

Hands-on Process Discovery with Python - Utilizing Jupyter Notebook for the Digital Assistance in Higher Education

Adrian Rebmann,¹ Alexander Beuther,¹ Steffen Schumann,¹ Peter Fettke¹

Abstract: The university course *Process Mining* aims at teaching contents as practically as possible. Students should therefore be able to apply the theoretical knowledge acquired in the course to real-world scenarios. In order to bridge the gap between these theoretical foundations and real-world applications, we propose Jupyter Notebook as an interactive learning environment for the introduction to process model discovery. To this end, we present an exercise, which consists to a large extent of the data preparation of process execution data, distributed in database tables. From this, a meaningful event log is to be generated and consequently, process model discovery techniques are to be applied. All descriptions and data that are necessary to complete the exercise, are provided through a single notebook. This guides the students throughout the process and also eases the grading for the instructors. The proposed exercise is meant as a pilot for evaluating the teaching approach in the course. We conducted initial tests with university staff, resulting in positive feedback. An in-depth evaluation is planned during this semester's edition of the course.

Keywords: Jupyter Notebook; Interactive Data Science; Modelling Course; Process Model Discovery

1 Introduction

1.1 Motivation

In recent decades, university teaching has increasingly demanded a change of perspective from lecturer-centered to student-centered teaching, which allows students to take an active role to practically apply the theoretical content of a lecture [MF16]. Especially in process model discovery, proven concepts, existing models in different representations, real-world data and software technologies must be combined to create process models. Moreover, not only the learning content itself, but also the skills of project management, collaboration and flexible processing need to be addressed. Digital solutions are discussed in traditional lecture formats but platforms and approaches that support an integrated representation of knowledge as well as technical prerequisites for solution finding are still sparse in university teaching.

¹ German Research Center for Artificial Intelligence (DFKI) and Saarland University, Institute for Information Systems, Campus D3.2, 66123 Saarbrücken, Germany, firstname.lastname@dfki.de

1.2 Challenges in Creating Complex and Realistic Process Discovery Exercises

This work addresses several challenges when creating realistic and complex exercises in the context of process mining, especially process model discovery. In order to address the challenge of creating a plausible and realistic data basis, first, a method for the generation of event log data from real business process models is presented. Event logs should be converted into data tables, as they are frequently implemented in ERP-systems. This would also allow for sharing data for teaching purposes without data protection problems, but still with relation to a real-world scenario. Secondly, problems, which also occur with real company data (data acquisition and integration from multiple systems; data storage in different tables and formats; data transformation; data preparation; etc.), should be addressed but are often difficult to introduce in a single exercise at once. Therefore, students should be guided through the process selectively in order to ensure success in processing the overall task, to avoid frustration, and to spark interest. Thirdly, apart from these specific challenges of the creation of exercises, they should also foster project management skills, self-discipline and self-organization.

1.3 Guided Learning with Notebooks

For tackling these challenges, we create a complex exercise within a Jupyter notebook. It offers selective support by providing guidance in sub-tasks, and simultaneously gives the possibility of inserting practical implementations using the Python programming language. The concept corresponds to a guided programming project in Python for the *Process Mining* course, specifically for process (model) discovery. A web interface presents all necessary conceptual basics, data sets and concepts for programming. Students are guided step-by-step throughout the tasks. In appropriate cells of the notebook, students can insert their own implementations and inspect the results immediately to compare them with the expected result. Thus, using the notebook for process (model) discovery presented in this paper, students can be guided through the integration of knowledge, data sets and programming in a single interface to obtain hands-on experiences with process mining. Currently, the prototype is in the test phase and first evaluation results have been received. These first results will be used for improvements in further iterations. In addition, continuous expansion of the task corpus is planned. Furthermore, students train the application of digital learning contents, which is one of the goals following the agile Manifesto of education [Kr17]. Additionally, it offers possibilities for communication, location-independent cooperation, and time-flexible processing of tasks.

1.4 Underlying Perspective on Teaching Process Model Mining

The paper at hand is written from a practitioner's perspective. The authors designed, organized and conducted an introductory course on process mining at Saarland University

which has been held for several years. In an effort to make the course more practical and teach students a more data science oriented approach to process mining, python programming was introduced into the course concept. The paper at hand presents a particularly interesting and complex exercise that was created for this purpose. This exercise was created to imitate a real-world scenario of process model discovery from data extraction to model visualization. For this category of exercises students are not necessarily expected to deliver homogeneous results, which are equally measurable, but rather intends to follow an agile teaching approach. The *Agile Manifesto in Teaching and Learning* [Kr17] describes the experiences of six universities that have built up a learning community over two years to explore agile ways of teaching and learning. Agile rules that we follow in our approach are especially *Meaningful learning over the measurement of things*, but also *Stakeholder collaboration over complex negotiation* and *Responding to change over following a plan*.

1.5 Structure of this Paper

The paper at hand is structured into seven sections. Section 2 describes the theoretical foundations of the presented concept, its relation to e-learning and introduces requirements for the developed concept. Section 3 describes the general structure of the *Process Mining* course and shows the integration potential of tasks like the one presented in this paper. Section 4 documents the conception and creation of the exercise. Section 5 describes the procedure for creating the data basis of the task. Subsequently, the implementation of the notebook is presented and illustrated with excerpts of the notebook. Section 6 introduces related work and Section 7 concludes the paper, discusses limitations and gives an outlook on further development and applications.

2 Relation to Digital Learning

This section presents the relation to digital learning of the teaching approach presented in this paper. To this end, an overview of learning requirements and learning objectives is given and the Jupyter Notebook environment and its application potential in higher education is outlined.

2.1 Learning Requirements and Objectives

Besides the design of the learning environment, it is important to define the measurements of learning success. This work follows the de-facto standard *Kirkpatrick's Model for Evaluation* [Ki98, Po15], which was updated with further specifications [Po15]. Goals were derived for every level of evaluation. The four levels of the model are:

1. Reactions or Motivation [MR89]: Are learners satisfied with the exercise? Traditional course evaluations are one example for measuring the reaction level. It measures the reaction to the learning content, sizes, like engagement, relevance and satisfaction and also whether students found the content relevant, interesting, etc. The main question is: What do students need to perceive in order to perform and learn?
2. Learning: In this level, the degree to which students have acquired knowledge based on their solution of the exercise. This evaluation needs to be performed before, during, and after the exercise. The main question for the task-design is: What skills, knowledge, attitudes and resources are required?
3. Behaviour or Performance [Gi98]: Measures the degree for students behaviour or performance change as a result of the exercise. E.g. how they perform on solving similar questions. Main question for the exercise: What do students have to perform to solve practical issues, relevant for their future working life?
4. Results: This level determines tangible results of the teaching, like improved quality, efficiency, productivity or higher morale. This requires pre- and post-measurements of the learning objective. Main question is: How have students improved their productivity and quality of work?

2.2 Application of Notebooks in Higher Education

Jupyter Notebook is part of the Jupyter Project², which is a nonprofit organization that develops and provides open-source software and services for interactive programming, and supports over 40 programming languages. A Jupyter notebook is a web-based, interactive programming environment. To support university teaching, open-source licensing, free availability and integration capabilities are particularly promising features. For lecturers and students, the following functions are of particular interest: Support of several programming languages, different cell formatting possibilities, and the various export functions for results are advantageous. In general, notebooks consist of cells that can contain text, tables, links, graphics, images, videos, or formulas, and cells that contain and are able to execute program code. In addition, the Jupyter Notebook environment in conjunction with the Anaconda framework³ are a powerful data and process analytic tool. A notebook contains all relevant information, including the program code and the support of necessary data set imports. It also allows the user to execute and document their analysis, as well as their hypotheses and assumptions [Pe18]. Moreover, Jupyter as well as all enabling tools are open source. However, potential issues with notebooks exist, like the lack of logical organization of code, module reuse, and limited testing capabilities [Pe18]. Considering the focus of the course (see Section 3), this is not an issue here.

² <https://jupyter.org>

³ <https://anaconda.org>

3 Integration into the Course *Process Mining*

The lecture *Process Mining* is based on the eponymous book by Wil van der Aalst [vdA16], and teaches Petri nets, process modeling, process mining and data mining. The course format consists of a total of six online lectures, which alternate with six exercise sheets and associated classroom tutorials at Saarland University. The theoretical basics are made available online to the students for independent and self-responsible studying. The use of the online content is not monitored, which is intended to promote students' autonomy, time and self-management. During the attendance weeks, students are provided with an exercise sheet that is adapted to the theoretical content of the previous week. This addresses the areas of theory, practical process mining applications, and practical process mining implementation of individual solutions. The tutorials support students in the selection of online contents, the development of problem-solving strategies, and the handling of a wide variety of information sources and implementation tools. There is a number of tools suitable for analyzing and manipulating process data. Python is particularly useful for these purposes. It is freely accessible and there are numerous libraries and frameworks that can be used out of the box. For example, Python is superior to Excel in the analysis of temporary data. [SI17]. This is an important feature for process mining, as it essentially requires temporal data. Jupyter notebooks on the other hand are intended as a kind of logbooks of analyses that are conducted sequentially. It intermediately displays each analysis step and result, or even other content such as explanatory texts or graphics in a meaningful sequence. This advantage is very important for the tracability and replicability of analyses. Furthermore, if used online, the notebooks introduce a valuable collaboration opportunity. Hence, we consider Python in conjunction with the Jupyter Notebook environment an ideal candidate for teaching process mining concepts in the lecture. During the lecture, these technologies were already used for explaining process mining concepts and to introduce tasks in the exercise sheets. Yet, these tasks were built on existing well-formed process logs either in XES or CSV formats. These types of tasks were adopted well by students. Yet, the case at hand is a complex case, not a standalone analysis one probably would start with. It requires students to, first, develop a data understanding and then perform a series of data preparation and transformation steps.

4 Conception of the Modelling Exercise

The conceptual design of the task described in this paper focuses on the creation of an event log from a database and the application of established process discovery methods to create a de-facto process model [vdA16].

When executing a business process, various transactions are executed in IT systems, for example ERP-systems, whose data is then stored in database systems. The database, which serves as the basis for the task, contains records of an order-to-cash process of a company offering both physical products and services. It was created using a simulation of an existing

process model. The overall task is to reconstruct the process model from the database. This task was chosen primarily because companies rarely have well-formed event logs in the format, which is necessary for the immediate application of process mining techniques, e.g. the standard format XES [GV14]. As in many data mining projects, the main work is the development of a data understanding and data preparation. The actual modelling part is often rather small in the overall project [Sh00].

The task to be implemented in the Jupyter notebook, thus, consists to a large extent of the data preparation of process execution data distributed in database tables. From this, a meaningful event log is to be generated and then one or more process models are derived using process discovery methods. The overall task is divided into small-step sub-tasks. Each cell of the Jupyter notebook describes what the respective step on the way to a process model is and which tools are to be used. The partial tasks can be summarized as follows:

1. **Import of Required Libraries:** The first step is to import libraries that will be used in the exercise. The main scientific libraries are *NumPy*, *Pandas*, and *Matplotlib*, as well as the process mining library *pm4py* [BvZvdA19], and the interface to the SQL database, *SQLite*
2. **Develop Data-Awareness:** First the students should get an overview about the data structures, describe them and make an initial suggestion for attributes that are necessary for process mining, namely an ID for each process instance, activities, and times of every respective execution.
3. **Guided Creation of Partial Event Logs:** Partial event logs initially represent only a small portion of the overall process, namely the first activity. Specifically, in this case the *customer inquiry*. The general procedure for setting up an event log is explained using this example.
4. **Incremental Development of Event Logs:** Constructing the complete event log to cover the end-to-end process.
5. **Writing the Event Logs:** Transformation of the event log into the XES format. This provides the foundation for discovering the process model.
6. **Process Discovery:** Applying various process discovery approaches results in different process model visualizations. By comparing the results of the different methods, e.g. the Alpha Miner with the Heuristics Miner, more meaningful process models can be derived.

5 Implementation

The implementation of the exercise consists of the creation of a data basis and the implementation of the actual Jupyter notebook.

5.1 Creation of the Data Basis

The following procedure was used to create the data basis for the described task. The approach is depicted in Fig 1:

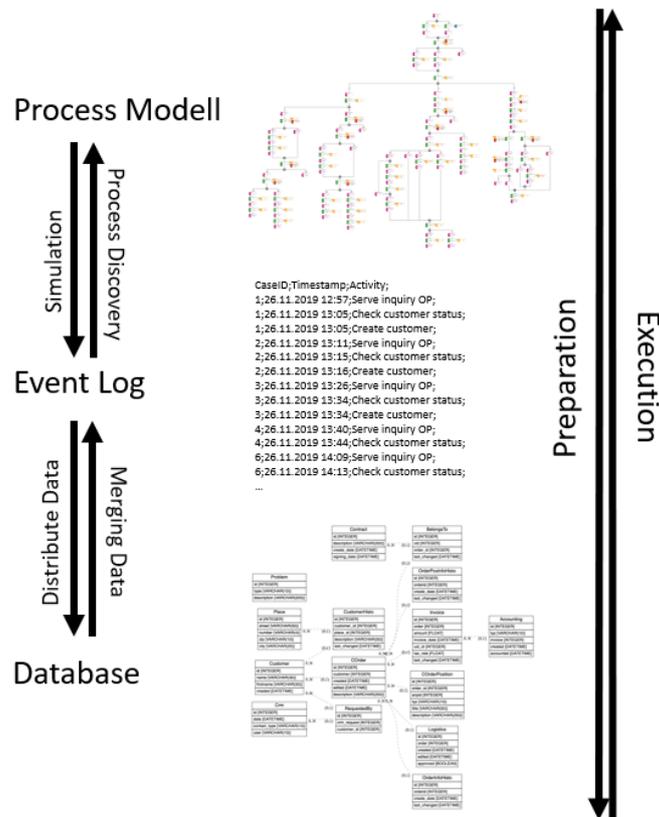


Fig. 1: General approach for the preparation and solution of the exercise.

1. **Modelling a Business Process Model:** A real-world business process model of a complete end-to-end process of a fictitious company, from handling customer inquiries to receiving payment builds the foundation for the task. The process was modelled using the ARIS Architect using the Event-Driven Process Chain notation. The model represents a *Best Practice* of an Order-to-Cash process. Thus, the model generally excludes aborts, unnecessary loops and errors, which frequently occur in the real-world process behaviour. Second, several functions and activities are annotated to the model, to identify and quantify diverse performance indicators. These elements are not directly related to the process model but are important for statistical measures with regard to the simulation.

2. **Simulation of the Process Execution:** The next step is the simulation of the business process execution. The use of simulated data of a process model eliminates availability issues as well as the need for data protection concerns. Similar approaches generate event data for obtaining the appropriate evaluation data for discovery techniques in process mining [JD19]. A simulation over 100 days was executed, which yielded 10,000 cases, where every case varies in duration, execution times of activities, resources, and outcomes. The execution data is stored in a spreadsheet format. The model also includes cases that were not completed, as the simulation is time-based and leaves a number of instances unfinished. So, slightly malformed data is present, which also reflects real-world behaviour of many processes. However, the presence of such malformed behaviour could be extended, e.g. in conformance-checking exercises.
3. **Transformation into Event Logs:** The simulated process executions are converted to CSV format in order to execute all further steps. All ARIS-specific columns are removed, because they are irrelevant for the task.
4. **Transformation of the Event Logs into Data Tables:** The final step is to derive a data structure and generate plausible data sets for the simulated process instances. Therefore we have developed a database schema, which reassembles a simplified version of a customer inquiry management system. The respective database tables are then filled by a developed Python program and the simulated event log.

This completes the data basis for the modeling task. As described in 4, the task consists of executing the steps in reverse, as illustrated in Fig. 2.

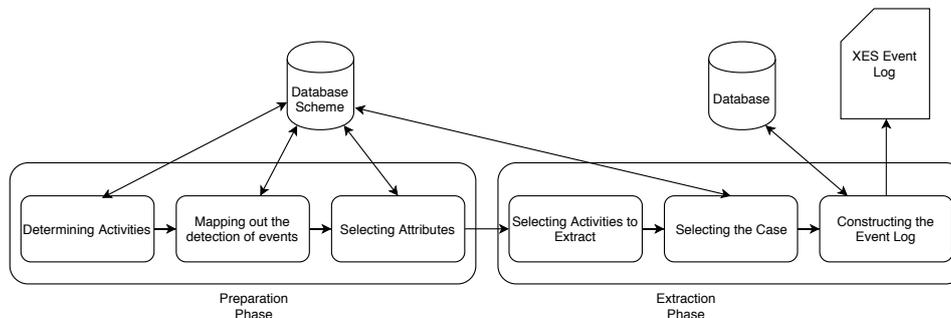


Fig. 2: Procedure for extracting event data from a database and constructing an event log. C.f. [Pi11].

The event log used to fill the data tables and the model can therefore be used as a basis for evaluation. However, it would be false to assume that only one possible model can result. There are basically infinitely many possible models. In this specific case, it depends mainly on how many activities are extracted from the database and how they are grouped into cases. The choice of case is essential for the process model that results. Therefore for assessing the students work, first, the final result counts. Therefore, a meaningful model should be present. If this is not available, then the intermediate results are evaluated. Since, in the

specific task at hand, there are a variety of possible results, depending on the rigour of the pre-processing. As a minimal requirement, the code in the notebook should be executable without errors and produce visualizations wherever requested.

5.2 Implementation of the Jupyter Notebook

All materials that are necessary to complete the exercise are stored in a single folder. The respective cells inside the Jupyter notebook reference files, in particular, a SQL database and an XES log later in the exercise, whenever they are of interest in a sub-task. The task was implemented according to the specification introduced in Section 4. Fig. 3 and Fig. 4 depict excerpts of the implemented notebook.

6 Related Work

The Jupyter Notebook environment is already used for general programming courses [GGC19]. However, no teaching applications were found in the modeling domain, even though this area is very suitable for individual implementations because it follows well researched and standardized procedures. In the area of process mining, the authors of [Sh18] introduced a learning module using process mining tools such as Disco⁴ and ProM⁵ to teach workflow and bottleneck analysis. They used the developed module in a graduate course and first results and feedback from students show positive results. [DP19] present a game-based approach, which teaches various aspects of business process management. Students can play the game in groups using e.g. a web-based process modelling interface. The game has been used for several years in a course at Eindhoven University of Technology. In [GGC19] an approach for teaching lab classes in a computer science course is introduced. The authors use the Scrum framework, for creating student-centered learning activities in course classes. They use Jupyter Notebook for online experimentation.

7 Discussion and Outlook

We presented an exercise focusing on the extraction of an event log from a database containing data of the order-to-cash process of a company. After building the log, process (model) discovery is the general goal of the task. All steps can be performed within a single Jupyter notebook, which guides students through the exercise while referencing all necessary materials.

Due to a missing long-term and in-depth evaluation of the proposed concept we cannot prove the fulfillment of all the objectives discussed in 2 but can provide first-hand experiences we made as course organizers.

⁴ <https://fluxicon.com/disco/>

⁵ <http://www.promtools.org/>

Building the event log (CRM-part)

The next step would be to create a dataframe representing the event log. For this, the tables and fields of the database, which contain relevant event data need to be merged. First, as an example, we want to focus on the CRM-Process: We import and merge the tables Crm, requestedBy and Customer using Pandas' SQL Interface. All values of table Crm should be present in the imported data set. Pandas features many importing functionalities. Data fields are automatically assigned suitable data types. However, special functions can be specified before data import, such as parsing of dates, specifying a custom header and defining a separator. Also different import functions are available to create a DataFrame:

- `pd.read_csv()` CSV
- `pd.read_excel()` Excel
- `pd.read_sql_query` SQL

We use the `read_sql_query` function

```
query_all_customer_requests_with_account_creation = """select * from Crm crm LEFT join requestedBy rb on crm.id = rb.c
header = ['id', 'crm_request_time', 'contact_type', 'crm_user', 'join_table_id', 'merge_id_customer', 'merge_id_crm',
dataset = pd.read_sql_query(query_all_customer_requests_with_account_creation, conn, parse_dates=['date', 'created'])
# setting a new header
dataset.columns = header
dataset.head()
```

	id	crm_request_time	contact_type	crm_user	join_table_id	merge_id_customer	merge_id_crm	customer_id	name	firstname	customer_create_time
0	1	2019-01-01 09:42:48	call center	user_107	1.0	1.0	1.0	1.0	Hans	Ing.	2019-01-01 10:12:36
1	2	2019-01-01 09:57:12	call center	user_102	NaN	NaN	NaN	NaN	None	None	NaT
2	3	2019-01-01 10:11:36	online	None	NaN	NaN	NaN	NaN	None	None	NaT
3	4	2019-01-01 10:26:00	online	None	2.0	4.0	2.0	2.0	Stahr	Heinz-Werner	2019-01-01 10:28:41
4	5	2019-01-01 10:40:24	online	None	3.0	5.0	3.0	3.0	Karz	Sinaida	2019-01-01 10:46:48

```
# in order to avoid naming issues later, we can use the attribute names defined by the XES standard for case id, activ
def prepare_customer_request(x):
    log_line = {'case:concept:name': x.id}
    log_line['concept:name'] = 'online request' if x.contact_type == 'online' else 'call center request'
    log_line['time:timestamp'] = x.crm_request_time
    log_line['org:resource'] = x.crm_user if x.crm_user else 'online platform'
    return log_line
```

Fig. 3: Sub-task to extract events from the customer-request handling of the process.

Initial tests with institute staff yielded positive feedback. An emphasis was on the fact that no switching between different media or documents was necessary to solve the task. Yet, a prior introduction into the use case and some instructions regarding the intended workflow were necessary. Also, in case of a first-time-use, the tool setup is a factor to be considered. A possible limitation for using the Jupyter Notebook environment as teaching support for process discovery in this context, could be that the task presented is based on only one existing database scenario. Thus, the students could be less independent in their solution finding. Nevertheless, complex real world process analytic problems like the one presented in this paper can be conveyed to students while easing the grading of programming assignments for the instructors. This is because a more fine-grained evaluation becomes possible based on the individual sub-tasks organized as cells.

The effort for creating this specific task is quite significant. Especially the model creation and simulation are very complex. However, the general notebook approach is unaffected by this, here the effort of task creation is lower, because all tasks and other materials can be created in one "place". Data generation is always problematic. Once the underlying model has been created and the program for dividing it into data base tables has been adapted to the use case, arbitrarily large log-files can be generated. Smaller model adaptations are not problematic but a change of the use case leads to significant effort. Jupyter Notebook supports sequential processing in terms of guidance, which enables students to work on such complex tasks.

Converting and Using the Log

As already seen in the lecture and other tasks XES is

- the de facto standard log format for Process Mining
- extends the XML format
- is supported by commercial as well as academic Process Mining tools This is why we use the created DataFrame to create a XES compliant log in this step.

- Task:**
- take the data frame created before and create a XES compliant log using the pm4py function

```
from pm4py.objects.log.exporter.xes import factory as xes_exporter
from pm4py.objects.conversion.log import factory as conversion_factory
# convert the dataframe here
```

Discovery

For this step, we import the XES-Log we created in the previous step. After this, we use the heuristics miner [1], the discover a model from our log. Heuristic mining in contrast to the Alpha-Algorithm [2] uses causal nets. It can handle concurrency and considers frequencies of events, so traces that do not happen frequently are abstracted from, in other words, noise is filtered out. Using the frequency of two activities following one another, dependency measures are computed and a dependency graph aka. causal net is created using certain thresholds for the dependency measure and frequency of activities directly following one another. The input is an event log that requires at least case identifiers and event labels. A causal net representing the process is the output of the algorithm.

Task:

Please use pm4py to apply

- the heuristics miner algorithm. Please explain the following measures. Also use different configurations.
 - dependency threshold of the Heuristics Miner
 - AND measure threshold of the Heuristics Miner
- the alpha algorithm.

```
from pm4py.algo.discovery.heuristics import factory as heuristics_miner
from pm4py.objects.log.importer.xes import factory as xes_importer
```

Fig. 4: Sub-task to convert the created event log and applying process discovery techniques.

The created notebook will be used in one of the next exercise sheets of the lecture *Process Mining* at Saarland University. After the students completed the sheet, focused feedback will be gathered from them. Additionally, the overall lecture evaluation conducted by the university will certainly also be included in the evaluation process. Furthermore, a comparison with other learning approaches, e.g. without providing a frame, would also be useful in the course of an evaluation in order to demonstrate the advantages of this approach, especially with regard to the targeted preparation of all necessary content in a single medium.

In addition to the actual model generation, the use of other methods of process mining is planned to be taught in this form and has already been done in small tasks in past exercise sheets. The software that was implemented to create the database from the simulated event log, can be reused flexibly to create other scenarios for teaching process mining, focusing not only on process model creation, but also on use cases like compliance and conformance-checking.

References

- [BvZvdA19] Berti, Alessandro; van Zelst, Sebastiaan J; van der Aalst, Wil: Process Mining for Python (PM4Py): Bridging the Gap Between Process-and Data Science. p. 13–16, 2019.

- [DP19] Dijkman, Remco; Peters, Sander: The Business Process Management Game. In (Depaire, Benoît; De Smedt, Johannes; Dumas, Marlon, eds): Proceedings of the Dissertation Award, Doctoral Consortium, and Demonstration Track at BPM 2019 co-located with 17th International Conference on Business Process Management (BPM 2019). CEUR Workshop Proceedings. CEUR-WS.org, pp. 119–123, 1 2019.
- [GGC19] Guerra, H.; Gomes, L. M.; Cardoso, A.: Agile approach to a CS2-based course using the Jupyter notebook in lab classes. In: 2019 5th Experiment International Conference (exp.at'19). pp. 177–182, June 2019.
- [Gi98] Gilbert, T: A leisurely look at worthy performance. The 1998 ASTD Training and Performance Yearbook, 1998.
- [GV14] Günther, Christian W.; Verbeek, Eric: XES Standard Definition. BPMcenter.org, 2014.
- [JD19] Jouck, Toon; Depaire, Benoît: Generating Artificial Data for Empirical Analysis of Control-flow Discovery Algorithms. Business & Information Systems Engineering, 61(6):695–712, Dec 2019.
- [Ki98] Kirkpatrick, Donald I. : Evaluating Training Programs: The four levels. 2nd Editio, 1998.
- [Kr17] Krehbiel, Timothy C; Salzarulo, Peter A; Cosmah, Michelle L; Forren, John; Gannod, Gerald; Havelka, Douglas; Hulshult, Andrea R; Merhout, Jeffrey: Agile Manifesto for Teaching and Learning. Journal of Effective Teaching, 17(2):90–111, 2017.
- [MF16] Michael Fellmann, Andreas Schoknecht, Meike Ullrich: Workshop zur Modellierung in der Hochschullehre. In: 2016 Modellierung 2016-Workshopband. Gesellschaft für Informatik eV, p. 45, 2016.
- [MR89] Markus, Hazel; Ruvolo, Ann: Possible selves: Personalized representations of goals. 1989.
- [Pe18] Perkel, Jeffrey M: Why Jupyter is data scientists' computational notebook of choice. Nature, 563(7732):145–147, 2018.
- [Pi11] Piessens, David: Event log extraction from SAP ECC 6.0. Event Log Extraction from SAP ECC, 6:71, 2011.
- [Po15] Pollock, Roy VH; Jefferson, Andrew McK; Wick, Calhoun W; Wick, C et al.: The six disciplines of breakthrough learning. Wiley Online Library, 2015.
- [Sh00] Shearer, Colin; Watson, Hugh J; Grecich, Daryl G; Moss, Larissa; Adelman, Sid; Hammer, Katherine; Herdlein, Stacey a: The CRISP-DM model: The New Blueprint for Data Mining. Journal of Data Warehousing, 5(4):13–22, 2000.
- [Sh18] Shahriar, Hossain: Teaching of Clinical Workflow Analysis with Process Mining : An Experience Report. 2018.
- [Sl17] Slater, Stefan; Joksimović, Srećko; Kovanovic, Vitomir; Baker, Ryan S.; Gasevic, Dragan: Tools for Educational Data Mining: A Review. Journal of Educational and Behavioral Statistics, 42(1):85–106, 2017.
- [vdA16] van der Aalst, Wil M. P.: Process Mining: Data Science in Action. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.