Extending BPMN 2.0 for Modelling the Combination of Activities That Involve Data Constraints

Luisa Parody, María Teresa Gómez-López, and Rafael M. Gasca

University of Seville, Computer Languages and Systems Department,
Avda. Reina Mercedes s/n, 41012 Seville, Spain
{lparody,maytegomez,gasca}@us.es
http://www.lsi.us.es/~quivir/

Abstract. The combination of activities to achieve optimal goals sometimes has a complex solution. Business Process Model and Notation (BPMN) 2.0 facilitates the modelling of business processes by providing new artifacts, such as various types of tasks, source of data and relations between tasks. Sometimes, although the order of the activities can be known, the concrete data values that the activities interchange to optimize their behaviour needs to be found, specially when input parameters of an activity affect to the input parameter of the others. Taking into account the lack of priority and clear sequential relationship between the activities of such combination, a deep analysis of possible models and data input values for the activities is necessary. For that reason, an extension of BPMN 2.0 with a new type of sub-process and its associated marker is proposed. The aim of this new sub-process is to define, in an easy way, a combination of several activities to find out, in an automated way, the concrete values of the data handling that optimize an overall objective.

Keywords: Business Process Management, Business Process Model and Notation, Combination of Activities, Data Input, Data Constraint.

1 Introduction

A business process, henceforth referred as BP, consists of a set of activities that are performed in coordination in an organizational and technical environment. These activities jointly perform a business goal [20]. The combination of activities is very important from the point of view of BP management, since it can increase customer satisfaction, reducing business investment and establishing new products and services at low cost.

Generally, a combination between various activities or business processes is almost always complicated, since every stakeholder focuses on their own interests. The degree of complexity becomes even greater when the objective of the combination is to optimize a common business goal. The ideal scenario arises when all the activities can obtain an optimal result based on the decisions made by each activity in an independent manner. The problem comes when values used as data input of an activity depend on the values taken for other activities in the process. Moreover, when the data input given by the user have interval domains (range of possible values), then obtaining the best local result does not necessarily imply that the best global result follows. In that case, how to combine the activities/sub-processes? This problem is worst when there is no priority and clear sequential relationship between the activities, since the artifacts in BPMN 2.0 required a decision order at design time (even in the ad-hoc sub-process the performer has to decide an activity order). Furthermore, the construction of models, in terms of activities order, implies an off-line analysis where the different combinations are studied. However, this off-line analysis is not enough when the behaviour and functionality of the activities are unknown, since they are external services which relationships between data input and output are unknown. For example, to known the price of a flight for a date and a pair of cities, it is necessary to call the service, then an off-line analysis is not enough to find out the best data input for the service. In that case, the relations between data inputs and outputs of activities have to be discovered, thus an analysis of all the possible data combinations and the behaviour of these activities in run-time is needed. In addition, the data combinations must take into account a set of existing data constraints that relate them and an objective function to be optimized. Therefore, the aim of the problem is to decide how to model with BPMN a BP whose activities (i) must be executed many times to search the concrete values for their data input that optimize the overall objective, and (ii) satisfy all the constraints that relate the values of the data input and output. This implies that the model has to be adaptive and flexible to manage the data handling in each instance.

The main standard used to model BP is Business Process Model and Notation (BPMN), proposed by OMG. The new version 2.0 solves the majority of the modelling problems. However, it remains as yet insufficiently powerful since, among other requirements, there is a significant need for the representation of the combination of activities where, the concrete data of each activity depends on the concrete data of the others, since they are related by means of constraints. To the best of our knowledge, there are no solutions with BPMN that enable to represent problems where the explicit order of the activities are not defined, and where the model is influenced by data constraints. Although it represents business processes, by means of Sequence Flows and Data Flows, the objective function and the constraints that relate the data input can only be added through annotations. Moreover, the annotations cannot be mapped to executed code in an automatic way.

For this reason, the aim of this paper is to extend the expressiveness of BPMN 2.0 with a new sub-process. The proposal is focused on the support of this combination of activities, whose main purpose is to optimize an objective function, where there is a set of possible data input for the activities that are related between them by means of constraints. Furthermore, this new sub-process enables the creation of executable code in an automatic way.

The rest of the paper is organized as follows: Section 2 motivates and explains the necessity of a combination of activities to search the concrete values of data to achieve a common goal. Section 3 summarizes the main issues about the combination of activities specification. Section 4 defines the BPMN extension to represent the combination of activities from the business descriptions perspective. Section 5 includes certain relevant related work. And finally, conclusions are drawn and future work proposed in Section 6.

2 Motivating Example

In this section we present a scenario to illustrate the combination of activities in the kind of problems addressed in this paper. Later on, it will be also used to explain our approach.

Our scenario is the Travel Booking Example presented in [13]. This example is proposed to show in-line event handling via event sub-process constructs. Although this is not the purpose of this work, the example highlights the lack of graphic representation for the combination of activities to search the concrete values for the data handling that achieve a common objective.

If we look through the Travel Booking Diagram (Chapter 9, page 28 in [13]), the customer wants to book a flight and a hotel room. The resulting business process (i) starts with the travel booking reservation request, (ii) follows with the search and evaluation of both flights' and hotel rooms' availability, (iii) selected alternatives are packaged and offered to the customer and (iv) the customer can select a proposed alternative or cancel the request. We are focused on the second step, where the searches of flights and hotel rooms are based on the customer request and are made in a parallel way. This is only possible when the customer request is formed by a concrete data input, in other words, when each data input has only one possible value (atomic value) and there is no relation between the customer criteria and the final result of both activities. It means that there are no relations between the activities, then they can be executed in a parallel way. For example, the customer wants to organize a trip, where he wants to depart on 2012-09-15 and return on 2012-09-30, with a flight from Madrid to London, and wants to book a hotel room in London during these days. The dates and the locations given by the customer are the data input. These data input have atomic values for the flight and hotel room searches, hence both searches can be performed in an independent way. Generally, the customer searches a trip for a concrete dates and location. Then, if necessary and also cheaper, the customer tries with another departure or return dates, thereby expanding the range of dates when leaves or arrives.

For that reason, we focus on problems when the customer wants to obtain the best result with a specific criteria by providing constraints between the data given. In that case, executing the activities in a parallel way is not enough, since obtaining the optimized value in an independent manner does not necessarily mean that the overall result will be the best option. For example, the customer wants the cheapest trip with the data input presented before, in this case also providing an interval (either range) of dates (for example, the departure date can be between 2012-09-15 and 2012-09-17, and the return can be either on 2012-09-30 or on 2012-10-01). Both, the flight and hotel room searches, have to combine their results and decisions to obtain the common objective. These possible values are also determined by a set of constraints that relates these data input, for example, (i) the depart date has to be a value between 2012-09-15 and 2012-09-17, (ii) the return date has to be a value between 2012-09-30 and 2012-10-01, and (iii) the return date of the flight and the check-out date at the hotel have to be the same. Therefore, this is an example where the activities share their data input. On the other hand, the objective function is a combination of the data outputs of these activities, given by their executions. In this example, the sum of the cost of buying an airline ticket and stay in a hotel room for these dates and locations. The aim of the problem is to know which is the best set of values for the interval data input that optimize the objective function and satisfy all the constraints.

3 Formal Definitions

In order to clarify the problem, a formalization is introduced. With the aim to understand how the relation between the activities data can be defined, the constrains definition that involve them and the objective function are essential. This definition enables the identification, description and definition of the type of problems that need the modelling for the combination of activities.

Let a **Combination of Activities that involves Data Constraint to Optimize an Objective Function (CAO)** be a business process whose activities $(A_1, ..., A_n)$ are independent and there is no priority order and clear sequence relationship between them. For each activity A_i, a set of input variables (I_{A_i}), and a set of output variables (O_{A_i}) are defined. Then, the following concepts are introduced for this CAO:

Definition 1. ***Activities Data Input (ADI):*** *The set of variables that represents the union set of all the data input of the* n *activities of the* CAO*.*

$$ADI = \bigcup_{i:1..n} I_{A_i}$$

Definition 2. ***Activities Data Output (ADO):*** *The set of variables that represents the execution result of the* n *activities* (O_{A_i})*.*

$$ADO = \bigcup_{i:1..n} O_{A_i}$$

Definition 3. ***Process Data Input (PDI):*** *The set of variables that represents the data introduced to the BP. Every variable* $x_j \in PDI$ *could have multiple values* $v(x_j) \in D(x_j)$*, where* $D(x_j)$ *is the domain of* x_j *(*$D(x_j)$ *is a finite set comprising all possible values that can be assigned to variable* x_j*).*

Definition 4. ***Process Data Output (PDO):*** *The set of variables returned by the BP. These variables represent the concrete values for the* ADI *that optimize the objective function. These concrete values will be provided to the user of the BP or another external process.*

Definition 5. ***Objective Function (ObjFun):*** *The global optimization function to be satisfied. This function can be defined in terms of* ADO*,* ADI *and* PDI*, and can be maximizing or minimizing.*

In a CAO, there is a set of **constraints (C)**, where each $C_k \in C$ relates a subset of variables $(x_m, .., x_z)$ belonging to the union of the ADI and PDI sets. And it represents a **subset of the cartesian product** $D(x_m) \times ... \times D(x_z)$ that specifies allowed combinations of values for the variables $x_m ... x_z$. The **result** of a CAO is an assignment v for that, each instance is a mapping that assigns to every variable $y_d \in PDO$ an element $v(y_d) \in D(y_d)$. This assignment v satisfies all the constraints belonging to C, such that $\langle \{y_{k_1}, ..., y_{k_g}\}, C_k \rangle \in C \; iff \langle v(y_{k_1}, ..., y_{k_g}) \rangle \in C_k$ and optimize the global function.

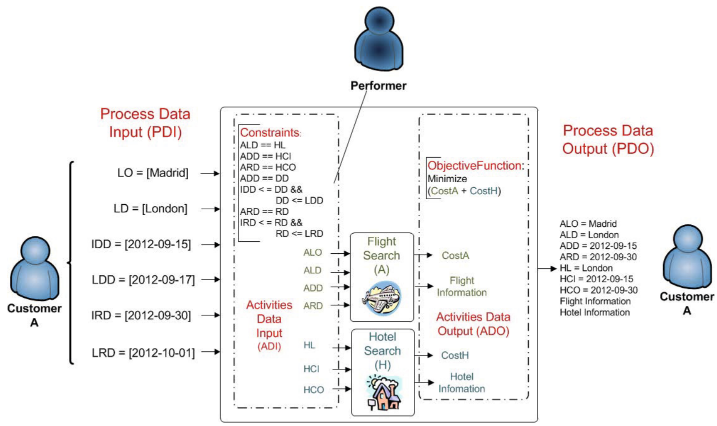

Fig. 1. The Travel Booking Example Structure

In the Travel Booking Example (see Figure 1), there are two activities: the Flight Search (A) and the Hotel Search (H). The PDI is the set of dates and locations given by the customer (location origin (LO), location destination (LD), initial depart date (IDD), last depart date (LDD), initial return date (IRD) and last return date (LRD)). The variables of the ADI are: airplane location origin (ALO), airplane location destination (ALD), airplane departure date (ADD), airplane return date (ARD), hotel location (HL), hotel check-in (HCI) and hotel check-out (HCO). There are many constraints that relate the ADI variables: the flight lands at the airport of the same city where is located the hotel, the departure and return date of the flight must match with the check-in and check-out date of the hotel, respectively. Moreover, there are constraints that relate

ADI with the *PDI*: for example, the value of the flight depart date has to fit with one of the depart dates given by the customer. On the other hand, there are two variables of the *ADO*: the cost of the flight and the cost of the hotel (CostA and CostH respectively). The objective function is to minimize the sum of both prices. Finally, the *PDO* is the set of the concrete values that optimize the objective function for the *ADI*.

4 BPMN Extension for Combining Activities That Involves Data Constraints

In business process modelling, one of the main goals is to facilitate the description of process model, shielding the user from unnecessary implementation details. However, BPMN 2.0 does not explicitly consider mechanisms to represent *CAO* as is defined in Section 3. For that reason, in order to capture these combination of activities within the business process, a notation supported by a set of graphical concepts is essential. This new notation enables a semantic representation and a graphical modelling. The extension presented in this section details this new notation and its marker associated.

BPMN 2.0 specification wants to stress the different stages in which the modeling process is composed: description, analysis and execution. The Description stage concerns the visible elements and attributes used in high-level modeling, in other words, the closest stage to the human level. The analysis stage contains all of the description stage and others, and it is closer to a software engine level. Both stages are focused on visible elements and a minimal subset of supporting attributes/elements. On the other hand, the execution stage focuses on what is needed to execute process models. To do this, following Subsections 4.1, 4.2, 4.3 and 4.4 define the metamodel, the new sub-process description, the operational semantics and the event handling, respectively. Finally, Subsection 4.5 presents a proposal to the executable stage. The typography, linguistic conventions and style of BPMN are also following to define this new extension.

4.1 Meta-model of the Combination of Activities Sub-Process

The Combination of Activities Sub-Process (CombA Sub-Process) meta-model proposed enables the analyst to describe and specify all aspects of *CAO* functionality and usage.

BPMN offers a way to represent the activities that have no REQUIRED sequence relationships through Ad-Hoc Sub-Process [7]. The Ad-Hoc Sub-Process semantics description is not enough for the combination of activities where the concrete values for the data input have to be found to optimize the goal, and there is no way to determine them at design time. Although there is no explicit process structure in Ad-Hoc Sub-Process, some sequence and data dependencies can be added to the details of the process. In addition, the performers[1]

[1] A Performer defines the resource that will perform or will be responsible for an activity. The performer can be specified in the form of a specific individual, a group, an organization role or position, or an organization [7].

determine when the activities will start, what the next activity will be, and so on. However, this is not valid when there are no sequence relationships, the data dependencies are on the data input of the activities and the performers cannot determine the sequence, as occurs in *CAO*. Furthermore, the list of BPMN elements that MUST NOT be used in an Ad-Hoc Sub-Process includes: Start and End Events, Conversations (graphically), Conversations Links (graphically) and Choreography Activities, which are useful to combine the activities to achieve an optimal goal.

In order to include these new requirements, an extension of Sub-Process definition [7] is necessary. A new contextual scope is defined to achieve an optimization agreement between different parts to search the data input values of the activities that optimize a goal. The new meta-model related to Sub-Process, adding the new type of Combination of Activities Sub-Process, called CombA Sub-Process, is presented in this paper.

4.2 CombA Sub-Process Definition

First of all, and according to the BPMN methodology, a descriptive stage is necessary to define the new sub-process. This description enables a high-level modeling through a visible element and a set of attributes.

CombA Sub-Process is a specialized type of Sub-Process which is a set of activities[2] that have no REQUIRED sequence relationships. This new sub-process has to combine the activities in order to search the concrete values for the interval data input handle that optimize a common objective. The set of activities can be defined in the process. However, the sequence and the number of performances for the activities cannot be determined by the performers of the activities, since the behavior of an activity depends on the data input belonging to other activities, and vice versa.

The main purpose of the combination of activities is to search the values of the data input that optimize a goal, not as much as the execution of the activities. However, the execution of the activities is a consequence since the objective function to optimize depends on the data output of the activities. Therefore, the activities have to be executed several times to agree which are the best values of the data input. If there were no intervals in the data input (since they have atomic values), the activities have to be executed only once to answer the customer request.

The formal definition presented in Section 3 is translated into BPMN elements in order to define the CombA Sub-Process. Therefore, the CombA Sub-Process is composed of:

- Activities (A in the formal definition): generally, these activities are tasks. A task is a unit of work, the job to be performed. However, each activity, in turn, can be another process. The unique requirement is that each activity has different and independent functionality.

[2] An Activity is a Process step that can be atomic (Tasks) or decomposable (Sub-Processes) [7].

– Data Object: it represents the information flowing through the process. The CombA Sub-Process handled mainly two types of data:
  • Data Input: there are two types of data input in the CombA Sub-Process: the first corresponds to the external input for the entire process. It can be read by an activity and is given by a performer (*PDI* in the formal definition); and the second type corresponds with the data input of each activity that participate in the optimization agreement (*ADI* in the formal definition).
  • Data Output: there are two types of data output: the first one corresponds to the variable available as result of the entire process and the answer of the customer request (*PDO* in the formal definition). And the second type corresponds to the output value of each activity (*ADO* in the formal definition).
– Constraints (*C* in the formal definition): a set of Formal Expressions[3] that relate both types of data input.
– Optimization Function: The Formal Expression used to define the function to be optimized. This function relates the data output of the activities.

As the visible element for the modelling, we propose to use a puzzle piece symbol like the marker for the CombA Sub-Process and it can be used in a collapsed and expanded way. The reason of choosing this symbol is that the activities in the CombA Sub-Process have to fit as the pieces of a puzzle, that can be a right visualization for the business level. The circular structure in the expanded CombA Sub-Process represents the lack of predefined order and sequence relationship, centered on the objective function, which is the CombA Sub-Process Core.

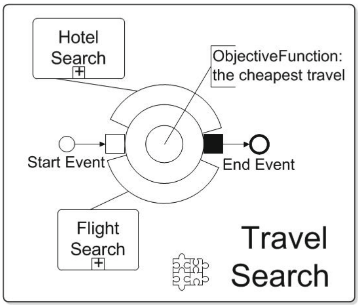

Fig. 2. Expanded Search Travel CombA Sub-Process

Figure 2 depicts the way to represent the combination of activities between Airline Searching and Hotel Searching activities with the new CombA Sub-Process marker.

---

[3] A Formal Expression is used to specify an executable Expression using a specified Expression language. A concrete constraint language is the one proposed in [6].

This CombA Sub-Process can be added to the diagram presented in [13] replacing the parallel search of flights and hotel rooms with this new Travel Search Sub-Process (see Figure 3).

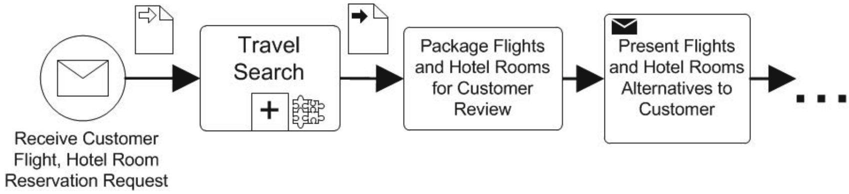

Fig. 3. Collapsed Travel Search CombA Sub-Process included in the example presented in [13]

The CombA Sub-Process element inherits the attributes and model associations of activities through its relationship to sub-process [7]. Table 1 presents the additional attributes and model associations of the CombA Sub-Process.

Table 1. CombA Sub-Process model associations

Attribute Name	Description - Usage
objectiveFunction: Formal Expression	Definition of the global optimization goal.
constraints: set of Formal Expressions	Definition of the constraints that relate the data input of the activities, limiting their possible behaviours.
numberSolutions: Integer	Attribute to determine if the CombA Sub-Process searches one of the best solutions or all the best solutions (values 0 or 1 respectively). The default value is 0.

Activities within this sub-process are generally disconnected from each other. During execution of the Process, all the activities are activated asynchronously and have to communicate their decisions to the others to achieve an agreement.

The formal definition given in Section 3 for the Trip Planner Example is valid for the CombA Sub-Process definition. Therefore, the attributes of CombA Sub-Process for the example are:

- **objectiveFunction:** To get the cheapest trip, the objective function is:
  $$min(costA + costH).$$
- **numberSolutions:** All the best solutions regarding the objective function to optimize. According to the values given in Table 1, the value is 1.
- **constraints:** Two of the possible constraints of the problem are: the depart date has to be a value between the initial depart date and the last departure date given by the user (C1) and the destination airport is at the same location than the hotel (C2).

$$(C1) IDD \leq DD \;\&\&\; DD \leq LDD \qquad (C2) ALD == HL$$

4.3 CombA Sub-Process Operational Semantics

As the definition of Sub-Processes presented in [7], a CombA Sub-Process is an activity that encapsulates a Sub-Process that is in turn modeled by Activities, Gateways, Events and Sequence Flow. Moreover, the activities involved in the combination could be composed of Conversations, Choreographies and other Sub-Processes. The CombA Sub-Process is instantiated when it is reached by a Sequence Flow token[4] through an unique Start Event. The CombA Sub-Process instance is completed when the activities achieve the optimal goal and there are no more tokens in the sub-process and none of its activities are still activated, in other words, when the End Event is reached.

As long as the activities do not find the optimal goal, and therefore, the concrete values for the data input, the sub-process will not end. The activities could find various best results since various combination of values can produce the same result. After the activities obtain an optimal value of the objective function, the value specified through the integer *numberSolutions* attribute is evaluated.

4.4 CombA Sub-Process Handling Events

One of the main problem of the combination of activities is the execution time. It can occur that the activities take a long time to find the combination of values that optimize the overall goal. In order to control this kind of problems, BPMN provides a set of Timer Events. If after a period of time no optimal goal is obtained, then the operation is canceled. Although, the CombA Sub-Process could return the best solutions found until that moment, it is not guaranteed that the optimal solution is among them. Sometimes, this timer requirement is provided by the customer who does not want to wait. However, timer requirement can also be used to provide quality products from the service provider point of view.

In general, BPMN provides several event handlers that help to manage and solve situations that happen during the course of a Process instance. Various of these event handlers can be applied to the CombA Sub-Process.

An Intermediate Event indicates where something happens (an Event), somewhere between the start and the end of a process. It will affect into the flow of the process, although this event will not start or (directly) terminate the process [7]. The Intermediate Timer Event, specifically the interrupting type, interrupts the activity, to which is attached, changing the normal flow into an exception flow.

Applied to the CombA Sub-Process (see Figure 4), it implies that there will be two outcomes: successful completion and failed completion.

Although in this paper only Timer Events are considered, a CombA Sub-Process can be combined with the set of Events given by BPMN 2.0 (e.g. Error,

[4] The concept of a token is used to facilitate the discussion of how Sequence Flows are used within a Process. A token will traverse the Sequence Flows and pass through the elements in the Process. A token is a theorical concept that is used as an aid to define de behavior of a Process that is being performed.

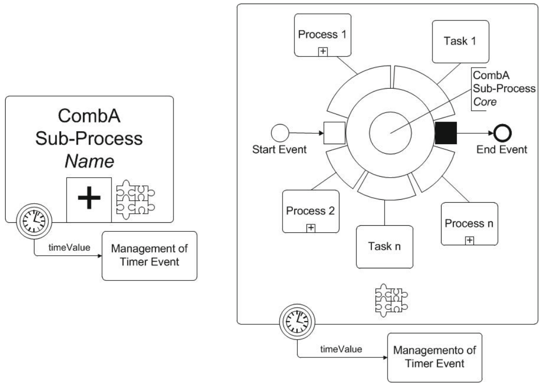

Fig. 4. A collapsed and expanded CombA Sub-Process with Timer Event

Message, Conditional, etc) [7] and completed handling these possible events as it is done by sub-processes. For example, an Error Event could warn when the problem is over-constrained. Since, although some over-constrained problems could be detected statically by the problem formulation, in the majority of cases it depends on the data returned by the activities. For example, if there are no flights for a concrete dates given by the user, then the problem is over-constrained.

4.5 CombA Sub-Process Execution Semantics

BPMN 2.0 pays special attention to the execution semantics since there is an important need of executable process models. In this subsection, a proposal is presented to implement and execute this CombA Sub-Process semantics.

A CombA Sub-Process contains a number of embedded inner activities and is intended to be executed with more flexible ordering, compared to the routing of Processes or Ad-Hoc Sub-Processes. The contained Activities in CombA Sub-Process are generally disconnected from each other and executed asynchronously. There are several ways to combine the activities and all of them depend on the performer (designer) criteria.

The combination of activities is formed by a set of activities whose objective is to search the concrete values of the data input that optimize an output. This problem is similar to Distributed Constraint Satisfaction Problems (DisCSP) [23], where the information is spatially and/or semantically distributed among various nodes where no one knows the whole information and the behaviour of each node. In [15], a methodology to model and execute combination of activities in business processes based on DisCSP is proposed. In order to solve the mentioned combination, it has developed an adaptation of the centralized backtracking algorithm existing for DisCSP. In a centralized solution, there is a coordinator that has the knowledge of the whole problem and ensures that the data input of the activities satisfy all the constraints and finds out the optimal value of the objective function. Thanks to this adaptation, the activities achieve the coordination to obtain an overall goal.

5 Related Work

One of the main ideas of BPMN is the imperative representation of business processes with their activities and the execution constraints between them. However, the execution order and the constraints that relate the activities cannot be always described with a traditional business process model.

There are many business process modelling techniques [1]: Event-Driven Process Chain (EPC) [17], UML [16] [9], Integration Definition for Function Modelling (IDEF) and Petri-Nets [3] [20]. Furthermore, Ruopeng Lu and Shazia Sadiq in [10] present a comparative of business process modelling approaches including graph-based modelling language and rule-based formalism. However, the most used graphical standard for modeling BP is Business Process Model and Notation (BPMN) [12] proposed by OMG, recently released the version 2.0 [7], which can be used for a wide range of problems [21]. This notation must also permit the incorporation of various perspectives giving place to various diagrams. The diagrams must show the rules, goals, objectives of the business and not only relationships, but also interactions [4]. A great part of the success of the modelling is the capacity to express the various needs of the business, as well as to have a notation in which these needs can be described. However, although it could not be an easy task, the business process, the environment features and the intended use of the model must be taken into account to make a successful choice of an approach and/or notation [2].

On the other hand, Yunzhou Wu et al. in [22] show how the constraints that necessitate coordination may be represented in the Business Process Execution Language for Web Services (BPEL). BPEL is an OASIS standard executable language for specifying actions within business processes with web services [11]. Processes in BPEL export and import information by using web service interfaces exclusively. But there is not a protocol defined in BPEL to combine the activities nor to select the concrete values of the data input according to the optimization function. The authors in [22] use a generalized adaptation and constraint enforcement models to transform the traditional BPEL process into an adaptive process. However, the authors solve the combined adaptation and constraint enforcement models in order by obtaining a policy that recommends adaptive actions while respecting the constraints. Therefore, there is no combination to search the concrete values for the interval data input in terms defined in our work. One way to represent the constraints is by means of the syntax of the Semantic Web Rule Language (SWRL) [19]. SWRL is used to specify the constraints and embed the constraints based on SWRL within BPEL.

Service-oriented systems have emerged as the paradigm to provide such automated support for business processes. Van der Aalst et al. in [18] and Papazoglou et al. in [14], present Web Services as the infrastructure to foster business processes by composing individual Web Services to represent complex processes.

There are several studies on the composition of services that can be extrapolated to the composition of activities in business processes since services are specific activities in the business process. To the best of our knowledge, none of these studies represents graphically or solves the type of coordination that this paper presents: a combination of independent activities to find the concrete values of their data input that optimize an overall objective.

Malinda Kapuruge et al. in [8] provide a systematic analysis of the requirements for process flexibility in the context of service compositions, and they analyze the existing approaches against these sets of requirements. Based on this analysis, some general observations are defined for certain critical issues for future investigation into business process flexibility support in service composition. However, a combination of the activities is possible against the lack of priority in the order of the activities presented in this paper.

Otherwise, Umeshwar Dayal et al in [5] define the coordination as a collaborative process, where the best service from among a set of existing and available services is chosen in order to fulfil a common need. Umeshwar Dayal et al. have analyzed and identified the requirements for business process flexibility in service composition and compared how existing process modelling and enactment approaches fulfil these requirements. However, in this work, each service that participates in the combination has an independent and distinct functionality. It is assumed that each of these services is the best at obtaining this functionality since the main objective in our work is that these services search the concrete values for their data input. Therefore, the way that these best services are chosen is irrelevant to this paper.

There are also various studies on planning algorithms for Web Services. The planning [24] has similar features with the same kind of combination as that presented in this paper. The input and output parameters of the distinct participating Web Services provide the basis of the planning. These relationships involve sequence relationships between the numerous Web Services. However, in this paper, there is no possibility of sequence relationships since only the data input are related.

6 Conclusions and Future Work

In this work, an extension of BPMN 2.0 for modelling the combination of activities that involve data constraints is proposed. The aim of this combination is to search the concrete values for the data handle by the activities in order to optimize an overall objective. This proposal arises from the need to combine various activities or sub-processes that belong to a business process and work concurrently achieving an optimization of an overall goal.

A CombA Sub-Process with its associated marker enables to incorporate combination requirements into a business process diagrams. These combination requirements will increase the scope of the expressive ability of business description. The CombA Sub-Process describes graphically the features of this combination of activities: there is a set of activities whose data input have a range of possible values, and have to optimize a common objective assigning concrete values to these data input. The main problem in this type of combination of activities is that the Activities have no required sequence relationships and their data input are related, by means of constraints, making possible several input and output data for each activity in an instance. There is no way to represent this with BPMN 2.0, hence the CombA Sub-Process element is proposed.

Future work will be oriented to enrich the combination of activities specifications and study in depth the analytic and common executable process modelling.

Acknowledgments. This work has been partially funded by the Junta de Andalucía by means of la Consejería de Innovación, Ciencia y Empresa (P08-TIC-04095) and by the Ministry of Science and Technology of Spain (TIN2009- 13714) and the European Regional Development Fund (ERDF/FEDER).

References

1. Aguilar-Saven, R.S.: Business process modelling: Review and framework. International Journal of Production Economics 90(2), 129–149 (2004)
2. Bidel, I.: Choosing approach to business process modeling - practical perspective (2007)
3. Bosilj-Vuksic, V., Hlupic, V.: Petri Nets and IDEF diagrams: Applicability and efficacy for business process modelling. An International Journal of Computing and Informatics 25(1), 123–133 (2001)
4. Castela, N., Tribolet, J.M., Silva, A., Guerra, A.: Business process modeling with uml. In: ICEIS (2), pp. 679–685 (2001)
5. Dayal, U., Hsu, M., Ladin, R.: Business process coordination: State of the art, trends, and open issues. In: Apers, P.M.G., Atzeni, P., Ceri, S., Paraboschi, S., Ramamohanarao, K., Snodgrass, R.T. (eds.) VLDB, pp. 3–13. Morgan Kaufmann (2001)
6. Gómez-López, M.T., Reina-Quintero, A., Martínez Gasca, R.: Model-Driven Engineering for Constraint Database Query Evaluation. In: First Workshop Model-Driven Engineering, Logic and Optimization: Friends or Foes? (MELO 2011) (June 2011)
7. Object Management Group. Business Process Model and Notation (BPMN) Version 2.0. OMG Standard (2011)
8. Kapuruge, M., Han, J., Colman, A.W.: Support for business process flexibility in service compositions: An evaluative survey. In: Australian Software Engineering Conference, pp. 97–106. IEEE Computer Society (2010)
9. List, B., Korherr, B.: A UML 2 Profile for Business Process Modelling. In: Akoka, J., Liddle, S.W., Song, I.-Y., Bertolotto, M., Comyn-Wattiau, I., Cherfi, S.S.-S., van den Heuvel, W.-J., Thalheim, B., Kolp, M., Bresciani, P., Trujillo, J., Kop, C., Mayr, H.C. (eds.) ER Workshops 2005. LNCS, vol. 3770, pp. 85–96. Springer, Heidelberg (2005)
10. Lu, R., Sadiq, S.: A Survey of Comparative Business Process Modeling Approaches. In: Abramowicz, W. (ed.) BIS 2007. LNCS, vol. 4439, pp. 82–94. Springer, Heidelberg (2007)
11. OASIS. Business Process Execution Language for Web Services. OASIS Standard (2005)

12. OMG. Business Process Model and Notation (BPMN) 1.2. OMG Standard (2009)
13. OMG. BPMN 2.0 by Example. Version 1.0 (non-normative). OMG Standard (2011)
14. Papazoglou, M.P., van den Heuvel, W.-J.: Service oriented architectures: approaches, technologies and research issues. VLDB J. 16(3), 389–415 (2007)
15. Parody, L., Gómez-López, M.T., Martínez Gasca, R., Borrego, D.: Using distributed csps to model business processes agreement in software multiprocess. In: 3rd International Conference on Agents and Artificial Intelligence (2011)
16. Sinogas, P., Vasconcelos, A., Caetano, A., Neves, J., Mendes, R., Tribolet, J.M.: Business processes extensions to uml profile for business modeling. In: ICEIS (2), pp. 673–678 (2001)
17. Tsai, A., Wang, J., Tepfenhart, W., Rosea, D.: Epc workflow model to wifa model conversion. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, SMC 2006, vol. 4766, pp. 2758–2763 (2006)
18. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business Process Management: A Survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
19. W3C. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C (2004)
20. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer (2007)
21. Wolter, C., Schaad, A.: Modeling of Task-Based Authorization Constraints in BPMN. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 64–79. Springer, Heidelberg (2007)
22. Wu, Y., Doshi, P.: Making bpel flexible - adapting in the context of coordination constraints using ws-bpel, pp. 423–430 (2008)
23. Yokoo, M., Durfee, E.H., Ishida, T., Kuwabara, K.: The distributed constraint satisfaction problem: Formalization and algorithms. IEEE Transactions on Knowledge and Data Engineering 10(5), 673–685 (1998)
24. Zhang, J.F., Kowalczyk, R.: Agent-based dis-graph planning algorithm for web service composition. In: CIMCA/IAWTIC, p. 258. IEEE Computer Society (2006)