Bridging the Gap between Processes and Data
Proposing and Evaluating Activity Views

Carlo Combi[1], Barbara Oliboni[1], Mathias Weske[2], and Francesca Zerbato[1]

[1] Department of Computer Science, University of Verona
[2] BPT Group, Hasso Plattner Institute, University of Potsdam

Abstract. Business processes constantly generate, manipulate, and consume data that are managed by organizational databases. Despite being central to business process modeling, the link between processes and data is often handled by developers during process implementation, thus leaving the connection unexplored during conceptual design. However, supporting process designers in understanding the structure and semantics of the conceptual data related to a process may result in better communication with stakeholders and improved data-aware process models. In this paper, we introduce, formalize, and experimentally evaluate a novel conceptual view that bridges the gap between process and data models, and show some kinds of interesting insights that can be derived when reasoning about such connection.

1 Introduction

The role played by data in business process design, implementation, and execution is gaining considerable attention within the Business Process Management (BPM) and database communities [1].

Both the connection between processes and persistent data managed by organizational database systems and the notion of *data–aware* process modeling have been investigated by recent studies in BPM considering both data-centric [8, 25] and activity-centric paradigms [3, 5, 14, 16].

However, despite being known that processes and data are "two sides of the same coin" [21], these two assets are still conceived separately in most organizational realities. On the one hand, activity-centric process modeling languages, such as the well-established Business Process Model and Notation (BPMN) [18], traditionally focus on modeling the control flow of a process by emphasizing the role of activities and their dynamic behavior. BPMN allows one to define business process models at multiple levels of abstraction, starting from a high-level conceptual viewpoint to the specification of technical aspects needed for implementation. On the other hand, database design consists of three consolidated and distinct phases, namely conceptual, logical, and physical design [7]. For each design phase, designers make use of different data models and schemata to capture the aspects of interest at a particular level of abstraction. At the highest level, conceptual data models are used to create conceptual schemata that concisely represent how the information of interest for a specific domain is organized.

In this scenario, the connection between processes and data is often handled by developers who implement activities, thus leaving the conceptual gap between processes and data open [5, 6].

However, being close to the human perception of the represented domain, conceptual approaches bring several advantages to process designers: they foster the visualization of processes and related data, support conceptual reasoning, and improve system flexibility in terms of preventive detection of issues and data inconsistencies that may arise during process implementation [3].

In this paper, we address the problem of connecting processes and data at the conceptual level, by using BPMN to represent processes at a suitable level of abstraction, tailored to meet conceptual data schemata. More specifically, we propose a novel Activity View, aimed to capture the connection between a BPMN process model and UML Class Diagram [19], the latter one being the conceptual schema of a database. Activity Views are meant to support process designers in modeling the operations performed by process activities on *persistent data* stored in a database, at a conceptual level, and to enable basic reasoning on the interplay between a process and a related database. Our approach is based on existing modeling standards to avoid defining yet another conceptual model and to ease the mapping of devised concepts to known (logical) frameworks [7]. Indeed, sitting in-between process models and data schemata, the Activity View provides a novel connected perspective, while leaving process and data models unchanged.

The main novelty of this paper is the formalization and experimental evaluation of the Activity View that can be used (i) to support the specification of data operations during process modeling and re-engineering, and (ii) to provide interesting insights on the interplay between process models and conceptual data schemata.

The remainder of this paper is structured as follows. Sect. 2 provides the motivations of our approach. Sect. 3 introduces the Activity View, while Sect. 4 describes how our proposal fosters new conceptual insights on processes and related data. Sect. 5 describes the experimental evaluation of our approach. Sect. 6 discusses related work. Finally, Sect. 7 concludes the paper.

2 Motivating Example and Open Research Questions

Starting from the BPMN [18] and UML [19] standards, in this section we introduce a sample scenario to motivate our approach.

Let us consider the process conducted by a professor to examine and grade students, as the one represented by the simple BPMN process of Fig. 1. In general, a process model is mainly composed of *activities* and *events*, which may have associated data and are connected by *sequence flows*, that denote their precedence relations and whose branching and routing is controlled by *gateways*.

The process of Fig. 1 begins with a start event s, depicted as a circle, which is followed by activity Check attendance, represented as a rectangle with rounded

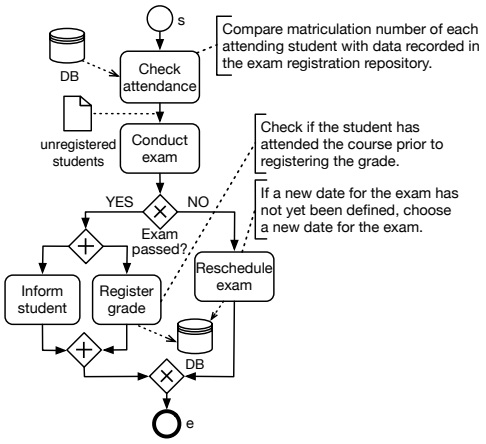


Fig. 1. BPMN process diagram representing the main actions performed by a professor to examine students. Operations on persistent data are described by text annotations.

corners. To check student attendance, the professor must compare the matriculation numbers of the attending students with the data retrieved from the exam registration repository. The latter one represents a source of persistent data and it is represented as a *data store*, named DB. Data stores are connected to one or more activities through directed *data associations*. The attendance of unregistered students is also recorded and this volatile information is passed along to activity Conduct exam through a *data object*. Then, each student is examined individually. Based on how well the student responded, the professor decides how to proceed. This decision is represented by exclusive gateway Exam passed? that splits the flow into two branches. For those students who failed, the exam is rescheduled, whereas, for those who passed, the grade is registered in the repository. While registering the grade, the professor informs the student about the result. Thus, activities Inform student and Register grade are executed in parallel, as shown by the enclosing gateways. Finally, end event e concludes the process.

In order to properly model the process of Fig. 1, designers must understand how the information is conceptually structured and organized within the exam registration repository and how the process interacts with it.

The UML class diagram of the exam registration repository is shown in Fig. 2. Classes Student, Exam, and Course represent the main concepts of interest, related by associations grade, with multiplicity $(0..*, 0..*)$, registration, examination, and attendance. Associations grade and registration have a related association class. Reflexive association representative links students with their student representative. Finally, classes Bachelor and Master specialize students.

Despite capturing informational aspects through data objects and data stores, BPMN process models provide little or no detail about the operation performed by process activities on a database and about the organization of such persistent data. This lack of knowledge complicates the modeling of process-related data

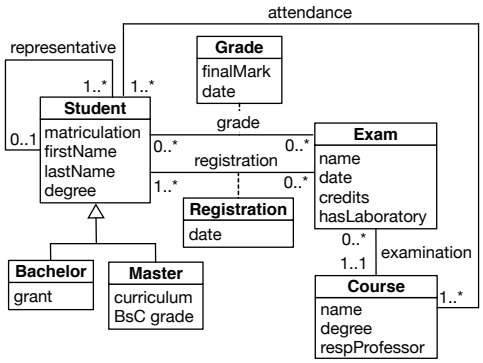


Fig. 2. UML class diagram representing a conceptual model of a database collecting data about students, course attendance, and examinations.

from different standpoints. In detail, we identified the following open conceptual issues **I1**–**I3** in BPMN and exemplified with respect to the process of Fig. 1.

I1. BPMN data stores are underspecified. BPMN data stores are used to represent persistent data sources [18] and, most likely, to identify where is the process there is the need to access a database. However, BPMN defines them at a very high level, and there is currently no standard-compliant way of specifying the schema and the constraints of the represented database [9]. For example, it is not possible to know which is the conceptual schema of the database represented by data store DB of Fig. 1.

I2. Relation between data objects and databases is not clear. Data objects can be used to represent volatile data at different granularities, spanning from single data entries to structured documents [15]. As a result, the correspondence between data objects or conceptual information entities related to a process and persistent data is not so clear. In practice [14, 15] data objects are often seen as instances of database classes but, again, the standard is not clear about their use in this way [18].

For instance, we may ask ourselves whether data object unregistered students is related to data class Student and how. More in general, let us suppose that a professor wants to read a student's grade transcript, containing the whole academic record. In the data schema of Fig. 2, there is no class "transcript" that captures such information directly, since the concept of grade transcript corresponds to the portion of the schema composed of data classes Student and Exam, related through association grade, and association class grade. Therefore, the data object representing the transcript must correspond to a more complex conceptual object, which is identified by a portion of the accessed database.

I3. Specification of data operations is lacking. Last but not least, too little detail about data operations is provided as these are often encoded within activities or their labels [6, 15]. Despite directed data associations allow one to

visualize when data are read or written by an activity, it is not possible to distinguish the granularity of the conceptual object(s) or of the sets of objects needed by a process. This is often true when models are generated independently and have different data semantics and granularities. For example, when comparing matriculation numbers, it is not clear which data classes must be read as the semantics of the process may be different from the one of the referred data models.

As process models and data schemata are conceived separately, to foster data-aware process modeling it is necessary to support designers in understanding and capturing the connection between processes and persistent data sources.

3 Bridging the Gap between Processes and Data

In this section, we propose a novel solution aimed to capture the connection between BPMN process models and UML class diagrams at a conceptual level. To this end, we devised the *Activity View*, a novel approach linking the conceptual representations of process and data models by detailing which operations are performed by a process activity on a database and how.

We chose activities as a starting point, as data modeling in BPMN is often related to activities or whole processes. The final goal of the Activity View is to show which is the portion of a database schema (i.e., the view) that is accessed by a given process activity and to detail interesting aspects of the performed data operations.

Definition 1 (Activity View). Given an activity ac in a process model, its *Activity View* $av_{ac} = \{t_1, \ldots, t_n\}$ is a set of tuples $t_1, \ldots, t_n$, where each tuple t_i denotes a particular data access operation performed by activity ac on a subset of classes of a given data schema. The latter is composed of a set of classes Cl, a set of associations As, and a set of association classes AsC. Each tuple t_i is defined as follows:

$$t_i = \langle C_{set_i},\ A_{set_i},\ AccessType_i,\ AccessTime_i,\ NumInstances_i \rangle$$

where:

- $C_{set_i} = \{c_1, \ldots, c_j\} \subseteq (Cl \cup AsC)$, is the set of connected classes accessed by process activity ac. By "connected" we mean that each class $c_j \in C_{set_i}$ is reachable from at least another class $c_h \in C_{set_i}$ by navigating an association $a_f \in As$ that directly links c_h and c_j (i.e., c_h and c_j are at the ends of association a_f). Moreover, if a_f is associated to association class $c_l \in AsC$, then c_l must also belong to C_{set_i}. If a class $c_j \in C_{set_i}$ specializes a more general class c_l, then it is sufficient that c_l is one end of association a_f for c_j to be considered connected to other classes of C_{set_i}. Instead, the opposite does not hold. Each class $c_j(attr_1, \ldots, attr_n) \in C_{set_i}$ is characterized by a unique name c_j and a set of attributes $attr_1, \ldots, attr_n$. If all the attributes of c_j are involved in the data operation, we write $c_j(*)$. Instead, if only a subset of attributes of c_j is accessed by the activity, we explicitly specify this subset as $c_j(attr_g, \ldots, attr_m)$ with $1 \leq g < m \leq n$.

– $A_{set_i} = \{a_1, \ldots, a_r\} \subseteq C_{set_i} \times C_{set_i} \subseteq As$ is a set of binary associations that directly link any two classes of C_{set_i} (i.e., the ends c_h and c_j of association a_i with $1 \leq i \leq r$ must belong to C_{set_i}). $A_{set_i} = \{*\}$ is the set of all associations that directly link any two classes of C_{set_i}.
– $AccessType_i \in \{R,\ I,\ D,\ U\}$ defines the type of access to the related information. R denotes a *read* of elements of C_{set}, whereas I, D, and U denote different kinds of write operations, namely I indicates an *insertion*, D stands for a *deletion*, while U denotes an *update*.
– $AccessTime_i \in \{start,\ end,\ during\}$ denotes when a data operation is performed with respect to activity execution. This qualitative information refines the description of data operations by specifying the moment they are performed. Access time defines a partial order among Activity View tuples.
– $NumInstances_i = (min,\ max)$, where $min \in \{0,\ 1,\ *\}$ and $max \in \{1,\ *\}$, denotes the number of objects involved in the considered operation. ndeed, activities must be able to access collections of objects and cardinalities should be properly managed [10]. Values $0, 1, *$ have the same meaning as in UML multiplicity specification, where $*$ means "multiple objects".

Consistency constraints. In order to be consistent, an Activity View must be defined by observing the following constraints.
– If $|C_{set_i}| = 1$ and no (reflexive) association is accessed by the process, then $A_{set_i} = \varnothing$;
– If $|C_{set_i}| > 1$, then $A_{set_i} \neq \varnothing$ and all data classes that are ends of $a_f \in A_{set_i}$ and the association class $asc_j \in AsC$ related to a_f must belong to C_{set_i};

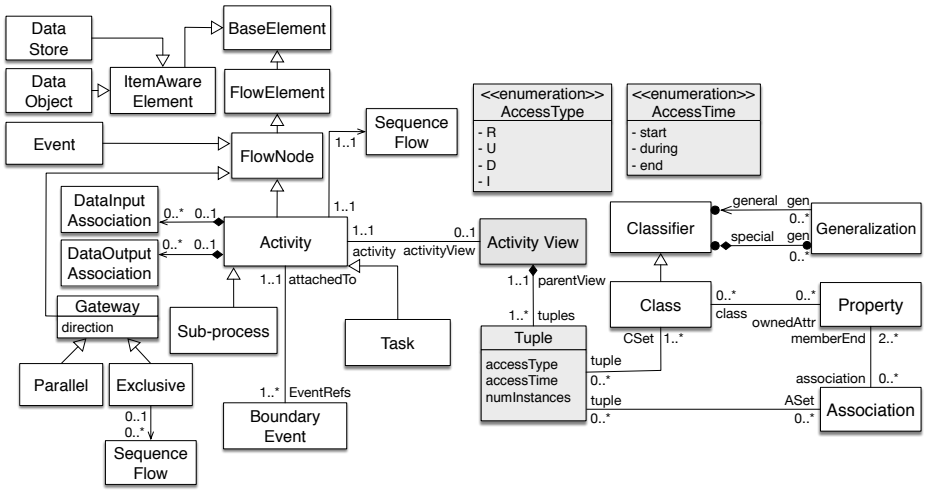


Fig. 3. Meta-model representing the link realized by the Activity View between the concepts belonging to BPMN processes and UML class diagrams considered in this paper. Novel introduced concepts are shaded.

– An association class $asc_j \in AsC$ may be accessed individually, as a other classes.
– If an activity has an associated data store, then it must also have a defined activity view, while the opposite does not hold.

Sitting in-between two well-established standards, the Activity View blends the concepts of activity, borrowed from BPMN, with those of class, attribute, and association taken from UML. Moreover, being defined independently from data stores and data objects the Activity View is able to provide a clear representation of the area of a data schema used by an activity as well as of the data operations performed on it, thus addressing open issues **I1**–**I3**.

Fig. 3 shows how the Activity View links the meta-model of BPMN processes [18] to the one of UML class diagrams [19,20]. For readability, we selected only a relevant subset of elements belonging to the two standards. One feature that immediately stands out from the proposed meta-model is the missing connection between class Data Store and classes Class and Association representing a database schema: this representation is standard compliant, as BPMN does not provide the possibility of specifying the schema of a data store, but only capacity and size limit. Similarly, existing modeling tools such as Signavio [22] and Camunda [2] do not allow one to associate conceptual data schemata or a database server name to data stores. Only during implementation, Java-based execution engines make use of the Java Persistence API to programmatically configure the access to the data stored in relational databases [5], but this is often done for each activity that requires data to be executed, regardless of the latter being connected to a data store.

Instead, at a conceptual level, the connection between activities and data schemata is realized by the Activity View, which can be defined even if in the BPMN process no data store is linked to the activity.

As an example, consider the previously described process of Fig. 1. To check students attendance, the professor must retrieve data regarding student enrollment from the exam registration repository, depicted as data store DB. Given the data schema of the exam registration repository, shown in Fig. 2, the described data access operation can be formalized by the following Activity View composed of a single tuple:

$$av_{CheckAttendance} = \{\langle\{Student(matriculation), Exam(name, date), Registration(*)\}, \{registration\}, R, during, (1,*)\rangle\}.$$

Similarly, activity Register grade performs the two following data operations:

$$av_{RegisterGrade} = \{\langle\{Student(matriculation), Course(name)\}, \{attendance\}, R, start, (1,1)\rangle, \langle\{Grade(*)\}, \varnothing, I, during, (1,1)\rangle\}.$$

Finally, activity Reschedule exam creates a new instance of class Exam.

$$av_{RescheduleExam} = \{\langle\{Exam(*)\}, \varnothing, I, during, (0,1)\rangle\}.$$

Activities Conduct Exam and Inform Student do not have a related Activity View as they do not require access to persistent data.

For better readability, the tuples of one Activity View can be represented in a tabular form, as shown in Fig. 4

$av_{RegisterGrade}$					
Tuple	C_{set}	A_{set}	*AccessType*	*AccessTime*	*NumInstances*
t_1	$\{Student(matriculation), Course(name)\}$	$\{attendance\}$	R	$start$	$(1,1)$
t_2	$\{Grade(*)\}$	$\varnothing$	I	$during$	$(1,1)$

Fig. 4. Tabular representation of the Activity View for activity Register Grade.

Graphically, the link constituted by the three described Activity Views can be visualized over a process diagram and a data schema, as shown in Fig. 5. Dashed arrows connect activities to the portion of the data schema specified in the Activity View. We used the same colors for the activity border, the dashed lines that frame the accessed data classes and the full lines that highlight associations. Connecting arrows are labeled with the information related to access type, access time, and number of objects involved in the operation. Whenever multiple tuples of one Activity View represent different data operations on the same area of the data schema, the dashed arrow connecting the activity with that area of the data schema may be associated to multiple labels.

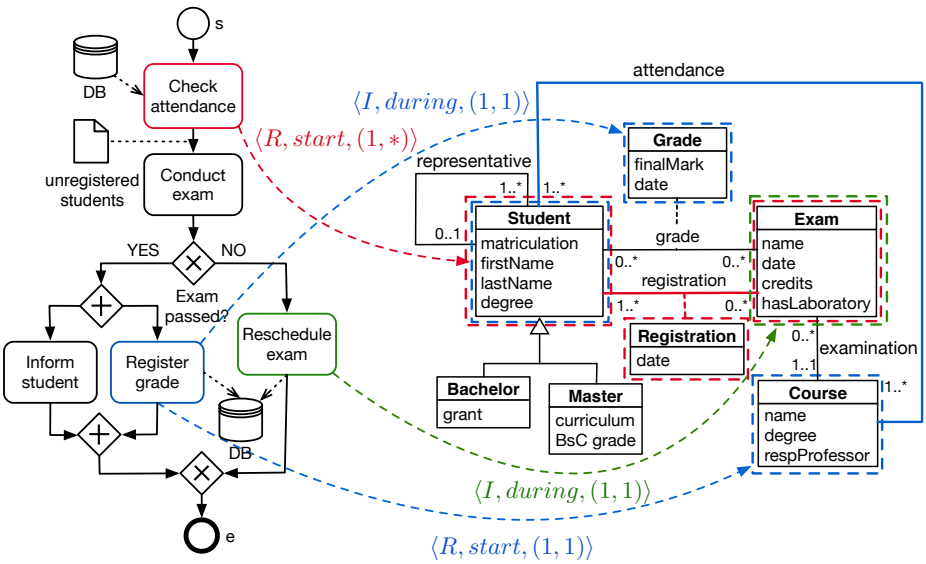


Fig. 5. Graphical representation of Activity Views linking a process to a data schema.

4 Novel Conceptual Insights

In this section we discuss some aspects of the chosen research line that lead to the definition of the Activity View and show the novel perspectives that can be discovered by using the Activity View during process design and analysis.

To come up with the definition of Activity View presented in Sect. 3, we considered several aspects related to linking process models and data schemata. *Data classes and attributes.* Data classes define the conceptual objects of interest that are needed by a process activity. An Activity View allows designers to specify that only certain attributes of a class are read or written. This situation is quite common whenever the data schema represents an organizational database that has not been designed for supporting only that specific process. Moreover, since the granularity of process activities and data classes is defined independently, the creation/modification of a certain object may be realized by multiple activities, in a stepwise manner. Finally, when considering process roles and data access privileges, it is plausible that certain attributes may have restricted access and, thus, a data class may not necessarily be accessed as a whole.

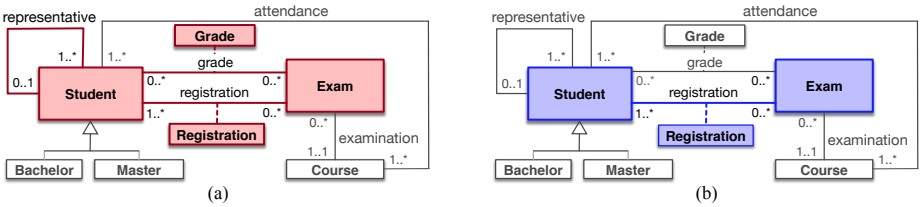


Fig. 6. Activity view when (a) excluding or (b) including associations between classes.

Data associations and association classes. Adding data associations to the specification of an Activity View substantially changes the level of detail provided by the Activity View itself, especially for those data schemata having reflexive or multiple associations between any two classes, as shown in Fig. 6. Specifically, we discerned two scenarios. If associations are not specified, the Activity View has a higher level of abstraction and it is not clear how any two classes of C_{set} are connected. In this case, depicted in Fig. 6.(a), we can assume that all reflexive associations defined on elements of C_{set} are included in the Activity View, together with all the possibly multiple associations that link any two classes of the C_{set}. Instead, by specifying class associations, we provide a more precise description of how the classes of C_{set} are related.

As an example, let us consider the process model and data schema of Fig. 5 and let us assume that the department secretary needs to have access to the information related to exam registration, that is, she will need to read objects of classes Student, Registration, and Exam, respectively. However, for privacy reasons, secretaries are not allowed to see the grades of students, stored in objects

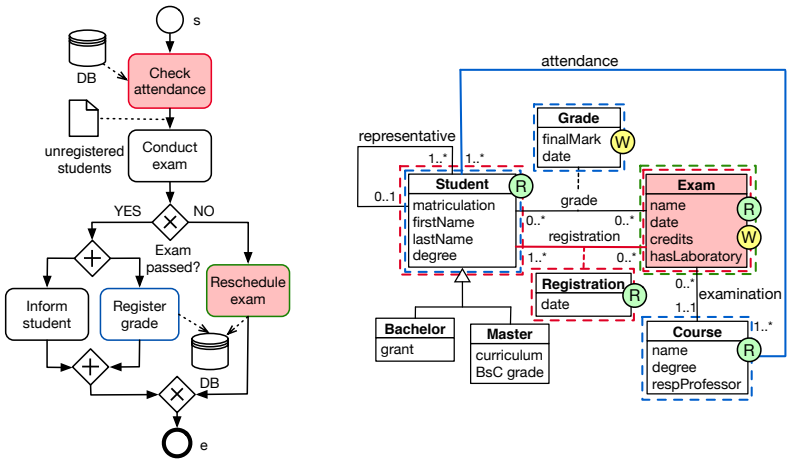


Fig. 7. Visualization of the conceptual insights provided by using Activity Views.

of association class Grade. In Fig. 6.(a) it is not possible to deal with the described setting due to the high level of abstraction. Instead, by adding data associations as done in Fig. 6.(b) we are able to distinguish how any two classes are related within an Activity View. For this reason, Def. 1 includes associations.

Beside supporting the modeling of the connection between process and data diagrams at a conceptual level, the Activity View provides other interesting insights, useful for analysis purposes as well as for improving the communication with stakeholders. In particular, using the Activity View can help designers to address issues **I1**–**I3** introduced in Sect. 2. In the following paragraphs, we discuss how the tuples of one or more Activity Views can be exploited to discover and visualize interesting aspects of the connection between processes and data.

Identifying the portion of a data schema accessed by a process activity. As described in Sect. 3, the Activity View allows one to identify which are the classes and associations of a data schema that are accessed by a certain activity, thus providing a better specification than data stores and responding to issue **I1**. However, to visualize the portion of the data schema accessed by an activity ac_k, all the tuples $t_{1,k}, \ldots, t_{n,k}$ of av_{ac_k} must be properly combined.

In detail, the comprehensive set of classes and association classes of a data schema accessed by ac_k is defined as $\bigcup_{j=1}^{n} C_{set_{j,k}}$, where j denotes the tuple and k the activity. Similarly, the set of all associations accessed by ac_k is $\bigcup_{j=1}^{n} A_{set_{j,k}}$.

In Fig. 5 and Fig. 7, the area of interest of the data schema is graphically rendered by framing classes with dashed lines colored as the activity border.

Detecting which activities operate on a certain data class. Under a different standpoint, Activity Views can be used to understand which among all process activities have access to objects of a certain data class. This is useful for several reasons, starting from easing the communication with domain experts during process modeling. Stakeholders are often interested in seeing where cer-

tain data are used in the process to understand which is the information that drives certain activities and which data are used to make decisions. This holds also for data compliance. Indeed, in some circumstances, the quality of activity execution may drastically improve if proper information is available. Under an engineering perspective, understanding how data are used during process execution provides hints for data management support and re-engineering.

For instance, class Exam of Fig. 7 is accessed by tasks Check attendance and Reschedule exam, as highlighted by the filled background. By taking a look at the structure of the process, we can easily see that if the student succeeds, class Exam is only accessed at the beginning of the process.

In order to retrieve the set of process activities $ac_g, \ldots, ac_l$ that manipulate a certain data class c_i, we shall go through all the Activity Views of the process and check whether c_i belongs to at least one class set $C_{set_{j,k}}$ of a tuple $t_j, k \in av_{ac_k}$. That is, given a class c_i the set of all activities that have access to it is given by $\{ac_k | \exists\, t_{j,k}\, (c_i \in C_{set_{j,k}})\}$.

Understanding which classes are either read or written by a process. The type of access to data allows designers to easily visualize when data classes have associated read or write operations and how these are distributed in the process (cf. **I3**). However, we can also retrieve which classes of the data schema are associated only to read or write accesses. This is particularly useful when speaking about data integrity, as several activities of one or more processes may operate on the same data class concurrently and, thus, transactional properties such as isolation must be discussed [16, 25]. Last but not least, certain sequences of read and write operations performed on the same data classes may lead to inconsistencies, as explained in [3].

For instance, in order to understand whether objects of a class c_k are only read by activities of a process, we shall go through all the Activity Views and ensure that there exists no tuple having $c_k \in C_{set}$ and access type of kind *I*, *U*, or *D*. This can be expressed as $\{c_k | \nexists j, i\, ((C_{set_{j,i}} \ni c_k) \wedge (AccessType_{j,i} =$ "I" $\vee\, AccessType_{j,i} =$"U" $\vee\, AccessType_{j,i} =$ "D"$))\}$.

In Fig. 7, for each data class related to the process, Ⓡ and Ⓦ denote if the class is read or written by process activities.

By combining the described insights provided by the Activity View, designers can understand and visualize, with the help of stakeholders, which is the key information needed to support process execution. This can be represented by one or more data classes, which we refer to as *core classes* for a given process.

Informally, given a data schema, a core class is a class of the data schema that represents valuable process-related data and has the following properties.

- It appears in a considerable number of Activity Views related to the process (i.e., it is shared by multiple process activities).
- Its objects are frequently accessed by the process, that is, it appears in a considerable number of Activity View tuples.
- Its objects are used by the most important activities of the process. By "most important", we refer to activities that are crucial for the chosen application domain or are executed in (almost) all the instances of a process, if any.

– Its objects are never deleted by the process.
– It is mostly subjected to mandatory access, that is, Activity View attribute $min \in NumInstances$ is never equal to 0.

With respect to the process of Fig. 7, classes Exam and Student are core classes, as they are the most accessed in the process. As for their use, they are accessed by exactly the same activities, but class Student is only read by process activities. Of course, to determine whether a read-only access is less important than a write access, domain experts should be consulted, as the idea is to exploit Activity Views in any manner, to retrieve information useful for conceptual design.

Indeed, if in such a simple example the identification of important data is quite straightforward, the concept of core classes becomes quite useful in complex and highly branched processes, where identifying the key information to support process execution is not straightforward. This latter issue is also open in the field of data-centric process modeling, since the same questions need to be answered to identify the data artifacts on which the processes are based [8, 25].

5 Experimental Evaluation: Design and Results

This section describes how we evaluated our approach and the obtained results. The detailed experimental setting, that is, questionnaires and raw results, are reported in Appendix A.

Experiment planning and design. In order to analyze the usability of the Activity View, we conducted a human-oriented single factor experiment by following the design principles described in [4, 12, 13, 26, 27]. The chosen factor is

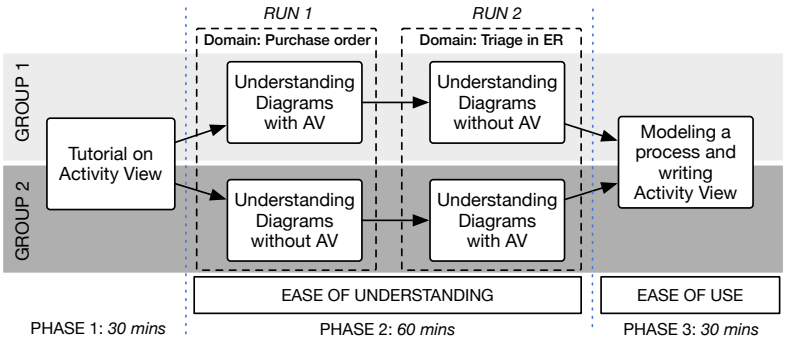


Fig. 8. Main steps of the experiment designed to evaluate the proposed Activity View.

the *Activity View*, which represents our controlled variable, with factor levels *present* and *absent*. Specifically, we aimed to evaluate how the use of Activity Views can improve both the modeling and the understanding of the interplay between a process and a persistent database. In order to analyze such improvement

quantitatively and qualitatively, we formulated the two following hypotheses.
H1 - Perceived ease of understanding. The Activity View improves the conceptual design of processes that operate on persistent data in terms of improved understanding of which data are needed by activities to be executed how they are used in a process. Improved understanding is quantified as better task performance in terms of increased speed and reduced error rate.
H2 - Perceived ease of use. It evaluates the ease of using the Activity View, that is, it assesses whether the Activity View can be easily read, understood, used, written, and adapted to different application domains.

Subjects are 21 students enrolled in the M.Sc. degree in Computer Science Engineering, 8 students enrolled in the M.Sc. degree in Medical Bioinformatics, and 4 researchers in the field of database design. All of the 33 subjects attended at least one information system course where BPMN is explained (about 8 teaching hours), and at least one complete database course (48 teaching hours). Among these, 8 subjects have working experience in the field of UML-based database design, whereas none of them has worked with BPMN at a professional level.

Objective	Evaluate the ease of understanding and ease of use of the Activity View to assess its effectiveness in terms of improved understanding of the interplay between process and data diagrams.
Independent Var.	Activity view (present or absent).
Dependent Var.	Time needed to execute the exercises, correctness of the answers.
Subjects	Trained students enrolled in the M.Sc. in computer science engineering and in the M.Sc. in medical bioinformatics.
Context	Process modeling and insights discovery using the Activity View.

Table 1. Setting of the performed experimental evaluation.

The experimental evaluation is organized as shown in Fig. 8. During *PHASE 1* the subjects attended a 30-minute tutorial on the Activity View, where fundamental concepts and motivations were explained. Then, for both *PHASE 2* and *PHASE 3* subjects were asked to execute an experimental task on paper. *PHASE 2* was divided into two runs, where each run was based on a questionnaire containing 7 questions regarding diagrams insights (cf. Sect. 4). We used a within-groups approach, that is, we randomly divided the number of subjects into two groups, and each group performed the task *with* and *without* the Activity View. In detail, we provided all the subjects with a textual description of a process and its related data operations, and with the corresponding BPMN and UML diagrams. At each run, one group was also provided with the Activity Views related to the process and data diagrams. During the first run, "Group 1" was asked to execute the experimental task using also the provided Activity Views while "Group 2" was asked to execute the same task but relying only the textual description of the context and on the BPMN and UML diagrams. During the second run we switched groups: "Group 1" executed the task without Activity Views, whereas "Group 2" used the Activity Views. For the second run, we changed the application domain in order to avoid potential learning. The

chosen domains were: Purchase order on a web-pharmacy (RUN 1) and triage in Emergency Room (RUN 2). The detailed text of both runs may be found in Appendix A.

The second experimental task was devised to test the actual usability of the Activity View during process modeling. All the subjects were asked to model a BPMN process and to write the related Activity Views, given a textual description of a process and the UML class diagram of the referred domain database. During this phase, we evaluated both the correctness of the designed processes and of the related Activity Views. Finally, we conducted a questionnaire-based interview to understand how the subjects perceived the Activity View, both for process modeling and conceptual insight discovery.

Evaluation results. Overall, the obtained results confirmed that the Activity View improves the integrated design and understanding of processes and related data, both in terms of reduced task times and increased task correctness.

The task executed during *PHASE 2* allowed us to quantitatively evaluate the use of the Activity View for diagram analysis. For each subject, we measured the task execution time and counted how many of the questions were answered correctly. We applied the most restrictive requirements for correctness, that is, answers were considered correct if answers were both right and complete. Finally, we analyzed the obtained results statistically by applying the *paired t-test*, where the execution times of the tasks carried out by one subject with Activity View were compared with those of the same subject without Activity View.

In the first run, subjects provided with the Activity View, took an average of 12,45 minutes and the 84,03% of the answers was evaluated correct. Instead, the group without Activity View took an average of 22 minutes to complete the task, and only the 39,28% of the answers was correct. Results related to the second run showed a reduction in answering times for both groups, especially for

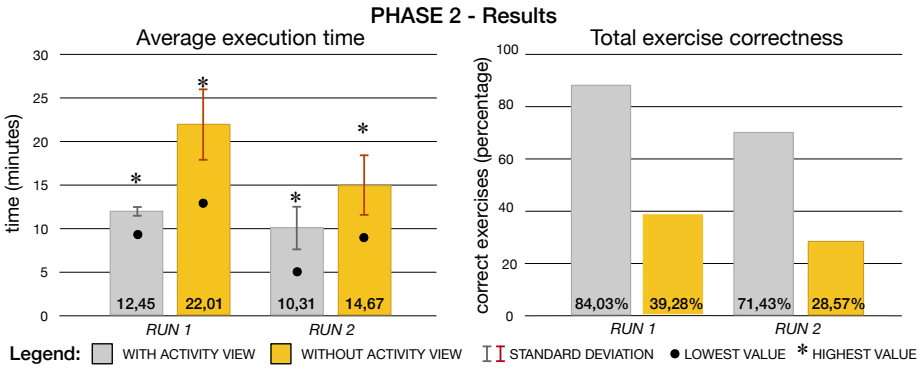


Fig. 9. Average execution time with standard deviation (left) and total percentage of correct answers (right) for the two runs of *PHASE 2*. Task execution with Activity Views is represented by the gray columns.

the those not having the Activity View, as reported in Fig. 9. Subjects claimed that learned what the questions were asking for. However, the correctness of the answers also decreased. By combining the results of both runs, we see that by using the Activity View task times decrease by 37,94% on average, while the number of correct answers improves by 43,80%. By applying the paired t-test to the measured execution times, we retrieved a p-value $< 0{,}005$ and, thus, the obtained results, sketched in Fig. 9, are very statistically significant and hypothesis **H1** is satisfied. The difference in measured execution times and answer correctness remains significant even when the correction requirements are relaxed and also partially correct answers are assigned a positive score (see Appendix for more details).

The experimental task of *PHASE 3* was reviewed by assigning one point for each correctly written Activity View tuple, thus considering each attribute of the tuple worth 0.20 points. Besides, we also considered the correctness of the BPMN process. Overall, the 58,89% of the written Activity Views was correct and the 83,94% of the BPMN process diagrams was designed correctly. The percentage of correct Activity Views increases to 61,11% when excluding incorrect use of the access time, which was not easy to be determined from the exercise text.

The results of the modeling exercise are coherent with the outcome of the interviews conducted at the end of *PHASE 3*. Indeed, all the subjects declared that executing the first experimental task without the help of the Activity View was more difficult, and the 93% of them answered positively when asked whether the Activity View improves the modeling of the link between processes and related data. Then, we asked subjects to answer questions related to the usability of the Activity View based on a rating scale from 1 to 5, with 1 meaning "strongly disagree" and 5 denoting "strongly agree". The average results of this questionnaire-based interview are reported in Fig. 10. Overall, the perception of the Activity View was more than satisfactory and this confirms hypothesis **H2**.

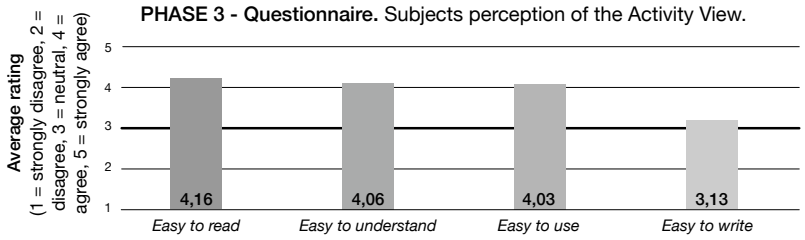


Fig. 10. Average rating of subjects perception of the Activity View.

Being most of the subjects computer science students, our evaluation approach has some limitations, as the result may not be easily generalizable to real organizational environments, where processes are more complex and people receive more professional training. Indeed, exercises were designed in a didactic

way to avoid task misunderstanding. Moreover, results are limited to the kinds of questions that were asked and also depend on the preparation and personal interest of the students in the fields of process and data modeling.

6 Related Work

The relationship between data and processes has been tackled by several research efforts within the fields of high-level Petri nets [11, 16], activity-centric process modeling [5, 14, 23], and data-centric process design [8, 10, 25].

In [16] db-nets are proposed as a novel three-layered approach to combine colored Petri Nets and relational databases, which communicate through an intermediate data logic layer. However, this approach goes beyond the conceptual modeling of data needed for process execution, as it focuses on modeling and verifying a "connected system", where an instance of a database is subjected to changes imposed by the control layer.

Activity-centric modeling paradigms, and especially BPMN, are by all means the most used in practice, despite their support for the data perspective being limited. However, this limitation is often perceived as a design choice and data are combined with processes, at a lower, engineering level [5]. The shortage of well-founded conceptual modeling frameworks supporting process and data integration was motivated by some recent proposals in the field [5, 14, 23].

In [5], a BPMN process diagram is linked through OCL (Object Constraint Language) expressions to the information model of the process, a class diagram incorporating a class "Artifact" which contains the process variables. In detail, the process diagram is formalized as a Petri net, BPMN activities are specified as OCL operation contracts and, then, OCL contracts are encoded into a set of logic derivation rules that can be easily translated into SQL queries. Another framework based on constraint logic programming for representing business processes and reasoning on their behavior and data properties is introduced in [23]. A logical language and a formal semantics are defined to describe data object manipulation and to explicitly represent the interaction of a process with an underlying database. Finally, a technique for automatically deriving SQL queries from annotated data objects is proposed in [14], in order to check data requirements for activity execution.

Despite recognizing the need of linking processes and data conceptually, the approaches introduced in [5, 14, 23] address the connection of processes and data at a lower (logical) level, by considering process variables and data instances, and by providing valuable details for formalizing and automating queries on process-related data. Instead, our contribution provides a higher-level, conceptual view of the connection between processes and data schemata, without excluding the possibility of mapping our approach to any of the introduced ones when moving down to a lower level and considering query specification. Moreover, the Activity View follows the idea presented in [16] that calls for leaving the original process and data models untouched (i.e., it is not meant to extend BPMN). Probably, the main weakness of the Activity View remains its graphical representation,

which may become hard to read due to the proliferation of connections. However, this issue seems to be a problem also in [14] and [16], as the number of data objects, respectively view places labeled transitions, tends to increase with higher numbers of data instances and operations.

The connection between process and data diagrams is tackled by previous research [3], yet considering only which data classes are accessed by a certain process activity and how, in order to detect potential inconsistencies between data classes accessed by multiple activities. Finally, an approach based on data-flow matrices for representing input and output data of workflow activities is presented in [24]. A data-flow matrix summarizes all the read and write operations performed on the process data objects.

As for the verification of data-aware processes, a formally grounded framework that considers also the effects of activities on data is presented in [6]. The framework combines Petri nets, relational data models, and data-centric dynamic systems for capturing process and data interactions.

7 Conclusion

Bridging the gap between process and data diagrams becomes necessary to support process designers in understanding the structure and semantics of conceptual data related to a process. In this paper, we introduced, formalized, and evaluated the Activity View as novel approach aimed to reaize the connection between a process model and a conceptual data schema, while allowing designers to detail also data operations. In particular, we showed how using the Activity View allows one to retrieve and visualize interesting insights related to this connection. For future work, we aim to enrich the Activity View with multiple abstraction levels in order to deal with sub-processes and whole processes. Moreover, we are working towards improving the graphical representation of the Activity View and its integration in existing process modeling tools.

References

1. Calvanese, D., De Giacomo, G., Montali, M.: Foundations of data-aware process analysis: a database theory perspective. In: 32nd ACM SIGMOD Symposium on Principles of Database Systems (PODS). pp. 1–12. ACM (2013)
2. Camunda services GmbH: Camunda BPM Platform. https://camunda.org
3. Combi, C., Oliboni, B., Weske, M., Zerbato, F.: Conceptual modeling of interdependencies between processes and data. In: ACM Symposium on Applied Computing (SAC). pp. 110–119. ACM (2018)
4. Cruzes, D.S., Vennesland, A., Natvig, M.K.: Empirical evaluation of the quality of conceptual models based on user perceptions: A case study in the transport domain. In: International Conference on Conceptual Modeling (ER). pp. 414–428. Springer Berlin Heidelberg (2013)
5. De Giacomo, G., Oriol, X., Estañol, M., Teniente, E.: Linking Data and BPMN Processes to Achieve Executable Models. In: 29th International Conference on Advanced Information Systems Engineering (CAiSE). LNCS, vol. 10253, pp. 612–628. Springer (2017)

6. De Masellis, R., Francescomarino, C.D., Ghidini, C., Montali, M., Tessaris, S.: Add data into business process verification: Bridging the gap between theory and practice. In: AAAI. pp. 1091–1099. AAAI Press (2017)
7. Elmasri, R., Navathe, S.B.: Fundamentals of database systems. Pearson (2015)
8. Hull, R.: Artifact-centric business process models: Brief survey of research results and challenges. In: OTM Confederated International Conferences "On the Move to Meaningful Internet Systems". pp. 1152–1163. Springer (2008)
9. Kossak, F., Illibauer, C., Geist, V., Kubovy, J., Natschläger, C., Ziebermayr, T., Kopetzky, T., Freudenthaler, B.and Schewe, K.: A Rigorous Semantics for BPMN 2.0 Process Diagrams. Springer (2014)
10. Künzle, V., Reichert, M.: PHILharmonicFlows: towards a framework for object-aware process management. Journal of Software Maintenance and Evolution: Research and Practice 23(4), 205–244 (2011)
11. Lenz, K., Oberweis, A.: Inter-organizational business process management with XML nets. In: Petri Net Technology for Communication-Based Systems, pp. 243–263. Springer (2003)
12. Maes, A., Poels, G.: Evaluating quality of conceptual models based on user perceptions. In: International Conference on Conceptual Modeling (ER). pp. 54–67. Springer Berlin Heidelberg (2006)
13. Mehmood, K., Cherfi, S.S.S.: Evaluating the functionality of conceptual models. In: International Conference on Conceptual Modeling (ER). LNCS, vol. 10445, pp. 222–231. Springer (2009)
14. Meyer, A., Pufahl, L., Fahland, D., Weske, M.: Modeling and enacting complex data dependencies in business processes. In: 11th International Conference on Business Process Management (BPM), LNCS, vol. 8094, pp. 171–186. Springer (2013)
15. Meyer, A., Weske, M.: Extracting data objects and their states from process models. In: 17th International Enterprise Distributed Object Computing Conference (EDOC). pp. 27–36. IEEE (2013)
16. Montali, M., Rivkin, A.: Db-nets: On the marriage of colored petri nets and relational databases. In: Transactions on Petri Nets and Other Models of Concurrency XII, LNCS, vol. 10470, pp. 91–118. Springer Berlin Heidelberg (2017)
17. Motulsky, H.: Intuitive biostatistics: a nonmathematical guide to statistical thinking. Oxford University Press, USA (2014)
18. Object Management Group: Business Process Model and Notation (BPMN), v2.0.2. Available at: http://www.omg.org/spec/BPMN/2.0.2/
19. Object Management Group: Unified Modeling Language, v2.5. Available at: http://www.omg.org/spec/UML/2.5/
20. Object Management Group: Meta Object Facility (MOF), v2.5.1. Available at: http://www.omg.org/spec/MOF/2.5.1/ (2016)
21. Reichert, M.: Process and Data: Two Sides of the Same Coin. In: 20th International Conference on Cooperative Information Systems. pp. 2–19. Springer (2012)
22. Signavio GmbH: Signavio Business Transformation Suite. www.signavio.com
23. Smith, F., Proietti, M.: Reasoning on data-aware business processes with constraint logic. In: 4th International Symposium on Data-driven Process Discovery and Analysis (SIMPDA). pp. 60–75 (2014)
24. Sun, S.X., Zhao, J.L., Nunamaker, J.F., Sheng, O.R.L.: Formulating the data-flow perspective for business process management. Information Systems Research 17(4), 374–391 (2006)
25. Sun, Y., Su, J., Wu, B., Yang, J.: Modeling data for business processes. In: 30th International Conference on Data Engineering (ICDE). pp. 1048–1059. IEEE (2014)

26. Wang, W., Indulska, M., Sadiq, S.W., Weber, B.: Effect of linked rules on business process model understanding. In: International Conference on Business Process Management (BPM). LNCS, vol. 10445, pp. 200–215. Springer (2017)
27. Wohlin, C., Runeson, P., Host, M., Ohlsson, M., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer-Verlag Berlin Heidelberg (2012)

A Experimental Evaluation

In this section, we integrate the concepts introduced in Sect. 5 and provide a more detailed explanation of the experimental planning and design underlying the empirical evaluation of the Activity View. In particular, we provide a summary of the main concepts explained during subject training, show the complete text of conducted exercises and related questionnaires, and describe the obtained raw results, their interpretation and the chosen correction methods.

The proposed three-phased experimental evaluation follows a *survey* approach, that is, it makes use of questionnaires to gather human attitudes, opinions, and impressions on the proposed modeling method. The phases of the experiment are detailed in Fig. 8 of Sect. 5.

As already mentioned, we followed and combined some of the main principles explained in [12,13,27] and took inspiration from case studies belonging to similar fields [4, 26].

A.1 PHASE 1 - Tutorial

PHASE 1 consisted of a 30 minutes tutorial addressing the problem of connecting business process models and database schemata. The tutorial was based on recent literature [1,3,5,14,16] and discussed the motivations behind two main research questions. The first one *(i) How is the connection between persistent data used by business processes and databases realized at a conceptual level?* addresses the importance of choosing an abstraction level that is suitable to sit between existing process and database models. The second one *(ii) How does the process diagram interacts with the database schema?* discusses which are common kinds of operations performed by a process on persistent data and how they can be represented. Then, the tutorial recalled which BPMN elements (such as data objects, process variables, message flows, and events that carry data) may be used to represent data within a process and, precisely, we stressed on the concept of *persistent data* and on the use of BPMN data stores in practice.

Once having addressed the research problem, we introduced the Activity View as a possible solution for bridging the gap between processes and data and explained its formalization (cf. Def. 1) step-by-step. Then, we discussed all the insights detailed in Sect. 4 brought by the Activity View, by also referring to practical examples. Last but not least, we explained the structure of the experimental evaluation as done at the beginning of Sect. 5, clarified subjects doubts, and randomly divided all of the participants in two groups, "Group 1" and "Group 2".

A.2 PHASE 2 - Evaluating Ease of Understanding

PHASE 2 was aimed at evaluating the Activity View, specifically considering perceived ease of understanding [12].

PHASE 2 consisted of two runs (*RUN 1* and *RUN 2* of Fig. 8) for an overall duration of around 60 minutes, considering that all subjects had to finish *RUN 1*

before we handed out the second exercise to everyone. Each one of the subjects was provided with two exercises, one for each run, having in common a textual description of a business process and its data accesses on a database, the BPMN process model, and the UML Class Diagram of the accessed domain database. However, at each run, one group was provided also with the tabular representation of the Activity Views related to the process and database, whereas the other group was asked to solve the same exercise without the Activity Views.

The design of the experiment provides that the same subjects solve the exercises having and not having the Activity View. This setting calls for a *paired t-test* that accounts for both the systematic variability between groups with variability between subjects [17].

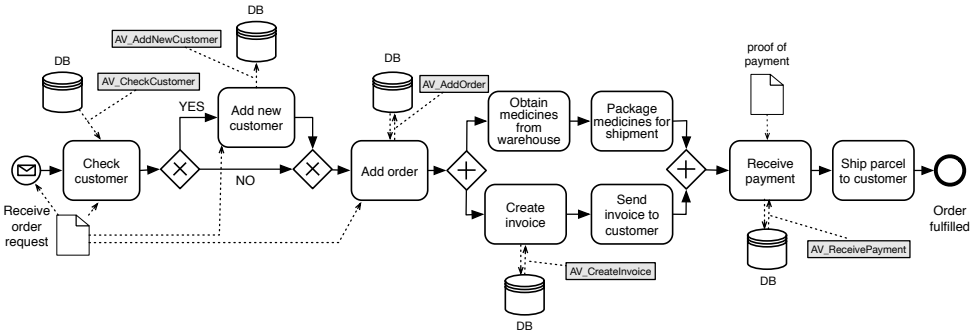
Our main goal was to evaluate whether and how much (i) subjects provided with the Activity View were faster in answering the questions, (ii) the accuracy of the answers improved with the help of the Activity View. The latter aspect is strongly related to the understandability of the proposed model, as the Activity View must be well-understood in order to be used correctly by the subjects. During the whole course of *PHASE 2*, the formal description of the Activity View was written on the blackboard so that participants could consult it.

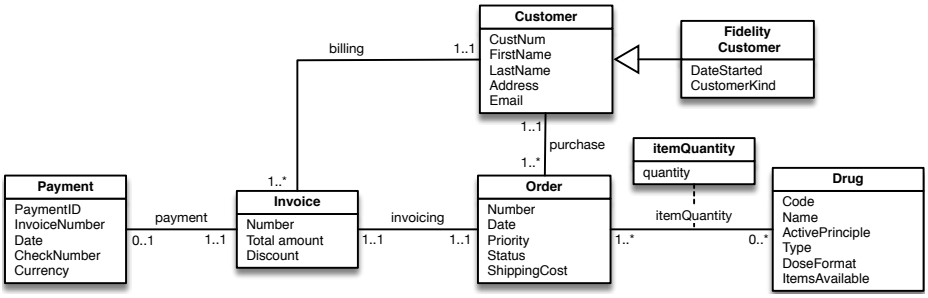
The texts and questionnaires of the two exercises of *PHASE 2* are reported below.

Exercise 1 - Purchase Order on a Web Pharmacy

The process describes a purchase order on a web-pharmacy.
A new process instance is created when an order request is received from a customer. The order request contains information about the customer, the order and the ordered items. Since all customers need to be registered in the database, the information about the customer contained in the order request is compared with the data of customers already saved in the database. If the customer is new, he or she must be added to the database of the pharmacy. Then, the order is added to the database. An order contains a preamble of general information, such as its number, priority, and shipping costs, but it also contains the list of purchased drugs and the related quantity. Afterwards the ordered drugs are obtained from the warehouse and are boxed for shipment. While obtaining the drugs and boxing them, an invoice is created and, then, it is sent to the customer. When invoice creation begins, the order number and the number of the purchasing customer must be checked and, then, the new invoice is created and added to the database. Then, the process waits for the payment to be received. The operator that receives the proof of payment must record it in the database, prior to updating the status of the order to "Ready for Shipment". Then, the parcel is shipped to the customer. The process ends when the order is fulfilled.

The BPMN diagram and the UML class diagram corresponding to the description above are provided below.





The provided Activity Views for Group 1.

AV_CheckCustomer

TUPLE	CLASS SET	ASSOC SET	ACCESS TYPE	ACCESS TIME	INSTANCES
T1	{Customer(CustNum, FirstName, Lastname)}	∅	R	During	(1,*)

AV_AddNewCustomer

TUPLE	CLASS SET	ASSOC SET	ACCESS TYPE	ACCESS TIME	INSTANCES
T1	{Customer(*)}	∅	I	During	(1,1)

AV_AddOrder

TUPLE	CLASS SET	ASSOC SET	ACCESS TYPE	ACCESS TIME	INSTANCES
T1	{Customer(Number)}	∅	R	Start	(1,1)
T2	{Order(*), itemQuantity(quantity), Drug(Code)}	itemQuantity	I	During	(1,*)

AV_CreateInvoice

TUPLE	CLASS SET	ASSOC SET	ACCESS TYPE	ACCESS TIME	INSTANCES
T1	{Customer(Number), Order(Number)}	purchase	R	Start	(1,1)
T2	Invoice(*)	∅	I	During	(1,1)

AV_ReceivePayment

TUPLE	CLASS SET	ASSOC SET	ACCESS TYPE	ACCESS TIME	INSTANCES
T2	Payment(*)	∅	I	During	(1,1)
T3	Order(Status)	∅	U	End	(1,1)

Exercise 1 - Questionnaire
Correct answers have been reported in between parentheses in blue, for completeness purposes.

Preliminary questions
- Have you ever had any working/internship experience in the field of BPMN modeling? _____
- Have you ever had any working/internship experience in the field of UML data modeling? _____

Exercise 1 - Purchase order on a web-pharmacy. Questions.

1. Which data classes does activity "Check customer" access?
   _____ (Customer)

2. Which data classes does activity "Receive Payment" access?
   _____ (Payment, Order)

3. (a) Do the sets of classes accessed by activities "Add order" and "Create invoice" intersect?
   YES ☐ NO ☐ (YES)
   If they do, which data classes belong to their intersection?
   _____ (Customer, Order)

4. Are there classes in the data schema that are never accessed by activities of the process? If so, which ones?
   _____ (Fidelity Customer)

5. Which are the classes of the process used by the highest number of activities?
   _____ (Customer)

6. Which are the activities of the process that access class "Order"?
   _____ (Add order, Receive Payment, Create Invoice)

7. Are there classes that are used only for read operations? If so, which ones?
   _____ (NO)

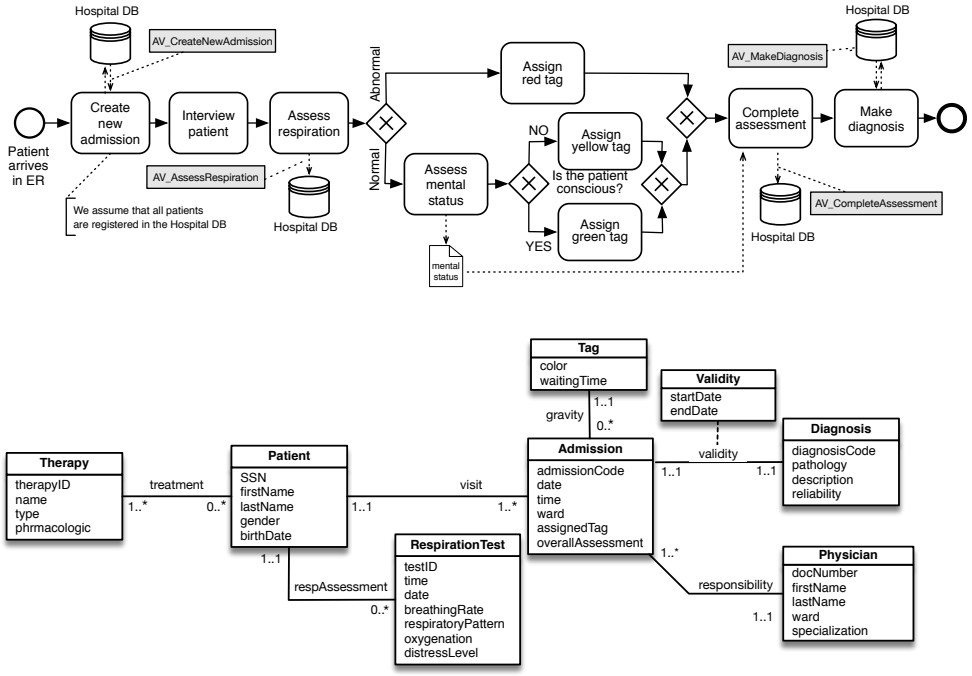
Total Time: _____

Exercise 2 - Triage in Emergency Room

The process describes the main steps performed by a triage nurse to prioritise patients coming to a hospital emergency room (ER).

A new process instance is created when a patient arrives in ER. Since we assume that all the patients are already registered in the database, when creating a new admission, the nurse must compare the information regarding the arrived patient with the database. Then, a new admission is created for that patient and stored in the database. Then, the nurse interviews the patient quickly and assesses his or her respiration through a breathing test, whose parameters are saved in the database. If the breathing rate is abnormal, a red tag must be assigned to the patient. Instead, if the patient does not present respiratory abnormalities, his or her cognitive status is checked and, depending on the degree of consciousness, either a yellow or a green tag is assigned. Information about assigned tags is recorded during assessment completion, during which also the collected information regarding the overall assessment of the patient is updated. Finally, a physician is called in to make a diagnosis, based on data regarding the current admission, on previous therapies, if any, and on the results of the respiration test. The diagnosis is saved in the database together with its validity, which holds starting from the moment it the diagnosis recorded.

The BPMN diagram and the UML class diagram corresponding to the description above are provided below.



The provided Activity Views for Group 2.

AV_CreateNewAdmission

TUPLE	CLASS SET	ASSOC SET	ACCESS TYPE	ACCESS TIME	CARDINALITY
T1	{Patient(*)}	∅	R	S	(1,1)
T2	{Admission(*)}	∅	I	D	(0,1)

AV_AssessRespiration

TUPLE	CLASS SET	ASSOC SET	ACCESS TYPE	ACCESS TIME	CARDINALITY
T2	{RespirationTest(*)}	∅	I	E	(1,1)

AV_CompleteAssessment

TUPLE	CLASS SET	ASSOC SET	ACCESS TYPE	ACCESS TIME	CARDINALITY
T1	{Admission(overallAssessment, assignedTag)}	∅	U	D	(1,1)

AV_MakeDiagnosis

TUPLE	CLASS SET	ASSOC SET	ACCESS TYPE	ACCESS TIME	CARDINALITY
T1	{Patient(*), Admission(*)}	{visit}	R	S	(1,1)
T2	{Patient(*), Therapy}	{treatment}	R	S	(1,*)
T3	{Patient(*), RespirationTest(*)}	{respAss}	R	S	(1,1)
T4	{Diagnosis(*)}	∅	I	D	(1,1)
T5	{Validity(startDate)}	∅	I	D	(1,1)

Exercise 2 - Questionnaire
Correct answers have been reported in between parentheses in blue, for completeness purposes.

Exercise 2 - Triage in Emergency Room. Questions.

1. Which data classes does activity "Create new admission" access?
   _____ (Patient, Admission)

2. Which data classes does activity "Complete Assessment" access?
   _____ (Admission)

3. (a) Do the sets of classes accessed by activities "Make Diagnosis' and "Create New Admission" intersect?
   YES ☐ NO ☐ (YES)
   (b) If they do, which data classes belong to their intersection?
   _____ (Patient, Admission)

4. Are there classes in the data schema that are never accessed by activities of the process? If so, which ones?
   _____ (Physician, Tag)

5. Which are the classes of the process used by the highest number of activities?
   _____ (Admission)

6. Which are the activities of the process that access class "Patient'?
   _____ (Create new admission, Make diagnosis)

7. Are there classes that are used only for read operations? If so, which ones?
   _____ (YES. Patient, Therapy)

Total Time: _____

Raw Results. Exercises have been corrected by adopting the most restrictive requirements for correctness, that is, exercises were considered correct if answers were both right and complete. That is, if one answer was only partially correct or incomplete, it was counted as being wrong.

The detailed results are shown in Fig. 11 and have been reported in the histograms shown in Fig. 9 of Sect 5.

EXERCISE 1 - Purchase order on a web pharmacy

GROUP 1 - WITH ACTIVITY VIEW

Q1	Q2	Q3	Q4	Q5	Q6	Q7	
Y	Y	Y	N	Y	Y	Y	14
Y	Y	Y	Y	Y	Y	Y	9,84
Y	Y	Y	Y	Y	Y	Y	9,87
Y	Y	Y	N	Y	Y	Y	11
Y	Y	Y	N	Y	Y	Y	11
Y	Y	Y	Y	Y	Y	N	13
Y	Y	Y	N	Y	N	Y	15
Y	Y	N	N	Y	N	Y	14
Y	Y	N	N	Y	N	Y	16
Y	Y	Y	Y	Y	Y	N	13
Y	Y	Y	Y	Y	Y	Y	13
Y	Y	Y	Y	Y	y	y	12
Y	Y	Y	Y	Y	Y	N	10
Y	Y	Y	Y	Y	Y	Y	11
Y	Y	Y	N	N	N	Y	14
Y	Y	N	N	Y	Y	Y	13
Y	Y	Y	Y	Y	Y	Y	12

AVERAGE TIME	12,45
% CORRECT ANSWERS	84,03
STANDARD DEV	1,832

GROUP 2 - WITHOUT ACTIVITY VIEW

Q1	Q2	Q3	Q4	Q5	Q6	Q7	TIME
Y	N	Y	N	Y	Y	N	23
Y	N	Y	N	Y	Y	N	24
N	N	Y	N	N	Y	N	24
Y	N	Y	Y	N	N	N	25
Y	N	N	Y	Y	N	N	25
Y	N	N	Y	Y	Y	N	24
Y	N	N	Y	N	N	N	24
Y	N	N	Y	Y	N	N	24
Y	N	N	Y	Y	N	N	24
Y	Y	Y	N	Y	N	N	13
Y	N	Y	Y	Y	Y	N	14
N	N	N	N	N	N	N	13
N	N	N	N	Y	N	Y	21
Y	N	N	N	Y	N	N	22,16
Y	N	N	N	Y	N	N	25
N	N	N	N	Y	N	N	27

AVERAGE TIME	22,01
% CORRECT ANSWERS	39,286
STANDARD DEV	4,502

EXERCISE 2 - Triage in Emergency Room

GROUP 1 - WITHOUT ACTIVITY VIEW

Y	N	Y	N	Y	N	N	13
Y	N	Y	N	N	N	N	12
N	N	N	N	N	N	N	14
Y	N	N	N	N	N	Y	14
N	N	N	N	N	N	N	14
N	N	N	Y	N	N	N	15
N	N	Y	N	N	Y	N	15
N	N	N	N	N	N	Y	15
N	N	N	N	Y	N	N	9
N	N	N	N	Y	N	N	10
Y	N	Y	N	Y	Y	Y	18
Y	N	Y	N	Y	N	N	22
Y	N	N	N	Y	N	N	19
N	N	N	N	Y	N	Y	19
Y	Y	Y	N	Y	N	N	17,45
Y	N	Y	N	Y	N	N	10
Y	N	N	N	Y	N	N	13

AVERAGE TIME	14,67
% CORRECT ANSWERS	28,57
STANDARD DEV	3,537

GROUP 2 - WITH ACTIVITY VIEW

Y	Y	Y	Y	Y	Y	N	8,5
Y	Y	Y	Y	Y	N	N	9
Y	Y	N	Y	Y	Y	N	8,6
N	N	Y	N	Y	N	N	5
Y	Y	Y	Y	Y	Y	N	12
Y	Y	Y	Y	Y	Y	N	13
Y	N	Y	N	Y	Y	Y	10
Y	N	Y	N	Y	Y	N	11
Y	N	Y	Y	N	N	N	14
Y	Y	N	N	Y	N	Y	13
Y	Y	Y	Y	Y	Y	Y	11,92
Y	Y	Y	N	Y	Y	N	7
Y	Y	Y	Y	Y	Y	N	14
Y	Y	Y	Y	Y	Y	N	8
Y	Y	Y	N	Y	Y	Y	9
Y	Y	Y	N	Y	N	N	11

AVERAGE TIME	10,314
% CORRECT ANSWERS	71,429
STANDARD DEV	2,6118

Fig. 11. Results of *Exercises 1* and *2* of *PHASE 2* corrected adopting the most restrictive requirements for correctness.

To evaluate the statistic significance of the obtained results, we applied the pared t-test to both execution times and answer correctness by matching the results of one subject without the Activity View, with those of the same subject with the Activity View. We obtained a p-value < 0.001 and thus, the results are statistically significant. The details of the calculation of p are as follows.

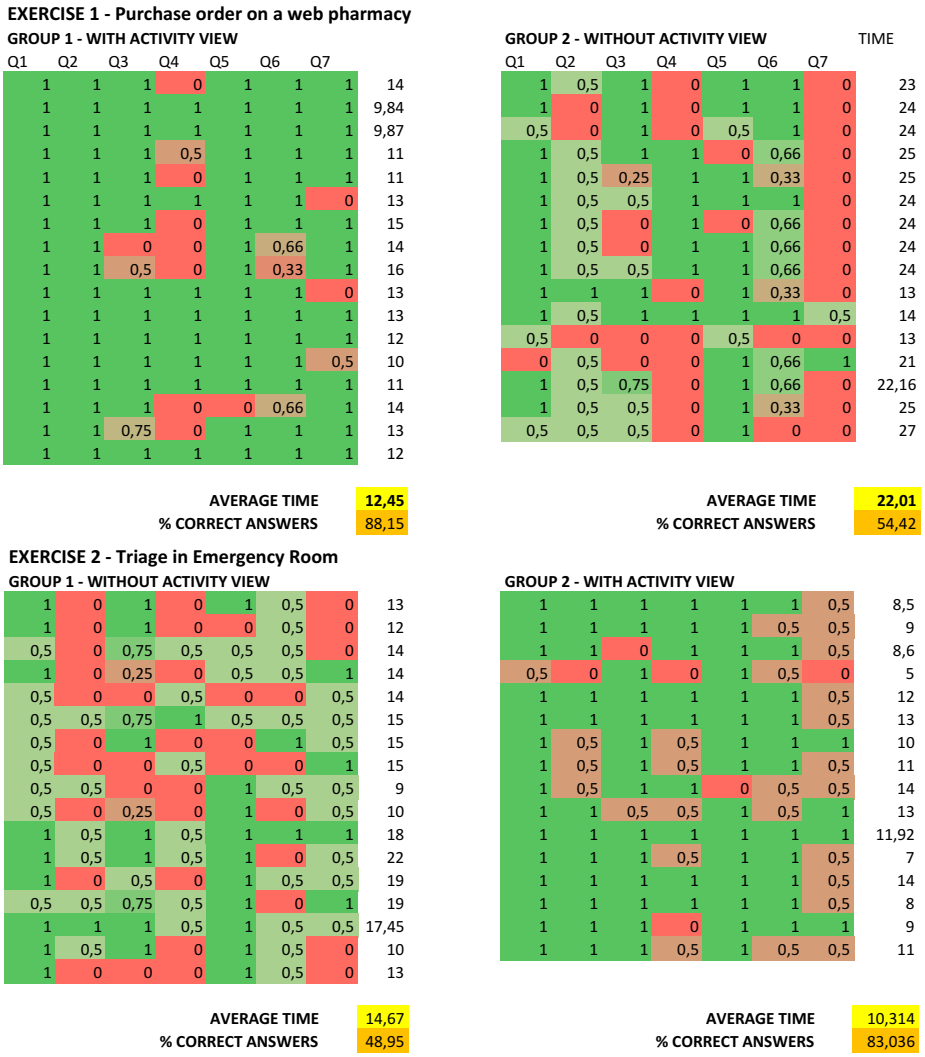
Paired t-test applied to Execution Times. The mean of the measurements without the Activity View minus the one with the Activity View is equal to -6.8145. The 95% confidence interval of this difference goes from -9.2361 to -4.3930. The intermediate values used in the calculations of the p-value are: t = 5.7322 degrees of freedom = 32, and standard error of difference = 1.189. Thus, the obtained two-tailed p-value is less than 0.001.

Paired t-test applied to Answer Correctness. The mean of the measurements without the Activity View minus the one with the Activity View is equal to -3.09. The 95% confidence interval of this difference goes from -3.74 to -2.45. The intermediate values used in the calculations of the p-value are: 9.5848 degrees of freedom = 32, and standard error of difference = 0.324 . Thus, the obtained two-tailed p-value is less than 0.001.

Once having corrected all the exercises, we organized a meeting with all the participating subjects to discuss results. The difference of execution times between *Exercise 1* and *Exercise 2*, especially for those not having the Activity View is probably due to some form of learning: Subjects claimed that they became familiar with looking at processes and data diagrams together, and they had learned how the task was structured.

Then, we discussed some of the questions that lead to the highest number of mistakes. In particular, we analyzed the results of questions *Q2* and *Q7* from *Exercise 1*, and question *Q4* and *Q7* of *Exercise 2*.

- Question *Q2* of *Exercise 1*: Most of the subjects of "Group 2" added class "Customer" to the answer, thinking that also the customer was needed to execute the activity. However, at a conceptual level, the operation involves only classes "Order" and "Payment", as the customer is known from the beginning of the process and the order already contains information about the customer identifier.
- Question *Q7* of *Exercise 1*: Most of the subjects of "Group 2" wrote that class "Drug" is only read by the process when, in reality, they are read by some operator to be inserted in the database, but there is no activity that reads them from the database.
- Question *Q4* of *Exercise 2*: Most of the subjects of "Group 1" forgot to report class "Tag" among those not used by the process and only put "Physician". Probably the fact the some tasks concern tag assignment was misleading. However, from the provided data schema it is clear that class "Tag" serves as data dictionary.



EXERCISE 1 - Purchase order on a web pharmacy

GROUP 1 - WITH ACTIVITY VIEW

Q1	Q2	Q3	Q4	Q5	Q6	Q7	
1	1	1	0	1	1	1	14
1	1	1	1	1	1	1	9,84
1	1	1	1	1	1	1	9,87
1	1	1	0,5	1	1	1	11
1	1	1	0	1	1	1	11
1	1	1	1	1	1	0	13
1	1	1	0	1	1	1	15
1	1	0	0	1	0,66	1	14
1	1	0,5	0	1	0,33	1	16
1	1	1	1	1	1	0	13
1	1	1	1	1	1	1	13
1	1	1	1	1	1	1	12
1	1	1	1	1	1	0,5	10
1	1	1	1	1	1	1	11
1	1	1	0	0	0,66	1	14
1	1	0,75	0	1	1	1	13
1	1	1	1	1	1	1	12

AVERAGE TIME 12,45
% CORRECT ANSWERS 88,15

GROUP 2 - WITHOUT ACTIVITY VIEW

Q1	Q2	Q3	Q4	Q5	Q6	Q7	TIME
1	0,5	1	0	1	1	0	23
1	0	1	0	1	1	0	24
0,5	0	1	0	0,5	1	0	24
1	0,5	1	1	0	0,66	0	25
1	0,5	0,25	1	1	0,33	0	25
1	0,5	0,5	1	1	1	0	24
1	0,5	0	1	0	0,66	0	24
1	0,5	0	1	1	0,66	0	24
1	0,5	0,5	1	1	0,66	0	24
1	1	1	0	1	0,33	0	13
1	0,5	1	1	1	1	0,5	14
0,5	0	0	0	0,5	0	0	13
0	0,5	0	0	1	0,66	1	21
1	0,5	0,75	0	1	0,66	0	22,16
1	0,5	0,5	0	1	0,33	0	25
0,5	0,5	0,5	0	1	0	0	27

AVERAGE TIME 22,01
% CORRECT ANSWERS 54,42

EXERCISE 2 - Triage in Emergency Room

GROUP 1 - WITHOUT ACTIVITY VIEW

1	0	1	0	1	0,5	0	13
1	0	1	0	0	0,5	0	12
0,5	0	0,75	0,5	0,5	0,5	0	14
1	0	0,25	0	0,5	0,5	1	14
0,5	0	0	0,5	0	0	0,5	14
0,5	0,5	0,75	1	0,5	0,5	0,5	15
0,5	0	1	0	0	1	0,5	15
0,5	0	0	0,5	0	0	1	15
0,5	0,5	0	0	1	0,5	0,5	9
0,5	0	0,25	0	1	0	0,5	10
1	0,5	1	0,5	1	1	1	18
1	0,5	1	0,5	1	0	0,5	22
1	0	0,5	0	1	0,5	0,5	19
0,5	0,5	0,75	0,5	1	0	1	19
1	1	1	0,5	1	0,5	0,5	17,45
1	0,5	1	0	1	0,5	0	10
1	0	0	0	1	0,5	0	13

AVERAGE TIME 14,67
% CORRECT ANSWERS 48,95

GROUP 2 - WITH ACTIVITY VIEW

1	1	1	1	1	1	0,5	8,5
1	1	1	1	1	0,5	0,5	9
1	1	0	1	1	1	0,5	8,6
0,5	0	1	0	1	0,5	0	5
1	1	1	1	1	1	0,5	12
1	1	1	1	1	1	0,5	13
1	0,5	1	0,5	1	1	1	10
1	0,5	1	0,5	1	1	0,5	11
1	0,5	1	1	0	0,5	0,5	14
1	1	0,5	0,5	1	0,5	1	13
1	1	1	1	1	1	1	11,92
1	1	1	0,5	1	1	0,5	7
1	1	1	1	1	1	0,5	14
1	1	1	1	1	1	0,5	8
1	1	1	0	1	1	1	9
1	1	1	0,5	1	0,5	0,5	11

AVERAGE TIME 10,314
% CORRECT ANSWERS 83,036

Fig. 12. Results of *Exercises 1* and *2* of *PHASE 2* corrected by giving scoring also to partially correct answers.

– Question *Q7* of *Exercise 2*: Most of the subjects of both groups forgot to report class "Therapy" beside "Patient".

In Fig. 12 we report the same results corrected by adopting a less strict method. That is, we considered each answer to be worth one point, but assigned partial scores depending on how many correct sub-answers were provided. Indeed, since some of the answers asked for multiple activities/classes, we could divide the one point assigned to each answer for the number of required sub-answers and assign partial scores accordingly.

For instance, let us consider question Q4 of Exercise 2. The correct answer is "Physician, Tag". According to our point assignment, each class is worth 0.5 points: 0.5 +0.5 = 1 point. If somebody wrote only "Tag" we would score the answer with 0.5 points: 0 + 0.5 = 0.5.

However, to avoid counting as correct answers that included wrong sub-answers beside the correct ones, we subtracted 0.5 points from the overall answer score for each additional wrong sub-answer. If somebody answered question Q4 of Exercise 2 "Physician, Tag, Therapy", we would give the question 0.5 points: 0.5 + 0.5 - 0.5 = 0.5. However, we did not allow scores to be negative, that is, if the combination of sub-answers was completely wrong to reach a negative score, we still gave 0 points to the overall answer.

Compared to the results of Fig. 11, relaxing the requirements for correctness leads to an increased percentage of correct answers, as expected. However, the difference between the two groups remains significant. We applied the *paired t-test* and compared the correctness of the exercise with the Activity View to the one of the exercise solved by the same subject but without the Activity View.

The computed p-value remains < 0.001, with a mean correctness of 85.67 for exercises solved with the Activity View, and a mean correctness of 51.60 for exercises solved without the Activity View.
Paired t-test applied to Answer Correctness. The mean of the measurements without the Activity View minus the one with the Activity View is equal to -34.0692. The 95% confidence interval of this difference goes from -40.6335 to -27.5049. The intermediate values used in the calculations of the p-value are: t = 10.5719 degrees of freedom = 32, and standard error of difference = 3.223. Thus, the obtained two-tailed p-value is less than 0.001.

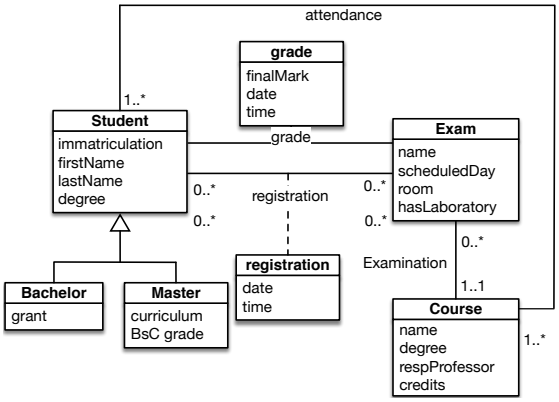
A.3 PHASE 3 - Evaluating Ease of Use

The last phase of the experimental evaluation, *PHASE 3* was meant to evaluate the use of the Activity View during process design. This time, all the subject were given the same exercise. In detail, they were asked to model a simple BPMN process according to a provided textual description, and to write the Activity Views that connected the modeled process with the provided schema of the domain database. The text of *Exercise 3* is provided below.

Exercise 3 - Modeling a student examination

Design a BPMN process diagram that corresponds to the following description.
Let us consider the process of student examination, from the perspective of a professor. For simplicity, let us assume to have a single student, willing to take an oral exam, and a single examining professor.
The first activity of the process is the definition of the day of the exam. The professor must add to Esse 3 all the possible dates for the exam, considering the exam name, scheduled day, room, and the possibility of having a lab session.
Then, when the student comes on the exam day, the professor must check if the student has registered. This is done by checking that in Esse3 there is some registration corresponding

to the students immatriculation number for the given exam. (For simplicity, we assume that all students are registered prior to take the exam). Then the professor examines the student. Finally, when the exam is over, the professor grades the student and registers all the details of the grade in Esse3.



Task: In the space below, design the BPMN diagram of the process and model persistent data access with Activity Views.

Exercise 3 was corrected by assigning a maximum of one point for the correct BPMN process, and a maximum of one point for each correct Activity View. We expected the subjects to write a process having three main activities and three simple Activity Views.

A sample solution of the process is provided in Fig. 13, while a couple of solutions designed by the students are shown in Fig. 14.

Results of corrected *Exercise 3* are reported in Fig. 15. Common mistakes stemmed from the use of a different access times, which were probably hard to understand from the text of the exercise. Moreover, several subjects included more classes and associations than needed. The results of this last exercise were in line with the closing questionnaire, where we asked to provide an overall opinion of the proposed model and to suggest possible improvements.

The questionnaire is provided in Fig. 16, while answers are summarized in Fig. 10 of Sect. 5. Overall, all the subject found the exercise with the Activity View easier than the one without the model. As for attributes of the Activity View perceived as superfluous, two people found attribute *AccessTime* not useful. Instead, the attributes perceived as hardest to understand/write were *AccessTime* (6 people) and attribute A_{set} (5 people).

Last but not least, we asked for suggestions related to the graphical representation of the Activity View, as our goal is to provide a compact representation of the Activity View that could be easily blended within existing process modeling tools. Somebody suggested to encode data operations trough graphical symbols,

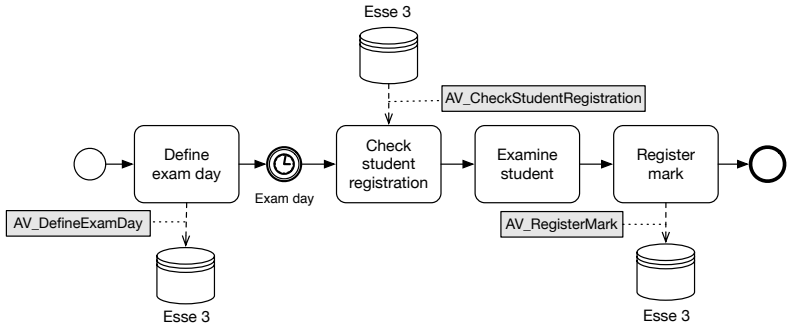
to shorten their description. As for other comments, a couple of people pointed out that a minimum of experience in database design is required to be able to understand the Activity View completely, while one person expressed the concern that the Activity View may become more complex for bigger and articulated processes.

A.4 Final Considerations

Overall, the results of the experimental evaluation were encouraging and provided a good starting point for understanding practical needs related to the joint design of processes and data.

In general, providing a mapping towards consolidated approaches at the logical level could help clarifying the need for some attributes, such as Access Time and NumInstances, which were identified as the hardest to understand and use. Moreover, the integration of the Activity View in existing modeling tools would be a great step in terms of conveying its meaning and need.

Our goal is to continue with the refinement and evaluation of the Activity View by involving people coming from more variate backgrounds.



AV_DefineExamDay

TUPLE	CLASS SET	ASSOC SET	ACCESS TYPE	ACCESS TIME	INSTANCES
T1	{Exam(*)}	∅	I	During	(1,*)

AV_CheckStudentRegistration

TUPLE	CLASS SET	ASSOC SET	ACCESS TYPE	ACCESS TIME	INSTANCES
T1	{Student(immatriculation), registration(*), Exam(name)}	registration	R	During	(1,1)

AV_RegisterMark

TUPLE	CLASS SET	ASSOC SET	ACCESS TYPE	ACCESS TIME	INSTANCES
T1	{grade(*)}	∅	I	During	(1,1)

Fig. 13. Sample solution with timer events.

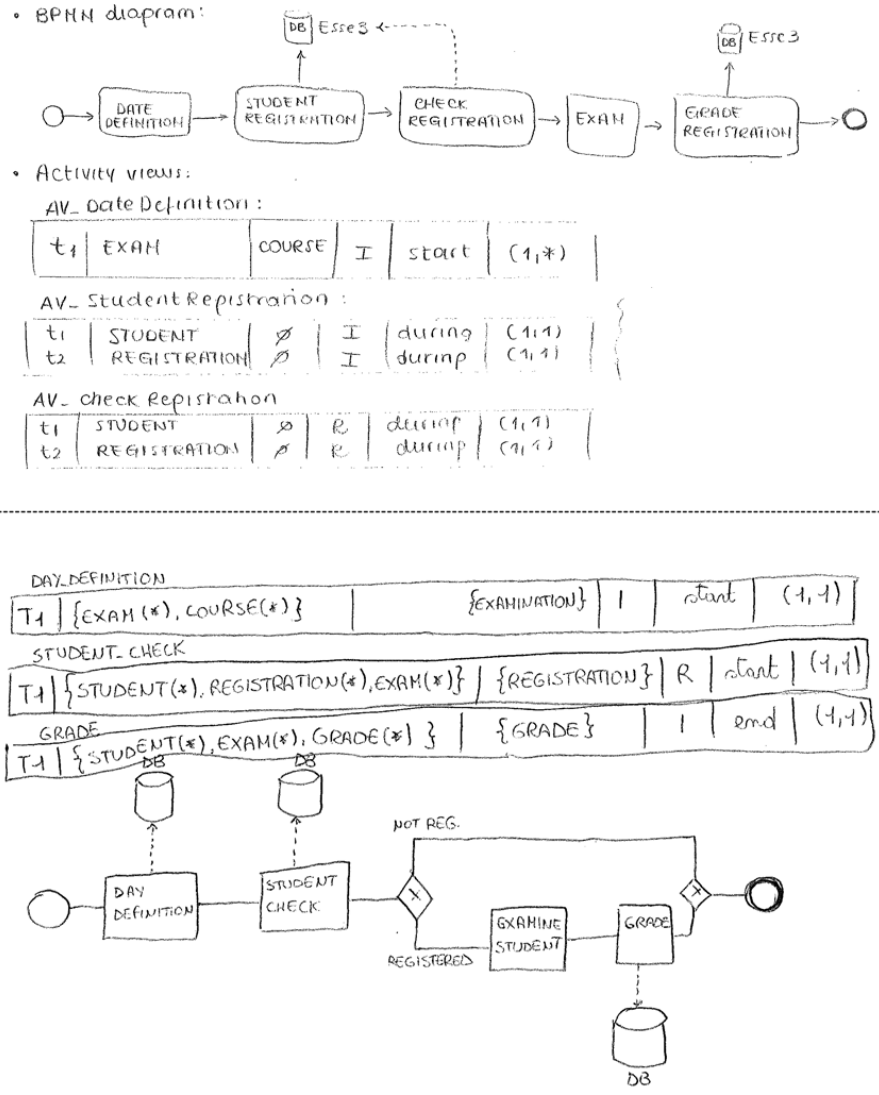


Fig. 14. A couple of solutions to *Exercise 3* provided by the participating students.

Modeling Exercise

BPMN	AV 1	AV 2	AV 3	Tot	Tot AV
1	0,6	0	0,2	1,8	0,8
1	0,8	0,2	0,6	2,6	1,6
1	1	1	0,6	3,6	2,6
1	0	0	0	1	0
1	1	1	0	3	2
1	0,8	0,8	0	2,6	1,6
1	0,8	1	0,8	3,6	2,6
1	0,6	0,8	0,6	3	2
0,5	0,4	0,2	0	1,1	0,6
1	0,8	1	0,6	3,4	2,4
0,8	0	0	0	0,8	0
0,8	0	0	1	1,8	1
0,8	0,8	0,6	0,8	3	2,2
0,8	0,8	0,8	0,8	3,2	2,4
0,8	0,8	0,8	0,6	3	2,2
1	0,6	1	0,8	3,4	2,4
1	1	0,8	0,4	3,2	2,2
1	1	1	0,6	3,6	2,6
1	1	1	1	4	3
0,8	0,8	0,4	0,4	2,4	1,6
1	0,8	0,6	0,6	3	2
1	0,6	1	0,6	3,2	2,2
1	1	1	0,8	3,8	2,8
0,8	0,6	0,4	0,4	2,2	1,4
1	0,6	0,6	0,6	2,8	1,8
1	0,4	0,2	0,4	2	1
0,8	0,4	0,4	0	1,6	0,8
1	1	0,8	0,6	3,4	2,4
0,8	0,2	0	0,6	1,6	0,8
1	1	0,4	0,6	3	2

Overall Accuracy	**67,25**
Activity View Accuracy	**58,89**
	83,94

Fig. 15. Results of corrected *Exercise 3*.

Summary Questions

Which exercise did you find more difficult to execute? The one with or without the activity view?

.....

Express your opinion.
Give a score of 1 to 5 to the following statements, where 1 = strongly disagree, 2 = disagree, 3 = neutral, 4 = agree, 5 strongly agree

Do you think that the activity view can improve the integrated modeling of processes and data?

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Do you think it is necessary to have knowledge of database modeling to understand/use the activity view?

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

How much to you find the activity view…

Generalisable (adaptable to the context of different application domains).

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Easy to read.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Easy to understand.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Easy to write.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Easy to use.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Answer the following questions and fill in the blanks, when needed.

Is there any attribute of the activity view that you perceived as superfluous?

No ☐ Yes ☐ .....

Is there any attribute of the activity view that you consider more difficult to understand/ write than others?

No ☐ Yes ☐ .....

Comments

Do you have any suggestions to improve the graphical representation of the activity view?

.....
.....
.....

Other comments

.....
.....
.....

Fig. 16. Questionnaire conclusive of the whole experimental evaluation