Enhancing BPMN models with ethical considerations

Beatrice Amico, Carlo Combi, Anna Dalla Vecchia, Sara Migliorini, Barbara Oliboni, and Elisa Quintarelli

Department of Computer Science - University of Verona
Strada Le Grazie, 15 – 37035 Verona (Italy)
{name.surname}@univr.it

Abstract. Fairness has recently emerged as a challenging topic in many areas of computer science, as it is related to algorithms supporting decision-making, experimental research, and information access and processing. As (decision-intensive) business processes are inherently using information to reach their goals, their fairness possibly depends on the kind of information they are allowed to access. In this paper, we elaborate on this aspect and propose some criteria to consider when conceptually specifying business activities and their related information seamlessly through a recently proposed approach based on the concept of Activity View. More specifically, we distinguish equality and equity as two aspects of fairness and discuss how to enforce them in business process design. Their expression according to the specification of Activity Views is formally proposed and discussed in the paper.

1 Introduction

Ethics in data management has become a challenging topic in recent years [17] since it is essential to ensure the responsible treatment of information. Fairness is becoming of interest in many different computer science research areas: for example, fairness is required (i) when we design decision-support systems, where algorithms have to be fair in their design, (ii) when we perform experimental research, as the derived results have to be explicitly related to some possible bias that could limit their generality, (iii) when data access authorizations have to be granted for a decision-making task, and (iv) when user assignments have to be managed in process-oriented systems where some intertwined activities may create conflicts of interest [2, 8, 9, 15, 17].

In this paper, we will focus on the last two aspects of fairness, which are related to the context of process-aware information systems (PAISs). Indeed, information management is required, directly or indirectly, by all organizational activities. PAISs explicitly deal with business processes that use, produce, and manipulate organizational data stored in databases. Business processes, often represented through BPMN (Business Process Model and Notation) [13], and data are intertwined [16], and each of them plays a crucial role in PAISs.

Integrating ethical considerations in business processes is, thus, important to promote trust, fairness, satisfaction, and sustainability and thus to realize the organization's long-term success thanks to a positive reputation. More specifically, in this paper, we will consider the fairness-related concepts of *equality* and *equity* in performing both single tasks and complete (data-intensive) business processes. To this end, a suitable existing methodology proposed in [4, 5] will be considered, as it allows the design of BPMN models, together with access to data, in the context of PAISs.

This paper highlights the importance of integrating fairness in PAISs where data-intensive business process models must be suitably designed, considering fairness criteria instead of revealing unfair policies with successive data analysis. The main original contributions of the paper can be summarized as follows.

- We extend a recent formal model for the conceptual design of data-intensive business processes [5], to integrate fairness requirements in process modeling.
- We deal with different aspects of fairness by highlighting its connection to both the considered activity and the data required for the activity. Such an aspect, to the best of our knowledge, has not been considered in other research contributions.
- We introduce fairness, particularly equity and equality aspects, both for specific activities and for an overall execution path, where fairness may depend on many different intertwined activities.

The structure of the paper is as follows. Sect. 2 discusses some related research directions. Sect. 3 provides a motivating example taken from the university domain and discusses some fairness requirements. Sect. 4 contains the proposal of a new formal model, where activity views are extended to represent different kinds of fairness requirements. In Sect. 5, we discuss how to evaluate some fairness properties of data-intensive business processes, while in Sect. 6, we sketch some concluding remarks.

2 Related Work

This section investigates how fairness has been applied in the literature in different fields, capturing some of the issues and solutions addressed. Although the debate on ethical topics has received growing attention in recent years, especially with the widespread use of machine learning and artificial intelligence techniques in everyday life, there is still a lack of consensus on ethical guidelines [17]. However, a widely accepted factor is that ethical rules depend on the specific circumstances in which they are applied, highlighting their context-sensitive nature.

Regarding the state of the art on ethical issues related to machine learning (ML), in [2], the authors outline different approaches proposed in the literature to enhance fairness in ML, highlighting existing methodologies to avoid possible ethical biases and inequities. They conclude by presenting five dilemmas for future research. It confirms that, although the problem has been explored for years, it is not easy to delineate in a uniform way the concept of fairness [7, 12]. The authors in [9] examine the definition of fairness as the absence of discrimination for individuals with the same "merit" and fairness in algorithms as the absence of discrimination. However, they point out three weaknesses of this definition: disparities justified by "merit", the limitation to the algorithm, and the ignoring of the disparities within groups. Furthermore, in [11], the authors survey the presence of bias in various real-world applications and define a taxonomy for the definition of fairness in artificial intelligence (AI) systems. Specifically, they identify two primary sources of unfairness in ML outcomes (i.e., data and algorithms). In [6], the authors discuss how data bias should be managed. From their point of view, it is not always necessary to completely remove the bias. Otherwise, this process may lead to other types of bias. A possible solution could be to provide the users with a tool that allows them to adjust existing biases, enabling them to leverage the benefits of fairness for certain tasks. According to the emerging evidence that ML algorithms can make discriminatory decisions, researchers have been investigating computational techniques that make ML algorithms unbiased and non-discriminatory. Fairness focusing on distributive justice has been a central research topic in computer theory, artificial intelligence, and machine learning. As already highlighted in [6], the authors in [10] propose a procedural justice framework for algorithmic decision-making, which explains algorithmic assumptions and properties displaying inputs and outcomes, allowing interactively adjusting the outcome.

An additional interesting facet of the fairness concept is its relationship with transparency/explanation in AI-assisted decision-making, an issue that numerous studies have emphasized. AI-assisted decision-making that affects individuals brings up essential issues related to transparency and fairness in AI. In [1], the authors extensively analyze this relationship, observing that, according to their experiments, AI explanations increased user trust in AI-informed decision-making, and different explanation types did not show differences in affecting user trust. Furthermore, they explain that AI explanations increased users' perceptions of fairness. Another aspect related to fairness present in literature is the fairness in ranking. In the past few years, research communities have worked a lot on incorporating fairness requirements into algorithmic rankers. They focused on data management, algorithms, information retrieval, and recommender systems. In [18, 19], the authors extensively overview the state-of-the-art literature on fair ranking in score-based and supervised learning-based ranking domains. They present a selection of approaches that were developed in several fields.

To the best of our knowledge, the concept of fairness has not been completely investigated in the context of business processes. An initial example of how the concept of fairness can be integrated into the Business Process Modeling Notation (BPMN) [13] is presented in [15]. The authors propose a BPMN-based framework that takes into account different aspects: (i) the design of business processes considering security, data-minimization and fairness requirements; (ii) the encoding of such requirements as reusable, domain-specific pattern; (iii) the checking of alignment between the encoded requirements and annotated BPMN models based on these patterns; (iv) the detection of conflicts between the specified requirements in the BPMN models based on a catalog of domain-independent anti-patterns. They specify the security requirements, data minimization, and fairness in BPMN models, using existing security annotations from the *SecBPMN2* modeling language and introducing new data minimization and fairness annotations. This extension facilitates the alignment checking of security, data minimization, and fairness requirements with their specifications in BPMN models. The process is automated by extending a graphical query language, *SecBPMN2-Q*, formulating the requirements as reusable procedural patterns that can be matched to BPMN models. Additionally, considering different pairs of requirements in BPMN models, they propose an automated conflict detection technique that uses encoded knowledge about conflicts and potential conflicts between the requirements. They do not explicitly consider the relationship between the activities and the accessed data, and for this reason, in this paper, we try to overcome this limitation.

3 Running Example

This section proposes a running example related to the university domain. In particular, we consider activities associated with managing various student career cases, such as student enrollments, exams, graduation exams, and scholarships. Fig. 1 shows a simplified process model related to the considered scenario and represented by using the Business Process Model and Notation (BPMN) [13]. We will give a formal definition of Process Model in Def. 1 of Sect. 4.

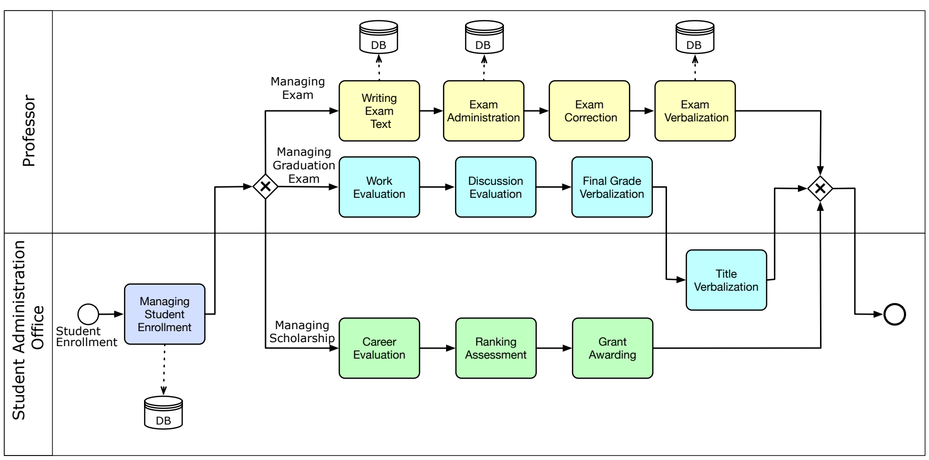

Fig. 1. A simplified business process model representing some activities associated with the management of students' careers.

In BPMN, a process is defined as a sequence of *activities* and *events*, connected by *sequence flows*, defining their execution order. *Gateways* allow to split the sequence flow into multiple paths and merge them, thus realizing routing. These BPMN elements can be briefly described as follows.

- **Activities** identify work performed within the process. Activities can be either *tasks* (atomic units of work) or *subprocesses* (compound activities). Graphically, tasks are depicted as rectangles with rounded corners and labels specifying their names. An example of *task* in Fig. 1 is "Managing Student Enrollment".
- **Events** represent instantaneous facts that impact the sequencing or timing of process activities. They are depicted as circles, which may contain a marker to diversify the kind of event trigger. *Events* can be *start events*, *intermediate events*, and *end events*. In Fig. 1, the process starts with the "Student Enrollment" *start event* and finishes with an *end event*.
- **Gateways** allow controlling the divergence and convergence of the sequence flow according to data-driven conditions or event occurrence. Graphically, they are represented as diamonds with an internal marker that differentiates their routing behavior. In Fig. 1, the ✕ split gateway represents a data-driven exclusive gateway, i.e., the point in the process where a condition must be evaluated to choose one path out of more. In this example, each path is related to managing a different aspect, such as "Managing Exam", "Managing Graduation Exam", and "Managing Scholarship".

BPMN also provides elements for representing data involved in the process and process participants.

- **Data Stores** are visualized as a database symbol and represent data accessed by activities. Some process activities in Fig. 1 are related to a database named "DB" for representing the operations performed on the database.
- **Pools and Lanes** allow the representation of resources (people/roles) involved in the process. The process participants in Fig. 1 are represented through the two lanes "Professor" and "Student Administration Office".

The process in Fig. 1 starts when the student applies for enrollment. The first activity, "Managing Student Enrollment," is carried out by the Student Administration Office and requires access to the "DB" database to insert student information[1]. After the student enrollment, the process presents three exclusive paths, each dedicated to managing a different phase in the student's life cycle. For Managing Exam (yellow path), the professor performs the following activities: "Writing Exam Test", "Exam Administration", "Exam Correction", and "Exam Verbalization". When Writing an Exam Test, the professor needs to access the database to retrieve data about the number of students involved in the examination and the possible special needs of students with disability, if any. During the Exam Administration, the Professor accesses the database to have the list of students with special needs at her disposal, in order to properly

[1] For sake of simplicity, in the following we will focus only on the data accessed for the exam management – yellow path.

identify the most suitable exam conditions and duration. Finally, the professor corrects the exam and records the obtained results in the database. Thus, the "Exam correction" activity does not require any access to the data store, while the "Exam Verbalization" activity is connected to the data store to represent the writing operations.

Among the different fairness-related issues, we focus on ensuring an ethical treatment of data accessed while performing activities. More precisely, we need to consider different aspects: (i) the portion of accessed data, (ii) the user who accesses data, and (iii) the ethical dimension to satisfy. Moreover, the ethical treatment of data inside a process model can be evaluated with respect to: a set of activities, such as those belonging to a given path (path evaluation), or all the activities belonging to the whole process (global evaluation). In our process model, for example, data accessed during the "Writing Exam Test" activity can differ according to the considered ethical dimension. When considering *equality*, the professor needs to know only the number of students involved in the examination, while when considering *equity*, the professor also needs to know the possible presence of students with disability, thus acquiring some sensitive information.

To describe in more detail the data accessed by each single activity, we can rely on a recent extension of the BPMN model called Activity View [4, 5]. This formalism allows the connection of a conceptual representation of a process model to a portion of a database schema by detailing the operations performed by a process activity on the database. For this purpose, the following section introduces an extension of the Activity View model, which allows the definition of ethical properties in a process model.

4 BPMN and Fairness-driven Activity View

Starting from the well-known concept of *Activity View* [3], this section formalizes the proposed extension, which allows the integration of fairness in process design.

4.1 Describing Process and Data by Means of Activity View

Business process activities need to access and manage data stored in databases. The connection between processes and data is usually handled at the implementation level and is often left implicit at the conceptual level. However, modeling processes and data at the conceptual level supports improving business process models and identifying requirements for data management [4]. For this reason, this paper assumes a formal definition of the process model, as shown below, which combines both control-flow and data-flow aspects and explicitly defines the relationship between the activities and the accessed data.

Definition 1 (Process Model).
A process model m is a tuple $m = \langle N, C, DN, F \rangle$ consisting of:

- *a finite non-empty set of flow nodes N.*
  The set $N = Ac \cup G \cup E$ of flow nodes consists of the disjoint sets:
  - *Ac set of activities;*
  - *G set of gateways.*
    The set $G = G_s^x \cup G_m^x \cup G_s^a \cup G_m^a$ is partitioned according to the routing behavior of its nodes into the disjoint sets G_s^x of xor split nodes, G_m^x of xor merge nodes, G_s^a of and split nodes, and G_m^a of and merge nodes, respectively.
  - *$E = \{s, e\}$ of start and end events.*
- *a finite non-empty set C of control flow edges.*
  The control flow $C \subseteq N \times N$ connects the elements of N. Given a flow node $n \in N, \bullet n \subseteq N$ ($n \bullet \subseteq N$) denotes the set of direct predecessor (successor) nodes of n.
- *a finite set DN of data nodes.*
  $DN = DO \cup DS$ is the set of data nodes, consisting of the disjoint sets DO of data objects, i.e., volatile pieces of information exchanged between activities, and DS of data stores, i.e., persistent data sources such as enterprise databases.
- *a finite set F of data associations.*
  $F \subseteq (DN \times Ac) \cup (Ac \times DN)$ is the data flow that connects data nodes with activities.

The remainder of this paper will concentrate on the subset of the data nodes represented by the data stores DS. In particular, we concentrate on a representation of DS by means of a relation model since it allows us to refer to a set of abstract common operations on data, like projections (π), selections (σ), and joins ($\bowtie$), and write in a compact way the views each activity has to access to.

Definition 2 (Database schema and instance).
A database schema $\mathcal{DS}$ is a set of relation schemes:

$$\mathcal{DS} = \{R_i(X_i)\}_{i=1}^n = \{R_i(A_{1_i}, A_{2_i}, \ldots, A_{m_i})\}_{i=1}^n$$

where R_i is a relation name and $X_i = (A_{1_i}, A_{2_i}, \ldots, A_{m_i})$ is a list of attributes.

Given a relation schema $R_i(X_i)$, where $X_i = (A_{1_i}, A_{2_i}, \ldots, A_{m_i})$, a relation instance r_i is a set of m_i-tuples $r_i = \{t_1, \ldots, t_{k_i}\}$ each one defined on X. Similarly, the instance $\mathcal{DI}$ of $\mathcal{DS}$ is the set of instances r_i of the relation schemata belonging to $\mathcal{DS}$.

Given two database schemata $\mathcal{DS}_1$ and $\mathcal{DS}_2$, we need to define the concept of containment $\subseteq$ between them to identify the fact that a database schema (or portion of it) extends another one by revealing some additional information. Notice that this notion is particularly useful when database schemata are used to describe different portions of a relational database accessed by a process activity.

Definition 3 (Database schema containment).
Given two database schemata $\mathcal{DS}_1 = \{R_i(X_i)\}_{i=1}^n$ and $\mathcal{DS}_2 = \{P_j(Y_j)\}_{j=1}^m$, we say that $\mathcal{DS}_1 \subseteq \mathcal{DS}_2$ iff $\forall i \in [1..n]\ \exists j \in [1..m]\ R_i(X_i) \subseteq P_j(Y_j)$, where $R_i(X_i) \subseteq P_j(Y_j)$ means that $R_i = P_j \wedge X_i \subseteq Y_j$.

From this definition, we say for instance that $\{\pi_{\mathsf{CourseID,studentID}}(\mathsf{EXAM})\} \subseteq \{\pi_{\mathsf{CourseID,studentID,Date}}(\mathsf{EXAM})\}$, since in the second database schema there is an additional attribute. Another example of containment is

$$\{\pi_{\mathsf{CourseID,studentID,Date}}(\mathsf{EXAM})\} \subseteq \{\pi_{\mathsf{CourseID,studentID,Date}}(\mathsf{EXAM}), \pi_{\mathsf{sID,requirement}}(\mathsf{SDISABILITY} \bowtie_{\mathsf{sID=studentID}} \mathsf{EXAM})\}$$

since the latter schema contains an additional relation with respect to the former.

Given the generic notion of the database schema and the containment relation defined on it, we can introduce the concept of *Activity View* [4], which provides a formal representation of the operations performed by a process on a database. In particular, an Activity View describes which database subsets (or portion of the schema) are accessed by a particular process activity and which data operations are performed on them.

Definition 4 (Activity View).
Let $m = (N, C, DN, F)$ be a process model and $\mathcal{DS} = \{R_i(X_i)\}_{i=1}^n$ a database schema with its instance $\mathcal{DI}$, representing a data store $ds \in DS \subseteq DN$ inside m. An Activity View $av_{ac} = \{t_1, \ldots, t_m\}$ *of an activity $ac \in Ac \subseteq N$, such that ac is connected to a data store according to F, is a set[2] of tuples $t_1, \ldots, t_m$, where each tuple t_k denotes a particular data access operation performed by ac on data in the given database instance $\mathcal{DI}$. Formally, each tuple of the Activity View has the form*

$$t_i = \langle Q_i, AccessType_i, AccessTime_i \rangle$$

where:

- $Q_i = \{q_1, \ldots, q_j\} \subseteq \mathcal{DI}$ *is the set of relational algebra expressions specifying the data ac needs to access. In this paper, we will consider only projections and joins, as the main focus in on the attributes accessed by different activities. $Att(Q_i)$ is the overall set of attributes appearing in the relational algebra expressions of Q_i.*
- $AccessType_i \in \{\mathrm{R, I, D, U}\}$ *defines the type of access to the related information.* R *denotes a* read *of elements of $\mathcal{DI}$, whereas* I, D, *and* U *denote an* insertion, *a* deletion, *and an* update *operation, respectively.*
- $AccessTime_i \in \{\text{start, during, end}\}$ *denotes when a data operation is performed w.r.t. the activity execution.*

This paper aims to elaborate more on the notion of Activity View to represent and identify different fairness requirements for managing data [8]. More specifically, we concentrate on two main principles "equality" and "equity".

[2] We represent data access operations as a set as the same activity can imply the execution of different queries in many possible orders.

4.2 Introducing Fairness inside Activity View

The previous section introduces the concept of Activity View to describe how the activities inside a process model access data. In this section, we provide a step forward by discussing how two of the main fairness principles, equality and equity, can be incorporated into a process model for ethical data access and management.

Equality is defined in literature as "the state or quality of being equal". This means providing everyone with the same opportunities. For instance, when you assign offices to two new PhD students and equip them equally, you are practicing equality. However, this does not necessarily mean you are being fair, as this behavior disregards their individual needs and differences. Consider if one of the students has a physical disability that prevents them from sitting at a desk all day. In this case, their office setup does not meet their specific needs. To this purpose, equity is more appropriate as fairness behavior in this case.

Equity means "the quality of being fair or impartial". It involves recognizing that people face different circumstances and adjusting to ensure everyone has the same opportunities. Regarding the example above, the benefits of diversity in the workplace are numerous, making fairness and justice essential considerations from the early stages of process design.

As already observed in Sect. 3, equity implies that some sensitive data need to be known to understand specific contexts and situations. To suitably deal with sensitive data, we introduce the concept of sensitivity-aware database schema.

Definition 5 (Sensitivity-aware database schema and instance). *A sensitivity-aware database schema $\mathcal{S}$ is a tuple $\langle \mathcal{DS}, \mathcal{SA} \rangle$, where:*

- *$\mathcal{DS}$ is a database schema defined as in Def. 2, namely, a set of relation schemes $\mathcal{DS} = \{R_i(X_i)\}_{i=1}^{n}$, where R_i is a relation name and $X_i = (A_{1_i}, A_{2_i}, \ldots, A_{m_i})$ is a list of attributes;*
- *$\mathcal{SA} \subseteq \bigcup_{i=1}^{n} X_i$ is the set of sensitive attributes inside the database schema.*

We say that a relation schema $R_j(X_j)$ is a sensitive relation schema if at least one attribute of its schema is sensitive, i.e. $X_j \cap \mathcal{SA} \neq \emptyset$.

A sensitive relation instance is an instance of a sensitive relation schema. Similarly, a sensitive database instance $\mathcal{DI}$ of a sensitivity-aware database schema $\mathcal{S}$ is the set of (possibly sensitive) instances r_i of some $R_i \in \mathcal{DS}$.

Tab. 1 reports a fragment of a relational database related to the university domain where sensitive attributes are represented in boldface.

A process model can realize fairness in different ways. For the purpose of this paper, we concentrate on the fact that each activity can access all and only the information needed to achieve one of the main principles introduced above.

Definition 6 (Fairness-driven Activity View). *Let $m = (N, C, DN, F)$ be a process model and a sensitivity-aware database schema $\mathcal{S} = \langle \mathcal{DS}, \mathcal{SA} \rangle$ with its sensitive instance $\mathcal{DI}$, representing a data store*

STUDENT(sID, **Surname**, **Name**, **DateOfBirth**, **Gender**, **Citizenship**, **Revenue**, **Working**, ...)

APPLICATION_CANDIDATE(sID, AnonymizedCV)

SDISABILITY(**sID, D_code**, **requirement**) *D_code=Disability code

SJOB(**sID**, **type**, **full-time**, **timeslot**, . . .)

PROFESSOR(pID, **Surname**, **Name**, **Dateofbirth**, **Gender**, **Citizenship**, **Role**, Sector, ...)

PDISABILITY(**pID**, **D_code**, **requirement**)

HEALTHRECORD(**pID, I_code**, **requirement**) *I_code=Illness code

EXAM(CourseID, studentID, Date, Time, Room)

COURSE(CourseID, AYear, pID)

EXAMRECORD(StudID, CourseID, Date, Mark)

CAREERRECORD(StudID, **Avgmark**, **Internship**, . . .)

GRADUATION(StudID, **Careermark**, ThesisMark, FinalMark)

. . .

Table 1. A simple Relational Database schema related to the management of students' careers. Underlined attributes represent primary keys, while the asterisk symbol denotes an enumerated domain. Bold attributes represent possibly sensitive information.

$ds \in DS \subseteq DN$ inside m. Given an activity $ac \in Ac \subseteq N$, the Fairness-driven Activity View *eav_{ac} of ac is a set composed of at least one of the following activity views:*

- *$av^{=}_{ac} = \{t_1, \ldots, t_m\}$, which denotes the activity view related to ac when the Equality principle need to be implemented,*
- *$av^{\doteq}_{ac} = \{t'_1, \ldots, t'_m\}$, which denotes the activity view related to ac when the Equity principle need to be implemented.*

We will denote as $av^{=}_{ac}[\mathcal{S}]$ and $av^{\doteq}_{ac}[\mathcal{S}]$ the set $\bigcup_{i\in\{1,\ldots,m\}} Att(Q_i)$, i.e., the schema attributes of $\mathcal{S}$ each tuple of an activity view needs to access, according to the equality- and equity-fairness principles, respectively. Moreover, notations $av^{=}_{ac}[\![\mathcal{S}]\!] = av^{=}_{ac}[\mathcal{S}] \cap \mathcal{SA}$ and $av^{\doteq}_{ac}[\![\mathcal{S}]\!] = av^{\doteq}_{ac}[\mathcal{S}] \cap \mathcal{SA}$ are introduced to identify the attributes containing sensitive information an activity view needs to access.

Any equality-related activity view $av^{=}_{ac}$ cannot contain sensitive attributes, while an equity-related one $av^{\doteq}_{ac}$ needs to have at least one sensitive attribute, which allows the distinction of different cases equity criteria have to consider. More formally, it always holds $av^{=}_{ac}[\![\mathcal{S}]\!] = \emptyset$ and $av^{\doteq}_{ac}[\![\mathcal{S}]\!] \neq \emptyset$.

It is essential to highlight that when the process is executed, and a certain activity is associated with more than one activity view because more fairness principles can be implemented for that activity, only one will be selected during the execution, depending on the specific application needs.

Referring back to the example introduced in Sect. 3, Fig. 2 enriches the process in Fig. 1 with the Fairness-driven Activity View related to both the equality and the equity, taking the schema in Tab. 1 as a reference data store. In particular, let us consider the "Writing Exam Text" (WET) activity; in this case, the professor can need to access two distinct portions of the database schema

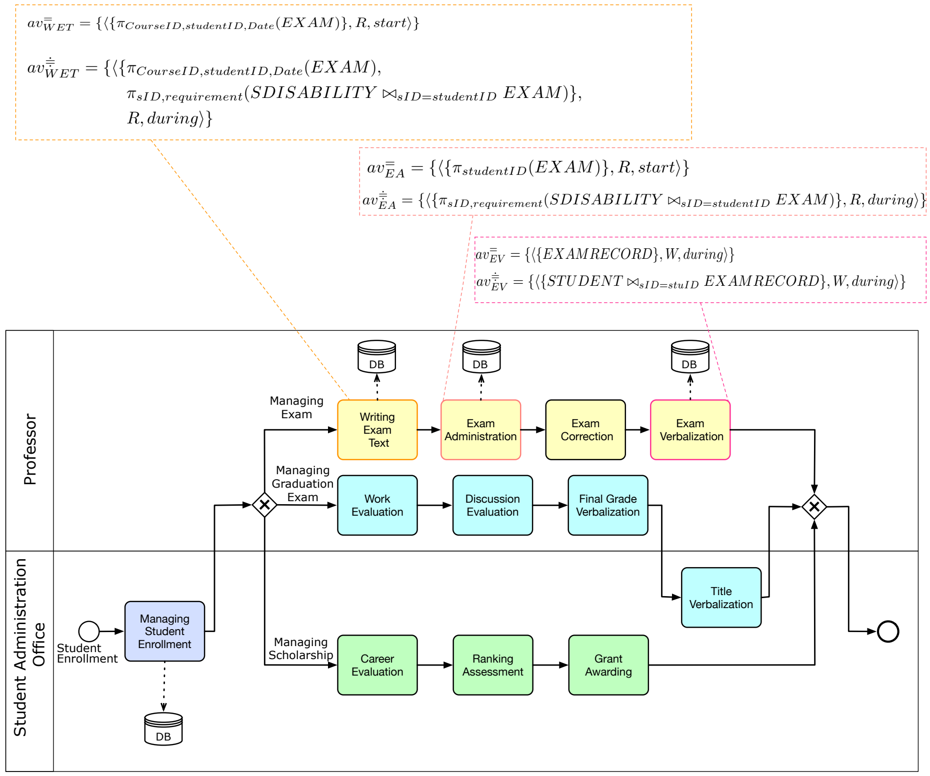

Fig. 2. The business process model for the management of students' careers completed with its Fairness-driven Activity View.

depending on the fairness principle we want to implement. In the case of equality $av^{=}_{ac}$, the exam paper will be the same for all the students and thus, it is enough to access information for the exam organization, such as the course and student identifier and the date of the session, to count the number of students who will be present during the exam. Conversely, if the equity principle is taken into consideration, the professors need to access some additional and sensitive information about the student, like his/her possible disabilities, to accommodate specific and tailored needs. This kind of information can also be useful during the "Exam Administration" (EA) since also in this case, the special needs of some students need to be carefully considered. The activity "Exam Correction" (EC) does not have any access to the data store since it does not need any additional information. Finally, "Exam Verbalization" (EV) can be performed with respect to both ethical principles. Thus, according to the proposed formalization, stakeholders may specify during the conceptual design of data-intensive process models different fairness-compliant data accesses through *fairness-driven activity views*.

5 Evaluating Fairness Properties

The proposed extension of the Activity View allows the designers to explicitly specify two different fairness features –equality and equity– for an activity with respect to the information it needs to access.

According to the fairness-driven characterization of activity views, we are now able to specify and evaluate some ethical properties at different resolutions, with the scope ranging from a single activity to the entire process.

Starting from the example in Fig. 2, we can easily envision a first property.

Property 1 (Activity Fairness). Given an activity ac, it holds that:

$$av^{=}_{ac}[\mathcal{S}] \subseteq av^{\doteqdot}_{ac}[\mathcal{S}].$$

The rationale under the property is that if an activity can be implemented following more than one fairness principle, then equity needs access to a broader set of data, i.e., additional attributes, some of them being sensitive (e.g., presence of disability) or additional relations, with respect to equality [14]. For example, in Fig. 2, activity "Writing Exam Test" needs to access only a projection of the EXAM relation when considering equality, and in addition to that, also to a projection of the join of EXAM and SDISABILITY when considering equity.

The notion of fairness can be extended from a single activity to an entire path inside a process model. More precisely, we have to consider, for each path, those activities assigned to the same actor (i.e., in the same lane). While, in principle, for equality, any equality-related activity view may be independent of the other ones, even when related to activities executed by the same actor, specific attention has to be paid to two different aspects:

- how equality-related activity views are related to equity-related ones associated to activities performed by the same actor;
- how equity-related activity views are related to other equity-related activity views for activities performed by the same actor.

As for the first aspect, the following property holds.

Property 2 (Path Equality Fairness). A path of a process model m, given by the sequence of activities inside the same lane, is equality fair if for any activity $a \in Ac$ such that there exists $av^{=}_{a}[\mathcal{S}]$ or a does not access any data store, it holds $\neg\exists b \in \circ\, a\ (av^{\doteqdot}_{b}[\![\mathcal{S}]\!])$, where $\circ\, a$ denotes all the activities preceding a.

The intuition behind this property is that any actor cannot use an equality fairness policy if, in some previous activity, he/she adopted an equity-based approach, as it for sure allowed him/her to know sensitive information, which may influence equality.

As for the second aspect, the following property holds.

Property 3 (Path Equity Fairness). A path of a process model m, given by the sequence of activities inside the same lane, is equity fair if for any activity $a \in Ac$ it holds $av^{\doteqdot}_{a}[\![\mathcal{S}]\!] \subseteq av^{\doteqdot}_{b}[\![\mathcal{S}]\!]$ for all activities b in $a\,\circ$, where $a\,\circ$ denotes all the activities following a.

This property states that to ensure equity on a path of a process model, all the successors of activity a, assigned to the same actor executing a, must work on a superset (or the same set) of attributes that activity a accesses.

This property enforces that the same actor cannot work on data that, in successive activities, cannot be available if we want to preserve equity.

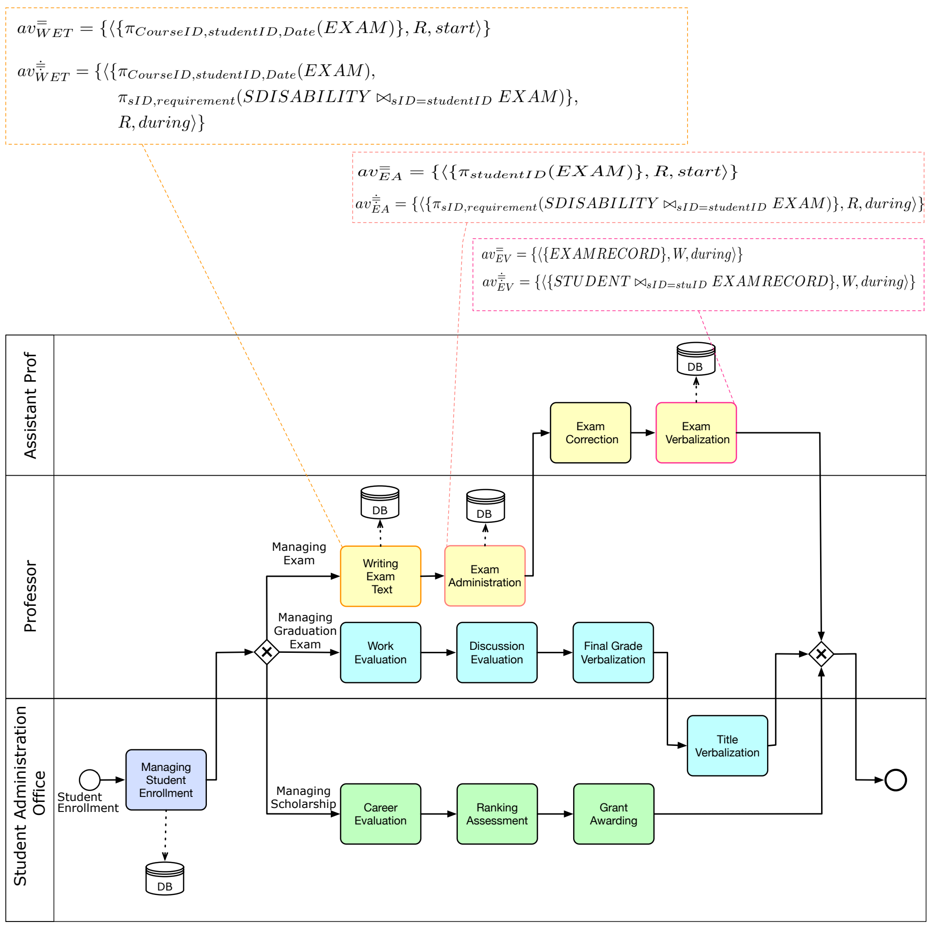

Fig. 3. The re-engineered business process model for the management of students' career equipped with Fairness-driven Activity Views.

Let us consider again the example in Fig. 2, and in particular, the yellow path representing the activities carried out for managing a university exam. In this case, if the equity principle is implemented for the first two activities, the professor has access to sensitive information regarding the disability of some students. However, these pieces of information cannot be forgotten by the same professor when he/she executes the last two activities related to the exam correction and verbalization. More specifically, during the correction, the professor should not know any information about the student's health status or career. This is represented in the activity view by the fact that the accessed database schema is empty. Therefore, this path cannot be considered entirely fair. For this reason, Fig. 3 proposes a re-engineered version of the process that corrects such an unfair situation. In particular, the solution includes the introduction of a new actor, represented by an additional lane identified by the name "Assistant Professor", which will perform the final two activities requiring a more strict fairness principle. Let us notice that in this case, BPMN lanes are used not only to identify different roles but also to identify different actors covering them.

The property of equity fairness can finally be extended to the entire process model in a straightforward manner.

Property 4 (Process Equity Fairness). A process model m is equity fair if all its possible paths inside the same lane are equity fair.

It follows that when gateways are present in the process model, the fairness of paths needs to be verified on all the possible paths obtained on the base of the gateway semantics.

Let us notice that the concept expressed with respect to a model path can also be true when different instances of the same process model are considered and are under execution over time. The extension from the fairness of process schema to the fairness of the single instances is out of the scope of this paper but can be considered a valid extension point for future work.

6 Conclusions and Future Work

In this paper, we proposed an integrated approach to data and process modeling, which explicitly considers fairness in data and process management. More specifically, we focused on the concepts of equity and equality in the context of PAISs. We considered fairness issues both at the level of single activities and related data, moving up to the overall process model. Moreover, we provided some examples of how different fairness requirements related to data access are also connected to the task-assignment policies for different actors and agents.

As for future work, we plan to study fairness-related issues, together with the characterization of sensitive data, also in the context of more complex role-based activity assignments. In addition, the distinction between individual fairness and group fairness notions can guide the re-engineering of unfair BPMN models.

References

1. A. Angerschmid, J. Zhou, K. Theuermann, F. Chen, and A. Holzinger. Fairness and explanation in ai-informed decision making. *Mach. Learn. Knowl. Extr.*, 4(2):556–579, 2022.

2. S. Caton and C. Haas. Fairness in machine learning: A survey. *ACM Computer Surveys*, 56(7), 2024.
3. C. Combi, B. Oliboni, M. Weske, and F. Zerbato. Conceptual modeling of interdependencies between processes and data. In *ACM Symposium on Applied Computing (SAC)*, pages 110–119. ACM, 2018.
4. C. Combi, B. Oliboni, M. Weske, and F. Zerbato. Conceptual modeling of processes and data: Connecting different perspectives. In *37th International Conference on Conceptual Modeling, ER 2018*, volume 11157 of *Lecture Notes in Computer Science*, pages 236–250. Springer, 2018.
5. C. Combi, B. Oliboni, and F. Zerbato. Integrated exploration of data-intensive business processes. *IEEE Trans. Serv. Comput.*, 16(1):383–397, 2023.
6. G. Demartini, K. Roitero, and S. Mizzaro. Data bias management. *Commun. ACM*, 67(1):28–32, dec 2023.
7. B. Hutchinson and M. Mitchell. 50 years of test (un)fairness: Lessons for machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 49–58, New York, NY, USA, 2019. Association for Computing Machinery.
8. H. V. Jagadish, J. Stoyanovich, and B. Howe. The many facets of data equity. *ACM J. Data Inf. Qual.*, 14(4):27:1–27:21, 2022.
9. M. Kasy and R. Abebe. Fairness, equality, and power in algorithmic decision-making. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 576–586, New York, NY, USA, 2021. Association for Computing Machinery.
10. M. K. Lee, A. Jain, H. J. Cha, S. Ojha, and D. Kusbit. Procedural justice in algorithmic fairness: Leveraging transparency and outcome control for fair algorithmic mediation. *Proc. ACM Hum. Comput. Interact.*, 3(CSCW):182:1–182:26, 2019.
11. N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM Computer Surveys*, 54(6), jul 2021.
12. A. Narayanan. Translation tutorial: 21 fairness definitions and their politics. In *Proc. conf. fairness accountability transp., new york, usa*, volume 1170, page 3, 2018.
13. Object Management Group. Business Process Model and Notation (BPMN), v2.0.2. Available at: http://www.omg.org/spec/BPMN/2.0.2/, 2014.
14. D. Pujol and A. Machanavajjhala. Equity and privacy: More than just a tradeoff. *IEEE Secur. Priv.*, 19(6):93–97, 2021.
15. Q. Ramadan, D. Strüber, M. Salnitri, J. Jürjens, V. Riediger, and S. Staab. A semi-automated BPMN-based framework for detecting conflicts between security, data-minimization, and fairness requirements. *Softw. Syst. Model.*, 19(5):1191–1227, 2020.
16. M. Reichert. Process and Data: Two Sides of the Same Coin? In R. Meersman, H. Panetto, T. Dillon, S. Rinderle-Ma, P. Dadam, X. Zhou, S. Pearson, A. Ferscha, S. Bergamaschi, and I. F. Cruz, editors, *20th International Conference on Cooperative Information Systems*, pages 2–19. Springer, 2012.
17. M. Steen. Ethics as a participatory and iterative process. *Communications of the ACM*, 66(5):27–29, 2023.
18. M. Zehlike, K. Yang, and J. Stoyanovich. Fairness in ranking, part II: Learning-to-rank and recommender systems. *ACM Computer Surveys*, 55(6), 2022.
19. M. Zehlike, K. Yang, and J. Stoyanovich. Fairness in ranking, part I: score-based ranking. *ACM Computer Surveys*, 55(6):118:1–118:36, 2023.