




Evaluation of Study Plans using Partial Orders

Christian Rennert[✉] , Mahsa Pourbafrani , and Wil van der Aalst 

Chair of Process and Data Science (PADS), RWTH Aachen University, Germany
{rennert,mahsa.pourbafrani,wvdaalst}@pads.rwth-aachen.de

Abstract. In higher education, data is collected that indicate the term(s) that a course is taken and when it is passed. Often, study plans propose a suggested course order to students. Study planners can adjust these based on detected deviations between the proposed and actual order of the courses being taken. In this work, we detect deviations by combining (1) the deviation between the proposed and actual course order with (2) the temporal difference between the expected and actual course-taking term(s). Partially ordered alignments identify the deviations between the proposed and actual order. We compute a partial order alignment by modeling a study plan as a process model and a student’s course-taking behavior as a partial order. Using partial orders in such use cases allows one to relax the constraints of strictly ordered traces. This makes our approach less prone to the order in which courses are offered. Further, when modeling course-taking behavior as partial orders, we propose distinguishing intended course-taking behavior from actual course-passing behavior of students by including either all terms in which a course is attempted or only the term that a course is passed, respectively. This provides more perspectives when comparing the proposed and actual course-taking behavior. The proposed deviation measuring approach is evaluated on real-life data from RWTH Aachen University.

Keywords: Educational Process Mining · Conformance Checking · Event Data · Partial Order · Campus Management System.

1 Introduction

In Germany, universities must provide students with the knowledge, skills, and methods required for their subjects.¹ Furthermore, regulations state that (1) every degree program must offer a study plan completable within the degree’s standard duration, and (2) universities must create individual study plans for students if required.² In this work, a study plan comprises several courses whose completion is assessed by examination activities. Here, only the passing of final exams is considered as such activities, although this is not a general restriction.

¹ In Germany, states regulate higher education. For example, North Rhine-Westphalia defines the purpose of study plans in §58 (1) Hochschulgesetz (HG) NRW.

² According to §58 (3) Hochschulgesetz (HG) NRW.

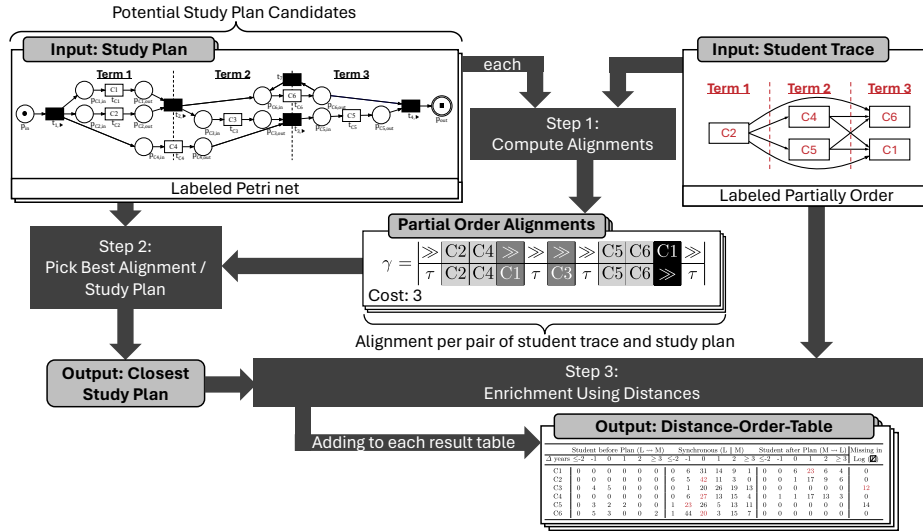


Fig. 1: Framework for the evaluation of study plans using partial orders.

Across all study programs offered in 2022 in Germany, nearly 247,000 students received their bachelor’s degree³. First-time graduates received their degree on average after around 4 years of studies⁴. However, the standard period of study for bachelor’s degree courses in Germany is three years. This indicates an average extension of studies by around one year in 2022.

In [8], potential reasons for an individual to extend the duration of their study are categorized. In detail, they identify (1) study conditions, (2) individual characteristics and entry requirements, and (3) personal living conditions and context factors as categories of reasons. Although there are many reasons for an extended study duration, if possible, study planners should implement measurements to avoid an extended study duration. Therefore, a systematic analysis of student data and study plans helps identify and improve study plan shortcomings such that study conditions are improved.

One approach is to use process mining to understand students’ interactions with their study plan. In detail, we model study plans as process models and store exam data in event logs to allow for systematic analysis.

The complete approach presented in this work is the following. Workflow nets are used to model all study plans for a given time period, as there may be multiple study plans available due to changes applied to a degree over time. Partial orders are used to model the student’s course-taking order. Both are compared using partial order alignments. The alignment with the lowest cost is then identified. For its study plan and all its courses, we aggregate counts for

³ <https://www.statista.com/statistics/584454/bachelor-and-master-degrees-number-universities-germany/>, 2024, last access 2024-07-05

⁴ <https://www.statista.com/statistics/584277/average-study-duration-graduates-germany/>, 2023, last access 2024-07-05.

each combination of (1) the course’s relative position between the log and model parts of the alignment, indicating whether a course is moved forward, backward, or if it is in the right position in the study plan, and (2) the absolute difference between the expected and actual terms that the course is taken. This procedure is visualized in Figure 1.

The remainder of this paper is structured as follows. Section 2 details the related work, Section 3 gives the preliminaries for this work, Section 4 describes the approach introduced in the paper, Section 5 evaluates the results obtained by applying the approach to a real-life dataset, and Section 6 concludes the paper.

2 Related Work

Process mining in the context of education, also known as Educational Process Mining (EPM), is the subject of several papers. An extensive survey by Bogarín et al. [4] discusses educational process mining as an educational data mining (EDM) technique. Here, the authors show that there are some works using conformance checking techniques to identify the agreement between students’ exam histories as event traces and their study plans as process models. Alternatively, the authors in [4] also discuss that conformance checking can be applied using rule modeling for expected student behavior, which is another conformance checking technique not investigated in this paper. For more details on EPM, we refer to the survey in [4]. We limit our scope to works that target the course-taking process of students and their interactions with the study plan.

Hobeck et al. [6] apply process mining and in particular the PM² methodology on exam data from the university information system of the TU Berlin. For a group of students, the authors investigate the question of whether students follow their study plan by comparing its conformance with the full student’s exam history. In the work of Bendatu and Yahya [3], they also compare exam histories with a single study plan by computing the temporal distance between the expected and actual point in time when an exam is taken. We use the general idea and extend the study plan to allow for several expected terms.

In [5], Diamantini et al. model study plans as blocks of exams to be taken to compare the ordering of exams in the student’s exam history and in the study plan. They use their approach to understand bottlenecks and to make a distinction between successful and non-successful students. Similar to abstracting study plans into course blocks, some works model student exam histories as blocks, using a set abstraction per term. In [9], Priyambada et al. use this approach to compare and cluster students by detecting course patterns for the relative term course sets. In [10], Rafiei et al. exploit atomic ordering and educational KPI information in exam data to derive labels for rule extraction with the aim to assist study planners. In their work, they derived partial orders by considering DFGs to directly represent a partial order.

Usually in process mining, the ordering based on timestamps between events and a suggested ordering of activities in process models can be compared using alignments as first introduced by Adriansyah [2]. Later, in [7], Lu et al. extended

alignments to also be applicable for partial orders. Nevertheless, so far no existing work incorporates partial order alignments in its research or case study. Therefore, this is among the first works to incorporate partial orders.

In the following section, we discuss the preliminaries of this work.

3 Preliminaries

Basic Notations. A *multiset* generalizes the concept of a set and allows for multiple occurrences of the same element, e.g., $[x, y, y, x, x, z] = [x^3, y^2, z]$. The set of all multisets over a set X is $\mathbb{M}(X)$. Similarly to a multiset, a *sequence* can contain multiple occurrences of the same element. However, within a sequence, all elements are *totally ordered*, e.g., $\sigma = \langle x, y, y, x, z \rangle$ is a sequence over the set $X = \{x, y, z\}$ and thus it is in its *Kleene closure* X^* , i.e., $\sigma \in X^*$. Given a sequence σ and a value $i \in \mathbb{N}$ with $1 \leq i \leq |\sigma|$, we denote by $\sigma(i)$ the element at the i -th position of the sequence σ , e.g., for $\sigma = \langle x, y, z \rangle$ it holds that $\sigma(3) = z$. Given an n -tuple (a_1, a_2, \dots, a_n) , we define π_i to be the selector function that returns the i -th entry, e.g., $\pi_2((a, b, c)) = b$. We define that applying a single element function $f : X \rightarrow Y$ to a sequence $\sigma \in X^*$, applies the function f to each element in the sequence, i.e., $f(\sigma) = \langle f(x) \mid x \in \sigma \rangle$. The domain of a function f is described by $dom(f)$, e.g., $dom(f) = \{1, 2, 3\}$ for $f(1) = 4$, $f(2) = 5$, and $f(3) = 3$.

Table 1: Example educational event log L_1 .

Student	Course	Term	State
S1	C2	1	1
S2	C1	2	1
S1	C4	2	1
S2	C4	2	0
S2	C4	2	1
S1	C5	2	1
S1	C6	3	1
S1	C1	3	1

Educational Event Log. In process mining, *event data* are used to observe and keep track of *behavior*. In our use case, an event exists for each exam try of a student. Therefore, we first introduce the universe of student matriculation numbers \mathcal{U}_{Mat} and the universe of course IDs \mathcal{U}_{cid} . An event $e \in \mathcal{U}_{Mat} \times \mathcal{U}_{cid} \times \mathbb{N} \times \{0, 1\}$ is a quadruple of a student’s matriculation number, a course ID of a taken course, a natural representing the term in which the event occurred relative to the start of the student’s studies, and a state indicating whether the exam attempt is passed (1) or failed (0). We may also refer to the matriculation number as student ID. The universe of events is denoted by \mathcal{U}_e . An example of an event is $e_1 = (S2, C4, 2, 1) \in \mathcal{U}_e$ which shows that the course C4 is passed by the student S2 in its second term of studies. An event log $L \in \mathbb{M}(\mathcal{U}_e)$ is a multiset of events, as there can be several exam attempts within a term.

Given an event e , for better readability, we define $\pi_{mat}(e) = \pi_1(e)$, $\pi_{cid}(e) = \pi_2(e)$, $\pi_{term}(e) = \pi_3(e)$, and $\pi_{pass}(e) = \pi_4(e)$, e.g., $\pi_{cid}(e_1) = C4$. In this work, we restrict each educational event log L to only contain courses that are eventually passed, i.e., $\forall e \in L \exists e' \in L : ((\pi_{cid}(e) = \pi_{cid}(e')) \wedge (\pi_{pass}(e') = 1))$. An example of an educational event log L_1 is given in Table 1 where $e_1 \in L_1$ holds.

Partial Orders. We use *partial orders* to model temporal relations between student events, these are used because multiple events can occur for the same term and we do not consider the order within a term. In detail, labeled partial orders are used, as we may have to model students who took several courses with the same course ID. This might be the case for a student that is required to take distinct seminars with the same course ID. A labeled partial order lpo is a quadruple $lpo = (V, \prec, A, l)$ where V is a set of nodes, \prec is a partial order relation, A is a set of activities, and $l: V \rightarrow A$ is a labeling function. Figure 2 shows the labeled partial order for the student with **Student ID S1** from event log L_1 . The next section describes how to derive labeled partial orders from an event log. First, we introduce labeled Petri nets as process models so that we can model study plans.

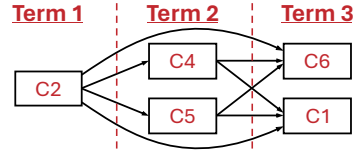


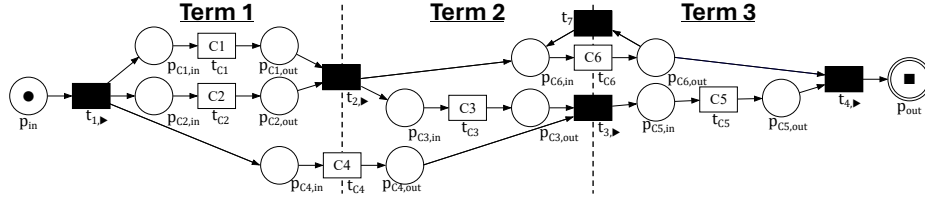
Fig. 2: **Labeled partial order** lpo_{S1} of passed courses for the student with student ID **S1**.

Petri and Workflow Nets. A labeled Petri net N is a quintuple $N = (P, T, F, A, l)$, where P is a finite set of *places*, T is a finite set of *transitions* such that $P \cap T = \emptyset$, $F \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs, the so-called *flow relation*, A is a set of activities, and $l: T \rightarrow A \cup \{\tau\}$ is a *labeling function* with $\tau \notin A$ being a reserved silent activity label corresponding to no activity being performed.

A *workflow net* is a labeled Petri net $N = (P, T, F, A, l)$ for which there is exactly one input place $in \in P$ such that $\{(p, in) \in F \mid p \in P\} = \emptyset$, one output place $out \in P$ such that $\{(out, p) \in F \mid p \in P\} = \emptyset$, and for which modifying it to be short-circuited makes it strongly connected. An example workflow net is shown in Figure 3. For more details on Petri nets as process models and their semantics, we refer to [1]. Workflow nets and labeled partial orders describing observed behavior can be checked for conformance using alignments.

Alignments. A partial order alignment $\gamma(lpo, N)$ relates the behavior described by a labeled partial order lpo with the behavior of a workflow net N . Therefore, an alignment consists of several moves that distinguish whether the behaviors agree and are in synchronization or whether they disagree and deviate. In detail, we distinguish the following four move types. (1) *Synchronous moves* (lightgray) indicate no deviation, (2) *log moves* (black) indicate a deviation such that a move was taken on the log side but not on the model side, (3) *model moves* (gray) indicate a deviation such that a move was taken on the log side but not on the model side, and (4) *invisible model moves* indicated by τ specify that a not observable model move is taken and that a misalignment on the log side is not required. In a deviation, the misalignment is denoted by “ \gg ”, to indicate that either the log or the model does not have any matching behavior.

An alignment that best matches the model and partial order is the so-called *optimal alignment* because it has a minimal number of mismatches. In Table 2, an optimal alignment $\gamma(lpo_{S1}, N_1) = \langle (\gg, \tau), (C2, C2), \dots, (\gg, \tau) \rangle$ between the labeled partial order lpo_{S1} and the workflow net N_1 is given. The optimal alignment $\gamma(lpo_{S1}, N_1)$ has three misalignments, i.e., one log and two

Fig. 3: Example workflow net N_1 .

model moves. The upper row indicates the log behavior, while the lower row indicates the model behavior. For an alignment γ , we define $\gamma_L = \pi_1(\gamma)$ and $\gamma_M = \pi_2(\gamma)$, e.g., $\gamma_L(lpo_{S_1}, N_1) = \langle \gg, C2, C4, \gg, \gg, \gg, \gg, C5, C6, C1, \gg \rangle$ and $\gamma_M(lpo_{S_1}, N_1) = \langle \tau, C2, C4, C1, \tau, C3, \tau, C5, C6, \gg, \tau \rangle$ for the behavior on the log and model side, respectively. We can now consider the impact of using labeled partial orders instead of ordering courses within a term. Consider a labeled partial order lpo'_{S_1} that requires $C4$ and $C6$ to always occur before $C5$ and $C1$, respectively, although events $(S1, C4, 2, 1)$ together with $(S1, C5, 2, 1)$ and $(S1, C6, 3, 1)$ together with $(S1, C1, 3, 1)$ each have the same term, which otherwise indicates no strict ordering. Computing an optimal alignment $\gamma(lpo'_{S_1}, N_1)$, as shown in Table 3, between lpo'_{S_1} and N_1 results in a worse alignment than the alignment $\gamma(lpo_{S_1}, N_1)$, since its cost is higher by a value of two. Therefore, abstracting course orderings within the same term affects alignment computation. For details on the computation of partial order alignments, we refer to [7].

Table 2: An optimal alignment $\gamma(lpo_{S_1}, N_1)$ for lpo_{S_1} and N_1 .

Table 3: An optimal alignment $\gamma(lpo'_{S_1}, N_1)$ for lpo'_{S_1} and N_1 .

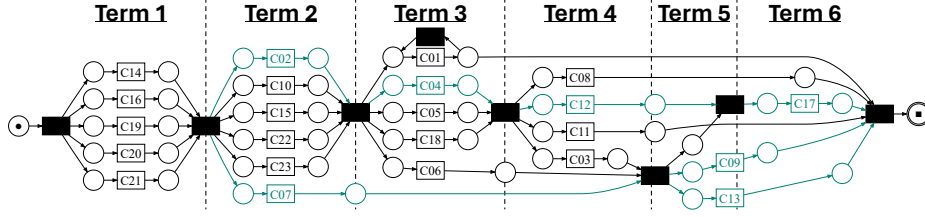
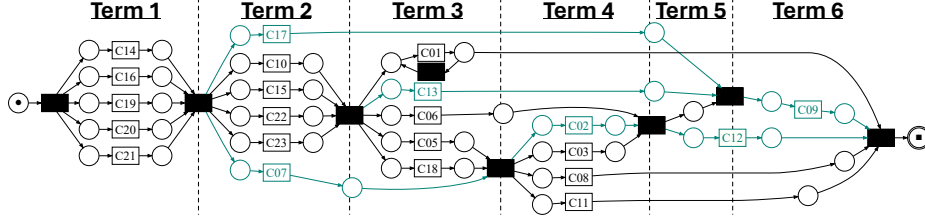
$$\left| \begin{array}{cccccccccccc} \gg & C2 & C4 & \gg & \gg & \gg & C5 & C6 & C1 & \gg \\ \tau & C2 & C4 & C1 & \tau & C3 & \tau & C5 & C6 & \gg & \tau \end{array} \right|$$

$$\left| \begin{array}{cccccccccccc} \gg & C2 & C5 & C4 & \gg & \gg & \gg & \gg & C6 & C1 & \gg \\ \tau & C2 & \gg & C4 & C1 & \tau & C3 & \tau & C5 & C6 & \gg & \tau \end{array} \right|$$

4 Approach

Study Plan. For each degree program, study planners provide a study plan. It organizes the degree program into terms, stating when each course should be taken. In addition, it differentiates between mandatory courses, required to complete a degree program, and elective courses, of which only a subset must be taken. We take electives into account by mapping all elective course IDs to a reserved placeholder elective course ID $C_{el} \notin \mathcal{U}_{cid}$. A study plan process model should allow for the elective course ID C_{el} to reoccur multiple times, modeling an unknown number of electives to be taken by an individual student. Formally, we define a study plan SP as $SP: (\mathcal{U}_{cid} \cup \{C_{el}\}) \rightarrow \mathbb{N} \times \mathbb{N}$ mapping a course ID $cid \in (\mathcal{U}_{cid} \cup \{C_{el}\})$ to its first expected term T_{start} and its last expected term T_{end} , i.e., $SP(cid) = (T_{start}, T_{end})$. Next, we create a workflow net from a study plan to compute labeled partial orders.

Study Plan to Workflow Net. A workflow net $N_{SP} = (P, T, F, A, l)$ is constructed from a study plan SP as follows. **Introducing term start and end synchro-**


 Fig. 4: Workflow net N_{10} from the study plan SP_{10} .

 Fig. 5: Workflow net N_{18} from the study plan SP_{18} .

nizing silent transitions: For each term $term \in \mathbb{N}$ with $1 \leq term \leq k$ contained in the study plan SP , with $k = \max(\{\pi_2(SP(cid)) \mid cid \in \text{dom}(SP)\})$ being the highest term where \max is the maximum function, we add a silent transition $t_{(term, \blacktriangleright)}$ with $l(t_{(term, \blacktriangleright)}) = \tau$ starting each term. We define that each transition $t_{(term, \blacktriangleright)}$ starting a term $term$ is the transition $t_{((term-1), \blacktriangleright)}$ that closes the previous term, i.e., $\forall_{term \in [2, k]}: t_{((term-1), \blacksquare)} = t_{(term, \blacktriangleright)}$. Since there is a missing silent transition that closes the term k , we add $t_{(k, \blacksquare)}$ with $l(t_{(k, \blacksquare)}) = \tau$ to the set of transitions T . We connect the transition that starts the first term with the input place $in \in P$ and the transition that ends the last term with the output place $out \in P$, i.e., $\{(in, t_{(1, \blacktriangleright)}), (t_{(k, \blacksquare)}, out)\} \subseteq F$. **Adding and connecting course transitions:** The set of activities is identical to the set of course IDs contained in the study plan, i.e., $A = \text{dom}(SP)$. Finally, we add a transition t_{cid} with $l(t_{cid}) = cid$ for each $cid \in \text{dom}(SP)$ and connect them by their start and end term in the study plan, i.e., we add places $\{p_{(cid, in)}, p_{(cid, out)} \mid cid \in \text{dom}(SP)\} \subseteq P$ and arcs $\{(t_{(T_{start}, \blacktriangleright)}, p_{(cid, in)}), (p_{(cid, in)}, t_{cid}), (t_{cid}, p_{(cid, out)}), (p_{(cid, out)}, t_{(T_{end}, \blacksquare)}) \mid cid \in \text{dom}(SP) \wedge (T_{start}, T_{end}) = SP(cid)\} \subseteq F$. Additionally, we make the transition with the elective course ID $C_{el} \in \text{dom}(SP)$ repeatable, i.e., $\{(p_{(C_{el}, out)}, t_{C_{el}}), (t_{C_{el}}, p_{(C_{el}, in)})\} \subseteq F$, e.g., C01 in Figure 4.

We obtained two study plans SP_{10} and SP_{18} from the 2010 and 2018 study regulations for the RWTH Aachen University Computer Science B.Sc. program. In Figure 4 and Figure 5, we show the workflow nets N_{10} and N_{18} constructed from the study plans SP_{10} and SP_{18} , respectively. We highlight transitions labeled with course IDs missing in either study plan or differing in start/end terms in both workflow nets, i.e., $\{t_{cid} \mid (cid \in ((\text{dom}(SP_{10}) \setminus \text{dom}(SP_{18})) \cup (\text{dom}(SP_{18}) \setminus \text{dom}(SP_{10}))) \vee (cid \in (\text{dom}(SP_{10}) \cup \text{dom}(SP_{18})) \wedge SP_{10}(cid) \neq SP_{18}(cid)))\}$. The reserved elective course ID is C01 in both workflow nets.

Deriving Labeled Partial Orders. Next, we introduce how to derive labeled partial orders from an event log L . We derive two types of labeled partial orders: life-cycle-aware and atomic. Life-cycle-aware modeling considers all exam attempts for a course ID, including failed attempts, while atomic modeling only considers the passed attempt. This allows us to differentiate between students' intended conformance in exam-taking behavior and their actual conformance based only on passed attempts.

Let $L \in \mathbb{M}(\mathcal{U}_e)$ be an event log and $mat \in \mathcal{U}_{Mat}$ be a student ID. The filtered event log L_{mat} of all exam attempts by a student with student ID mat is the multiset $L_{mat} = [e \in L \mid \pi_{Mat}(e) = mat]$. For each course ID $cid \in \{\pi_{cid}(e) \mid e \in L_{mat}\}$ in the student's event data, we calculate the start term $t_{\blacktriangleright}(L, mat, cid)$ and the end term $t_{\blacksquare}(L, mat, cid)$. The end term is the passing term, i.e., $t_{\blacksquare}(L, mat, cid) = \pi_{term}(e)$ with $e \in L_{mat} \wedge \pi_{cid}(e) = cid \wedge \pi_{pass}(e) = 1$. The start term differs between life-cycle-aware and atomic modeling: atomic modeling uses a single point in time, i.e., $t_{\blacktriangleright}(L, mat, cid) = t_{\blacksquare}(L, mat, cid)$, whereas life-cycle-aware modeling uses the term of the first exam attempt, i.e., $t_{\blacktriangleright}(L, mat, cid) = \min(\{\pi_{term}(e) \mid e \in L_{mat} \wedge \pi_{cid}(e) = cid\})$ with \min being the minimum function.

Let $N = (P, T, F, A, l)$ be a workflow net constructed from a study plan SP . A labeled partial order $lpo_{mat} = (V_{mat}, \prec_{mat}, A_{mat}, l_{mat})$ for a student ID $mat \in \mathcal{U}_{Mat}$ that agrees with the workflow net N is derived as follows. First, we ensure that the workflow net N and the labeled partial order lpo agree by sharing the same activities, i.e., $A_{mat} = A$. Next, we add a node $v_{mat, cid} \in V$ for each course ID $cid \in \{\pi_{cid}(e) \mid e \in L \wedge \pi_{mat}(e) = mat\}$ in the event log L_{mat} . During labeling, we differ between mandatory and elective courses, i.e., $\forall v_{cid} \in V: l(v_{mat, cid}) = cid$ if $cid \in (A \setminus \{C_{el}\})$ and $l(v_{mat, cid}) = C_{el}$ otherwise. Any two partial order nodes $v_{mat, cid}, v_{mat, cid'} \in V$ share an edge if one course's start term $t_{\blacktriangleright}(L, mat, cid')$ follows another course's end term $t_{\blacksquare}(L, mat, cid)$, i.e., $\forall v_{mat, cid}, v_{mat, cid'} \in V: (t_{\blacksquare}(L, mat, cid) < t_{\blacktriangleright}(L, mat, cid')) \Rightarrow ((v_{mat, cid}, v_{mat, cid'}) \in \prec)$. Next, we define a function to calculate term distances between the courses in the event log and their expected times in the study plan.

Term Distance. Let SP be a study plan, L an event log, mat a student ID, and cid a course ID. For better readability, we conclude (1) $exp(\blacktriangleright) = \pi_1(SP(cid))$ if $cid \in dom(SP)$ and $exp(\blacktriangleright) = \pi_1(SP(C_{el}))$ otherwise, and (2) $exp(\blacksquare) = \pi_2(SP(cid))$ if $cid \in dom(SP)$ and $exp(\blacksquare) = \pi_2(SP(C_{el}))$ otherwise, the expected start and end term in the study plan, respectively. The term $act = t_{\blacksquare}(L, mat, cid)$ is the actual term in which the student with the student ID mat passed a course. We define the distance function $\Delta_t(SP, L, mat, cid)$ as follows:

$$\Delta_t(SP, L, mat, cid) = \begin{cases} 0, & \text{if, } exp(\blacktriangleright) \leq act \leq exp(\blacksquare), & \text{(in time)} \\ act - exp(\blacktriangleright), & \text{if, } act < exp(\blacktriangleright), \text{ and} & \text{(early)} \\ act - exp(\blacksquare), & \text{if, } act > exp(\blacksquare). & \text{(late)} \end{cases}$$

Thus, the distance function $\Delta_t(SP, L, mat, cid)$ indicates if a course was taken on time, early, or late, showing the distance to the expected start or end

term(s). In our example, we know that all terms are semesters in the study plans and in the event log. Thus, we simplify the distance function to a year distance $\Delta_y(SP, L, mat, cid) = \left\lceil \frac{\Delta_t(SP, L, mat, cid)}{2} \right\rceil$, summarizing two terms as up to one year. Next, we align each course ID in the study plan and the labeled partial orders by their relative positions. Both are combined later to derive insights.

Alignment Order Relation. Let $\gamma(lpo, N)$ be a partial order alignment between a labeled partial order $lpo = (V, \prec, A', l')$ and a workflow net $N = (P, T, F, A, l)$. By construction, all activities in the labeled partial order lpo are contained in the workflow net N , i.e., $A' \subseteq A$, and each workflow net activity occurs at least once on the model side γ_M of the alignment γ . Thus, for each activity, or more precise course ID $cid \in \mathcal{U}_{cid}$, on the log side γ_L , we can determine its position relative to its position on the model side γ_M of the partial order alignment γ .

We distinguish four cases: (1) A course ID occurs at the same position in the alignment and therefore the student’s behavior and the study plan are aligned ($L \parallel_{cid} M$). (2) A course ID occurs first on the log side then on the model side, indicating that the course is moved forward compared to the study plan ($L \rightsquigarrow_{cid} M$). (3) A course ID occurs first on the model side and then on the log side, indicating that the course is moved back compared to the study plan ($M \rightsquigarrow_{cid} L$). (4) A missing entry on the log side ($\not\parallel_{cid}$).

Formally, for $i \in [1, |\gamma|]$ a position in the alignment where the i -th entry on the model side is a course ID $cid \in \mathcal{U}_{cid}$, i.e., $\gamma_M(i) = cid$, we define:

- $L \parallel_{cid} M$, if $\gamma_L(i) = cid$, (Synchronous)
- $L \rightsquigarrow_{cid} M$, if $\exists_{j \in [1, i-1]}: \gamma_L(j) = cid$, (Student before Plan)
- $M \rightsquigarrow_{cid} L$, if $\exists_{j \in [i+1, |\gamma|]}: \gamma_L(j) = cid$, and (Student after Plan)
- $\not\parallel_{cid}$, if $\forall_{j \in [1, |\gamma|]}: \gamma_L(j) \neq cid$. (Missing in Log)

Next, we apply our approach to a real-life data set to demonstrate its applicability and gain insights.

5 Evaluation

We conducted the evaluation using PM4Py ⁵, an open-source Python library for process mining. For evaluation, we used a real-life data from three German universities collaborating in the BMBF-funded joint research project “AIStudy-Buddy”. As part of the project, two software tools are being developed to assist students in organizing their studies and study planners in analyzing study programs. The data includes 1,162 RWTH Aachen University Computer Science Bachelor’s graduates who began earliest in 2010 and have at least 150 ECTS. As mentioned in Section 4, in Figure 4 and Figure 5 the corresponding plans for 2010 and 2018 are modeled as workflow nets N_{10} and N_{18} . For each student from the event log, a labeled partial order is created using the life-cycle-aware and atomic approach. The code for modeling study plans and translating educational event logs is publicly available on GitLab⁶. In addition, the published

⁵ <https://github.com/pm4py/pm4py-core>

⁶ <https://git.rwth-aachen.de/christian.rennert/po-based-SP-eval>

Table 5: Distance in years for the 2010 study plan and life-cycle-aware student courses.

Δ years	Student before Plan (L \rightsquigarrow_{cid} M)						Synchronous (L \parallel_{cid} M)						Student after Plan (M \rightsquigarrow_{cid} L)						Missing in Log (\mathbb{I}_{cid})
	≤ -2	-1	0	1	2	≥ 3	≤ -2	-1	0	1	2	≥ 3	≤ -2	-1	0	1	2	≥ 3	
C05	0	9	29	1	1	0	1	2	570	27	5	2	0	0	0	11	3	2	0
C12	13	0	1	0	0	0	2	16	493	100	25	6	0	0	3	4	0	0	0
C13	0	2	4	5	0	0	2	20	428	138	34	20	0	0	0	0	0	0	10
C16	0	0	0	0	0	0	4	7	530	13	3	1	0	0	0	60	22	23	0
C17	77	21	3	2	0	1	0	119	111	65	10	4	0	0	0	0	0	0	250
C18	0	3	3	0	1	0	1	2	456	21	7	1	0	0	1	69	57	41	0

Table 6: Distance in years for the 2018 study plan and life-cycle-aware student courses.

Δ years	Student before Plan (L \rightsquigarrow_{cid} M)						Synchronous (L \parallel_{cid} M)						Student after Plan (M \rightsquigarrow_{cid} L)						Missing in Log (\mathbb{I}_{cid})
	≤ -2	-1	0	1	2	≥ 3	≤ -2	-1	0	1	2	≥ 3	≤ -2	-1	0	1	2	≥ 3	
C05	2	3	25	2	1	0	1	6	371	46	9	1	0	0	3	19	6	4	0
C12	7	54	32	8	7	2	1	52	260	53	16	7	0	0	0	0	0	0	0
C13	0	0	0	0	0	0	0	1	250	160	51	33	0	0	0	0	0	0	4
C16	0	0	0	0	0	0	6	5	326	16	2	0	0	0	1	75	44	24	0
C17	3	3	2	0	0	0	1	14	262	50	13	11	0	0	0	0	0	0	140
C18	0	4	3	1	0	0	0	6	282	37	5	4	0	1	1	87	30	38	0

code includes mock student data and the two study plans used in this work. We computed alignments for both workflow nets using both, the atomic or life-cycle-aware labeled partial orders, and selected the study plan with the lower alignment cost for further result aggregation. We determined the expected study plan based on whether a student started before or from 2018 onward. Table 4 shows a confusion matrix for the expected and the most conforming study plan.

Of the 860 students, 237, or around 27.5%, expected to follow the 2010 study plan conform most to the 2018 plan. In contrast, only a fraction of around 13.2% of those expected to follow the 2018 plan conform most to the 2010 plan. An explanation may be that in an educational setting, students may adopt to a new study plan within their studies. It is to note, that the confusion matrix does not alter between the life-cycle-aware and atomic partial order modeling approach.

The alignment of the most conforming study plan is used to compute the alignment order relation and year distance for each course. We count the frequency of each combination for the two study plans. Tables 5 and 6 show the results for both study plans, respectively. Complete and further tables are available on Gitlab⁷, showing all distances in terms and years and all courses being included as both life-cycle-aware and atomic. Although present, the difference between atomic and life-cycle-based modeling is marginal in our results. However, we expect that this may vary when future experiments are applied, and therefore requires further investigation. In the following, we focus only on life-cycle-aware modeling.

Expected	Most conforming		
	2010	2018	Σ
2010	623	237	860
2018	40	262	302
Σ	663	499	1162

Table 4: Confusion matrix between expected study plans and most conforming study plans.

⁷ See Footnote 6.

Next, we analyze the results to derive some potentially interesting insights for study planners. We select representatives from two groups of courses represented in both the 2010 and 2018 study plans: (1) Courses moved in the 2018 study plan compared to 2010, i.e., courses with course IDs C12, C13, and C17, and (2) courses not moved between the 2010 and 2018 study plans, i.e., courses with course IDs C05, C16, and C18.

For the former, we use our method to check if the changes are supported from results of the original 2010 study plan and if conformance improves. Therefore, we detail every change, describe the evidence, and evaluate the change in conformance. In the 2010 study plan, Course C12 is taken in the fourth to fifth term, but in 2018, it shifts to the fifth to sixth term. We do not identify direct results supporting this change. After the change, instances of students taking a course earlier than its expected position increased from 14 to 110. This holds even in relative counts, with more students conforming to the 2010 study plan than to the 2018 study plan. This suggests the change may need more time to take effect or may be impractical for students. For Course C17, the opposite holds. In the 2010 plan, it is expected in the sixth term only, but in the 2018 plan, in the second to fifth term. There is evidence of students taking the course earlier than planned and typically over 2 years ahead. Additionally, more events are missing in the logs for the 2010 study plan than for the 2018 study plan, strengthening the finding. Thus, we expect the change to positively impact conformance, as shown in the 2018 study plan results. Course C13 moves from the fifth and sixth terms to the third to fifth terms. There is minimal evidence of students taking the course early before the change and none after. In conclusion, we expect our approach to be usable for detecting and evaluating potential changes effectively using the available evidence.

For the second group, we investigate the general misalignment quota and its stability after other course changes. The misalignment quota may assess the course conformance as shown in the following. Courses C16 and C18, preliminary courses in the first and third term, are often delayed due to their difficulty. Life-cycle-aware course modeling together with “student after plan” counts shows individual delays of first exam attempts, i.e., for course C16, 75 students postpone by 1 year, 44 by two years, and 24 by three years. This information is relevant to study planners because preliminary courses contain essential knowledge for other courses and may lead to a longer study duration, which planners aim to avoid. We expect this method to provide study planners with a measurement of the frequency and role of delayed courses, serving as a basis for countermeasures like improved mentoring. Toward stability, the relative frequency of students taking courses C16 and C18 has increased, from 105 out of 558 to 143 out of 355 students. This may highlight the increasing importance of individual study planners’ countermeasures and how to find them.

Future Work. We plan to apply the approach to more study plans and event data to further investigate the insights from the proposed method. In addition, an adaptation is to cluster students by key performance indices such as GPA or study duration to explore their relation to (non)conforming behavior. Finally,

we plan to add the relative alignment positions to the event data to identify frequent itemsets of mismatched courses.

6 Conclusion

This work contributes to the field of educational process mining by combining conformance checking methods to account for the relation between course order and the difference between expected and actual course passing terms. We model study plans with workflow nets and student course-taking behavior with labeled partial orders, providing a basis for further experiments in educational process mining. Partial order alignments are used to compare expected and actual course orders. Using real-life data, initial experiments show our contribution's usability and may suggest valuable insights for study planners.

Acknowledgments. We thank the Alexander von Humboldt (AvH) Stiftung for supporting our research (grant no. 1191945). The authors gratefully acknowledge the financial support by the Federal Ministry of Education and Research (BMBF) for the joint project AIStudyBuddy (grant no. 16DHBKI016). Further, we thank all reviewers for their valuable feedback



References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016)
2. Adriansyah, A.: Aligning observed and modeled behavior. Phd thesis 1 (research tu/e / graduation tu/e), Mathematics and Computer Science (2014)
3. Bendatu, L.Y., Yahya, B.N.: Sequence matching analysis for curriculum development. *Jurnal Teknik Industri* **17** (2015)
4. Bogarín, A., Cerezo, R., Romero, C.: A survey on educational process mining. *WIREs Data Mining Knowl. Discov.* **8** (2018)
5. Diamantini, C., Genga, L., Mircoli, A., Potena, D., Zannone, N.: Understanding the stumbling blocks of italian higher education system: A process mining approach. *Expert Syst. Appl.* **242** (2024)
6. Hobeck, R., Pufahl, L., Weber, I.: Process mining on curriculum-based study data: A case study at a german university. In: *Process Mining Workshops - ICPM 2022*. LNBIP, vol. 468. Springer (2022)
7. Lu, X., Fahland, D., Van Der Aalst, W.M.P.: Conformance checking based on partially ordered event data. In: *BPM Workshops*, vol. 202. Springer International Publishing, Cham (2015)
8. Penthin, M., Fritzsche, E.S., Kröner, S.: Gründe für die überschreitung der regelstudienzeit aus studierendensicht (Reasons for exceeding the standard period of study from a student perspective). *Beiträge zur Hochschulforschung* **39**(2) (2017)
9. Priyambada, S.A., ER, M., Yahya, B.N., Usagawa, T.: Profile-based cluster evolution analysis: Identification of migration patterns for understanding student learning behavior. *IEEE Access* **9** (2021)
10. Rafiei, M., Bayrak, D., Pourbafrani, M., Park, G., Helal, H., Lakemeyer, G., van der Aalst, W.M.P.: Extracting rules from event data for study planning. In: *Process Mining Workshops - ICPM 2023*. LNBIP, vol. 503. Springer (2023)