahp/ahp-altcalc.php?n=5&t=License&c[0]=Apromore+Community+Edition&c[1]=Celonis&c[2]=Disco&c[3]=myInvenio&c[4]=ProM) (accessed on 14 June 2020).
36. Goepel, K.D. AHP Group Results. 3 October 2019. Available online: <https://bpmsg.com/ahp/ahp-group.php?sc=qybaru> (accessed on 24 December 2020).