Towards a Holistic Method for Business Process Analytics

Gianna Reggio, Maurizio Leotta, Filippo Ricca, Egidio Astesiano

Abstract:

In this paper, we propose a holistic approach aimed at combining business process modelling and data-driven business process improvement. The first step requires to develop a "precise" model of the processes of the organization using the UML. Precise means that all business entities involved in the process are determined as well as all the tasks composing the process executions, and all relevant data about them are modelled. Then, a model of the data space of the process will be derived taking into account also the quantitative aspects of each process (e.g. how many instances of the process will run each day?, how many possible instances of some business entities will be around or have ever been created?). In this way it is possible to conceive and design various analyses and improvements of the process based on its data, since all the aspects related to each business process have been explicitly modelled, and in a sufficiently formal way. We will introduce our approach using a small case study: the Buying process having as participants manufacturers, dealers, shippers, and payment systems.

Digital Object Identifier (DOI):

`http://doi.org/10.1007/978-3-319-61994-1_16`

Towards a Holistic Method for Business Process Analytics

Gianna Reggio, Maurizio Leotta, Filippo Ricca, Egidio Astesiano

Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi (DIBRIS)
Università di Genova, Italy

gianna.reggio@unige.it, maurizio.leotta@unige.it,
filippo.ricca@unige.it, egidio.astesiano@unige.it

Abstract. In this paper, we propose a holistic approach aimed at combining business process modelling and data-driven business process improvement. The first step requires to develop a "precise" model of the processes of the organization using the UML. Precise means that all business entities involved in the process are determined as well as all the tasks composing the process executions, and all relevant data about them are modelled. Then, a model of the data space of the process will be derived taking into account also the quantitative aspects of each process (e.g. how many instances of the process will run each day?, how many possible instances of some business entities will be around or have ever been created?). In this way it is possible to conceive and design various analyses and improvements of the process based on its data, since all the aspects related to each business process have been explicitly modelled, and in a sufficiently formal way. We will introduce our approach using a small case study: the Buying process having as participants manufacturers, dealers, shippers, and payment systems.

1 Introduction

In the last decade, the availability of massive storage systems, large amounts of data and the advances in several disciplines related to data science such as data mining, data analytics, business intelligence, and machine learning provided powerful tools for potentially improving the business activities of the organizations. Indeed, the management can leverage the large amounts of data for extracting, by means of different techniques, useful information with the aim of improving the business processes and related activities.

To this end a variety of Business Process Management Systems (BPMS) have been investigated and proposed, that usually include a component dealing with Process Business Analytics (BPA), for collecting and analysing the process-related data to answer some process-centric questions (see, e.g. [16] and [2]).

Anderson [1] hints at the assumed canonical steps to be followed for a data analytics task: "*a) developing questions to be answered, b) curating the potential data sources, c) collecting data from these sources, d) cleaning the collected data, e) storing it, f) processing/analysing the data, and then g) displaying and visualizing the data in response to queries.*".

Most often, organizations start introducing "analytics" from bottom, i.e. by trying to do some analysis on data that they have already at hand and found inside the many different systems and storage means available. However there has been a lot of work for organizing process data in a way suitable to answer the questions relevant to BPA (see for references Ch. 5 in [2]). As a notable example, in [3] a process data warehouse schema is presented that is derived as an abstraction from the experimental analysis of a variety of effectively running business processes, together with their related BPA questions. That work "makes possible a unified approach to reporting and analysis, so that the effort for setting up the analysis for a new process or customer consists mostly of customization rather than development". From a different angle, the decision-making viewpoint, in [7] a goal-oriented approach is pursued addressing the first point raised by Anderson "developing questions to be answered". We are looking at the problem, in particular, from still a different perspective, trying to see whether the level of abstraction can be raised further. We have in mind what happens in the software engineering field where best practices require structured development processes composed of various steps such as domain modelling, capture and specify requirements, design the software, and then implementation, made using high-level programming languages, reusing design patterns, and putting together large pieces of software, such as components and services, when not just by transforming models into running code (e.g. when following a model-driven approach [13]).

The lack of a well-defined engineered approach can lead to various problems:

- the collected data could be inadequate or excessive for useful analysis; in the first case it is not possible to answer the relevant questions, while, in the latter case, the organizations sustain unnecessary storage costs;
- connections between business processes, stakeholder's goals, and stored data could be ill-defined;
- the right questions for improving the business activities could be difficult to find.

In this paper, we propose a first step towards a holistic approach aimed at combining business process modelling and data-driven business process improvement and able to overcame the problems mentioned above.

The remainder of the paper is organized as follows. Section 2 provides an overview of the method while the subsequent Sections 3, 4, 5 sketch the first steps of our method (i.e. respectively the creation of the Precise Business Process Model, of the Business Process Data Space Model, and definition of the relevant questions). Finally, Section 6 concludes the paper.

2 Method Overview

An attempt at analysing an already running business process may start just by (a) looking at the available, more or less raw, data, (b) discussing with the stakeholders relying on their know-how, (c) examining the existing documentation of the process (e.g. a model of the process itself). The first point (a) may lead to concentrate on analysing aspects of the process expressed at a low-level (e.g. the mean time between two low-level messages, or expressing relationships between the codes classifying different categories of clients). The second point (b) may generate the classical problems associated with ambiguous terminology, hidden assumptions and misunderstandings between stakeholders and the analyst. The third point (c), if the documentation (e.g. a process model expressed using some notation) is of good quality and quite extensive, may provide a mean to conceive the relevant questions on the process and communicate them to the analyst, and in the meantime could also help present the results of the analysis in a way suitable for the stakeholders.

For the above reasons and as well-recognized in the literature (see e.g. [2]), we think that a good starting point for the analysis of a process may be its model (i.e. not limiting only to the (a), (b), (c) activities). Nowadays there are many different proposals for business process modelling, mainly based on graphical notations, e.g. BPMN [5], EPC [15, 14], and UML [6]. UML is widely known and used [12, 10], and provides many different diagrams suitable to cover all aspects of a business process such as data and participant features (in contrast, e.g. to BPMN, whose specification [5] explicitly states "*Therefore, the following are aspects that are out of the scope of this specification: ... Data and information models ...* "); furthermore, by means of the profile mechanism, UML provides a way to define its own variants, thus it may be easily tailored for this specific modelling task. The above considerations lead us to prefer the UML, which allows to model any aspect of a business process.

Any business process modelling notation, or better modelling method [8] represents the process dynamics as a workflow of basic chunks of activity, that may be either actions (not better qualified) or message exchanges, or events either instantaneous or having a duration. The granularity level of these chunks will be reflected in the granularity of the elicited questions, and thus of the designed process analysis. If the chosen modelling method allows to decompose and/or compose these chunks, it will be possible to arrange their granularity.

In some previous papers, we have developed some modelling methods for business processes based on the precise use of the UML, following the service oriented paradigm [9], where the basic unit of activity was the message exchange between two business entities [11], and another one based instead on tasks to which various entities may cooperate that we will briefly sketch in Sect. 3.

Fig. 1 summarizes the activities required by our method to enhance a business process using analytics technique on its data (here, for the sake of simplicity, we consider only a business process but usually there are many of them sharing participants and data).

The first step is to produce a precise UML model of the process itself (and this can be done either on a process "to-be" or an existing one). Precise means that all business entities involved in the process are defined, as well as all the basic actions composing the process executions, and all relevant data about them are modelled.

Then, a model of the *data space of the process* [2] will be derived taking into account also the quantitative aspects of each process (e.g. how many instances of the process will be run each day?, how many possible instances of some business entities will be around or have ever been created?). The data space is related to all possible data that can be generated by the executions of the process, as defined by its model, i.e. by its instances. Moreover, it is described at the same abstraction level, following the same style, and using the same terminology of the business process model. Thus it is suitable to express all possible questions on the process itself in a way understandable by the business process model's stakeholders.

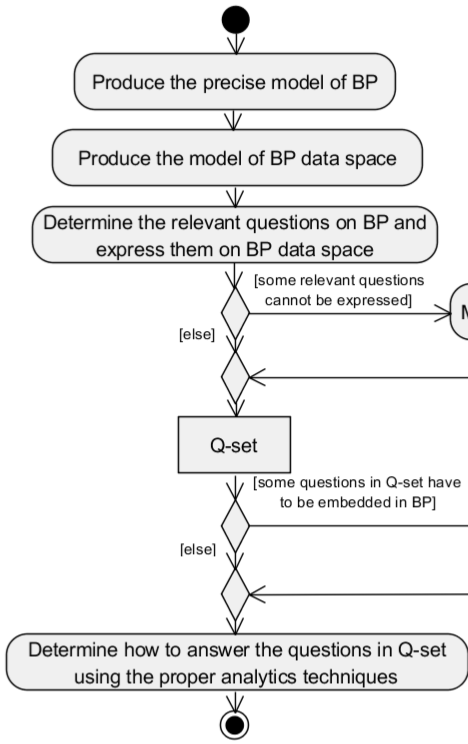


Fig. 1. Our Method Proposal for Business Process Analytics

Subsequently, it is required to determine the relevant questions on the business process expressed using the process data space and following one of the two strategies provided by our approach ("Bottom-up" and "Top-down").

Once the list of questions to answer is produced, we may modify the process to be sure to collect and store all the relevant data, and these modifications will be reflected on its model. Moreover, the activity of generating the relevant questions can be considered also a kind of inspection of the process model. This could lead to discover, already at this point, some problems in the process, before to start the analytics job.

Finally, the method requires to determine which theories, algorithms and analytic approaches allow to answer the selected questions, and to select the best among them. To support this task, we intend to develop what we call *analytics patterns*, i.e. patterns aimed at proposing a solution to a business process analytics problem (as the famous design patterns [4] used in software development).

At the end there will be available an abstract description, akin a kind of specification, of the data to be collected and to be analysed, together with the needed analyses and how to perform them. Later, all these elements should be implemented over the systems supporting the process, but having a guide to select which data to collect, and which kind of analytics tool to integrate.

Having modelled in an integrated way business process and their data spaces, defined the analytics activities, allows to trace the changes among them and thus simplifies the maintenance activities and helps in case of business evolution.

3 Business Process Precise Model

The first step of our method requires to produce a model of the process itself (and this can be done either of a process "to-be" or of an existing one). The term *business process* denotes the activities performed by some businesses, or more in general by some organizations, for achieving specific goals; such activities will be structured in terms of basic chunks of activity (*tasks*) related by the classical control flow operators, such as sequencing, parallelism, and conditional choice. In a business process there are a number of entities that perform activities or are manipulated by such activities, that we call *business entities*. We classify the business entities in *workers*, *systems*, and *business objects* (they will be detailed in Section 3.1). A business is usually structured in several processes, which may be mutually interfering since they may involve the same business entities, here for simplicity we consider the modelling of a single process in isolation.

The structure of the models of the business processes is defined in Fig. 2 by means of a UML class diagram[1].

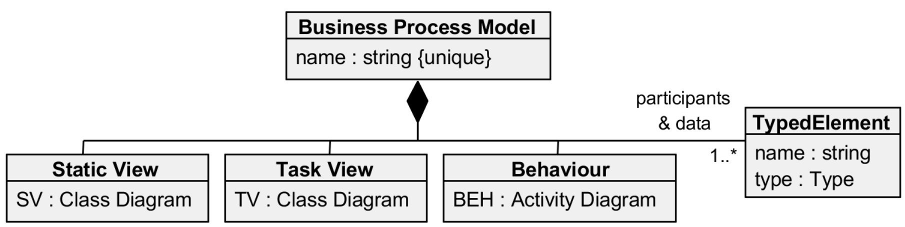


Fig. 2. Business Process Model Structure

In the following, we will detail the various parts of a business process model, exemplifying them on the business process Buying. The Buying business process concerns a business community including three primary kinds of parties: the dealers, the manufacturers and the shippers. They are independent parties, but they want to work together. All the manufacturers sell a unique kind of product (e.g. gasoline), the dealers buy that product from the manufacturers, and the shippers deliver it to the dealers. When a dealer wants to buy a certain quantity of the product from a manufacturer, it may before ask to the manufacturer for the current price, and decides if it is convenient. In the case of an affirmative answer, it will place an order to that manufacturer to buy some quantity of the product. When a manufacturer receives an order, it will check the product availability. If the ordered amount of product is available, the order will be accepted, otherwise, the order will be rejected. The dealer will pay for the order. If the payment ends successfully,

[1] In this paper, we follow the convention that in a UML class diagram the multiplicity of an association will be omitted whenever it is equal to 1.

the manufacturer will send a confirmation to the dealer, and will ask a shipper to deliver the product to the dealer; otherwise, i.e. if the payment failed, the manufacturer will cancel the order. When a shipper receives a shipping request from a manufacturer, it may refuse, otherwise it will inform the manufacturer of the date for picking up the shipment. Once the shipment is delivered to the dealer, the shipper will send a confirmation to the manufacturer.

3.1 Static View

The static view is a UML class diagram defining the classes typing the business entities (i.e. they will be modelled by objects typed by such classes), and their mutual relationships (modelled by associations among the corresponding classes).

In the static view, we use the following class stereotypes to classify the business entities in the three categories:

- ≪worker≫: (human) entities that perform the basic actions of the business;
- ≪system≫: entities corresponding to hardware or software systems taking part in the business;
- ≪object≫: those entities over which the basic actions of the business are performed;

The three stereotypes defined above are mutually exclusive. We name *entity class* any class stereotyped with either ≪worker≫ or ≪system≫ or ≪object≫.
The business entities (types) relative to our running example Buying are modelled by means of the classes appearing in the static view shown in Fig. 3. More in detail:

- Manufacturer, those that provide the product sold in the network (remind that a unique kind of product is commercialized, e.g. gasoline);
- Dealer, those that buy the product from the manufacturers;

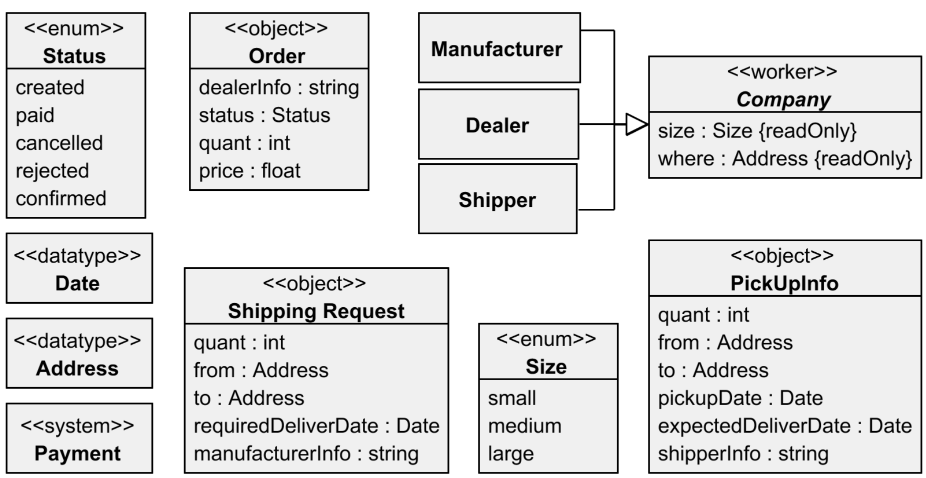


Fig. 3. Business Process Buying Model: Static View

– Shipper, those able to move the product from the manufacturers to the dealers;
– Payment not further detailed system supporting the handling of payments between two parties, e.g. by means of credit card or Paypal;
– Order the orders sent by the dealers to the manufacturers;
– ShippingRequest the documents sent by the manufacturers to the shippers for asking the transportation of products to the dealers;
– PickUpInfo the info sent back by a shipper when accepts a request.

Notice that only the relevant info about the various entities is considered in the model, that reflects the view of the stakeholders, and will influence the subsequent analytics activities.

3.2 TaskView

The task view is a class diagram containing classes stereotyped by ≪task≫ modelling the basic activities of the process, and any other class/datatype needed to define them, but the latter must have been already introduced in the static view.

The participants of a task, that are roles typed by entity classes and not specific instances, are represented by means of associations connecting the task class with participant classes, and the association end attached to the entity class should be the participant name.

If a task participant should be instantiated before the task may start (whereas others may be determined by the task activities, e.g. when an order is created or a shipper selected), then it will be stereotyped by ≪in≫ (short for input). A task class may have any number of attributes typed by datatypes; they represent the data used by the task; similarly to the participants they may be stereotyped with ≪in≫. A task class may be characterized by means of pre/post/invariant constraints concerning its participants and used data (expressed using the OCL).

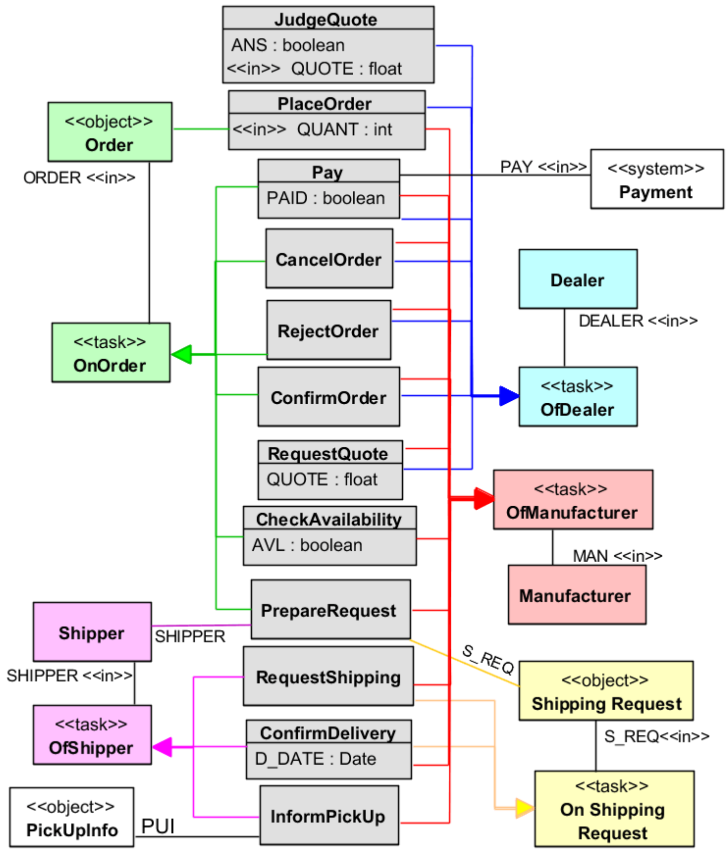
The task view of the process Buying is shown in Fig. 4. For instance, both Dealer and Manufacturer take part in the RequestQuote task given that Dealer may before ask to Manufacturer for the current price of the product (i.e. executing the task RequestQuote). Once the QUOTE has been received from Manufacturer, Dealer performs the JudgeQuote task. Such task has a pre-condition constraint concerning the admissible values of the QUOTE (i.e. QUOTE must be greater than 0).

3.3 Behaviour and Participants/Data List

The behaviour of a business process is modelled by a UML activity diagram. The behaviour of the business process Buying together the list of participants and used data is shown in Fig. 5.

Notice that the participants/data of a business process are roles and not specific individuals, and are modelled by pairs consisting of an identifier and a class/datatype (see Fig. 2); their list is inserted in a UML note[2] put near the activity diagram modelling the process behaviour (see Fig. 5 bottom). A process participant/used data role, similarly

[2] Represented by the icon of the paper leaf with a bended corner



context JudgeQuote
  pre: QUOTE$>$0
context PlaceOrder
  pre: QUANT$>$0 and PRICE$>$0
  post: ORDER.status=created and ORDER.price=PRICE and ORDER.quantity=QUANT
context CancelOrder
  pre: ORDER.status = created   post: ORDER.status = cancelled
context RejectOrder
  pre: ORDER.status = created   post: ORDER.status = rejected
context ConfirmOrder
  pre: ORDER.status = created   post: ORDER.status = confirmed
context RequestQuote
  post: QUOTE$>$0
context RequestShipping
  pre: ORDER.status = confirmed

Fig. 4. Process Buying Model: Task View

to the case of the tasks, may stereotyped by ≪in≫ in the case it is mandatory to provide it for the process being able to be executed.

Our method suggests to use only the following subset of the available activity diagram constructs, that are enough to model the business processes: action node, control node (i.e. initial, decision/merge, fork/join, flow and activity final), time event, control flow, and rake construct (that allows to reuse an activity defined elsewhere by means of another activity diagram).

Action nodes in the activity diagram modelling the behaviour of a process correspond to basic tasks, and will be labelled by instances of the task classes.

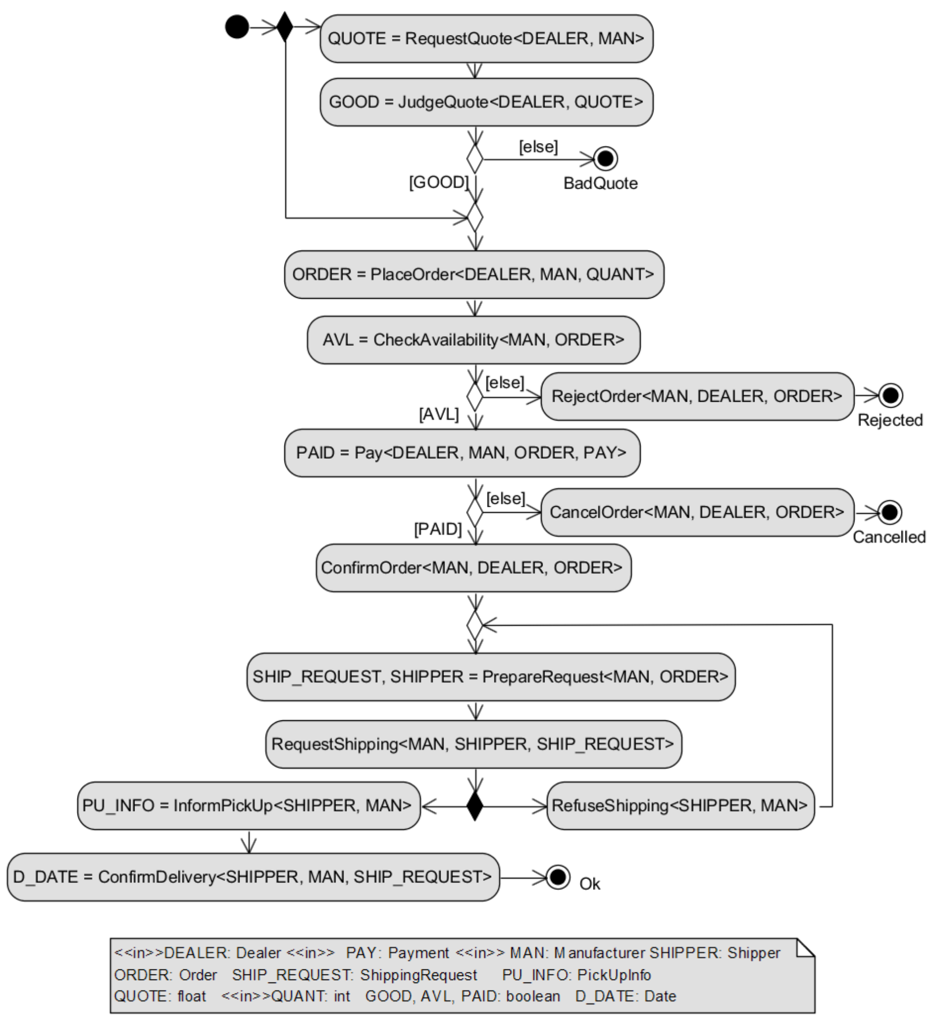


Fig. 5. Business Process Buying Model: behaviour plus participants and used data list

Task instances, used to label action nodes, are defined as follows. Let TC be a task class whose participants and used data are $\mathsf{X}_1 \colon \mathsf{T}_1 \ll\mathsf{in}\gg, \ldots, \mathsf{X}_n \colon \mathsf{T}_n \ll\mathsf{in}\gg, \mathsf{X}_{n+1} \colon \mathsf{T}_1, \ldots, \mathsf{X}_m \colon \mathsf{T}_n$, $\mathsf{E}_1, \ldots, \mathsf{E}_n$ be well-formed OCL expressions having type $\mathsf{T}_1, \ldots, \mathsf{T}_n$ respectively, and $\mathsf{Y}_{n+1} \colon \mathsf{T}_{n+1}, \ldots, \mathsf{Y}_m \colon \mathsf{T}_m$ be participants/used data of the process, then

$\mathsf{Y}_{n+1}, \ldots, \mathsf{Y}_m = \mathsf{TC}<\mathsf{E}_1, \ldots, \mathsf{E}_n>$

represents the instance of TC determined by $\mathsf{E}_1, \ldots, \mathsf{E}_n$ whose outputs will be assigned to $\mathsf{Y}_{n+1}, \ldots, \mathsf{Y}_m$ respectively.

A name must be assigned to each final node, to discriminate between successful and failed process terminations, and among the different reasons for terminating or failing.

Furthermore, business processes require often non-deterministic choices among several alternatives and, since the UML activity diagram does not offer a specific construct to represent it, but it may be obtained by a choice where all the guards of the alternatives are true, we introduce the black diamond to represent such construct (see for instance the black diamond near the initial state in Fig. 5).

4 Business Process Data Space

A business process model, as the one presented in Sect. 3, is in some sense a template for the many possible process executions, since it describes how the process works, precisely defining who is acting, which actions are performed and over what. A business process execution is named *(business) process instance*. Thus, for each business process several instances can exist, that will be executed at different times as well as simultaneously.

Let BP be a business process whose precise model is BPMod. A BP's *instance* is determined by instantiating all the process participants and data marked by ≪in≫ by business entities and values of the proper types, and will be described by the flow of task instances and consequently modifications on participants/data as defined by BPMod.

The *data space* of BP at a certain time is given by the all the possible data about all its instances (terminated or in execution) existing at that moment.

The data space of BP is modelled by a UML model (consisting of just a class diagram) that can be automatically derived by means of a model transformation from BPMod (the model of BP), and that it is sketched in Fig. 6 (it describes the generic structure of the Business Process Data Space Model). Obviously, some well-formedness constraints have to be added (e.g. the end time of an instance should be later than its begin time, or the type of the entity instances related with the participants are those expressed by BPMod). The classes whose name is slanted are abstract classes.

The data space of BP includes:

- *Entity Instance*: the data about the business entities involved at least in one BP's instance, a class of instances for each entity class in BPMod (in Fig. 6 we have reported only the sample class of the instances of EC1). The data about an entity instance are a set of snapshots (determined by the values of the entity class attributes) and of task instances (those in which the entity takes part).
- *Task Instance*: a class of task instances for each task class present in BPMod (in Fig. 6 we have reported only the sample class of the instance of TC1), each one

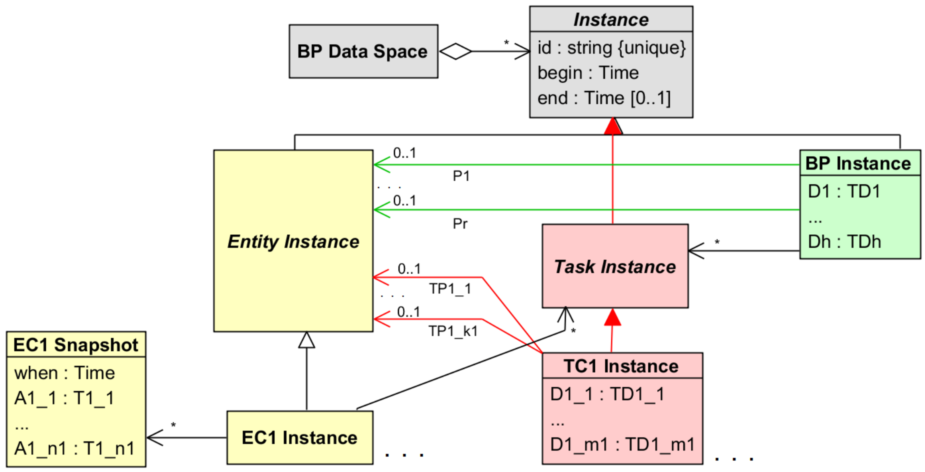


Fig. 6. Generic Business Process Data Space Model structure

characterized by its participants (represented by associations, for example TP1_1, . . . , TP1_k1 in Fig. 6) and data (represented by attributes, for example D1_1, . . . , D1_m1 in Fig. 6).

– BP *Instance*: each process instance is characterized by the participants and used data, as for the task instances, and by a set of task instances.

More in details (again referring to Fig. 6):

– P1, . . . , Pr and D1, . . . , Dh are respectively the participants and used data of BP.
– EC1 (with attributes A1_1, . . . , A1_n1), . . . are the entity classes in the static view.
– TC1 (whose participants and used data are TP1_1, . . . , TP1_k1 and D1_1, . . . , D1_m1 respectively), . . . are the task classes part of the task view.

All the instance info part of the process data space are enriched with a unique identity, and two time stamps, precisely the begin and the end of each process/task/business entity (end time is optional since the instances may be still running), whereas the entity snapshots are annotated with the time are taken.

Once produced the model of the data of the business process to enhance, the method requires to perform a dimensional analysis of it, that requires to elicit and state conditions on the minimum/maximum number of the various kinds of instances part of the process data space along the time. The dimensional analysis will be useful to understand whether the cardinality of the collected process data is so high to require big data techniques or not, and to decide which techniques to use to perform the data analysis.

Considering a hypothetical business process Buying, and assuming the following business community, the result of the dimensional analysis is:

Instances	Estimation
Buying instances	$\leq$ 15.000 created each day for up to 10 years
Manufacturer instances	$\leq$ 100
Shipper instances	$\leq$ 10.000
Dealer instances	$\leq$ 25.000
Payment instances	4
business object classes instances	3 $\times$ the number of the Buying instances

Notice that the workers and the systems instances are neither created nor destroyed by the Buying process, and thus their number may be assumed constant; the 4 payment systems are Visa, Mastercard, Bancomat and PostPay, and each process instance at most creates three business objects (i.e. order, shipping request, and pickup info).

So, we see that in this case the volume of the process data is not "big" (only 15.000 Buying instances per day), and so traditional storing and handling technologies may be used.

It is however easy to consider different processes, for instance concerning global e-commerce transactions, where the data space will be truly "big" (in the order of tens of billions of transactions/year).

5 Business Process Analytics

At a very abstract point of view, the application of analytics techniques to a business process (or more in general to all the processes of a business) means to be able to answer some *questions* concerning the process itself in a sound way. Analytics techniques can be used to:

– *discover hidden knowledge useful to improve the process itself* (e.g. referring to the Buying process, does the shipper company size (small/medium/big) influence the delivery date);
– *detect problems* (e.g. the rate of the Buying instances failing because the order is rejected due to a lack of product is too high);
– *forecast how the process will behave in the future* (e.g. is the quantity of product bought each day going to increase?).

To make the question elicitation a systematic activity, and to ground future work on supporting tools, such as a wizard guiding the elicitation, we give a conceptual model for the *questions* in Fig. 7.

A question should concern some *knowledge item* definable on the process data space, either *indicator*, i.e. a quantity computable on the process data space, or a *relationship*, i.e. a mapping between some quantities (at least one) and another one computable on the process data space. For example, the mean duration of the process instances, and the rate of female clients are indicators, whereas the number of items sold each day/month[3], the mapping between age and gender of the clients and the number of process instances in which they took part, and the mapping between zip code of the clients and the number of failed process instances are relationships.

[3] Mapping between days/months and number of items

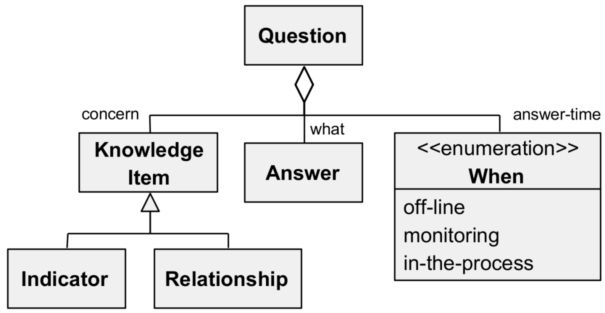


Fig. 7. Form of the questions

To ensure that the found indicators/relationships are computable over the process data, they should be expressed by means of OCL expressions defined over the model of the BP data space, also enriched with auxiliary classes, operations, and attributes. For example the duration of an instance may be modelled by means of a derived attribute Duration: Time of the class *Instance* defined by the OCL expression:

if not temination.isUndefined() then termination–begin else undefined.

The *answer* component of a question expresses what should be said about the knowledge item, e.g. in the case of an indicator: which is its value?, is its value larger than some specific value? whereas for a relationship the answer may either require to visualize such relationship, or to say whether the input values influence the output value (e.g., the gender and the age of the clients affect and how much the number of placed orders), or whether the available data allows to make some forecast on the evolution of the mapping.

Each question must also include the information on *when* its answer is needed, i.e. *off-line*, when it is needed to support the strategic decisions of the management (e.g. is true that the big size dealers have less failed orders for not being able to pay in time than the small size one?), *monitoring* when it is needed to generate some alarm while the process is running (the number of failed order during the current day is greater than 50%?), and *in-the-process* when it is needed to make some decision inside the process (e.g. to decide the quotation to propose depending on some analysis of the data on the past quotations followed or not by an order). The use of analytics techniques should allow to answer the above defined questions.

As shown in Fig. 1, the starting point of any application of analytics technique to a business process is to determine the relevant questions on the business process; we propose two different strategies to perform this activity: bottom-up (or data-driven) and top-down (or goal-driven), illustrated in the following subsections.

The Bottom Up Strategy The bottom-up strategy suggests to systematically look for the questions on the process data space, and then the analyst will have to select among them those relevant from the business point of view.

The method asks to look for the relevant indicators and relationships definable on the process data space, see Fig. 6, and then to build over them the interesting questions. The elements of the data space are essentially process, entity and task instances.

Trying to elicit the relevant indicators and relationships will lead also to a kind of inspection of the process, and possibly to modify it, and to discover that there are other relevant indicators/relationships that cannot be computed using the process data space, again this will lead to a modification of the process.

The Top-Down Strategy In the case a preliminary set of questions is already available prepared by the stakeholders, we can also follow a different strategy. Usually the questions will be expressed informally, thus the work of the analyst should consist in trying to express them using indicators and relationships definable on the process data space. So all the terms appearing in the questions should be associated with indicators/ relationships, and then each question should be accommodated in the schema of Fig. 7.

Examples of questions on the process Buying, found following our method, include: which are the manufacturers that cancel an order (their behaviour includes instances of Place Order immediately followed by Reject Order) with a percentage larger than 50%? Which is the rate of the dealers asking for a quotation before placing an order? Which is the rate of the dealers not placing an order after getting a quotation?

6 Conclusions

In this work, we have proposed a first step towards a holistic approach aimed at combining business process modelling and data-driven business process improvement. We briefly detailed its first steps (i.e. respectively creation of the Business Process Precise Model, of the Business Process Data Space and definition of the relevant questions).

As future work we plan to: (1) better refine the two strategies provided by our approach (i.e. "Bottom-up" and "Top-down") for determining the relevant questions on the business process expressed using the process data space, (2) investigate how integrating the existing analytic approaches in our method with the goal of providing answers the selected questions; (3) validate the applicability of our method on an industrial case study, and finally (4) derive a wizard tool that hints to the analyst one after the other all the possible knowledge items and the possible questions over them to make the procedure suggested by our method less cumbersome; moreover, since the process data space model is a UML model and a model of the questions can be also given, we plan to investigate the (semi-)automatic generation of the questions by means of model transformations.

References

1. K. M. Anderson. Embrace the challenges: Software engineering in a big data world. In *Proceedings of 1st IEEE/ACM International Workshop on Big Data Software Engineering*, BIGDSE 2015, pages 19–25. IEEE, 2015.

2. S. Beheshti, B. Benatallah, S. Sakr, D. Grigori, H. Motahari-Nezhad, M. Barukh, A. Gater, and S. Ryu. *Process Analytics: Concepts and Techniques for Querying and Analyzing Process Data*. Springer, 2016.
3. F. Casati, M. Castellanos, U. Dayal, and N. Salazar. A generic solution for warehousing business process data. In *Proceedings of 33rd International Conference on Very Large Data Bases*, VLDB 2007, pages 1128–1137. VLDB Endowment, 2007.
4. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley Longman Publishing Co., Inc., 1995.
5. Object Management Group (OMG). Business Process Model and Notation (BPMN). http://www.omg.org/spec/BPMN/.
6. Object Management Group (OMG). Unified Modeling Language (UML) Specification. http://www.omg.org/spec/UML/.
7. A. Pourshahid, G. Richards, and D. Amyot. *Toward a Goal-Oriented, Business Intelligence Decision-Making Framework*, pages 100–115. Springer, 2011.
8. G. Reggio, E. Astesiano, and C. Choppy. A framework for defining and comparing modelling methods. In R. D. Nicola and R. Hennicker, editors, *Software, Services, and Systems*, volume 8950 of *LNCS*, pages 377–408. Springer, 2015.
9. G. Reggio, M. Leotta, D. Clerissi, and F. Ricca. Service-oriented domain and business process modelling. In *Proceedings of 32nd ACM/SIGAPP Symposium on Applied Computing*, SAC 2017, pages 751–758. ACM, 2017.
10. G. Reggio, M. Leotta, and F. Ricca. Who knows/uses what of the UML: A personal opinion survey. In *Proceedings of 17th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS 2014)*, volume 8767 of *LNCS*, pages 149–165. Springer, 2014.
11. G. Reggio, M. Leotta, F. Ricca, and E. Astesiano. Business process modelling: Five styles and a method to choose the most suitable one. In *Proceedings of 2nd International Workshop on Experiences and Empirical Studies in Software Modelling*, EESSMod 2012, pages 8:1–8:6. ACM, 2012.
12. G. Reggio, M. Leotta, F. Ricca, and D. Clerissi. What are the used UML diagram constructs? a document and tool analysis study covering Activity and Use Case diagrams. In *Model-Driven Engineering and Software Development*, volume 506 of *CCIS*, pages 66–83. Springer, 2015.
13. T. Stahl, M. Voelter, and K. Czarnecki. *Model-Driven Software Development: Technology, Engineering, Management*. John Wiley & Sons, 2006.
14. W. van der Aalst. Formalization and verification of event-driven process chains. *Information and Software Technology*, 41(10):639 – 650, 1999.
15. Wikipedia. Event-driven process chain (EPC). https://en.wikipedia.org/wiki/Event-driven_process_chain.
16. M. zur Mühlen and R. Shapiro. *Business Process Analytics*, pages 137–157. Springer, 2010.