Computer Science and Information Systems 00(0):0000–0000 https://doi.org/10.2298/CSIS123456789X

Decision-Making Support for Input Data in Business Processes according to Former Instances

José Miguel Pérez Álvarez[1], Luisa Parody[2], María Teresa Gómez-López[3], Rafael M. Gasca[3], and Paolo Ceravolo[4]

[1] NAVER LABS Europe
6 Chemin de Maupertuis, 38240 Meylan, France
jm.perez@naverlabs.com
[2] Universidad Loyola Andalucía
Avda. de las Universidades s/n. 41704 Dos Hermanas (Sevilla), España
mlparody@uloyola.es
[3] Universidad de Sevilla
Av. Reina Mercedes s/n, 41012 Sevilla, España
{maytegomez,gasca}@us.es
[4] Universita' degli Studi di Milano
Via Festa del Perdono, 7, 20122 Milano, Italy
paolo.ceravolo@unimi.it

Abstract. Business Processes facilitate the execution of a set of activities to achieve the strategic plans of a company. During the execution of a business process model, several decisions can be made that frequently involve the values of the input data of certain activities. The decision regarding the value of these input data concerns not only the correct execution of the business process in terms of consistency, but also the compliance with the strategic plans of the company. Smart decision-support systems provide information by analyzing the process model and the business rules to be satisfied, but other elements, such as the previous temporal variation of the data during the former executed instances of similar processes, can also be employed to guide the input data decisions at instantiation time.
Our proposal consists of learning the evolution patterns of the temporal variation of the data values in a process model extracted from previous process instances by applying Constraint Programming techniques. The knowledge obtained is applied in a Decision Support System (DSS) which helps in the maintenance of the alignment of the process execution with the organizational strategic plans, through a framework and a methodology. Finally, to present a proof of concept, the proposal has been applied to a complete case study.

Keywords: Business processes, Input Data, Decision-making support, Evolution Models of variables, Constraint Programming, Process Instance Compliance

1. Introduction

The operational plans of organizations are documents that include in detail all technical and organizational aspects related to the development of their products or services. Organizations frequently perform their operations by using *Business Processes* to support the services offered. A *Business Process* consists of a set of activities that are performed in coordination within an organizational and technical environment to achieve an objective [43]. In process orientation, business processes are the main instrument for the organization [18], where commercial *Business Process Management Systems* are incorporated to facilitate the automation and monitoring of their daily processes, and that they are aligned with their objectives and data at the same time [32]. These systems support the implementation, coordination, and monitoring of the business process executions, and produce a great quantity of data that can be stored for its later application in deriving a data evolution pattern temporal variation.

In order to maintain the correct management defined in the strategy plans, companies evaluate the status of the organization by analyzing the suitability of their *Key Process Indicators* (henceforth referred to as KPI) [34]. However, some of the decisions made during the process instances take into account information not available when the decisions are made, because it depends on the activities that will be executed later in the process and how the KPIs will evolve in the future. The process model describes activities that will be executed until the process instance ends. However, to know how the KPIs will evolve, it is necessary to analyze former instances, and the values of the variables that represent the measurements concerning the evolution of the process instances. This data allows a target value to be monitored over specific periods through extraction this value from the execution of business processes. For this reason, this set of observational variables represents the stage of the business, in terms of measurements.

In this paper, we propose a methodology to support the decisions about the values of input variables in business process instances introduced by the business experts at runtime. Frequently, these decisions are made by the expert experience, that can be partial and subjective. But, what can be done when the process instance evolution is not aligned with the KPIs defined. For this reason in this paper, we support the decision-making for the alignment of the execution of the processes with the KPIs, guided by how previous instances of the same process evolved.

To clarify our contribution, the application of our proposal will be illustrated using an example. In the organization of a scientific conference, a set of decisions must be made. Months before the celebration and the attendees are registered, the business experts must decide the early and late registration fees, the venue, or the number of proceedings to print. These decisions affect the successful execution of the conference. Usually, to make these decisions, experts consult previous editions of the same or similar conferences. The analysis of how the number of registrations evolves, and how it is affected by the proximity to the end of the early registration period can avoid making incorrect decisions. Furthermore, the decision-makers can check if the information obtained is aligned with the business goals and organizational constraints.

1.1. Challenges

A methodology to guide decision point during process instantiation was addressed in previous work [15,14]. That methodology was based on modeling the restrictions extracted from the operational plans, by using a special type of *Business Rules* called *Business Data Constraint* [16].

However, the previous approximations fail in the following aspects, which are tackled in the current paper:

- Analyze the temporal variation of the relevant data during the process instances by consistently incorporating the fluctuating patters, their trends and recurring behavior.
- Incorporate into the decision point results obtained by cases with analog contextual patterns. The activities executed during each instance describe the stage of an instance, although other factors can influence the temporal variation of the data involved.
- The degree of uncertainty of the variables handled at the decision points, was not accounted in the previous methodology. It is essential to include this uncertainty since it is not the same to decide the value of a variable in an initial stage of the business process instance, in which its uncertainty is very high, as is in an advanced stage, in which the uncertainty is reduced.

1.2. Objectives

Taking into account the above challenges, in this paper we propose an extension of the previous study to achieve the objective: *The creation of a methodology that supports the decisions made during the process model execution about the input data values. The basis of the methodology is to provide the proper range of values taking into account the stage of the instance and the possible evolution of the variables in the future according to analog models, i.e. related former-instances.*

This objective implies the achievement of the three following sub-objectives:

1. **Problem Modeling:** During modeling time, the different parts of the problem are incorporated using the business *Expert Knowledge*. The elements that must be modeled are: The *Business Process Model*, *Dictionary*, *Stages* and *Business Rules*.
2. **Creation of evolution patterns of variables:** Before starting the instance in which decisions have to be made, it is necessary to create the the patterns of the data temporal variation, by analyzing how they evolved in analog models.
3. **Decision-making support for input data at runtime:** For each decision, the system uses the knowledge extracted from data temporal variation in previous instances, adapted to the current stage of the *Process-Observational Variables* in the decision-making moment. This is performed by using the Constraint Programming paradigm that provides different techniques to solve constraint problems.

Thanks to this new approach, the board and executive team of an enterprise can compare the current status of an instance with previous statuses, thereby helping to maintain the alignment of all business processes with the defined KPIs. On the other hand, thanks to the base of knowledge obtained, we will have the capacity to predict how the business instance will evolve, and how undesirable variation can be detected at an earlier stage.

The paper is organized as follows: Section 2 introduces a real world case study, used throughout the paper to illustrate the various aspects. Section 3 presents a graphical overview of the basic aspects of the methodology. The subsequent three sections describe the three phases of the methodology: Problem Modeling in Section 4, creation of variable evolution pattern in Section 5, and Decision-making support for the input data phase in Section 6. Section 7 describes related work. Finally, conclusions are drawn and future work is proposed in Section 8.

2. Case Study

In order to describe our proposal, a case study is presented to make the methodology accessible. The case study is a sample extracted from an event organization company focused on conferences. The company offers the service by using a web-based application, through which the customers can manage the conferences that they organize. Thanks to this service, the company has a large database of past events that can be consulted in order to improve decisions regarding conferences of the future. Figure 1 shows the process that support a conference organization that follows the operational business plans of the company. The example represents a research conference where participants can present a research work that has been previously reviewed and accepted. The process has been simplified to illustrate the example.

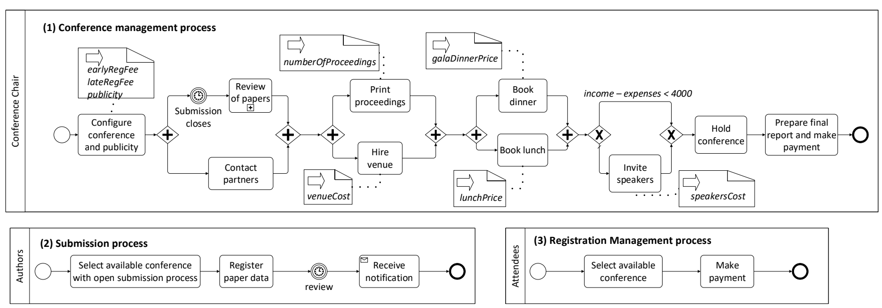


Fig. 1. Example of processes for Conference Organization.

The process *Conference management* is performed by the *Conference Chair*, and supports the operations for the establishment and management of the conference organization. The first task is to *Configure conference and publicity*, where the conference chair decides to set up the parameters of the conference, by defining the initial data values. Some of this data, represented by means of variables in the process, is related to registration fees at different times, available budgets, and the important dates. There is a data element associated with the activity *Configure conference and publicity* where the input variables, whose values must be established, are depicted (e.g. early registration fee, late registration fee, date of submission closes, date of early registration closes, etc.). Once the conference is configured, some values of the input variables cannot be changed, for example, the early registration fee, this is why to take into account how the process instance can evolve is so relevant. Afterwards, *Contact partners* is performed in parallel with the *Review of papers*, as shown in Figure 1. These two latter activities can start once the *Submission time* has expired. Subsequently, the *Print proceedings* and *Hire venue* tasks are executed in parallel. The subsequent steps are *Book gala dinner* and in parallel *Book lunch*. In the case where a profit of more than 4,000 euros is expected, speakers can be invited. This is performed in the task *Invite speakers*. Finally, the conference takes place when the task *Hold conference* is executed, and payments and final reports are performed in the task *Prepare final reports and make payments*.

The second process, *(2) Submission*, is carried out by *Authors*. The first task is *Select available conference with open submission process*, since several conferences can be available for the submission of contributions at the same time. The next task consists of collecting information on the author and the contribution in *Register the paper data*. The author must subsequently wait until the revision is completed. Finally, the author is notified once the task *Receive notification* is executed.

The third process is *(3) Registration management*, which is in a pool assigned to the *Attendees* for their registration in a conference.

This case study is challenging for several reasons: (1) it includes several actors in the process: organizers, authors, reviewers, and attendees; (2) there is a relationship between various processes with a significant number of shared variables, such as conference id, number of attendees, etc.; and (3) there are many decisions about variable values needed throughout the process that are key to the success of the process: registration fees, venue cost, lunch and gala dinner prices, etc. One of the main problematic issues of the conference organization is that several decisions must be made before a specific value is known. Many activities include decisions on the value of variables that are key to the successful celebration of the conference. For example, in the *Configure conference and publicity* activity, the values of registration fees are established without knowing the number of participants. A low registration fee and few attendees could mean that no speakers can be invited to give the keynotes or there may not be enough money to hire the gala dinner. In order to avoid this type of problems, these decisions tend to be made according to the previous experiences of the organizers, but frequently based on subjective aspects. The evolution of the number of papers sent, or the number of early registered can give clues about how the process instance can evolve in the future, making the current decisions more proper and accurate. Without the use of a decision-making support system, the organizer of a conference cannot come ahead of time when the number of attendance are not evolved as expected (in comparison with previous instances), and how the decisions of the future can be made to avoid an unwanted situation.

Thanks to our proposal, the organizers can obtain information of previous instances and make decisions according to this information at runtime. Following with the keynotes problem and let's assume that income is only dependent on registrations. When the organizers have to decide who to invite to give a keynote, they are going to have available not only the number of people registered up to that moment, but also how that value is evolving with respect to previous instances. If the evolution is very negative (many less registered), our framework will let the organizers know how much money they will have available approximately according to that evolution. How the framework is integrated in the business process and in which points the decisions are made are explained in the following sections.

3. Methodology for Decision-Making Support of Input Data

The main objective of our proposal is to assist the business experts in the decision-making during process execution. Figure 2 graphically summarizes our methodology. The summary is introduced in this section, and details are provided in the following sections.

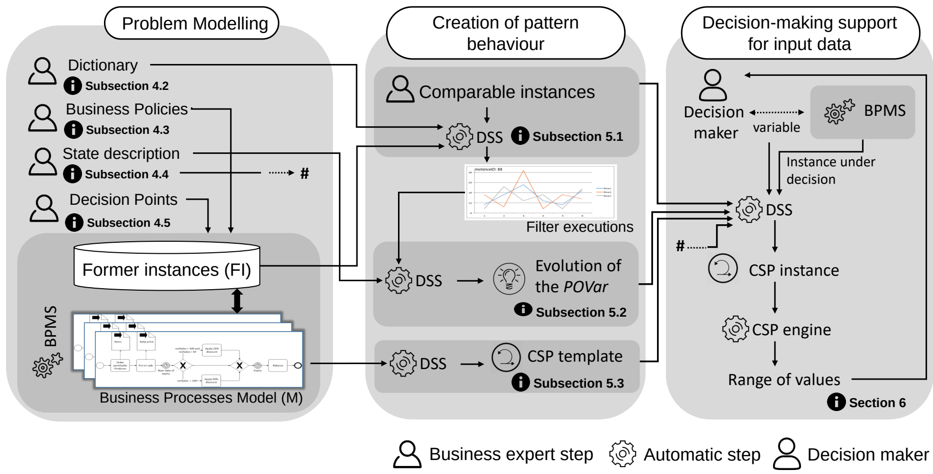


Fig. 2. Steps of the Methodology proposed.

1. **Problem Modeling:** This takes place during the definition of the business process by the business expert as detailed in Section 4:
   - The *Business Process Model* must be included to describe the behavior of the organization (Subsection 4.1).
   - The *Dictionary* of *Process-Observational Variables* is defined, with the objective of maintaining a common language that facilities the following steps of the methodology, explained in Subsection 4.2.
   - *Business Rules* enrich the *Business Process Model* to describe the business rules involving *Process-Observational Variables* as described in Subsection 4.3.
   - The set of relevant *Stages* (*S*), in which the *Business Process Instances* (*BPI*) can be, must be defined. Based on the strategic plans of the company and the business experts' knowledge, in Subsection 4.4 we propose a mechanism to model and compute the *S* that represents the *BPI* temporal variation.
   - Identify activities where decisions will be made, so that different decision points can be pinpointed in the business process (Subsection 4.5).
2. **Creation of the evolution pattern of the variables:** The temporal variation of the relevant variables and how each stage can affect them is fundamental in the later decision-making process. Since the *BPI* works in various ways, the selection of a subset of instances similar to the instance subject to a decision is essential. Section 5 details the following information:
   - A mechanism enables business experts to specify the *Comparable Instance* criteria, to select the former instances that are more related to the instance under decision. These aspects are discussed in Subsection 5.1.
   - According to the *Comparable Instance* and by analyzing the former instances, the temporal variations of the *Process-Observational Variables* are obtained as

detailed in Subsection 5.2. The discovery of the evolution pattern is crucial for the understanding of how the business works, thereby helping in the decision-making process.
   - A model template is automatically created by traversing *Business Process Models* and combining the *Business Rule* associated with each activity. Once the evolution patterns are created, we propose the use of the Constraint Programming Paradigm, thereby creating a *Constraint Satisfaction Problem* template that contains the temporal variation patterns obtained and the structure of the process, as detailed in Subsection 5.3.
3. **Decision-making support for input data:** During the execution of each *Decision Point* in an instance under decision, decisions regarding input data can be made. In order to determine the correct input domain, we propose the combination of the current stage with the *Constraint Satisfaction Problem* created in the previous step, as discussed in Section 6.

4. Problem Modeling

The model of the problem is composed of a Business Process Model, a Dictionary of Process-Observational Variables, the Business Rules, the Stage descriptions, and the Decision Points. Each component is described in the following subsections.

4.1. Business Process Model

Business Process Models permit the description of the activities developed by an organization. The standard BPMN 2.0 [31] is usually employed for this purpose. The example of Figure 1 includes three business processes. Business Processes can include: a start event, end events, activities, gateways, and conditions associated with the gateway branches (OR and XOR). These components are combined in a control flow structure that manages the order of execution of the activities.

4.2. Dictionary (D)

In order to describe the relevant control variables that will be involved in the business rules and in the KPIs, the business experts must describe the *Dictionary* of terms. A *Dictionary* is formed of a set of *Process-Observational Variables* $\{\text{POVar}_1, \ldots, \text{POVar}_n\}$, whose values are relevant for the organization to know whether their policies comply and their KPIs are reached or not. Each *Process-Observational Variable* is composed of a name of the variable and its description ($\langle$*name, description*$\rangle$). The description contains information that enables its value to be extracted from the *Business Process Management System* for each instance. In our case study, examples of the *Process-Observational Variables* include:

- `earlyRegFee`: Cost in the early registration period.
- `lateRegFee`: Cost in the late registration period.
- `maxPapers`: Maximum number of papers that can be accepted.
- `submissionOpen`: Date when the paper submission opens.

- `regOpen`: Date when the registration opens.
- `earlyRegClose`: Date when the early registration closes.
- `submissions`: Number of submissions to the conference.
- `earlyReg`: Number of registrations in the early registration period.
- `totalReg`: Number of attendees.
- `sponsorship`: Total amount reached by sponsors.
- `acceptedPapers`: Number of accepted papers.

In order to describe the meaning of the *Process-Observational Variables* and to automatically evaluate and monitor these variables, the use of *Process Instance Query Language (PIQL)* [33] is proposed. PIQL enables business experts to extract information from the process instances that match with specified criteria, and includes a set of operations over the selection. Not only does PIQL allow the selection of instances under execution, but also that of former instances. PIQL is oriented to business experts, offering a natural-language-like specification.

It should be borne in mind that although PIQL is the language selected to specify *Process-Observational Variables*, the study of PIQL itself remains outside of the scope of this paper. Certain details are provided below for a better understanding, but the formal syntax, and in-depth details are available in [33]. The types of operations that can be performed over processes and task instances can be seen in Table 1.

Operation	**PIQL Syntax**
Count all selected instances	The number of instances of processes
Obtain the value of a variable of the data-flow if just one instance matches	The value of *variable* of the process *p*
Obtain the average of values of a variable of the data-flow for the selected instances of a process	The average value of *variable* of the process *p*
Obtain the maximum value of a variable of the data-flow for the selected instances of a process	The maximum value of *variable* in the process *p*
Obtain the minimal value of a variable in data-flow for the selected instance of a process	The minimum value of *variable* in the process *p*
Count the task instances for all the selected instances	The number of instances of task *t*

Table 1. Operations over a set of process and task instances.

Additionally, the attributes of Table 2 can be used for the filtering of processes and task instances. Moreover, certain arithmetical, logical, and comparison operators can be used. Moreover, a set of predicates can also be used as detailed in Table 3.

By using PIQL, several of the *Process-Observational Variables* specified above can be modeled as follows:

- *maxPaper*: The value of *maxPaper* of the process “Conference Management”.
- *regOpen*: The value of *regOpen* of the process “Conference Management”.
- *submissions*: The number of instances of “Submission process” that are finalized.
- *earlyReg*: The number of instances of “Registration Management process” that end before *earlyRegClose*.

– *totalReg*: The number of instances of "Registration Management process" that are finalized.
– *acceptedPapers*: The number of instances of "Submission process" with "accepted" = "true".
– *notified*: The number of instances of "Author notifications" task of process "Conference Management process".

Attributes	**PIQL Syntax**
idCase	with a case id *idCase*
Process_Name	with a name *process_name*
Task_Name	with a name *task_name*
Start	with a start date *date*
End	with an end date *date*
Cancelled	*cancelled*
Who	executed by the user *user*

Table 2. Attributes of process and task instances.

Predicates	**Transformed pattern**
are finalized	end date is not equal to Null
are not finalized	end date is equal to Null
are cancelled	cancelled is not equal to Null
are not cancelled	cancelled is equal to Null
executed by {name}	the user is equal to {name}
start before {date}	a start date is less than {date}
end before {date}	an end date is less than {date}
start after {date}	a start date is greater than {date}
end after {date}	an end date is greater than {date}

Table 3. Predicates allowed

4.3. Business Rules

The strategic plan of a company is commonly a set of documents written in natural language. These documents must be translated into something computable in order to know whether a *Business Process Instance* has finalized correctly or not, according to the policies of the companies. Numerous studies propose a variety of taxonomies to classify the definition of business rules [7,19]. One of the most frequent definitions is: A specification, a policy or a standardized procedure, that represents a natural step towards the inclusion of semantic requirements between business functionality and data. However, if the relations between the variables that delineate the *Stages* of the business have to be described, then *Business Rules* have the capacity to describe Business Compliance Rules [5]. This is the focus of this subsection, the modeling of the *Business Rules* by using an adaptation of the *Business Data Constraints* introduced in [13,16].

A *Business Rule* is a Boolean combination of numerical constraints whose evaluation can be true (satisfied) or false (unjustifiable). These numerical constraints can be specified by means of operators of comparison and *Process-Observational Variables* defined in the *Dictionary*. *Business Rules* represent the semantic relation between the data values that are introduced, read, and modified during the execution of the *Business Process Instances* [12].

In our case study, the set of constraints associated with the main process includes: Expenses must be less than or equal to income for every conference; The maximum accepted papers cannot be exceeded; All attendees must possess a copy of the proceedings; It is forbidden to exceed the maximum capacity of the auditory.

Business Rules can be associated with an activity, a set of activities, or as an invariant of the whole process. For example, when the activity *Invite Speakers* is executed, then the constraint *the total cost of bringing a speaker to the conference cannot exceed 30% of the income reached by sponsorship* must be satisfied.

Business Processes describe relations between *Process-Observational Variables*. The scope of these *Process-Observational Variables* can involve various instances of numerous processes. *Business Processes* therefore take into account not only relations between the local status of individual instances of a process, but also the global status of the company. *Business Rules* have been addressed in [15], which is adapted to our proposal, where both the *Process-Observational Variables* and the variables defined in the data-flow of the process can be involved.

These constraints can appear from different sources, such as business strategic plans, business rules, and manager restrictions. The principal aspect here is to determine if these constraints are mandatory for the defined scope, since, in the case where they remain unsatisfied, the process cannot be considered as having been finalized successfully. One example of *Business Process* related to the case study is: "*expenses must be less than or equal to income for every conference*". This is defined by the manager, and in the case where it remains unsatisfied, the instance of the process in which it has been defined, will not finish properly.

In our case study, the set of constraints associated to the main process is:

– Expenses must be less than or equal to income: $expenses \leq income$.
– All attendees must possess a copy of the proceedings: $numberOfProceedings \geq totalReg$.
– It is forbidden to exceed the maximum capacity of the auditory: $venueCapacity \geq totalReg$.
– The cost of inviting a speaker must be less than 30% of sponsorship income: $speakersCost \leq sponsorship \times 0.3$.

On the other hand, there are also other *Business Rules* oriented towards the definition of intermediate variables, such as:

– **Income** is the result of multiplying the number of early registrations by the early registration fee, plus the result of multiplying the number of late registrations by the late registration fee, plus the income by sponsorship: $income = earlyReg \times earlyRegFee + lateReg \times lateRegFee + sponsorship$.
– **Expenses** is the result of adding the total cost of publicity, plus the number of copies proceedings printed multiplied by the price of each copy, plus the total venue cost, plus the gala dinner cost, plus the lunch cost, plus the total cost of inviting speakers to the conference: $expenses = publicity + numberOfProceedings \times proceedingPrice + venueCost + galaDinnerPrice + lunchPrice + speakersCost$.

BRs are stored and queried using a *Constraint Database* as described in [37,11].

4.4. Business Process Stages

The temporal variation of the *Process-Observational Variables* is frequently related to the execution of the activities that form the *Business Process Model*. However, certain external factors may also influence these variables. For instance, conference chairs know that the information about the number of registrations (`TotalReg`) of a conference increases fairly quickly once paper notification has been performed. However, this type of information cannot always be included explicitly in a BPMN model, and cannot be extracted directly from the business process. Since this information is crucial to know the variation at any moment, we propose a simple description of these *Stages*, to be used in the process instance analysis developed for the decision-making process. When a decision has to be made, the user will consult the status of the variables related to this decision.

As can be observed, and according to our case study, the definition of these *Stages* is not exclusively dependent on which activities are being executed, and is related to the variation of the values of some *Process-Observational Variables*. For example, in our case study, there is no specific activity to make a late registration, but there is a period of time where the registration is more expensive (after the *earlyRegClose*). Therefore, even though all registrations are made using the *Registration Management Process*, the variation of the *number of registrations* depends on the stage of the registration (early or late).

Let S be $\langle s_1 \dots s_n \rangle$, which represents all *Stages* defined by the business experts for a *Business Process Model*, where at least one stage must be defined. In our case study, the set of stages defined are:

- Paper submission: Period of time in which the paper submission is open to authors. The period between `submissionOpen` and `submissionClose`.
- Early registration: Period of time in which the registration is the cheapest. More specifically, the period between `regOpen` and `earlyRegClose`.
- Late registration: Period of time in which the registration is open in a late registration phase, which is the period between `earlyRegClose` and `regClose`.
- Notified: When the activity *Notify* is executed.
- Conference: Period in which the activity *Hold Conference* is executed.

In order to describe the *Stages*, we also propose the use of *Business Data Constraint* [15], by using the *Process-Observational Variables* defined in the *Dictionary*, as shown above. $currentDate$, $currentDay$ and $currentHour$ placeholders are also allowed, and the engine automatically sets their values at runtime.

Listing 1.1 shows an example of *Stages* using *Business Data Constraints*, where the *Stages paperSubmission*, *earlyReg*, *lateReg*, *notified*, and *conference* are defined.

```
[paperSubmission]
    submissions ≥ 0 AND notified = 0
[earlyReg]
    regOpen ≤ $currentDate AND earlyRegClose > $currentDate
[lateReg]
    earlyRegClose ≤ $currentDate AND regClose > $currentDate
[notified]
    notified > 0
[conference]
```

```
    holdConference > 0
```

Listing 1.1. Stages defined for the example

There are no restrictions regarding the number of stages defined by the business experts. They can define the set of stages that they consider relevant for the knowledge extraction phase. The overlapping of stages is of no consequence; this issue will be considered in the normalization phase explained in Subsection 5.2.

At moment (t) of the *Business Process Instance*, the possible *Stages* defined by a business expert can be in either of two positions: *Activated* or *Deactivated*. A status is *Activated* when the *Business Process Instance* meets the conditions defined for this *Stages* and the Boolean expression is *true*; it is *Deactivated* and *false* otherwise.

4.5. Decision Points

During the instantiation of a business process model, the activities can include forms to introduce values of variables of the process. The value of the input variables can be provided by the third party not being necessary a decision (e.g. the gathered sponsorship), or by a business expert as a product of decision-making about the most proper value for the variable (e.g. the number of copies of the proceedings according to the estimation of attendance before the final number is known). A *Decision Point* is associated to activity of the business process, where some values of a set of input variables must be introduced at instantiation time [6]. Each *Decision Point* is formed of a tuple ⟨*Decision Variables*, activity⟩.

For the first process of Figure 1, (1) Conference Management Process, there are six *Decision Points* (*Decision Variables*, activity):

- *Decision Point* 1: ⟨{*earlyRegFee*, *lateRegFee*, *publicity*}, Configure conference and publicity⟩
  - `earlyRegFee`: Cost of early registration fee.
  - `lateRegFee`: Cost of late registration fee.
  - `publicity`: Estimation of the cost for publicity actions.
- *Decision Point* 2: ⟨{*numberOfProceedings*, *proceedingPrice*}, Print a copy of the proceedings⟩
  - `numberOfProceedings`: The number copies of the of proceedings to print.
  - `proceedingPrice`: The price of printing a copy of the proceedings of one conference.
- *Decision Point* 3: ⟨{*venueCost*}, Hire venue⟩
  - `venueCost`: Venue cost.
- *Decision Point* 4: ⟨{*galaDinnerPrice*}, Book dinner⟩
  - `galaDinnerPrice`: The price of gala dinner.
- *Decision Point* 5: ⟨{*lunchPrice*}, Book lunch⟩
  - `lunchPrice`: The price of lunch.
- *Decision Point* 6: ⟨{*speakersCost*}, Invite speakers⟩
  - `speakersCost`: The total cost of inviting speakers to the conference.

When a process instance reaches an activity with a *Decision Point*, the DSS must evaluate the range of values for the input data that enable the aim of the *Business Process* to be achieved. This range is derived from the analysis of the former instances of the process to ascertain how the values of the *Process-Observational Variable* can evolve until the process ends according to the stage of the current process instance. The following section analyzes how the *Process-Observational Variable* evolution patterns are created in order to facilitate decision making.

5. Creation of evolution pattern of Process-Observational Variables

In order to make the decisions during process execution, it is important to take into account the context of the decision, in terms of the stages of the *Business Process Instance*. For example, to decide the most appropriate number of copies of the proceedings to print (`numberOfProceedings`), it is necessary to derive the number of attendees (`totalReg`). However, this decision must be made several months before the conference starts. The number of attendees can be derived from the number of those already registered and how this value evolved in previous and similar conferences (e.g. conferences in the same research area, city, or period). Obviously, the final `totalReg` depends on the current value of the variable, and the rest of POVar, or the stage of the instance (before or after the `lateReg` stage). How the POVar can evolve in the future can be derived by analyzing the BP, and how they evolved in previous process executions. However, in order to compare the *Business Process Instance* under decision with previous ones, it is necessary to extract only the *Comparable Instance* (e.g. conference of the same research area) as explained in Subsection 5.1, and to extract the evolution patterns (Subsection 5.2). The subsections below describe how this can be performed.

5.1. Specify Comparable Instances

A *Business Process Instance* (BPI) represents a specific case in an operational business process, and an execution of a *Business Process Model*. All $BPIs$ $\{bpi_1 \ldots bpi_n\}$ of a model, are individually described by the tuple $\langle M, Start, End, UpdateEvents\rangle$: M is the *Business Process Model*, $Start$ is the start data when bpi_i started; End is end date when the bpi_j finalized (empty if not finalized); and $UpdateEvents$ is a set of updated events, that is, the moments at which the value of any variable of the BP instance has been modified. We differentiate between two types of *Business Process Instance*:

- Non-former instances (NFI): *Business Process Instances* that are still under execution, that is, are not finalized. $NFI \in BPI$; $\forall nfi \in NFI, nfi.End = null$. In certain non-former instances, decisions regarding data must be made; these instances are called Instances Under Decision.
- Former instances (FI): *Business Process Instances* already finalized. $FI \in BPI$; $\forall fi \in FI, fi.End \neq null$. The set of former instances constitutes a major part of our proposal, since they represent historical information about the executions. In some of these former instances, decisions have been made.

The analysis of former instances will be very useful to understand the temporal variation of the variables in the various *Business Process Instances*, and the DSS is able to use this information to help business experts make better decisions regarding the instances under study. A subset of the former instances can be selected to create a evolution pattern for the decision point.

Let comparable instances $CI \ \langle ci_1 \ldots ci_n \rangle$ be the set of former instances that share certain characteristics with the instance under study. These related instances are able to analyze the *Process-Observational Variables* in former instances, to improve the decisions. In our case study, examples of characteristics defined by business experts include: (i) Topic: software engineering(ii) End date: 5 days before the current date; and (iii) Number of registrations: the final register will be less than 250 and greater than 100 attendees.

The former instances that comply with the criteria defined above are considered comparable instances to the non-former instances in which decisions must be made. Therefore, the former instances that comply are valid for analysis and for the extraction of useful knowledge to be used in the DSS.

By using the set of variables defined in the *Dictionary*, and by using the grammar defined for PIQL, business experts can define those characteristics as shown in Listing 1.2.

```
topic IS EQUAL TO 'software engineering' AND
holdConference IS GREATER THAN $currentDate – 5 years AND
totalReg IS LESS THAN 250 AND totalReg IS GREATER THAN 100
```

Listing 1.2. Example of the definition of Comparable Instances

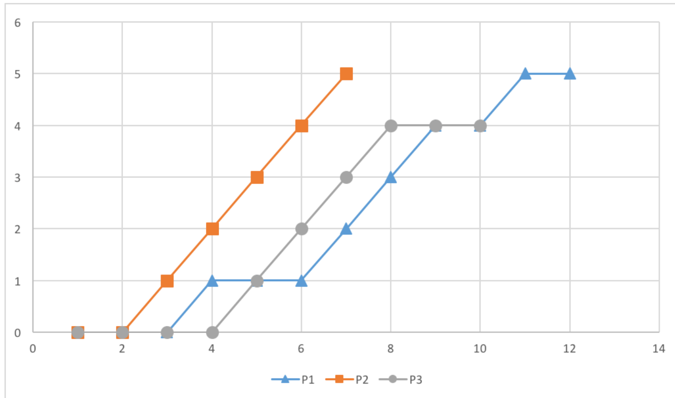
5.2. Extracting of patterns of evolution of Process-Observational Variables

This step consists of automatically creating a temporal variation model of the *Process-Observational Variables* of the business processes by analyzing the former instances.

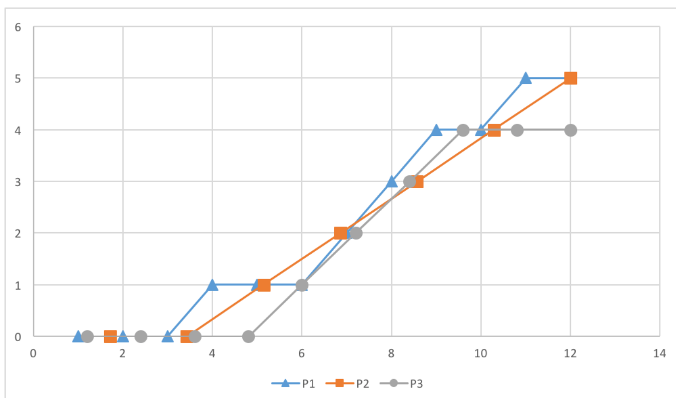
It should be borne in mind that each instance and its aforementioned *Stages*, can have a different duration. For instance, in the example presented in Section 2, conference managements can differ in their duration. Figure 3.a shows the variation of the *Process-Observational Variable* "number of registrations" (`totalReg` for the organization of three conferences (i.e., three different instances of "conference management process"). In order to facilitate the understanding of the example, these three processes start at the same time ($t = 1$). As can be observed, P1 has a duration of 40 units of time, P2 has a duration of 20 units of time, and P3 has a duration of 30 units of time.

The differences in the duration introduce complications into the comparison of the POVar variation between different *Business Process Instances* and into the creation of evolution patterns. However, this comparison is essential to obtain expected data value evolution. For this reason, it is necessary to establish mechanisms that enable the comparison of various instances with different duration and into various periods of activities or stages.

Normalization The most evident mechanism of comparison is that of data normalization [10]. A representation of this mechanism is shown in Figure 3, where the execution times are normalized according to the events that occur during the instance-life. Normalization involves the adjustment of the values in different scales to a notionally common scale.



(a) Variable not normalized



(b) Variable normalized

Fig. 3. Normalization Process of the execution time of a *Process-Observational Variable*

Once the *Business Process Instances* are established on the same scale, they can then be compared.

However, the evolution of the values of the *Process-Observational Variables* is not constant throughout the whole life of the *Business Process Instance*. The evolution can differ depending on the *Stages*. Moreover, the duration of each *Stages* in different instances can be different, being necessary the normalization of the *Process-Observational Variables* when they are compared. For this reason, our proposal includes a normalization process that homogenizes the evolution of the values of the *Process-Observational Variables* to be comparable, taking into account the *Stages* and the duration of each business instances.

Therefore, the normalization process consists of two phases: (1) Displace the start instant of all *Comparable Instances* to the same instant, *Start=1*; and (2) Set the different duration of *Business Process Instances* to a common scale to compare the temporal variation of each variable involved in the various instances.

Thanks to this normalization, every instance is scaled to the same duration, where this duration is the maximum duration of the former instances (12 units of time for this example), as shown in Figure 3.b. Once normalized, the *Business Process Instances* can be compared, and the degree of similarity among the temporal variations can be observed. Table 4 shows an application of the *Stages* defined by business experts over the set of data shown in Figure 3.a. In this case, we can observe three possible *Stages* represented with three colors. Three sets of data are obtained (because three stages have been reached), and the results once the stage is started are shown graphically in Figure 4.

t	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
P1	0	0	0	0	0	0	0	0	0	0	1	3	4	5	7	8	9	10	12	13
P2	0	2	3	5	6	8	10	11	13	14	16	18	19	21	22	28	35	43	54	67
P3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

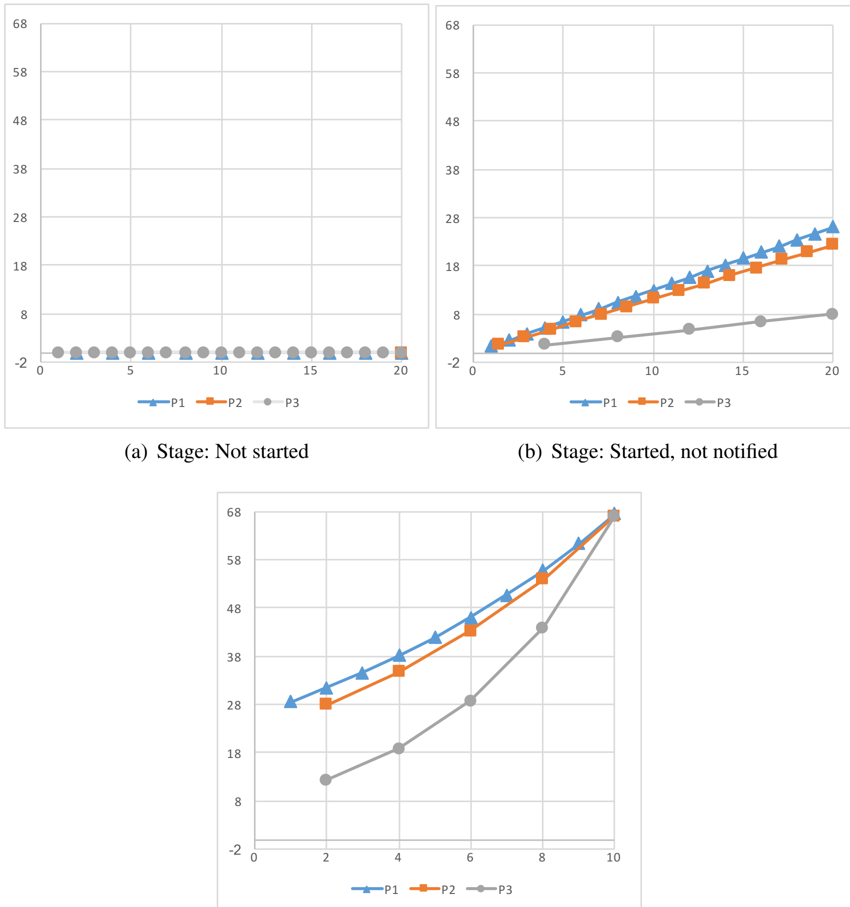
t	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**
P1	14	16	17	18	20	21	22	23	25	26	29	31	35	38	42	46	51	56	61	67
P2																				
P3	2	3	5	6	8	12	19	29	44	67										

Table 4. Variable *Registrations* with the stages colored in each instance

Finding out the Envelopment Temporal Variation of Process-observational variables
Once the temporal variation of the variables is normalized, techniques of knowledge extraction can be applied. For every detected possible combination of *Stages* and all variables in these *Stages*, this step consists of computing the minimum and maximum slope of the *Process-Observational Variable*. It represents the degree by which the variable increases or decreases, from the *Stages* understudy, to the final value of the variable, at the point of finalization. This implies the computing of the maximum and minimum percentage of increment or decrement of the variable from the *Stages* to the end of the process. Slopes for early stages contain more uncertainty than those for late stages; however, this information will be beneficial in the estimation of the potential final range of values of the *Decision Variable*.

There exist numerous references in the literature, where methodologies and techniques are proposed for the creation of models by using the observations, such as curve fitting [24], linear regression [29], and non-linear regression [30]. These models are needed for the computation of the potential increase or decrease of the variable, and hence, our proposal is based on the calculation of the slopes, although other techniques, such as those described below, could be applied.

The slopes are computed by dividing the difference between the final and initial value by the duration of the process. Minimum and Maximum slopes are calculated by taking minimum and maximum values into account, as shown in Figure 5 where the *Stages* is “started, not notified”. This consists of calculating the slopes by using the lowest value for any instance at the beginning of the stage, and the lowest final value of an instance for the computation of the minimal slope. On the other hand, by using the highest values at the



(a) Stage: Not started

(b) Stage: Started, not notified

(c) Stage: Started, notified

Fig. 4. Variable normalized according to the stage

beginning of the stage for any instance and the highest values at the end of the process, the maximum slope is computed.

5.3. Combination of the element of the Models by means of a CSP template

Once the different elements of the model have been described or derived from former instances, they must all be combined for overall reasoning. This implies to link the business rules to the activities, along with the *Decision Points* and the possible envelopment variation according to the stages. We propose to use an approximation similar to [15], where the elements were associated to the business process model according to the control-flow structure of the process. The various business rules associated to each activity or set of activities (as explained in 4.3) can be combined in a *Constraint Satisfaction Problem* to ascertain how the *Process-Observational Variables* variation can affect each decision.

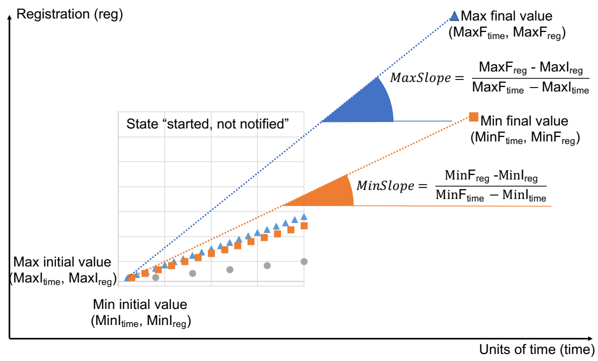


Fig. 5. Graphical representation of the max and min slope for one stage

Since the *Business Process Model* contains elements that route the execution flow of the process (such as gateways), it is necessary to analyze the control-flow structure and its associated business rules. The problem of analyzing the structure of the business process has been addressed in [15], which tackles how to obtain the set of *Business Rules* in accordance with the control-flow. The basic principles of how the Constraints are combined following the control flow are depicted in Figure 6. The set of patterns to combine the constraints are:

- Sequence (Figure 6.a). All the instances must execute these activities, hence all the Business Rules must be satisfied. Therefore they are put together with an AND Boolean relation between them.
- AND Split (Figure 6.b). Similarly with the AND split control flow, all the instances have to execute all the activities of the different branches, although the order is unknown. Therefore, the business rules of all these activities will be combined by means of an AND Boolean combination.
- XOR Split (Figure 6.c). In the case of the XOR control flow, where only one branch can be executed for an instance, the condition associated with each branch will be combined with the Business Constraints of the activities for each branch. The Business Constraints of the activities of a branch have to be satisfied only if the branch is executed. Therefore, the Business Rules of the activities of the different branches will have an OR Boolean relation between them, and the conditions are combined with an AND Boolean relation with the Business Rules of the activities for each branch. A special treatment is performed for the default branch, where the conditional flow of execution implies the non-compliance of the condition for the rest of the branches, thereby implying the negation of the rest of the conditions.
- OR Split (Figure 6.d). OR control flow is very similar to XOR, the only difference being that more than one branch can be executed, and hence the default option negating the rest of the branches does not appear since this would make no sense.

These algorithms enable to approach a *Business Processes* as a BPMN-graph that is traversed to build a Constraint Satisfaction Problem that guides during the decision-making process. For example, the BPMN-graph created for the process used in the paper

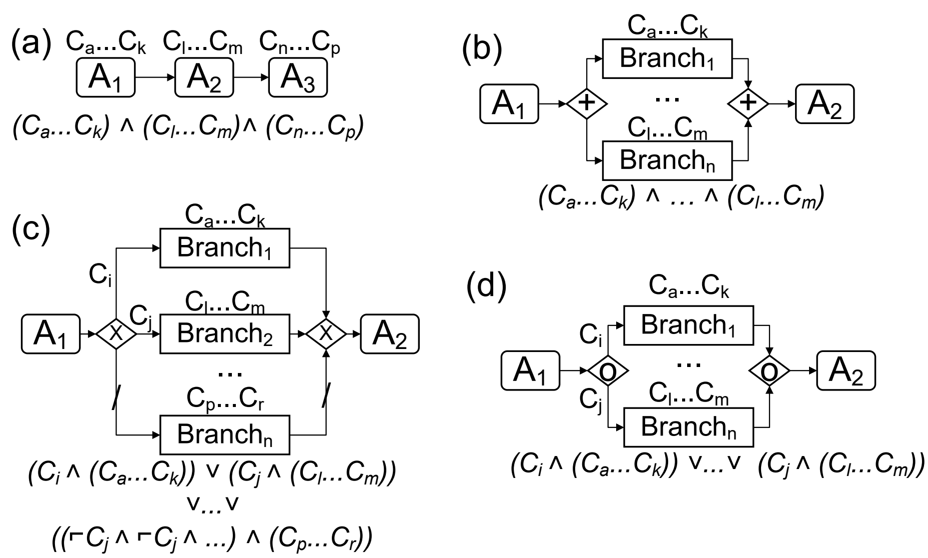


Fig. 6. Combination of *Business Rules* in terms of the control flows and their conditions

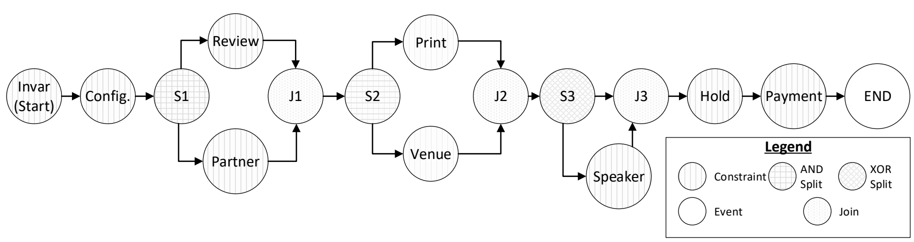


Fig. 7. BPMN Graph for Conference Organization.

is shown in Figure 7. Each node represents: start event (with the invariant constraints associated); an activity with the associated constraint, if correspond; gateways (and-split, and-join, xor-split, xor-join), and; end event.

The problem obtained has the following form:

$C_{Invariant} \wedge (C_{Config} \wedge (C_{Review} \wedge C_{Partner}) \wedge (C_{Print} \wedge C_{Venue}) \wedge (\neg(Income- expenses < 4,0000) \wedge C_{Speak}) \wedge C_{Hold} \wedge C_{Paym})$

The details about the traverse algorithm can be found in [15].

A *Constraint Satisfaction Problem* (CSP) represents a reasoning framework that consists of of variables, domains, and constraints. Formally, it is defined as a tuple $\langle X, DO, C\rangle$, where $X = \{x_1, x_2, \ldots, x_n\}$ is a finite set of variables, $DO = \{do(x_1), do(x_2), \ldots, do(x_n)\}$ is a set of domains of the values of the variables, and $C = \{C_1, C_2, \ldots,$

$C_m\}$ is a set of constraints. Each constraint C_i is defined as a relation R on a subset of variables $V = \{x_i, x_j, \ldots, x_l\}$, called the *constraint scope*. The relation R may be represented as a subset of the Cartesian product $do(x_i) \times do(x_j) \times \ldots \times do(x_l)$. A constraint $C_i = (V_i, R_i)$ simultaneously specifies the possible values of the variables in V in order to satisfy R. Let $V_k = \{x_{k_1}, x_{k_2}, \ldots, x_{k_l}\}$ be a subset of X. An l-tuple $(x_{k_1}, x_{k_2}, \ldots, x_{k_l})$ from $do(x_{k_1})$, $do(x_{k_2})$, ..., $do(x_{k_l})$ can therefore be called an *instantiation* of the variables in V_k. An instantiation is a solution if and only if it satisfies the constraints *C*.

In order to solve a CSP, a combination of search and consistency techniques is commonly used [9]. The consistency techniques remove inconsistent values from the domains of the variables during or before the search. During the search, a propagation process is executed which analyzes the combination of values of variables where the constraints are satisfiable. Several local consistency and optimization techniques have been proposed as ways of improving the efficiency of search algorithms.

In a CSP, the inclusion of a constraint in the set C has the same effect as including this constraint with an AND ($\wedge$) relation with the set C. For this reason, the CSP template is composed of:

- *X*: $\{$Related variables defined in the *Dictionary* $\} \cup \{$*Decision Variables*$\}$
- *DO*: Estimated ranges for each variable in *X*, in the *Stages* in which the decisions are going to be taken
- *C*: $\{$*Business Rules* defined for the whole *Business Process*$\} \cup \{$BDC obtained by traversing the *Business Process*$\}$

Not all *Process-Observational Variables* defined in *Dictionary* are included in the CSP template. Only those variables in the intersection between the *Process-Observational Variables* defined in *Dictionary* and the set of variables that have been used to define the *Business Rules*, are included in the CSP template. The reason is that, if *Process-Observational Variable* is defined in *Dictionary* but is not used in any *Business Rules*, then it implies that the variable exerts no affect, and therefore its inclusion can be omitted.

This CSP template represents the whole process since the template contains all relevant *Process-Observational Variables*, and the combined *Business Rules* are in accordance with the control flow. The only element in the preceding template that needs to be specified is the specific value of the *DO* of each *Process-Observational Variable* at the specific moment at which the decision must be made.

For the example, the CSP pattern built to analyze the possible valid values of a *Decision Variable* is presented in Figure 8. The orange parts can be defined with the business process analysis, although the green parts (instantiation of min and max ranges and definition of the input variable to be decided) will depend on the execution decision-making points.

Constraint Satisfaction Problems provide possible tuples of variables that satisfy the constraints. On the other hand, Constraint Optimization Problems provide the tuple of values that optimize a function. We can consider the utilization of an optimization in this context, however, it was dismissed for avoiding the reduction of the domain of the decision variables, necessary in uncertain scenarios. For the example, to minimize the outcomes, the person in charge of the decision can always select the lowest quantity of expenses and the highest of revenue. However, these decision could reduce the domain of the future decision provoking possible unsatisfiable business rules. This is why the proposal provides the possible range instead of a single value.

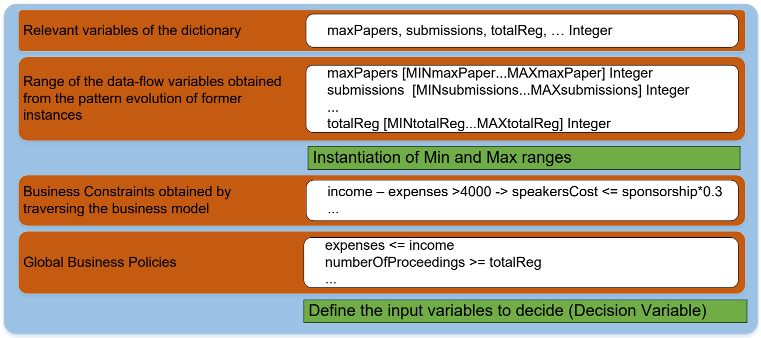


Fig. 8. Pattern of CSP for input-data decision-dmaking.

6. Decision-Making Support for Input Data in Decision Points

In this section, the whole DSS process is followed using a real set of *Process-Observational Variable* values applied to the example presented in Section 2. In the example, the number of copies of the Proceedings (`numberOfProceedings`) is an input variable whose value must be decided before the registration process is closed. However, our methodology can help in the decision-making regarding input data once the execution of an instance reaches a *Decision Point*, in this case, to decide the number of copies of proceedings to print in the activity *Print Proceeding*.

6.1. Ascertaining the current Stage

Table 5 shows the temporal variation of the most relevant *Process-Observational Variables* defined in the *Dictionary* of Section 4.2 for one simple former instance. This data has been processed with the aim of showing an understandable dataset, and, for this reason, every row has been summarized into weekly data, and several less relevant weeks have been removed. Several interesting aspects can be seen in this former instance, including aspects such as: The income for sponsorship remains unchanged from week 6; the submissions start in week 6, and their number remain unchanged after week 19; and the total number of registrations is 121 and this remains unchanged from week 36.

6.2. Estimation of the ranges for each *Process-Observational Variable*

The estimation of the ranges will be performed based on the current *Stage* at the moment of the decision, and the knowledge extracted regarding *Process-Observational Variable* temporal variation patterns. These ranges involve the possible ranges of potential final values of each *Process-Observational Variable*, based on the variation of *Process-Observational Variables* extracted from former instances of the business processes, as explained in Subsection 5.

week	sponsorship	submissions	totalReg	earlyReg	lateReg	acceptedPapers
0	0	0	0	0	0	0
1	100	0	0	0	0	0
2	10000	0	0	0	0	0
			...			
6	25000	1	0	0	0	0
7	25000	5	1	1	0	0
8	25000	5	4	4	0	0
9	25000	10	6	6	0	0
10	25000	25	8	8	0	0
			...			
19	25000	50	54	54	0	20
20	25000	50	66	66	0	20
21	25000	50	66	66	0	20
22	25000	50	67	66	1	20
23	25000	50	69	66	3	20
			...			
35	25000	50	114	66	48	20
36	25000	50	121	66	55	20
			...			
40	25000	50	121	66	55	20

Table 5. Temporal variation of the main POVars in the Dictionary

The *Stages* defined by business experts in Section 4.4 are applied with the experimentation data, and the data in each possible set of *Stages* that can be matched together are normalized to be comparable (shown in Section 5). The evolution patterns found are shown in Table 6. In order to create a comprehensible sample, only the most relevant *Process-Observational Variables* are shown.

Stages POVar \	sponsor.	submi.	totalReg	earlyReg	lateReg	accepted papers
earlyReg	0.00 - 625.00	1.25 -1.25	1.55 - 3.00	1.00 - 1.62	0.55 - 1.37	0.50 - 0.75
earlyReg paperSub	0.00 - 405.40	0.00 - 1.32	1.67 - 2.78	1.08 - 1.30	0.59 - 1.49	0.54 - 0.81
earlyReg notified	0.00 - 0.00	0.00 - 0.00	1.68 - 2.68	0.48 - 0.80	0.88 - 2.20	0.00 - 0.00
lateReg notified	0.00 - 0.00	0.00 - 0.00	0.00 - 1.10	0.00 - 0.00	0.00 - 1.10	0.00 - 0.00
lateReg notified confer.	0.00 - 0,00	0.00 - 0.00	0.00 - 0.00	0.00 - 0.00	0.00 - 0.00	0.00 - 0.00

Table 6. Knowledge extracted from Minimum and Maximum slopes for each POVar.

6.3. Instantation of the CSP template

When the possible ranges of the *Process-Observational Variables* have been obtained from the previous step, it is the moment to incorporate the obtained values into the CSP-template and to include the decision variable as the goal of the CSP. The way in which the template is instantiated is presented below.

Variable	Min Slope	Max slope	Current value	Estimated min value	Estimated max value
earlyReg	0.48	0.80	24	35.52	43.20
lateReg	0.88	2.20	0	21.12	52.80
totalReg	1.68	2.68	24	64.32	88.32
acceptedPapers	0.00	0.00	19	19	19
maxPapers	0.00	0.00	20	20	20
sponsorship	0.00	0.00	20000	20000	20000
earlyRegFee	0.00	0.00	300	300	300
lateRegFee	0.00	0.00	400	400	400
publicity	0.00	0.00	20000	20000	20000
venueCost	0.00	160.00	0	0	3840
venueCapacity	0.00	3.20	0	0	76.80
numberOfProceeding	0.00	2.80	0	0	67.20
proceedingPrice	0.00	3.20	0	0	76.80
galaDinnerPrice	1.60	2.80	0	38.40	67.20
lunchPrice	0.60	1.00	0	14.40	24
speakersCost	200.00	400.00	0	4800	9600

Table 7. Predicted minimum and maximum values for the related *Process-Observational Variables* in stage "earlyReg and notified"

Table 6 presents the knowledge regarding the temporal variation of *Process-Observational Variables* that has been extracted by analyzing the *Comparable Instances*. In Table 6, it can be observed how certain variables leave the value unchanged in certain stages, for instance, *earlyReg* has positive slopes when the set of stages defined by business experts is: *earlyReg* (Min: 1.00, Max: 1.65), *earlyReg and paperSubmission* (Min: 1.08, Max: 1.30), *earlyReg and notified* (Min: 0.48, Max: 0.80). This means that, in this situation, the variable has increased minimal and maximal values. However, in the stages *lateReg and notified*, and *lateReg and notified and conference*, the slopes are Min: 0.00, Max: 0.00, which means that, in these stages, the variable *earlyReg* remains unmodified.

Thanks to this information, once a decision regarding data has to be made, it is possible to estimate the final range of values of each variable, and a *Constraint Satisfaction Problem* can be built in order to verify that all constraints in the process are satisfied, and to inform the decision maker of the range of values of the *Decision Variable*.

In the aforementioned conference example, one instance of a decision is given in the establishment of the value of the variable *numberOfProceedings*, which is decided upon in the task *Print Proceedings*. For this example, the decision has been made at moment $t = 16$ weeks, and with the current values for the process following the current *Stages* of the *Process-Observational Variables* defined in the *Dictionary*: {submissions: 53, earlyReg: 24, lateReg: 0, totalReg: 24, acceptedPapers: 19, notified: 1, holdConference: 0, maxPapers: 20, submissionClose: 11, regOpen: 4, earlyRegClose: 23, regClose: 38, sponsorship: 20000, earlyRegFee: 300, publicity: 20000, lateRegFee: 400, venueCost: 0, numberOfProceedings: 0, proceedingPrice: 0, galaDinnerPrice: 0, lunchPrice: 0, speakersCost: 0}.

Therefore, by mapping this information with *Dictionary*, it is possible to observe that the stages activated are *earlyReg* and *notified*. As can be seen in Table 7, by using the previous knowledge extracted from former instances, it is possible to know how these variables will probably evolve. For instance, current *totalReg* is 24, and since the minimum slope and maximum slope calculated for these stages are: Min 1.68 and Max 2.68. Thereby, the final values will probably lie between:

- $EstimatedMinValue = (40t - 16t) \times 1.68\frac{totalReg}{t} + 24totalReg = 64.32\ totalReg$

- $EstimatedMaxValue = (40t - 16t) \times 2.68\frac{totalReg}{t} + 24totalReg = 88.32\ totalReg$

Other *Process-Observational Variables*, such as *sponsorship*, have a Min and Max slope of 0.00 in this set of stages and hence we consider that this value is fixed, in this case, to 20,000.

With this information, the CSP-template is instantiated according to the envelopers and the current stage, as shown in Figure 9. The instantiation of the Min and Max Ranges of the variables that evolve, and the input variable under decision is included (green part).

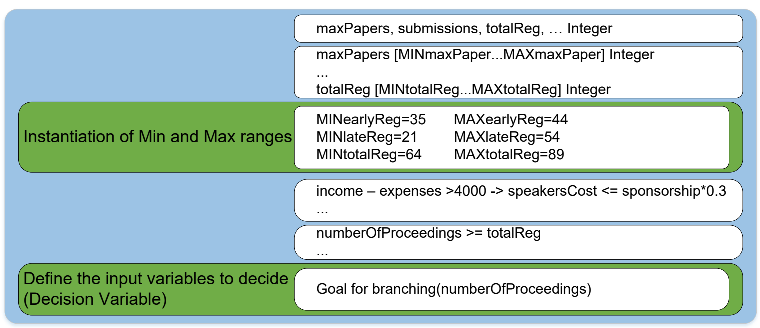


Fig. 9. Pattern of CSP for input at decision time.

The domain of the *Process-Observational Variables* depends on the stage of the variables when the decision is made, and how the value of the variables can evolve according to the envelopment temporal variation obtained in the previous subsection. The stage of the variables is known at decision time, and therefore the specific domain of the variables cannot be included in the CSP template, for the example *MINmaxPaper*, *MAXmaxPaper*, *MINsubmission*, *MAXsubmission*, *MINtotalReg* and *MAXtotalReg*.

Since the CSP solver returns all the possible values of the variables, it is necessary to limit it further to present only the values of the *Decision Variables*. To this end, the decision variables are defined as objectives during the propagation process where the variables are instantiated. This limitation enables the search to stop the propagation in those branches where no new values of decision variables can be found, thereby halting the unnecessary combinations of values. For each solution found, each value of the *Decision Variables* is stored in a sorted list. Each of these sorted lists is conditioned to return the list of intervals for each variable of decision. For example, if the values $\{1, 2, 3, 5, 8, 9, 10\}$ are found for the variable *x*, then the list of intervals built is $\{[1, 3], [5, 5], [8, 10]\}$.

6.4. Solution of the *Constraint Satisfaction Problem*

In order to compute the range of the *Decision Variable* to guarantee a successful execution, a Constraint Programming solver must solve the CSP, and obtain the possible range of the decision variable, *numberOfProceedings* for the example of the CSP above. Numerous commercial solvers are available. In this case, we have selected ChocoSolver [36].

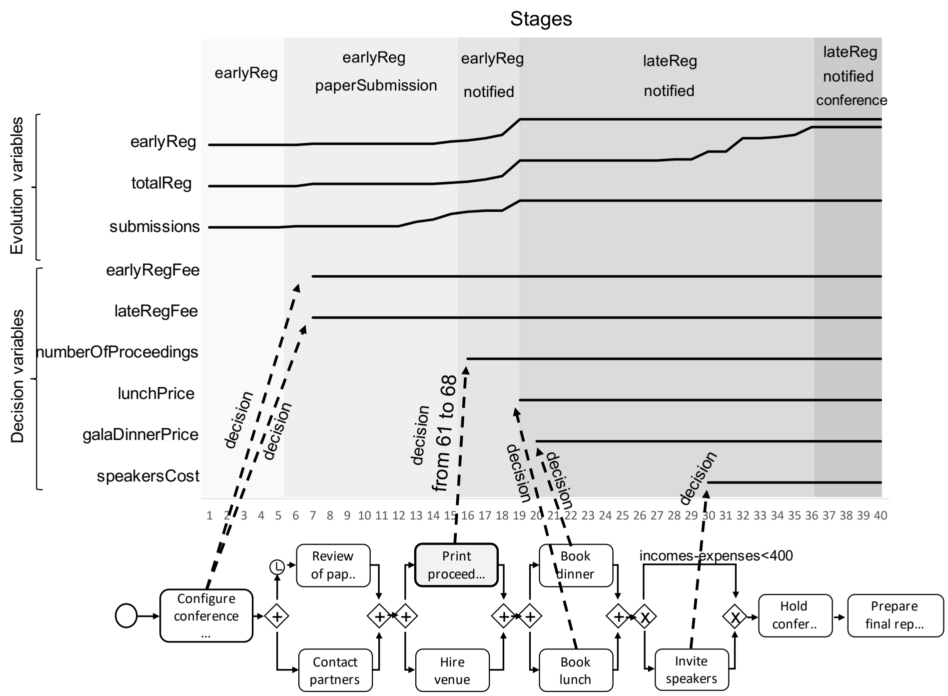


Fig. 10. Decisions and support system.

Figure 10 describes a possible process instance where the activities are executed and the decision are made, in accordance with the evolution of the evolution of the variables. For example, several decisions are made before the task *print proceedings* is executed, where the `numberOfProceedings` to print out has to be decided. Moreover, there is a set of evolution variables that are changing during the process instance. Following the resolution of the CSP, the values of the decision variables, that make possible the successful execution of the instance, are found, and the recommendation to the decision-maker is made. In the case of the `numberOfProceedings` to print out, the system determines that the value should lie between 61 and 68. With this information, the experts makes the last decision, and once it is made, as can be seen in Figure 10, the value of this variable is set. Of course, this value can also affect to future decisions.

As can be seen in Figure 10, for each decision, the system offers a range of values with which the process can be completed correctly, this is, in compliance with all the business rules. To ascertain better how the variable can evolve, the corresponding patter of evolution with the stage is used in each decision (represented with different color tones).

7. Related Work

In business processes, decision-making support contributes towards helping the business process designer choose the best combination of activities, to achieve a given objective. In the literature, **simulation-based** approaches have been proposed, such as [40] for complex dynamic systems and for the inclusion of uncertain data, or methods to optimize processes with fuzzy descriptions [41]. We observe that these types of methods ignore how the process works at runtime and fail to consider the importance of the variables of the data-flow. They are oriented towards the design of the model or the redesign of the business process [23] by analyzing the quality of the process at design time. **Data** has also been involved in other studies related to decision support; for example, [25] proposes operational decision support for the construction of process models based on historical data to simulate processes. That proposal includes a general approach to a business process for operational decision support and includes business process modeling and workflow simulation with the models generated, by using process mining. There are studies related to how to model the processes, such as that in [2], which proposes a framework of **assistance in the creation of models** by taking the necessary resources involved in the process into account. In that paper, the data that describes the resources of the execution of the process is used, but not the data that flows at run-time, nor is it considered how this assistance can function at run-time. Previous studious have faced the problem of optimal execution of decision models [4], where authors minimize and prioritize the acquisition of decision-related data by classifying decision inputs into decision trees, according to the degree of their influence on the outputs. The literature also contains methods for the discovering of the business process from logs, which can be used as a starting point for the business process design.

In general, we found that the literature is largely centered on the activity selection [3], or on the optimization of the process design [35], but not on the **assistance of the user for the input data**. Although work such as [21] is oriented towards auditing the process in order to detect gaps between the information system process flow and the internal control flow in the business process, the quality of the data values at run-time is not a cause of concern for the authors. Errors in the quality of data can be derived from the existence of an oversight in the description of the semantics of data in the business processes. The use of compliance rules has traditionally been used for the validation of the business process, and not for user assistance. The validation of business process traces has provided a field of intense research in recent years using business compliance rules; see [8] as an entry point into this literature. However, these types of proposals cannot be used in the decision-making support for input data since they are focused on compliance with the process model structure [38], [26].

Regarding how to **model data-aware compliance rules**, studies such as [42], [27], [20], and [1], have defined graphical notations to represent the relationship between data and compliance rules by means of data conditions. These types of compliance rules cannot, therefore, be employed to infer the possible values of the variables that are involved in the decisions. In [28], “semantic constraints” and the SeaFlows framework are proposed in order to enable integrated compliance support. Furthermore, in [22], a preprocessing step that allows the efficient verification of data-aware compliance is presented, whereby the data describes under what conditions the activities can be executed. In general, many examples can be found where data objects are used for compliance verification, for instance, the semantically annotating activities with preconditions and effects that may refer to data objects are introduced in [17], and the detection of tuples from different data-sources that refer to the same real-work entity can be found in [39], but none assist the user with this knowledge at run-time.

Summarizing, and to the best of our knowledge, only the preliminary studies [14] and [15] make use of the knowledge of the *Business Process Model* and the *Business Rules* for decision-making support for input data, while all other proposals are focused on the design or redesign of the model (*Business Process Model*). The current paper constitutes an improvement on these two previous papers by including not only *Expert Knowledge* but also the analysis of previous executions, in order to automatically extract knowledge that enables the evolution of the variables involved to be discovered, thereby offering better recommendations regarding input data to the decision maker.

8. Conclusions and Future work

Several decisions must be made at the operational level of an organization. Moreover, the are numerous situations that can affect the evolution of the organization. For this reason, when a decision is made, it is necessary to analyze the process model, the stage of the instance process, and to consider how other instances evolved in the past. All these elements are combined in the proposed DSS to help in the data input during process execution. This system provides a guide as to the correct management as defined in the business plans, by taking advantage of the information regarding former instances and business process knowledge. Thanks to this analysis, the information is set up to help in the decisions concerning input data in current instances, which for exceeds the simple use of this information for *Stages* reports or post-mortem analysis.

In order to develop the DSS, it is necessary to: model the problem, and include the business rules that define the goals of the organizations; create pattern of the temporal variation of the variables according to former instances; and create a CSP model that provides the correct domain for the input data that facilitates the correct finalization of the process.

Regarding future work, we consider three areas where our work could be more helpful: (1) through the improvement of the mechanism to ascertain the behavior of the business and its variables by applying data mining techniques; (2) through enriching the model with further components, such as the inclusion of external constraints related to services; and (3) through assistance in not only satisfying the strategic plan, but also in the optimization.

Acknowledgements

This work has been partially funded by the Ministry of Science and Technology of Spain ECLIPSE project (RTI2018-094283-B-C33) and the European Regional Development

Fund (ERDF/FEDER). Thanks to Lesley Burridge for the revision of the English version of the manuscript.

References

1. Awad, A., Weidlich, M., Weske, M.: Visually specifying compliance rules and explaining their violations for business processes. J. Vis. Lang. Comput. 22(1), 30–55 (2011)
2. Barba, I., Weber, B., Valle, C.D.: Supporting the optimized execution of business processes through recommendations. In: Business Process Management Workshops - BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I. pp. 135–140 (2011)
3. Batoulis, K., Baumgraß, A., Herzberg, N., Weske, M.: Enabling Dynamic Decision Making in Business Processes with DMN, pp. 418–431. Springer International Publishing, Cham (2016)
4. Bazhenova, E., Weske, M.: Optimal acquisition of input data for decision taking in business processes. In: Proceedings of the Symposium on Applied Computing. pp. 703–710. SAC '17, ACM, New York, NY, USA (2017)
5. Becker, J., Ahrendt, C., Coners, A., Weiß, B., Winkelmann, A.: Modeling and analysis of business process compliance. In: IFIP International Working Conference on Governance and Sustainability in Information Systems-Managing the Transfer and Diffusion of IT. pp. 259–269. Springer (2011)
6. Borrego, D., Gómez-López, M.T., Gasca, R.M.: Minimizing test-point allocation to improve diagnosability in business process models. Journal of Systems and Software 86(11), 2725–2741 (2013)
7. Cetin, S., Altintas, N.I., Solmaz, R.: Business Rules Segregation for Dynamic Process Management with an Aspect-Oriented Framework, pp. 193–204. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
8. Chesani, F., Mello, P., Montali, M., Riguzzi, F., Sebastianis, M., Storari, S.: Checking compliance of execution traces to business rules. In: Business Process Management Workshops. pp. 134–145 (2008)
9. Dechter, R.: Constraint Processing (The Morgan Kaufmann Series in Artificial Intelligence). Morgan Kaufmann (May 2003)
10. Eck, N.J.v., Waltman, L.: How to normalize cooccurrence data? an analysis of some well-known similarity measures. Journal of the American Society for Information Science and Technology 60(8), 1635–1651 (2009)
11. Gómez-López, M.T., Ceballos, R., Gasca, R.M., Valle, C.D.: Developing a labelled object-relational constraint database architecture for the projection operator. Data Knowl. Eng. 68(1), 146–172 (2009)
12. Gómez-López, M.T., Gasca, R.M.: Run-time monitoring and auditing for business processes data using contraints. In: International Workshop on Business Process Intelligence. pp. 15–25. BPI 2010, Springer (2010)
13. Gómez-López, M.T., Gasca, R.M.: Using constraint programming in selection operators for constraint databases. Expert Syst. Appl. 41(15), 6773–6785 (2014)
14. Gómez-López, M.T., Gasca, R.M., Parody, L., Borrego, D.: Constraint-driven approach to support input data decision-making in business process management systems pp. 457–469 (2011)
15. Gómez-López, M.T., Gasca, R.M., Pérez-Álvarez, J.M.: Decision-making support for the correctness of input data at runtime in business processes. Int. J. Coop. Info. Syst. 23(2), 29 (2014)
16. Gómez-López, M.T., Gasca, R.M., Pérez-Álvarez, J.M.: Compliance validation and diagnosis of business data constraints in business processes at runtime. Inf. Syst. 48, 26–43 (2015)

17. Governatori, G., Hoffmann, J., Sadiq, S.W., Weber, I.: Detecting regulatory compliance for business process models through semantic annotations. In: Business Process Management Workshops, BPM 2008 International Workshops, Milano, Italy, September 1-4, 2008. Revised Papers. pp. 5–17 (2008)
18. Hammer, M., Champy, J.: Reengineering the Corporation: A Manifesto for Business Revolution. Harper Business (1993)
19. Hay, D., Healy, K.A., Hall, J., et al.: Defining business rules-what are they really. The Business Rules Group 400 (2000)
20. Hoffmann, J., Weber, I., Governatori, G.: On compliance checking for clausal constraints in annotated process models. Information Systems Frontiers 14(2), 155–177 (2012)
21. Huang, S., Yen, D.C., Hung, Y., Zhou, Y., Hua, J.: A business process gap detecting mechanism between information system process flow and internal control flow. Decision Support Systems 47(4), 436–454 (2009)
22. Knuplesch, D., Ly, L.T., Rinderle-Ma, S., Pfeifer, H., Dadam, P.: On enabling data-aware compliance checking of business process models. In: Conceptual Modeling - ER 2010, 29th International Conference on Conceptual Modeling, Vancouver, BC, Canada, November 1-4, 2010. Proceedings. pp. 332–346 (2010)
23. Kock, N., Verville, J., Danesh-Pajou, A., DeLuca, D.: Communication flow orientation in business process modeling and its effect on redesign success: Results from a field study. Decision Support Systems 46(2), 562–575 (2009)
24. Lancaster, P., Salkauskas, K.: Curve and surface fitting. an introduction. London: Academic Press, 1986 1 (1986)
25. Liu, Y., Zhang, H., Li, C., Jiao, R.J.: Workflow simulation for operational decision support using event graph through process mining. Decision Support Systems 52(3), 685–697 (2012)
26. Ly, L.T., Rinderle, S., Dadam, P.: Integration and verification of semantic constraints in adaptive process management systems. Data Knowl. Eng. 64(1), 3–23 (2008)
27. Ly, L.T., Rinderle-Ma, S., Dadam, P.: Design and verification of instantiable compliance rule graphs in process-aware information systems. In: CAiSE. pp. 9–23 (2010)
28. Ly, L.T., Rinderle-Ma, S., Göser, K., Dadam, P.: On enabling integrated process compliance with semantic constraints in process management systems - requirements, challenges, solutions. Information Systems Frontiers 14(2), 195–219 (2012)
29. Montgomery, D.C., Peck, E.A., Vining, G.G.: Introduction to linear regression analysis. John Wiley & Sons (2015)
30. Motulsky, H.J., Ransnas, L.A.: Fitting curves to data using nonlinear regression: a practical and nonmathematical review. The FASEB journal 1(5), 365–374 (1987)
31. OMG: Object Management Group, Business Process Model and Notation (BPMN) Version 2.0. OMG Standard (2011)
32. Pérez-Álvarez, J.M., Gómez-López, M.T., Eshuis, R., Montali, M., Gasca, R.M.: Verifying the manipulation of data objects according to business process and data models. Knowledge and Information Systems (Jan 2020)
33. Pérez-Álvarez, J.M., López, M.T.G., Parody, L., Gasca, R.M.: Process instance query language to include process performance indicators in DMN. In: 20th IEEE International Enterprise Distributed Object Computing Workshop, EDOC Workshops 2016, Vienna, Austria, September 5-9, 2016. pp. 1–8. IEEE Computer Society (2016)
34. Pérez-Álvarez, J.M., Maté, A., Gómez-López, M.T., Trujillo, J.: Tactical business-process-decision support based on kpis monitoring and validation. Computers in Industry 102, 23–39 (2018)
35. Pivk, A., Vasilecas, O., Kalibatiene, D., Rupnik, R.: Ontology and SOA based data mining to business process optimization. In: Information System Development - Improving Enterprise Communication, [Proceedings of the 22nd International Conference on Information Systems Development, ISD 2013, Seville, Spain]. pp. 255–268 (2013)

36. Prud'homme, C., Fages, J.G., Lorca, X.: Choco Documentation. TASC - LS2N CNRS UMR 6241, COSLING S.A.S. (2017), http://www.choco-solver.org
37. Revesz, P.Z.: Constraint databases and temporal reasoning. In: Eigth International Symposium on Temporal Representation and Reasoning, TIME-01, Civdale del Friuli, Italy, June 14-16, 2001 (2001)
38. Sadiq, S.W., Orlowska, M.E., Sadiq, W.: Specification and validation of process constraints for flexible workflows. Inf. Syst. 30(5), 349–378 (2005)
39. Vieira, P.K.M., Lóscio, B.F., Salgado, A.C.: Incremental entity resolution process over query results for data integration systems. Journal of Intelligent Information Systems 52(2), 451–471 (Apr 2019)
40. Völkner, P., Werners, B.: A decision support system for business process planning. European Journal of Operational Research 125(3), 633–647 (2000)
41. Völkner, P., Werners, B.: A simulation-based decision support system for business process planning. Fuzzy Sets and Systems 125(3), 275–287 (2002)
42. Weber, I., Hoffmann, J., Mendling, J.: Semantic business process validation. In: 3rd international workshop on Semantic Business Process Management (2008)
43. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2007)