Process Variability Modeling for Complex Organizations

Riccardo Cognini, Flavio Corradini, Andrea Polini, Barbara Re
Computer Science Division, School of Science and Technologies
University of Camerino, 62032 – Camerino (MC), ITALY
Email: {firstname.lastname}@unicam.it

***Abstract*—When organizations provide similar services and share the same mission they often behave similarly. This, in particular, is true in the context of Public Administration where different offices organize the provisioning of services to citizens in similar ways. This paper presents a novel notation and approach to support variability modeling for those scenarios in which it is difficult to fully foresee in advance how variability can affect the various process perspectives. Notation and approach are inspired on feature modeling where features are used to represents activities of a processes family that can be differently implemented. Applying the proposed approach it is possible to derive a kind of partially predefined process model variant, which is a set of fragments, that in a subsequent step needs to be augmented with additional elements to then fully define process behavior. Notation and approach has been validated on real case studies with encouraging results.**

I. Introduction

Nowadays services providers operate in complex socio-economic contexts. Heterogeneous needs are given by a multitude of requirements coming from internal and external causes with respect to the organization [1]. Such needs influence the organization structure and the way it operates. Services provision is slightly different even if sharing the same goals. This asks to support variability as a natural aspect toward services customization. We believe that services are the result of a running Business Process (BP). Methods explicitly supporting variability modeling and management for BPs are needed. Variability is the ability of deriving BP variants from a configurable BP model [2], that is a generic model integrating all the possible BP variations [3]. Given a configurable BP model relevant are the steps needed to derive BP variants starting with a possible configuration. Than individualization (selection) is performed in order to derive a specific BP variant. Each supported BP variant acts as a blueprint for a set of instances. This means that a configurable BP model guides the users to a solution that better fits to a specific working context [4].

Complex organizations can take advantage by the possibility to globally specify the expected behavior and than specialize according to internal need. In such a complex organization process variability, if suitably managed, can contribute to eliminate model redundancies by representing the commonalities of different BP variants only once. Furthermore the possibility to explicitly express variability fosters models reuse improving the number of possible target organizations for the same model.

Public services delivery fits in such scenario. Focusing on the BP in behind a given service all the Public Administrations (PAs), at least in the same country, share an abstract BP model since all of them need to satisfy law constraints. Taking the point of view of a PA more detailed activities have to be introduced in order to implement a service. The resulting BP models start to differentiate considering, among the others, organization dependent characteristics [5]. Indeed, organizational structure of PAs is similar but not identical. It is the case that in big municipalities the office organization is different than in small municipalities.

In this paper we presents a novel notation and a modeling approach, named Business Process Feature Model (BPFM), that are able to represent a configurable BP model and support the BP variants derivation. The BPFM notation is an extension of Feature Models (FMs) incorporating aspects of BP modeling, permitting to represent activities, their partial execution order, and Data Objects. The notation includes new constraints that differentiate between the static inclusion and the dynamic occurrence of an activity. With reference to the Data Object modeling the notation supports the part-of relation, while some Data Objects are primitives, other can be decomposed into more fine-grained object. A BPFM model includes all the BP variants as a configurable process model. The approach permits to model variability and to derive BP models applying four steps as will be detailed in the following. The key novelty of the approach is that the BPFM configuration results in a kind of pre-BP model, which is a set of fragments, that then in a subsequent step needs to be augmented with additional sequence or parallel flow elements by the designer. Encouraging results coming from in the application to the European Project Budget Report (EPBR) case study.

The paper is organized as follow. Section II reports some background materials. Section III presents related works. Than, Section IV summarizes the running example and Section V introduces BPFM notation and approach. Finally, Section VII closes the paper with some conclusion and opportunities for future work.

II. Background

A. Feature Modeling

Feature modeling is an approach emerged in the context of Software Product Lines to support the development of a product family from a common platform. It aims at lowering both production costs and time in the development of individual products sharing an overall configurable model, while allowing them to differ with respect to characteristics to serve, e.g. different markets [6].

A Feature Model is a graphical model in the form of a tree representation in which the root represents the generic product. In particular, in the first feature modeling approach, named Feature-Oriented Domain Analysis, *mandatory*, *optional* or *alternative* features can be represented [7]. *Mandatory* features introduce characteristics that each product must have (i.e. each mobile device has a screen) (Fig. 1-A). *Optional* features introduce characteristics that a product can have, but a fully functional product can also be derived without including such a feature (i.e. a mobile device can have a 4G connection or not) (Fig. 1-B). Finally *Alternative* features introduce characteristics that cannot be included together in a product (i.e. a mobile device can have a standard screen or a touch screen, not both) (Fig. 1-C). Researchers have proven that basic FM models are too restrictive to represent all the relationships between features which are useful to characterize a family of products [8]. As a result the FM notation has been extended for instance to express relationships such as *"at least one feature in a set of features is needed in each product"*. This is done via *OR features* (Fig. 1-D). Additionally, *include relationship* is used to express that a feature implies another feature that is on a different part of the tree (Fig. 1-E), and *exclude relationship* is used to express that a feature requires to discard another one that is on a different part of the tree (Fig. 1-F). Starting from a FM a specific product can be built satisfying each relation in the tree starting from the root to the leafs. The selection of the various element according to the defined relations is generally referred as a *configuration*.

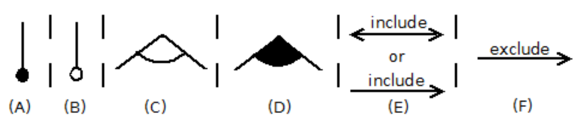

Fig. 1. Feature Model Graphical Notation.

B. Business Process and Related Perspective

A Business Process is defined as "*a collection of related and structured activities undertaken by one or more organizations in order to pursue some particular goal. Within an organization a BP results in the provisioning of services or in the production of goods for internal or external stakeholders. Moreover BPs are often interrelated since the execution of a BP often results in the activation of related BPs within the same or other organizations*" [9]. Different perspectives can impact BP modeling and execution [10] in particular with reference to variability issues [11]. They are following described and then referred in the rest of the paper.

- *Functional Perspective* specifies which units of work compose the BP. Units of work are the activities (atomic or composed) that may be performed to reach a particular goal.
- *Behavioral Perspective* specifies the behavior of BP via control flow relationship between activities. BP elements that impact in this perspective are gateways and control edge.
- *Organization Perspective* specifies the different actors and roles that are involved in the BP.
- *Information Perspective* specifies which information entities such data and Data Objects are used or generated during the BP execution.
- *Operational Perspective* specifies how atomic activities are implemented. For each atomic activity many implementations can exist.

The need to consider such an high number of perspectives, when modeling a BP, makes clear its complexity. Approaches and tools permitting to focus on different perspectives at different time could permit to get better results in BP modeling. Indeed this is even more important when also variability aspects have to be considered.

III. Related Works

Modeling variable BP is the ability to represent in a single model many alternative BPs sharing the same goal [2]. In order to describe variable BPs several approaches have been proposed, in some case extending already available notations. Relevant examples are certainly languages such as C-EPC [12], C-iEPC [13], vBPMN [14] or C-YAWL [15]. Also language independent approaches have been proposed. Among the others PROVOP [16] and PESOA [17] are the most used. Differently from our proposal such modeling languages permits to derive variants for which the control flow is fully determined. The configurable model includes all the possible control flow relations. These approaches cannot be easily applied when the characteristics to consider to derive the variant are not known a priori since they could depend from the specific organization in which the BP will be finally deployed. Indeed this is a quite general case for PAs. Our approach permits to derive variants for which the flow can be further refined after a configuration step taking into account information which are dependent from the deployment context.

The need of introducing BP variability in PA domain is also discussed [18] [19]. Differently from our work they do not consider the variability at the level of organization structure and they assume to have a fully structured control flow at the level of configurable BP model that as said can not be always easily defined. Moreover these works use the C-YAWL approach that does not permit to represent input-output data-objects.

Alternative approaches are those based on the declarative paradigm such as CMMN [20] and Declare [21]. Differently from our proposal such approaches do not intend to provide variants with a fully specified control flow and typically defer the definition of a precise order between the activities till their execution, and therefore are mainly meant to support knowledge intensive processes.

IV. The EU Grant Management Case

Since 1984 the European Commission (EC) finances research and innovation within the European Union (EU) through the "*Framework Programmes for Research and Technological Development*" generally referred as FP. The participation to a EU financed project obliges the beneficiary in grant management and related budget reporting activities as an evidence of the tasks performed within the project. The BP related to such activities has been chosen as the reference scenario of our work since it is enough simple but at the same time sufficiently complex to validate the proposed notation and approach. In particular, we refer to the reporting activities of Public Administrations in Italy (e.g. regions, municipalities, universities). European Commission itself recognized the complexity of the reporting procedures for FP, and has now simplified it in the new research framework program, referred as "Horizon 2020". Here we discuss only a small part of the overall *periodic budget reporting* BP, focusing on the sub-processes related to the reporting of different projects according to different founding schemes. The BP under analysis refers activities that a specific organization has to put in place in order to report on work done, and related costs, with reference to a specific project as described in the signed consortium agreement and in the corresponding Description of Work document (DoW).

At an abstract level the reporting activity is quite similar for all PAs participating to a successful proposal. The complexity emerges from the need to manage and report different types of project e.g. Network of Excellence, Collaborative Project and Coordination and Support Action. The following high level requirements can be identified and will be used as examples in the next sections to show how variability can be modeled. In particular, these requirements introduce variability needs with reference to the functional perspective, and partially to the behavioral perspective.

- **Req1** - all participants **must** fill the form C. This requires to report direct and indirect costs. The direct costs result from the sum of the personnel costs, **if there are**, subcontracting and other possible admissible costs (e.g. hardware) according to what specified in the consortium agreement. The indirect costs **should be** calculated using **four different possible strategies** such as *actual indirect costs*, *actual indirect costs using simplified method*, *flat rate of 20%* or *flat rate of 60% of the total direct eligible costs*.
- **Req2** - **If** the requested contribution exceeds 375K Euro, the participant **must** provide a copy of a "Certificates of Financial Statement" (CFS). This request can be ignored if the participant already acquired the "Certificate of Methodology" (CoM).
- **Req3** - Each participant **must** submit to the project coordinator the signed Form C together with a copy of the CFS **or** the CoM **when needed**.

With reference to the operational perspective some activities could be performed manually, others are completed using an IT system, and other could be completely automated. For instance, filling cost model, provide certificate, sign and submit form C are activities done manually even if they could be supported by IT systems. Activity such as calculate flat rate and filling the indirect cost are automatic activities.

Considering information perspective Data Objects variability can be also observed. It is the case that some Data Objects must, should, or should not be included (e.g. CFS or CoM could be needed or not). The following list introduces Data Objects relevant for periodic budget reporting BP.

- **Form C** is the main reporting document. It contains all the information related to the financial cost claim. The general structure of the data object is composed by two main parts: (i) the *Costs Form* containing all the costs that the participant has to declare with reference to direct (*Personnel Costs*, *Subcontracting Costs*, *Other Direct Costs*) and indirect cost (they are calculated considering the applicable flat rate of the participant and the direct costs); and (ii) the *Certificate Forms* containing the copy of the CFS or the CoM in case the financial contribution requested exceeds 375K Euro.
- **Time-sheets** can be used to calculate the personnel cost and they contain all the information related to activities done by the personnel.
- **Contracts** can be used to certify subcontracting costs.
- **Cost Doc** can be used to certify any other direct cost.
- **Original CFS** is the original version of the CFS.
- **Original CoM** is the original version of the CoM.

Considering the organization perspective several organizational models can be observed. Reporting can be managed for different projects by different offices in the same organization (in some case PAs introduce office dedicated to reporting, in other cases responsibilities are distributed among different offices), or it can be also possible that responsibilities could be delegated to an external organization.

V. Overview of the Approach

The proposed approach is organized in four main steps (Fig. 2). Regulatory frameworks as well as available reference manuals related to the provision of a service are the input of the proposed approach, while the output will be a BP variants that can be deployed according to the characteristics of the service under analysis.

- **BPFM model design** - The first step aims at defining a configurable BP model, this will be manually codified using the BPFM notation presented in Section VI-A. At this level a BPFM model directly permits to express variability aspects with respect to functional, information, (partially) behavioral and operational perspectives. A focus group or a competence center, including domain experts (eventually coming from more than one PA) and BP designers, performs the step for each service only once and for all the organizations aiming at using the approach.
- **Configuration** - The second step foresees the definition of a BPFM configuration according to a specific organization. The configuration permits to select which activities and Data Objects need to be included in the BP variant. Configuration respects the activities constraints specified in the BPFM model. This step is guided by include/exclude relationships and Data Object availability. This step is typically done manually involving BP designers working in the organization under consideration.
- **BP Fragment Derivation** - The third step takes in input the BPFM configuration, and then, through the

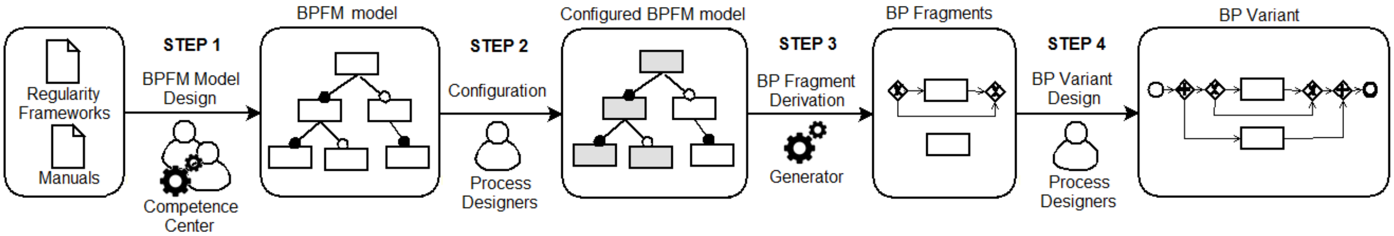

Fig. 2. Steps of the BPFM Approach.

application of the mapping rules (see Section VI-C) automatically derives BP fragments representing portions of the behavior that has to be completed to deliver the service under consideration.

- **BP Variant Design** - The last step concerns the derivation of the fully specified BP variant(s) starting from the generated BP fragments according to a more detailed view on the organization perspective. At this stage BP designers manually add, sequence flow, parallel execution constraints, events and organization details (swimlanes) permitting to derive a fully specified BP variant. It is worth mentioning that the same fragment (and activity) could be associated to different organizations and roles in different BP variants.

VI. From BPFM to BP Variant

A. Business Process Feature Model Notation

A BPFM model is constituted by a set of activities organized in a tree, where the root identifies the BP under analysis. Going from up to down on the tree BPFM introduces different levels of detail in the BP specification. In particular, each internal (non-leaf) activity denotes a sub-process, whereas the leaves represent atomic activities (task). In order to specify operational details, BPFM allows to type atomic activities using the same meaning and graphical representation given by BPMN 2.0 task types. In BPFM it is also possible to specify constraints between activities in two adjacent levels of the model. BPFM permits to model variability in terms of different activities available in a sub-process detailing an activity at an upper level, as well as some composition flow detail for the activities in the sub-process. Constraints are used to express (i) if child activities can or have to be selected in the configuration to be included in the BP variant; and (ii) if they can or have to be included in each execution path. In this way both the static and dynamic (execution-time) inclusion of activities can be specified. Constraints can be binary or multiple depending on the number of child activities connected to a parent activity. BPFM binary constraints are reported in the following.

- A *Mandatory Constraint* requires that the connected child activity must be inserted in each BP variant, and it has also to be included in each execution path (Fig. 3-A).
- A *Optional Constraint* specifies that the connected child activity can be inserted (or not) in each BP variant, and when included it is not necessary to include it in each execution path (Fig. 3-B).
- A *Domain Constraint* requires that the connected child activity must be inserted in each BP variant but it is not necessary to include it in each execution path (Fig. 3-C).
- A *Special Case Constraint* specifies that the connected child activity can be inserted (or not) in each BP variant. Nevertheless if selected it has to be included in each execution path (Fig. 3-D).

The following ones are the multiple BPFM constraints.

- An *Inclusive Constraint* requires that at least one of the connected child activities must be inserted in each BP variant, and at least one of them have to be included in each execution path (Fig. 3-E).
- A *One Optional Constraint* requires that exactly one of the connected child activities has to be inserted in each BP variant, and it is not necessary to include it in each execution path (Fig. 3-F).
- A *One Selection Constraint* specifies that exactly one of the connected child activities has to be inserted in each BP variant, moreover it has to be included in each execution path (Fig. 3-G).
- A *XOR Constraint* requires that all connected child activities must be inserted in each BP variant, and exactly one of them has to be included in each execution path (Fig. 3-H).
- A *XOR Selection Constraint* requires that at least one of the connected child activities has to be inserted in each BP variant, and exactly one of them has to be included in each execution path (Fig. 3-I).

Finally, as well as in "traditional" Feature Model, it is possible to specify *Include* and *Exclude* relationships between activities (Fig. 3-J and Fig. 3-K).

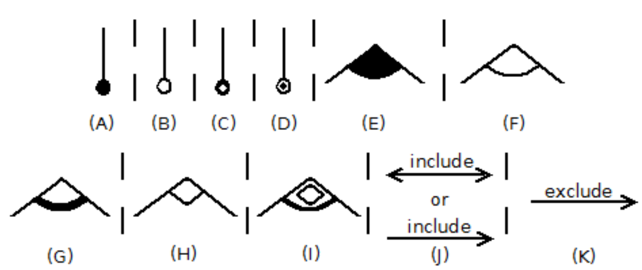

Fig. 3. BPFM Constraints.

BPFM manages all types of Data Objects introduced by BPMN 2.0 and it uses the same graphical representation [22]. As in BPMN 2.0 Data Objects can be connected as inputs and outputs to one or more activities. Successively to execute an activity all the Data Objects in input must be available, and as soon as all the Data Objects in output are generated than the activity can be terminated [23]. The BPFM notation gives also the possibility to model Data Objects since each BP variant could include completely different sets of Data Objects. A Data Object inherits the characteristics and constraints of the activities to which they are connected. For instance referring to the BPFM model in Fig. 4-A we can represent the following scenario.

- Data Input X and Data Store Y are optional in term of their presence in a variant, moreover when included in a variant it is possible that they will not be instantiated since they are input/output to the Activity 1 that is connected to the parent node with an *Optional Constraint* (i.e. the activity can be inserted (or not) in each BP variant and it is not necessary to include it in each execution path).
- Data Object Z is mandatory both in term of presence in a model variant and during the model execution since it is provided as input to Activity 2 that is connected to the parent node with a *Mandatory Constraint* (i.e. the activity must be inserted in each BP variant, and it has also to be included in each execution path).
- Data Object Collection A and Data Output B are mandatory in term of their presence in a variant, nevertheless they will not be necessarily instantiated during the execution since they related to Activity 3 that is connected to the parent node using a *Domain Constraint* (i.e. the activity must be inserted in each BP variant but it is not necessary to include it in each execution path).

BPFM permits also to include information concerning the state of a Data Object. Therefore an activity can specify a Data Object in a specific state. If the state is not explicitly reported the activity is state independent. A Data Object cannot be in two different states at the same time (Fig. 4-B). In case two different states of the same Data Object are indicated as input to the same activity the modeler will have to manually select just one of them.

BPFM supports the notion of composite and $Part - of$ Data Object that can be used for each type of BPMN 2.0 Data Object. The possibility of using this characterization for Data Objects is particularly useful to represent data in BP related to PA domain where documents are generally quite complex and can be decomposed in simpler parts.

- A composed Data Object includes a set of specific block of data, and it is marked with the letter *C*.
- A $Part - of$ Data Object is contained in a specific block of data, and it is marked with the letter *P*. It also explicitly refers to the Data Object of which it is part, reporting the name of it inside curly brackets.

Fig. 5 reports a simple example where a Data Object named *Composed DO* is composed by two parts *Part1* and *Part2*.

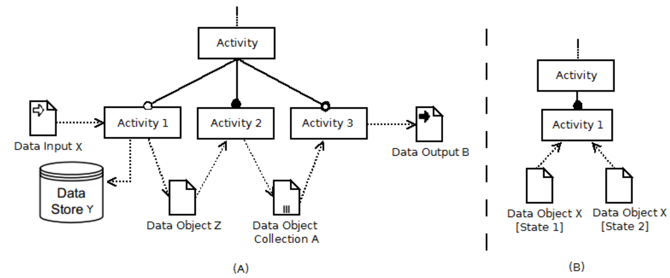

Fig. 4. Data Object in BPFM.

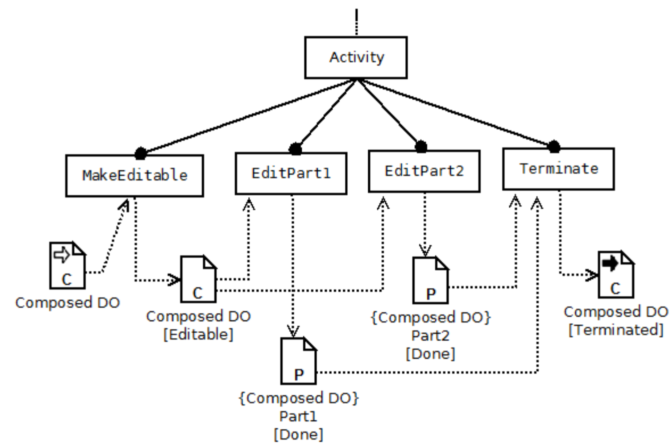

Fig. 5. Composed Data Object in BPFM.

Considering the editing of the two parts independent activities, even if it refers to the same Data Object, the notion of $Part - of$ gives the possibility to explicitly make such editing independent also from the data point of view.

The $Part - of$ notion can also be extended to the state of a Data Object. Nevertheless the state of a composed Data Object cannot be directly deduced from the state of the parts composing it. For instance Fig. 6 shows that the Data Object *Document* is composed by *Part 1* and *Part 2* that result as output of filling activities. As soon as the filling activities are fulfilled the state of *Document* is changed in *Filled*, then its state can change again in *Sent* without impacts on the states of *Part 1* and *Part 2*.

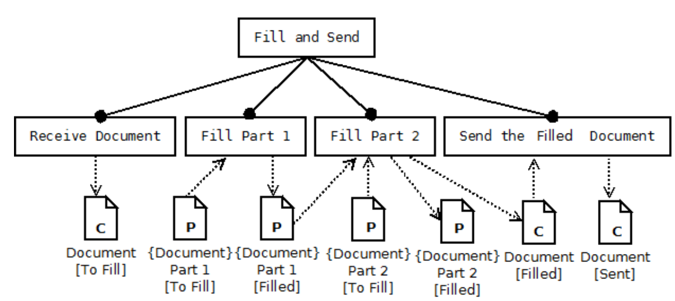

Fig. 6. Composed Data Object and States.

B. Modeling BPFM into Practice

In the following the description of the BPFM model resulting from the application of the notation is introduced. This is the output of the step 1 of the approach where details on functional, behavioral (partially), informational and operational perspective are introduced. The resulting BPFM model presents four levels. It includes 4 sub-processes and 12 atomic tasks. Many Data Objects are also represented. In the following description we refer to the BPFM model reported in Fig. 7. The first level activities are *Fill Form C*, *Provide Certificate*, *Sign Form C* and *Submit to Coordinator*.

Going into detail *Fill Form C* refers to the need to fill the cost form, it is an activity connected to the root via a *Mandatory Constraint* since it has to be inserted in each BP variant and it has to be always executed. It requires in input the Data Object *Costs Form* in the state *Empty* and it returns the same Data Object in the state *Filled* in output. The sub-process is composed of three sub-activities.

- *Fill Direct Cost Model* refers to the need to calculate and than fill the direct cost model. It is an activity connected to the parent via a *Mandatory Constraint*. It needs in input the Data Object *DCost* in the state *Empty* and it generates the same Data Object in the state *Filled*. It is composed by the sub-activities *Fill Personnel Cost*, *Fill Subcontracting* and *Fill Other Direct Cost*. They are related to the different parts of the direct cost model. *Fill Personnel Cost* is connected via a *Mandatory Constraint* since in many cases there are personnel costs to be reported, and *Fill Subcontracting* and *Fill Other Direct Cost* are connected via two *Domain Constraints* since they have to be included in the different variant but in some cases they are not reported.
- *Calculate Flat Rate* refers to the need to calculate the flat rate. It is an activity connected to the parent via a *Mandatory Constraint*. The activity requires in input the Data Object *DCost* in the state *Filled* and it generates the Data Object *Indirect Cost*. Indirect costs can be calculated considering four different flat rates depending on the specific organization. They are represented as sub-activities connected via a *One Selection Constraint* and they are *Calculate Actual Indirect Cost*, *Calculated Simplified Indirect Cost*, *Calculate Flat Rate 60%* and *Calculate Flat Rate 20%*.
- *Fill Indirect Cost Model* refers to the need to fill the indirect cost model. It is an activity connected to the parent via a *Mandatory Constraint*. The activity requires the Data Objects *Indirect Cost* and *ICost* in the state *Empty* in input and it generates the Data Object *ICost* in the state *Filled*.

Provide Certificate is an activity connected to the root via a *Special Case Constraint* since it has to be inserted only in some BP variants, nevertheless in case it is included it has to be executed. The activity requires the Data Object *Certificates Form* in the state *Empty* in input and it generates the same Data Object in the state *Filled*. In this activity the beneficiary has to provide its CFS, or its CoM. Two different sub-activities connected via a *One Selection Constraint* are modeled in order to provide the certificates (*Provide CFS* and *Provide CoM*).

Sign Form C is an activity connected to the root via a *Mandatory Constraint* since it has to be inserted in each BP variant and it has to be executed. In this activity the beneficiary has to fill the periodic report. The activity requires the Data Objects *Certificates Form* in the state *Filled* and *Costs Form* in input and it generates the Data Object *Form C* in the state *Signed*.

Submit to Coordinator is an activity connected to the root via a *Mandatory Constraint* since it has to be inserted in each BP variant and it has to be executed. In this activity the beneficiary has to submit the periodic report to the project coordinator. The activity requires the Data Objects *Form C* in the state *Signed* in input and it generates the Data Object *Form C* in the state *Submitted*.

C. Deriving BP Fragments and BP Variants

As soon as the BPFM model is designed it is possible to derive a configuration of the model in order to derive a BP variant (step 2 of the approach). The BPFM model is configured selecting the activities to include (represented in gray in the following figures). From the configuration a set of fragments is obtained (step 3 of the approach). The generation phase takes the root of the tree and maps it as a BP variant model, and then applies the defined rules from the first level to the leaves according to the configuration and the rules of the various constraints.

The mapping rules have been conceived considering that the connectors in a BPFM model are mainly related to the function perspective but nevertheless, as described above, they also implies simple behavioral constraints on the organization of the activities. Here we report the description of the rationale behind each mapping rule.

- A *Mandatory Constraint* rule asks for including the child activity in each execution path since it has to be always selected (Fig. 8-A).
- An *Optional Constraint* rule asks for a combination of an activity and gateway conditions when the child activity is selected (letter (i) Fig. 8-B), so that two execution paths of the fragment are possible, one including the activity and the other one not. When the activity is not selected it results with no mapping (letter (ii) Fig. 8-B).
- A *Domain Constraint* rule asks for a combination of the activity since it has to be always selected and gateway conditions, so that two execution paths of the fragment are possible, one including the child activity and the other one not (Fig. 8-C).
- A *Special Case Constraint* rule asks for including the child activity in the execution path of the fragment if selected (letter (i) Fig. 8-D). When the activity is not selected it results with no mapping (letter (ii) Fig. 8-D).
- A *Inclusive Constraint* rule asks for a combination of the selected child activities and inclusive gateway conditions with a default path, so that multiple paths in the fragment are supported. In case only one activity is selected it is mapped as an activity in the execution path of the fragment (letter (i) Fig. 10-E), otherwise if two (letter (ii) Fig. 10-E) or more activities are

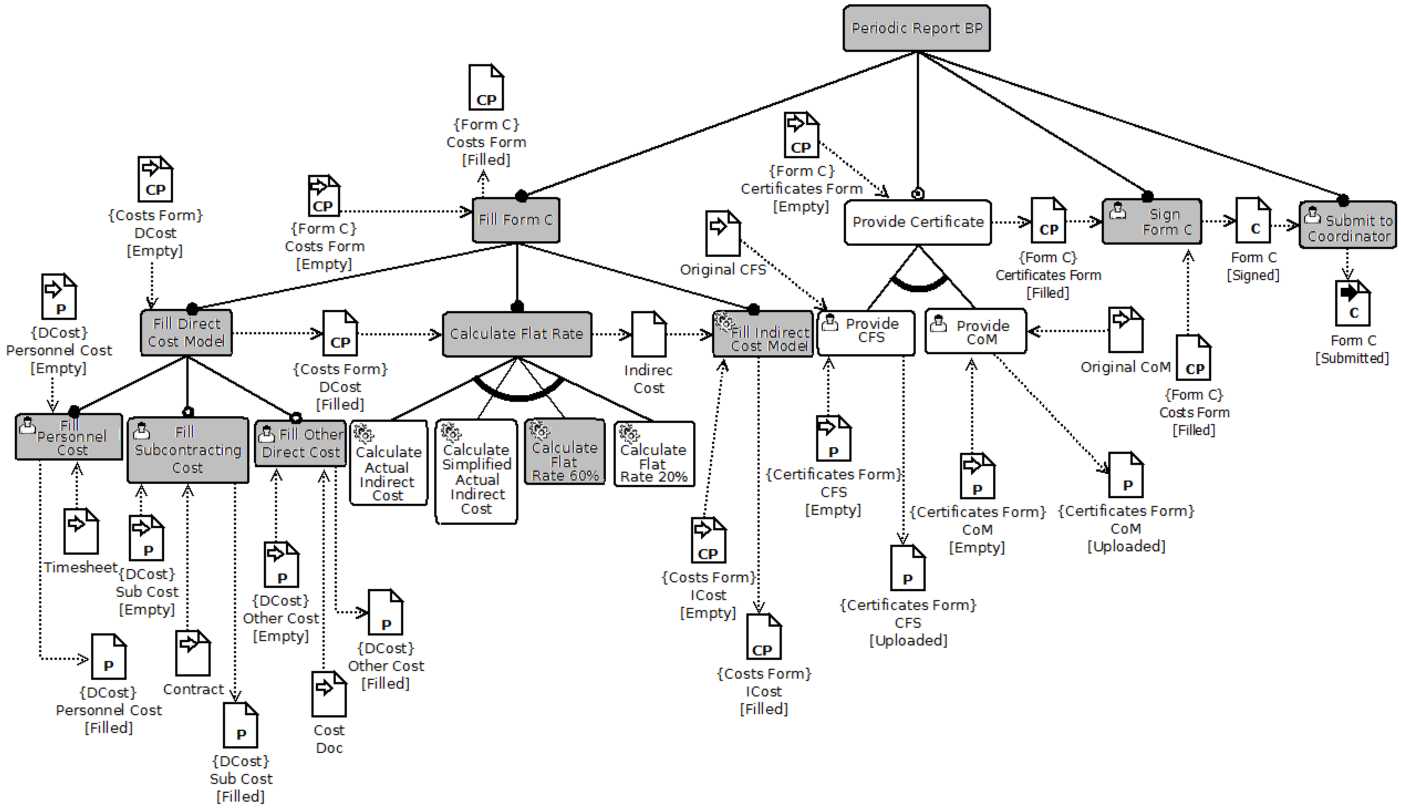

Fig. 7. EU Participant Periodic Report BPFM model - Configured.

selected all of them are included in the fragment. Finally, it could be possible that all the activities in the BPFM are selected (letter (iii) Fig. 10-E), then all of them are included in the fragment.

- A *One Optional Constraint* rule asks for a combination of an activity and gateway condition, when the child activity is selected (assuming that exactly one activity has to be selected) than two execution paths of the fragment are possible one including the activity and the other one not (Fig. 9-F).
- A *One Selection Constraint* rule asks for including the selected activity in the execution path of the fragment since at least one child activity has to be selected (Fig. 9-G).
- An *XOR Constraint* rule asks for a combination of the selected child activities and exclusive gateway conditions, so that alternative paths are supported in the execution path of the fragment (Fig. 9-H).
- An *XOR Selection Constraint* rule asks for a combination of the selected activities and exclusive gateway conditions, so that alternative paths are supported; in case only one child activity is selected rule asks for an activity in the execution path of the fragment (letter (i) Fig. 10-I), otherwise if two (letter (ii) Fig. 10-I) or more activities are selected all of them are included in the fragment. Finally, it could be possible that all the activities in the BPFM are selected (letter (iii) Fig. 10-I), then all of them are included in the fragment.

For include and exclude constraints no mapping rules are provided, since they only impact on the correctness of the configuration step. Moreover in case the activity is atomic it is mapped to a simple BPMN task. Otherwise, if the activity is a complex one, it is mapped as BPMN sub-process and its child activities are mapped inside the sub-process itself (Fig. 11).

Finally, Data Objects are mapped as input/output of the corresponding activities either in case they are atomic or composite. During the mapping states and data type information are preserved, and they can provide useful suggestions to complete the configuration as foreseen by the last step in the process. For instance Fig. 12 reports 4 mandatory activities leading to 4 separated fragments after the mapping. The variant designer will have then to consider the relations implicitly generated by the presence of the Data Objects, that for instance can be consumed only after they have been produced.

D. Deriving BP Fragments and BP Variants into Practice

In this section we report our experience in applying the proposed approach in the premises of the University of Camerino. The considered case study refer to the organization of the reporting activity with respect to EU projects. The application of the approach required to apply the different steps composing the approach to finally derive a fully defined BP.

- **Step 1 - BPFM Model Design.** This step refers to the case description already introduced in Section IV and results in the definition of the BPFM model introduced in Section VI-B.

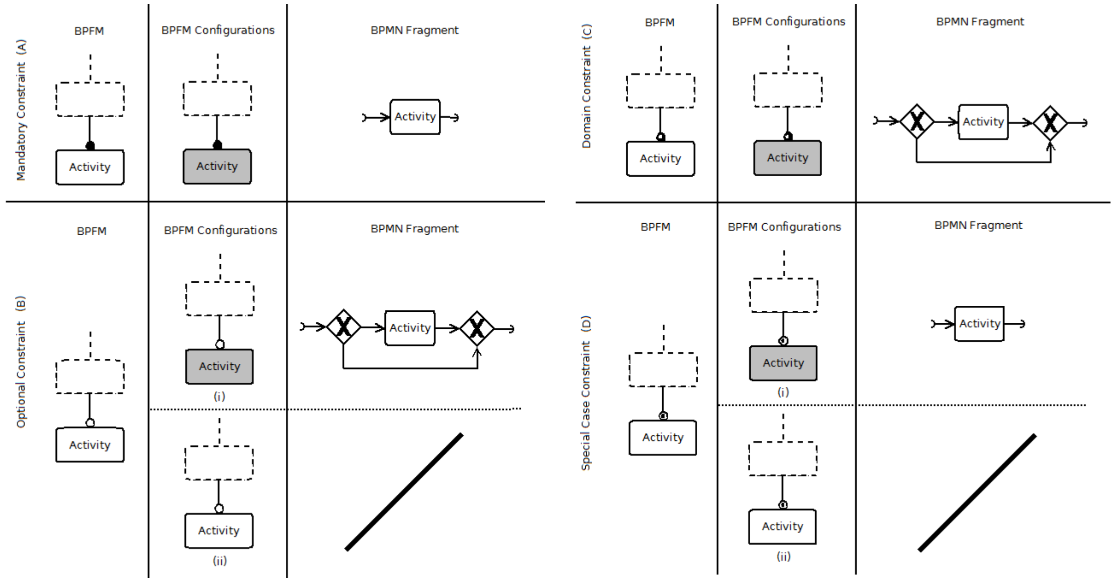

Fig. 8. Binary Constraints Mapping Rules.

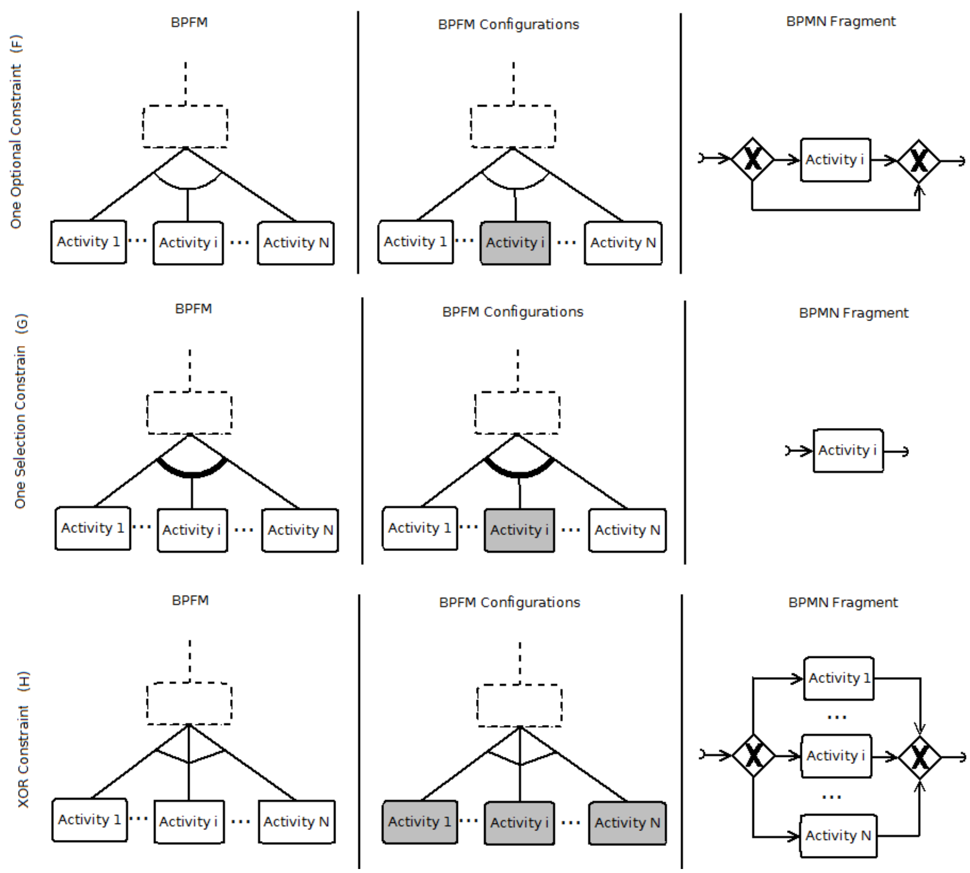

Fig. 9. Multiple Constraints Mapping Rules (Part I).

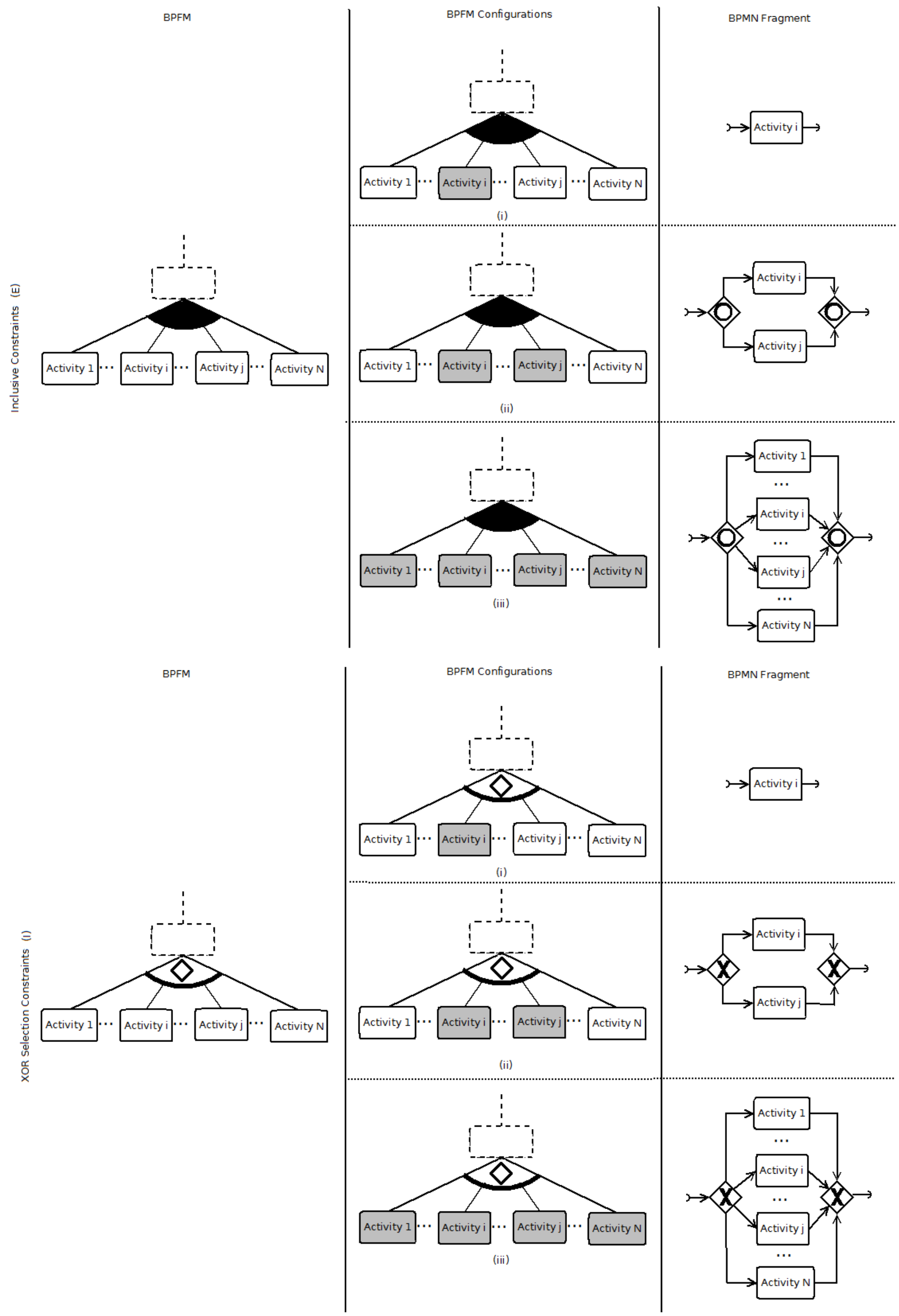

Fig. 10. Multiple Constraints Mapping Rules (Part II).

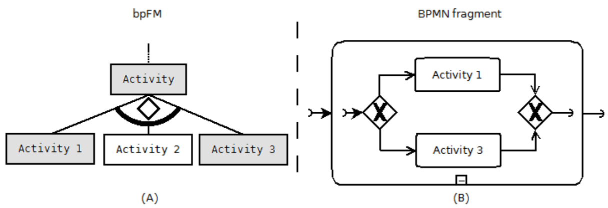

Fig. 11. Example of Composed Activity Mapping.

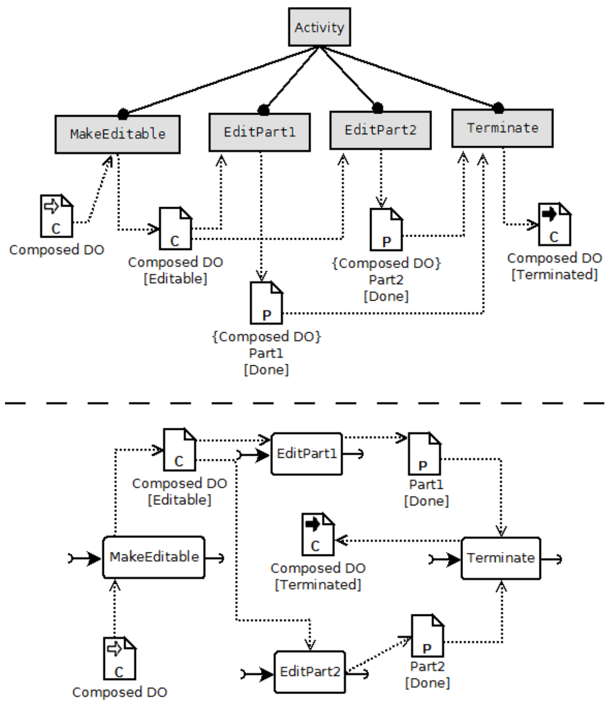

Fig. 12. Mapping Composed Data Object.

- **Step 2 - Configuration.** In this phase, starting from the given BPFM, the BP designer, that in our case was the head of the research and technology transfer Macro-Sector, has to define a novel configuration with reference to an EU project recently funded. He selected the activities and the Data Objects that are needed to define the reporting BP variant according to the specific characteristics of the internal organization of University of Camerino. The configuration includes all the mandatory activities as well as some optional activities taking into account that the needed flat rate is 60% and certificates are not needed, since the defined budget for the specific project is less than 375k euro (Fig. 7).

- **Step 3 - BP Fragments Derivation.** The selected configuration is automatically mapped into a set of BP fragments (Fig. VI-D-Left). The first level activity *Fill the Form C* is mapped as a sub-process using composed activity mapping rules, instead *Sign Form C* and *Submit to Coordinator* are mapped as tasks using the mandatory constraint rule. The *Fill the Form C* sub-process contains *Fill Indirect Cost Model* and *Fill Direct Cost Model* that are mapped as sub-processes using composed activity mapping rule and *Calculate Flat Rate* is mapped as tasks using mandatory constraint rule. *Fill Direct Cost Model* contains: *Fill Personnel Cost* that is mapped using mandatory constraint rule and *Fill Subcontracting Cost* and *Fill Other Direct Cost* that are mapped using domain constraint rule. Finally, *Calculate Flat Rate* sub-process contains the activity *Calculate Flat Rate 60%* that is mapped using one selection constraint mapping rule.

- **Step 4 - BP Variant Design.** At this stage the BP designer needs to further refine the process introducing missing sequence flows, parallel execution constraints, and events among the generated BP fragments, finally resulting in a fully specified BP variant. This step permits also to introduce more details in term of offices organization taking into account the possibility to allocate activities to different participants and roles. A possible variant generated to support reporting activities from the BPFM model and configuration is provided (Fig. VI-D-Right). The derived BP has three main phases all of them are done by the research and technology transfer Macro-Sector area of University of Camerino the first and the last referring to the filling form C and submitted form C are done by the Industrial and International Liaison Office, whereas the signing of the Form C is done by the head of the Industrial and International Liaison Office that is delegated by the Rector that is the university's legal representative.

In summary, as expected, the application of the approach permitted to reason on different aspects of the process at different times, and to reuse already taken modeling decisions to build a more specific BP model for a given organizational context.

VII. Conclusions and Future Work

A novel notation and approach have been defined in the paper. They are suitable to model variability of BP permitting to include in a single model many different variants of the same BP. The proposed approach has been conceived for situations in which activities composing the configurable BP model have to be successively refined to consider characteristics of the deployment context, such as the different arrangement of the organization supporting the BP itself. In such a case objectives and activities constituting the configurable BP model are general and independent from the specific characteristics of the organization delivering the service itself, nevertheless the precise definition of the BPs, in terms of availability, ordering of the activities and managed documents, depends from deployment related aspects and in particular relates to the organizational model. It is suitable in those context in which complex organizations, such as the PA, deliver the same service in many different contexts, and with procedures partially depending from the specific organization.

The experiments we carried on provided encouraging results and permitted to model quite easily complex scenarios, and to correctly derive the needed BP variants. Nevertheless the usability of the approach need to be further evaluated. In the next future we intend to add support for

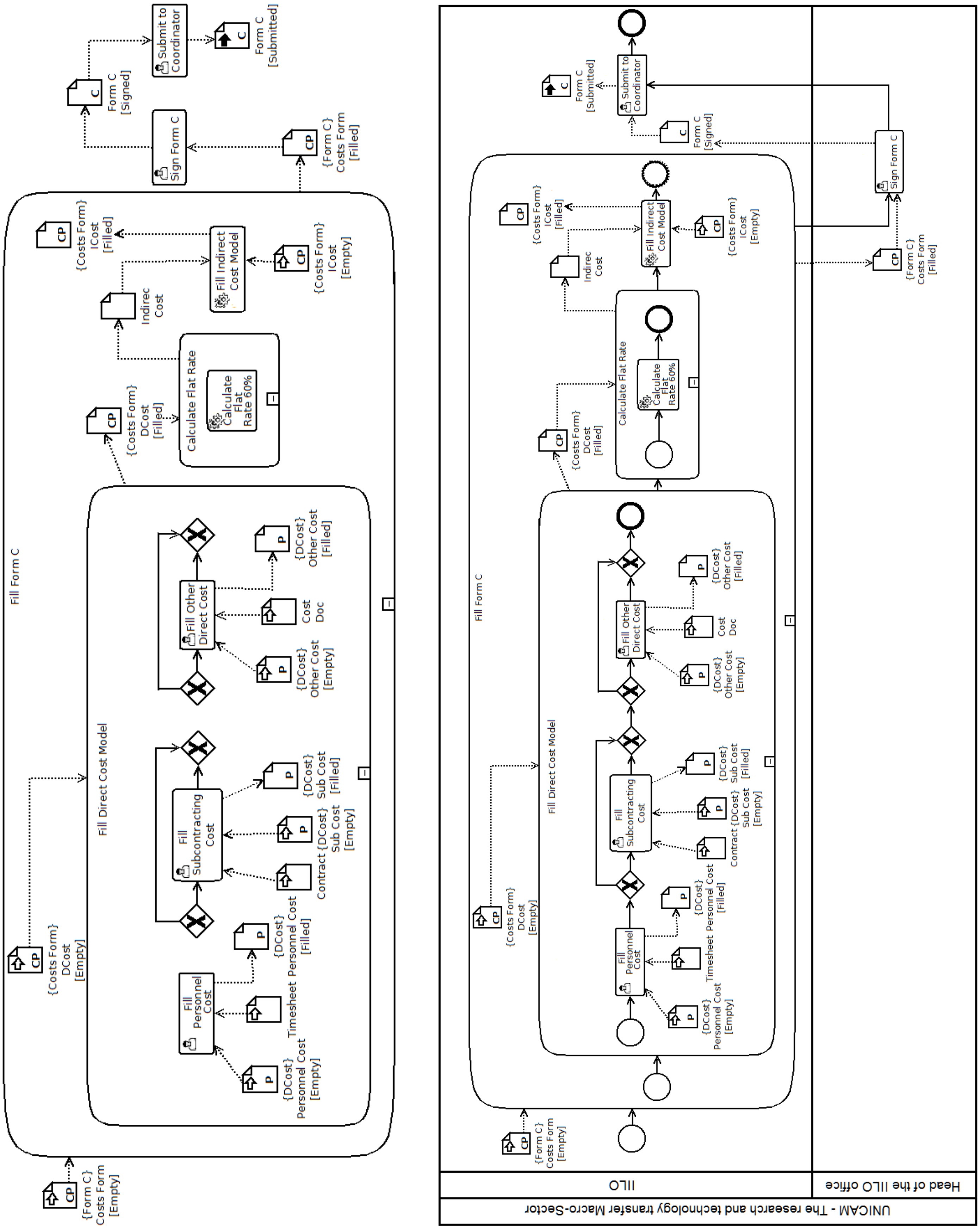

Fig. 13. Fragments generated (Left) and a possible BP variant (Right).

automatically checking the correctness of defined configurations, as well as the possible presence of errors in the modeling phase. In parallel further validation and modeling activities will continue in cooperation with Public Administrations in the context of the Learn PAd EU financed project (`http://www.learnpad.eu`).

ACKNOWLEDGEMENTS

This research has been partially founded by EU project LearnPAd (GA: 619583) and by the Project MIUR PRIN CINA (2010LHT4KM).

REFERENCES

[1] R. Cognini, F. Corradini, S. Gnesi, A. Polini, and B. Re, "Research challenges in business process adaptability," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. ACM, 2014, pp. 1049–1054.

[2] M. Reichert and B. Weber, *Enabling flexibility in process-aware information systems: challenges, methods, technologies*. Springer Science & Business Media, 2012.

[3] F. Gottschalk, W. M. van der Aalst, and M. H. Jansen-Vullers, "Configurable process models a foundational approach," in *Reference Modeling*. Springer, 2007, pp. 59 – 77.

[4] J. Recker, M. Rosemann, W. Van Der Aalst, M. Jansen-Vullers, and A. Dreiling, "Configurable reference modeling languages," *Reference modeling for business systems analysis*, pp. 22–46, 2006.

[5] E. Erkoçak and Ş. N. Açıkalın, "Complexity theory in public administration and metagovernance," in *Chaos, Complexity and Leadership 2013*. Springer, 2015, pp. 73–84.

[6] K. Pohl, G. Böckle, and F. J. v. d. Linden, *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer-Verlag, 2005.

[7] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-oriented domain analysis feasibility study," DTIC Document, Tech. Rep., 1990.

[8] R. Capilla, J. Bosch, and K. C. Kang, Eds., *Systems and Software Variability Management, Concepts, Tools and Experiences*. Springer, 2013. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36583-6

[9] A. Lindsay, D. Downs, and K. Lunn, "Business process – attempts to find a definition," *Information and Software Technology*, vol. 45, pp. 1015–1019, 2003.

[10] W. M. Van Der Aalst, "Workflow verification: Finding control-flow errors using petri-net-based techniques," in *Business Process Management*. Springer, 2000, pp. 161–183.

[11] C. Ayora, V. Torres, B. Weber, M. Reichert, and V. Pelechano, "Vivace: A framework for the systematic evaluation of variability support in process-aware information systems," *Information and Software Technology*, 2014.

[12] M. Rosemann and W. M. van der Aalst, "A configurable reference modelling language," *Information Systems*, vol. 32, no. 1, pp. 1–23, 2007.

[13] M. La Rosa, M. Dumas, A. H. ter Hofstede, J. Mendling, and F. Gottschalk, "Beyond control-flow: Extending business process configuration to roles and objects," in *Conceptual Modeling-ER 2008*. Springer, 2008, pp. 199–215.

[14] M. Döhring and B. Zimmermann, "vbpmn: event-aware workflow variants by weaving bpmn2 and business rules," in *Enterprise, Business-Process and Information Systems Modeling*. Springer, 2011, pp. 332–341.

[15] F. Gottschalk, W. M. Van Der Aalst, M. H. Jansen-Vullers, and M. La Rosa, "Configurable workflow models," *International Journal of Cooperative Information Systems*, vol. 17, no. 02, pp. 177–221, 2008.

[16] A. Hallerbach, T. Bauer, and M. Reichert, "Capturing variability in business process models: the provop approach," *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 22, no. 6-7, pp. 519–546, 2010.

[17] A. Schnieders and F. Puhlmann, "Variability mechanisms in e-business process families." in *BIS*, ser. LNI, W. Abramowicz and H. C. Mayr, Eds., vol. 85. GI, 2006, pp. 583–601.

[18] F. Gottschalk, T. A. Wagemakers, M. H. Jansen-Vullers, W. M. van der Aalst, and M. La Rosa, "Configurable process models: Experiences from a municipality case study," in *Advanced Information Systems Engineering*. Springer, 2009, pp. 486 – 500.

[19] C. M. Lönn, E. Uppström, P. Wohed, and G. Juell-Skielse, "Configurable process models for the swedish public sector," in *Advanced Information Systems Engineering*. Springer, 2012, pp. 190–205.

[20] OMG, *Case Management Model and Notation, Version 1.0*, Object Management Group Std., Rev. 1.0, May 2014. [Online]. Available: http://www.omg.org/spec/CMMN/1.0

[21] M. Pesic, H. Schonenberg, and W. M. van der Aalst, "Declare: Full support for loosely-structured processes," in *Enterprise Distributed Object Computing Conference, 2007. EDOC 2007. 11th IEEE International*. IEEE, 2007, pp. 287–287.

[22] R. Cognini, F. Corradini, A. Polini, and B. Re, "Using data-object flow relations to derive control flow variants in configurable business processes," in *Business Process Management Workshops - BPM 2014 International Workshops, Eindhoven, The Netherlands, September 7-8, 2014, Revised Papers*, 2014, pp. 210–221.

[23] A. Awad, G. Decker, and N. Lohmann, "Diagnosing and repairing data anomalies in process models," in *BP Management Workshops*. Springer, 2010, pp. 5–16.