Pre-Proceedings of 8th International Workshop on Engineering Multi-Agent Systems
(EMAS 2020 @ AAMAS), 8-9 May 2020, Auckland, New Zealand.
Eds. Cristina Baroglio, Jomi F. Hübner, and Michael Winikoff.

Goal-driven Adaptation of MOISE Organizations for Workflow Enactment

Massimo Cossentino, Salvatore Lopes and Luca Sabatucci

National Research Council of Italy (CNR)
Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
Via U. La Malfa, 153, Palermo (Italy)
{[name].[surname]}@icar.cnr.it
http://www.ecos.pa.icar.cnr.it

Abstract. The enactment of dynamic workflows may take advantage of the multi-agent system paradigm. The approach presented in this paper allows exploiting a high-level BPMN definition to generate several agent organisations that can enact the BPMN workflow. The availability of different solutions (i.e. organisations) enables the optimisation and adaptation features of the approach. The mapping of the initial workflow to organisations starts with the automatic generation of goals from the BPMN, and it exploits a metamodeling approach to generate MOISE organisation definitions.

1 Introduction

In the last two decades, web services enabled the application of business process working over the Internet [25]. Recently, business enterprises yield to redesign their information and process management systems to implement advanced features such as the adaptation ability [16, 10]. Business processes are defined as static, but, their enactment in a real environment often faces events and exceptions. Some of these may be anticipated, thus to be caught as special events leading to ad hoc deviations. Other ones cannot be planned: external events, failed tasks, incomplete or erroneous inputs and outputs. These produce situations in which a mismatch exists between the real processes and the software counterpart [20]. A taxonomy of motivations for facing dynamic workflows is provided in [11].

However, the objective to increase agility and flexibility of business processes is in conflict with the current trend of over-specifying workflow details. Moreover, the activity-diagram style makes hard to re-arrange the order of service. Some improvements consist in introducing decision/adaptation points [27, 14] inside the workflow model, where decisions are made by considering contextual data and pre-defined rules [13, 17]. Since these approaches require additional analysis skills, other solutions are to adopt smart workflow engines, like ADEPT [20], that can handle inserting, deleting and moving tasks at run-time.

Another direction is to use multi-agent systems. This is an alliance, very frequent in literature [4, 26, 5], in which the enactment of workflows takes great benefits from distinctive agent features like distribution, adaptation and smartness. Most of the agent-based approaches use 1) services and semantic web description (i.e. DAML-S [7]) as external behaviours for proactive agents, and 2) social abilities to coordinate the flow of tasks. It remains the problem of coordinating heterogeneous, autonomous agents, whose internal designs could not be fully known a-priori. To this aim, recently JaCaMo [2] emerged as in interesting solution for facing various aspects of agent programming. Briefly, JaCaMo represents the integration of three multi-agent programming dimensions (agent, environment, and organization levels) and provides Jason [3] for developing BDI agents [19], Cartago for defining artifacts [21], and MOISE [12] for specifying agent organizations.

Based on some recent results [24], this paper presents a novel approach for creating an agent-based adaptive workflow that can enact the description obtained from a high-level business process.

The proposed approach is based on three pillars:

1. The use of BPMN [6] for specifying the workflow; BPMN is a high-level language that focuses on analysis activity, differently from other execution languages such as BPEL4WS [1] (Business Process Execution Language for Web Services) that proposes a design-time bounding between tasks and services.
2. Goal-Driven enactment. A BPMN may be automatically translated into goals [24]. The advantage is that goals allow for breaking the rules: whereas sequence flows specify the precise order in which services are invoked, goals can relax strict constraints, widening the space for adaptation [23].
3. Workflow as a cooperative problem-solving. Multi-agent systems and services are a synergic alliance for achieving the generated goals.

The novelty of the paper is to link the goal generation of [24] to an automatic definition of social organisations for agents. We show that given a set of goals and a set of available services, it is possible to generate one or more social organisation for enacting the workflow. The number of organisations will depend on the availability of services and different plans that could be composed with them to pursue goals extracted from the workflow. The availability of different organisations will enable adaptation strategies for continuing goal pursuit in case of failure. To demonstrate this claim, we selected MOISE [12], an organisational model based on 1) a structural perspective (roles, groups and links), 2) a functional perspective (goals, plans and actions), and 3) a normative perspective (missions and norms). Therefore we map some of the concepts from [24] and from [23] into the MOISE metamodel, thus enabling the automatic generation of MOISE organisations.

The paper is structured as follows: Section 2 presents the general idea and a running example for presenting some preliminary result. Section 3 briefly reports the theoretical background at the basis of this work. The main contribution is discussed in Section 4, Section 5 provides an example, whereas Section 6 proposes some discussion. Finally, conclusions are sketched in Section 7.

2 Motivation and Proposed Approach

This section provides the motivation for converting BPMN into MOISE organizations. Indeed, it collocates the main contribution of the paper into a complete framework for enacting dynamic workflows (depicted in Figure 1). The whole framework is out of the scope of the paper, however, it is briefly described for allowing the reader to have an intuition of our idea of goal-directed agent-based adaptive workflows.

2.1 From Abstract Definition to Self-Adaptation

Figure 1 sets the use of BPMN (or similar notations) for specifying the standard flow of tasks shifting the complexity of error/exception management at run-time where self-adaptation techniques may be applied [23].

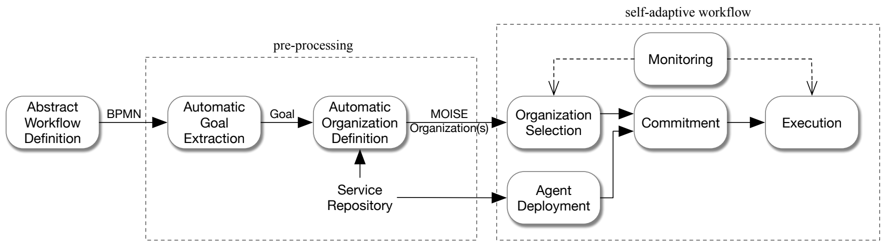

Fig. 1. Overview of the proposed approach

The proposed approach introduces a *pre-processing* phase between the workflow definition phase and its enactment, and two specific tools support this phase. The first one exploits the results from [24], for automatically generating BPMN workflow goals. Here, the novel contribution is the Automatic Organization Definition, which elaborates goals and available services for producing one or more executable MOISE specification(s) (intended as a replacement of languages such as BPEL4WS [1] i.e. Business Process Execution Language for Web Services).

The *self-adaptive workflow* compartment of Figure 1 represents the multi-agent system responsible for forming the organisation (by making each agent to play a role in it), commit to every single goal, and execute plans in order to enact the whole workflow. An internal monitoring activity must ensure estimating possible goal violations [9], raising adaptation signals when necessary.

The main advantage of this solution relies on the ***ability of self-adaptivity*** that agents may exhibit when something in the process goes wrong. When it is necessary, agents forming the MOISE organization may decide to dismiss the group and to generate a new one, even including other agents. Self-adaptation is based on the *online switch of agent organisations*: the Organization Selection phase evaluates all the available MOISE organisations (previously generated and sorted according to some metric), and it is responsible for selecting an alternative organisation.

So far, the *self-adaptive workflow* compartment is a work in progress, so its discussion is left as future work.

2.2 A Motivating Example

This paper uses a simplified version of the email-voting process, initially available in [18]. Figure 2 describes the process for mediating and coordinating voting members in resolving issues. Briefly, a manager prepares the issue list for a discussion. Participants will propose solutions via email. After one week, the discussion is closed with a voting session. Finally, the manager prepares the results and communicates the results. If the issue has not been resolved, then a new discussion cycle starts.

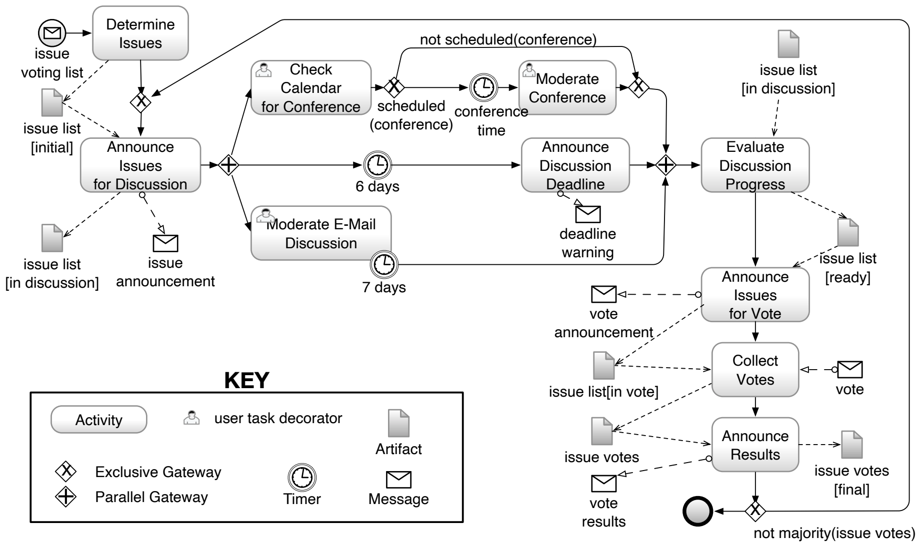

Fig. 2. The voting-by-email process, ri-elaborated from [18]

The execution of this business process involves several automatic and manual tasks. Automatic tasks are performed through web service invocation, whereas a human stakeholder performs manual tasks with the support of computer applications. An example of service task is the *Check Calendar for Conference*, whereas *Moderate Conference* is a manual task. The primary way to implement the communication between the issue manager and the voting members is via email through the *send_email_notification* external service.

The process has many failure points. A service could be temporarily unavailable, or it may produce wrong results. Also, the process may fail because a human fails in some manual activity. To cope with situations like these, changing workflow behaviour at run-time is an important requirement. The use of a dynamic workflow management system allows run-time modification of the original business process. Some examples of adaptation may occur:

1. when *Check Calendar for Conference* is unavailable, an alternative *Ask Data for Conference* could be used in place;
2. when a voting member misses reading the inbox, the system may use smartphone alerts to send her a reminder;
3. when the voting member inbox does not accept messages with big attachments, *send_email* may be replaced by using cloud file upload and sharing.

3 Theoretical Background

The work in this paper exploits the main result of [24] in which a BPMN is automatically processed for extracting procedural goals.

This section provides an overview of the theoretical background from [24] to enable the reader understanding the following phases.

It is worth noting the goals we aim at are ***implicit goals***, in the form of $trigger\_condition \rightarrow final\_state$ (or briefly $tc \rightarrow fs$), mainly describing functional aspects that are embedded in a traditional workflow definition. They represent the core of the procedure for automatically generating MOISE agent organisation for enacting an adaptive workflow.

The first step is the identification of the relevant states of a workflow when it is provided using the Business Process Model and Notation (BPMN) [6, 18], a very expressive graphical notation for representing business processes of diverse natures. BPMN is the de-facto standard choice to express a representation of a business process. It is basically an analysis instrument that allows discovering inefficiencies, inconsistencies, deadlocks, non-terminating conditions, and so forth [6]. The notation includes activities, gateways, events, data objects, and various kinds of relationships: the approach focuses on sequence and message flows.

- $available(\langle Data \rangle)$ describes the availability of a given input/output artifact;
- $received(\langle Message \rangle, \langle Actor \rangle)$ when a message incomes from a participant;
- $done(\langle Activity \rangle)$ when an activity has been completed with success.

The second step is to consider BPMN elements as state changes. Every single element (activities, events, and gateways) is something that, individually, contributes to the workflow state evolution. It is described by:

- a Waited State $ws(e)$, that is the state of the world soon before the element is executed;
- a Generated State $gs(e)$, that is the state of the world soon after the element execution.

WS and GS will consider all possible Input and Output Data Objects, Incoming and Outgoing Messages and Boundary Conditions. For instance, for an activity element:

$$\begin{cases} ws(e) = data\_in(e) \wedge mess\_in(e) \\ gs(e) = (data\_out(e) \wedge mess\_out(e)) \oplus \langle boundary \rangle\_termination(e) \end{cases} \quad (1)$$

Conversely, gateways are used to control how Sequence Flows interact as they converge and diverge within a process. The BPMN specification proposes several categories of gateways, but for what concerns the internal factors, they do not directly affect the state of the world. As a convention, for each kind of gateway $ws(e) = gs(e) =< unspecified >$.

The third step is to check the compatibility conditions

To this aim, the other two factors, Predecessors/Successors Influence studies the mutual incidence of predecessor and successor elements:

- the Predecessors Influence $pre\_inf(e)$ is the condition that must be true to make the element e compatible with its predecessor elements in the workflow;
- The Successors Influence $succ\_inf(e)$, is the condition that must be true to make the element e compatible with its successor elements in the workflow.

This analysis consists of observing each of the predecessor elements $Pre(e)$ (or successor elements $Succ(e)$), and evaluating their contribution propagates towards one of e input (output) ports. The analysis is particularly interesting for gateways (that are state-propagating element), where:

$$pre\_inf(e) = \underline{\bigvee}_{\forall k \in Pre(e)} (propag(k) \wedge flow(k, e)) \quad (2)$$

i.e. the exclusive disjunction of two components: 1) $flow(k, e)$ is the optional condition attached to the sequence flow from k to e and 2) $propag(k)$ is recursively defined as:

$$propag(k) = \begin{cases} gs(k) & \text{activity, event} \\ \underline{\bigvee}_{\forall r \in Pre(k)} (propag(r) \wedge flow(r, k)) & \text{converging exclusive gateway} \\ \bigvee_{\forall r \in Pre(k)} (propag(r) \wedge flow(r, k)) & \text{converging inclusive gateway} \\ \bigwedge_{\forall r \in Pre(k)} (propag(r) \wedge flow(r, k)) & \text{converging parallel gateway} \\ propag(pre) \wedge flow(pre, k) & \text{diverging gateway} \end{cases} \quad (3)$$

The last step is to calculate the workflow goals by solving the following boolean expressions

$$\begin{cases} \text{if } \exists M_{tc} \text{ such that } M_{tc} \models tc & \Rightarrow (M_{tc} \models ws) \text{ and } (M_{tc} \models pre\_inf) \\ \text{if } \exists M_{fs} \text{ such that } M_{fs} \models fs & \Rightarrow (M_{fs} \models gs) \text{ and } (M_{fs} \models succ\_inf) \end{cases} \quad (4)$$

where M_{tc} and M_{fs} are possible interpretations of the model.

The BPMN2Goals tool, available as a web service[1]:

- accepts BPMN files in the XMI format, according to the OMG specification and it generates a preliminary graph where nodes represent generic BPMN elements, i.e., tasks, events, and gateways.
- calculates waited state, generated state, successor influence, and predecessor influence for each node;
- solves Equations 4 by applying the simplest model that satisfies it: $tc = ws \wedge pre\_inf$ and $fs = gs \wedge succ\_inf$.

Some of the goals generated for the email-voting example are the following:

```
GOAL: Determine Issues
WHEN: received(issue_vote_list) THEN: initial(issue_list)

GOAL: Announce Issues for Discussion
WHEN: (((final(issue_votes) and sent(vote_results)) and
not agreement(issue_votes)) xor initial(issue_list))
THEN: in_discussion(issue_list) and sent(issue_announcement)

GOAL: Check Calendar for Conference
WHEN: in_discussion(issue_list) and sent(issue_announcement)
THEN: done(check_calendar_for_conference) and (scheduled(conference)
xor (not scheduled(conference) and in_discussion(issue_list)))

GOAL: Moderate Conference
WHEN: at(conference_time)
THEN: done(moderate_conference) and in_discussion(issue_list)
...
```

In the next section, two metamodels will be introduced to discuss how BPMN goals are mapped to MOISE organisations.

4 Mapping BPMN Goals to Organizations: Metamodels

In order to generate organizations for the adaptative execution of worklows, we need to map some concepts of the metamodel underlying the approach as proposed in [24] to the corresponding MOISE concepts. The metamodel implicitly adopted in [24] is represented in Fig. 3, the composing elements are:

- BPMN Workflow: this is the BPMN representation of the input workflow. Generally speaking this is the result of the work of a business analyst who draws a solution to the problem. This is composed by activities and flow relationships (used to organize the control flow of activities). For the purpose of this work we are now omitting some elements of a classical BPMN workflow (sequence flow relationships, data objects, events, messages, . . .).

[1] The web service is available, with a front-page, at `http://aose.pa.icar.cnr.it:8080/BPMN2Goal/`

- BPMN Activity: an activity can be a task or a sub-process. A task is an atomic activity (a portion of work to be done). A sub-process is a compound defined by a sub-flow of activities. We are currently omitting sub-processes in the proposed approach although their introduction would not affect that.
- Business Goal: a goal is a pair <tc,fs>where tc and fs are logical conditions combined by classical logic operators (AND, OR, NOT). The trigger condition (tc) describes when the goal may be pursued and the final state (fs) describes the desired state of the world. Roughly speaking we could say that goals are extracted from activities therefore a 'Goal Extraction' procedural relationship is drawn from the BPMN Activity element to the Goal element.
- Capability: it represents the concretization by an agent of one of the n possible strategies for fulfilling a goal. Frequently used as a wrapper to services.
- Concrete Plan: it is a flow of capabilities that can pursue a goal extracted from BPMN with the approach proposed in [24]. The plan may be produced by the Proactive Means End Reasoning algorithm [22] that composes a set of capabilities for satisfying the goal. A plan is composed of capabilities and relationships among them. Given a goal, many different plans may be found to satisfy that, if enough capabilities are available in the Yellow Pages repository. Each plan will achieve different performances because of the various behaviours of employed capabilities.
- Yellow Pages: a directory of the capabilities that can be provided by the agents populating the system. It is used to compose plans.

It is significant to note that we can identify three main zones in this metamodel: first the input part (on the left in Fig. 3). It consists of BPMN elements (BPMN workflow activity and others omitted for conciseness in this paper). This part of the model represents the workflow processed to obtain the central part of Fig. 3: Goal. Goal is the key abstraction of the approach. Goals are obtained by processing BPMN elements, mainly activities, as reported in Sect. 3. Finally, the right most part of Fig. 3 describes the solutions computed to pursue goals extracted from BPMN. Each solution consists of a Concrete Workflow that is in turn composed by Plans. Each Plan pursues one of the goals. Considering that the same goal may be fulfilled by composing different set of capabilities (and therefore by several different plans), it is possible to create several Concrete Workflows that can pursue the set of Goals extracted from the same BPMN workflow.

As discussed in sect. 2, suitable agent organizations are to be defined in order to enact the concrete workflows deduced from goals extracted by the input BPMN workflow. This is done by defining a corresponding number of MOISE organizations. For discussing that in details, we will now refer to the metamodel of the MOISE organizational framework reported in Fig. 4.

MOISE models organizations from three different perspectives: structural, functional, and normative[12]. The metamodel proposed in Fig. 4 only reports MOISE elements that are significant for the proposed approach. Indeed it also

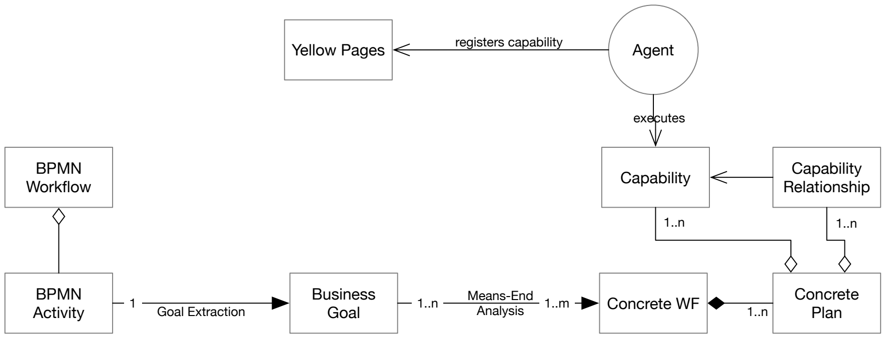

Fig. 3. The metamodel implicitly adopted in extracting goals from BPMN workflows

includes two elements (Collective Goal, Individual Goal) we introduced to better explicate the proposed approach; such elements are coded as MOISE goals.

Below a description of the MOISE metamodel elements based on definitions proposed by Hubner in [12]:

- Organization: A MOISE organization is a specialized Group that is devoted to pursue some goal. An organization is a collection of roles. Indeed, defining an organization implies the definition of its structural, functional and normative perspectives.
- Group: a group is composed by roles. The number of agents that can play a role is constrained by a minimum and maximum number. Links among roles in the group specify the type of relationship (Link element in the metamodel) between agents playing two roles. A set of compatibilities between roles may be specified as well.
- Goal: MOISE goals may belong to two different types: *achieve* and *maintain*. A plan is specified to pursue the goal and it may describe the decomposition of the goal into sub-goals by means of Plan operators (sequence, parallel, choice). Goal's specifications also include the time to fulfil (ttf) and the cardinality (how many agents have to achieve the goal to consider that as globally satisfied). For facilitating our mapping intents, we decided to specialize the goal concept into *Collective Goal* and *Individual Goal*. Indeed this is inspired by the MOISE framework itself that addresses the definition of a Scheme as a specification at the collective level and the Mission at individual level (see MOISE tutorial [15]).
- Collective Goal: It is a goal at the intermediate or root level in the goal decomposition tree. It is used as a label for intermediate parts of the scheme in the organization functional description.
- Individual Goal: It is a leaf goal of the goal decomposition tree. It is the objective of a MOISE mission and therefore some agent will pursue that by using some of its capabilities.

- Plan: a plan is composed of (sub-)goals related by operators (sequence, choice, parallel). It is used to create a sort of goal tree for decomposing the root goal (that is a kind of collective goal) into finer grained goals.
- Scheme: a Scheme is the collective specification of the goals to be pursued by an organization. It also includes Mission specifications.
- Mission: the specification at the individual level of the functional perspective on organizations. It lists the goals to be pursued by the agent(s) committing to the specific mission.
- Link: the type of relationship between roles. Allowed types are: acquaintance, authority, communication.
- Compatibility: the specification of the compatibility in playing one role and another by the same agent.
- Norms: there are two types of norms: obligation and permission. While the first obliges the agent playing a role to perform some mission, the latter allows for that but it is not compulsory.

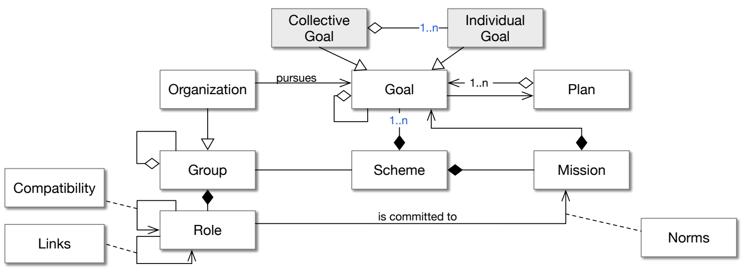

Fig. 4. The portion of MOISE metamodel adopted in the proposed approach for workflow enactment

4.1 Mapping Metamodel Elements for Workflow Enactment

In order to define the organizations that could enact concrete workflows while ensuring some degree of adaptation as discussed in sect. 2, we need to instantiate the elements of the MOISE metamodel reported in Fig. 4 starting from an instance of the metamodel reported in Fig. 3.

We base our organization definition on the mapping of elements reported in Table 1.

Each BPMN workflow generates several concrete workflows. Each concrete workflow is, in turn, mapped to one agent organization that will execute that. The availability of several concrete workflows (and corresponding organizations) for the same BPMN workflow allows for the choice of the one offering the best performance in terms of non functional requirements (usually evaluated by means of some quality function). In case of fail in the execution of the best concrete

Proposed Framework	**MOISE Metamodel**	**card.**	**Heuristics**
Concrete WF	Organization	1:1	N concrete workflows are built from each BPMN workflow. One organization is defined per each Concrete WF. The one with the best quality function is preferred. Others are invoked in case of failure (adaptation feature).
	Scheme	1	One for each organisation, it is a functional specification, and it lists goals, plans and missions.
	Group	1	One for each organisation, it is a structural specification. It is composed by a set of roles, links, compatibility constraints
BPMN Goal	(Collective) Goal	1:1	Name of (Collective) Goals is: [Goal]+"_goal" where [Goal] is the name of the Goal extracted from BPMN
Concrete Plan		1:1	Each Plan is devoted to fulfil one (Collective) Goal.
Capability	(Individual) Goal	1:1	Goals may have parameters. One (individual) goal per each capability. Name of (Individual) Goal is: [capability]
	Role	1:1	Role name is [capability]+"_role"
	Mission	1:1	One mission per each individual goal Mission name is: [capability]+ "_mission"
	Norm	1:1	One mission per each individual goal Mission name is: [capability]+ "_mission"
	Compatibility	1:1	Compatibility among all roles is supposed to be in place (if no information against that is deduced from application domain)
Capability Relationship	Plan	1:1	Moise Plans are generated by collecting Concrete Capability Relationships (+ ∥ ×) belonging to the same (Collective) Goal
	Link	1:n	Links are used to represent communication needs between two roles implementing concrete capabilities that need to exchange data

Table 1. Mapping the elements of the BPMN workflow into the MOISE metamodel.

workflow (organization), others may be instantiated in order to achieve the objective thus exploiting a adaptation feature.

Each organization is composed by one group and one scheme[2]. The first provides the structural specification of the organization (the list of roles, links and compatibility constraints among them). The latter provides the functional specification (goals, plans and missions). The list of roles is composed by looking at the capabilities used in the concrete workflow. One role is defined per each capability.

The objective of the organization is to fulfil the goal tree, in order to simplify the automatic construction of the organization specification, we suppose the tree composed by two different kind of goals (see Fig. 4): collective goals are intermediate nodes of the tree, they are decomposed into atomic level goals labelled individual goals. These are the leaves of the tree and may be fulfilled by one or more of the capabilities listed in the yellow pages. This means some agent exist in the system that could play the corresponding role and satisfy the individual goal.

[2] It is worth noting that the choice to create one schema for one workflow is motivated by the lack of information for separating goals in several schemas. We are considering to use this pattern when we will consider su-processes, however, to date, this is a ongoing work.

Capability relationships extracted from the concrete workflow allow for the definitions of MOISE plans. In fact such relationships emerge from the way each single BPMN goal is realised by using results from the Proactive Means-End Reasoning (PMR) algorithm [22]. Plans may decompose a collective goal in one or more other goals (collective or individual) that are connected by means of the three MOISE operators: sequence, choice, parallel. The need to exchange data between capabilities is converted in a communication link between the roles executing those capabilities. One mission is defined per each capability and a norm is introduced to oblige the role with the responsibility on executing that capability to perform the corresponding mission.

In other words, suppose goal *G1* is extracted from BPMN. This goal will be included in the solution organization as goal *G1_goal*. Now suppose the PMR algorithm combines two capabilities *C1* and *C2* in a sequence to purse goal G1. This means goal G1 is decomposed in two (individual) sub-goals *C1_goal* and *C2_goal*. These individual goals are combined in a plan using the MOISE *sequence* operator.

The organization will include roles *C1_role* and *C2_role*, each one responsible for the executing the corresponding capability (C1, C2). A mission will be created for each individual goal, the missions will be: *C1_mission* and *C2_mission*. One norm will be defined for each role:

```
<norm id="r1" type="obligation" role="C1_role" mission="C1_mission"/>
<norm id="r2" type="obligation" role="C2_role" mission="C2_mission"/>
```

Finally, supposing capability C1 provides input data to capability C2, a communication link is defined between the corresponding roles:

```
<link from="C1_role" type="communication" to="C2_role" scope="intra-group"/>
```

5 Case Study

This section reports an example of an application of the proposed approach to the case study introduced in subsection 2.2.

Suppose the list of goals (partially) reported in Section 3 has been automatically generated through the BPMN2Goals tool [24]. Moreover, suppose Figure 5 reports the list of services that have been registered into the Yellow Pages, and the results of the proactive means-end reasoning algorithm [23], i.e. how services could be combined for addressing the goals.

Many different situations may occur in employing services for achieving extracted goals. For instance, some services may be used to replace some others. For example, this happens for services s11 and s14, which address the same Moderate Discussion goal. Conversely, other services need to be composed in order to address a goal, and this happens for the sequence s1 + s4 that addresses the Determine Issues goal. Moreover, one service may address a combination of goals, as for s2 that addresses the two goals Determine Issues and Announce Issues for Discussion at the same time.

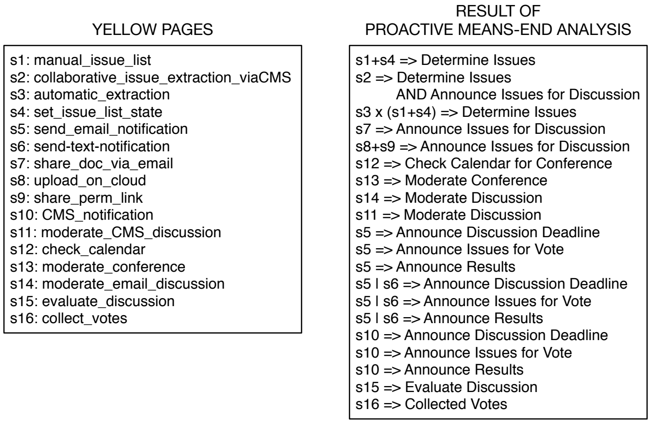

Fig. 5. On the left: the list of available services as registered in the system's Yellow Pages. On the right: the results of the PMR algorithm in terms of services and addressed goals

A concrete workflow is obtained by selecting enough items from this table to be sure that all goals are pursued by some plan (a service or a combination of several services). Therefore, by making different choices, it is possible to generate different concrete workflows and, consequently, different agent organisations may be defined to execute them.

As already described in Section 4, given a concrete workflow, the result of the mapping activity results in a MOISE organisation specification. Below an example of an organisation for the presented case study, will be detailed according to its three parts: 1) Structural Specification, 2) Functional Specification and 3) Normative Specification.

The Structural Specification describes roles and groups, where roles are derived from the services that realise the concrete workflow.

```
<role-definitions>
<role id="voting_role" />
<role id="manual_issue_list_role"><extends role="voting_role"/></role>
<role id="set_issue_list_state_role"><extends role="voting_role"/></role>
<role id="share_doc_via_email_role"><extends role="voting_role"/></role>
...
</role-definitions>

<group-specification id="voting_process_group">
<roles>
<role id="voting_manager_role" min="1" max="1"/>
<role id="manual_issue_list_role" min="1" max="1"/>
...
</roles>

<links>
```

```
<link from="manual_issue_list_role" to="set_issue_list_state_role"
type="communication"   scope="intra-group" />
...
</group-specification>
```

The Functional Specification decomposes a root goal in a tree of sub-goals. The leaf goals of this hierarchy are pursued by using services from the yellow pages. For each service, a mission specifies a group of services that must be assigned to the same role.

```
<scheme id="voting_sch">
<goal id="voting_root">
<plan operator="sequence">
<goal id="determine_issues_goal">
<plan operator="sequence">
<goal id="manual_issue_list" />
<goal id="set_issue_list_state" />
</plan>
</goal>

<goal id="share_doc_via_email"/>

<goal id="gateway1">
<plan operator="parallel">
<goal id="g1_1st_branch">
<plan operator="sequence">
<goal id="check_calendar"/>
<goal id="moderate_conference"/>
</plan>
...
<mission id="manual_issue_list_mission" min="1" max="1">
<goal id="manual_issue_list"/>
</mission>
...
</scheme>
```

The Normative Specification specifies which role is responsible for which mission.

```
<norm id="n2" role="manual_issue_list_role" mission="manual_issue_list_mission"
type="obligation"/>
<norm id="n3 " type="obligation" role="set_issue_list_state_role"
mission="set_issue_list_state_mission" />
...
```

6 Discussion

The automatic MOISE generation centres the composition of the organization upon goals and agents' capabilities (required to satisfy goals). As a consequence,

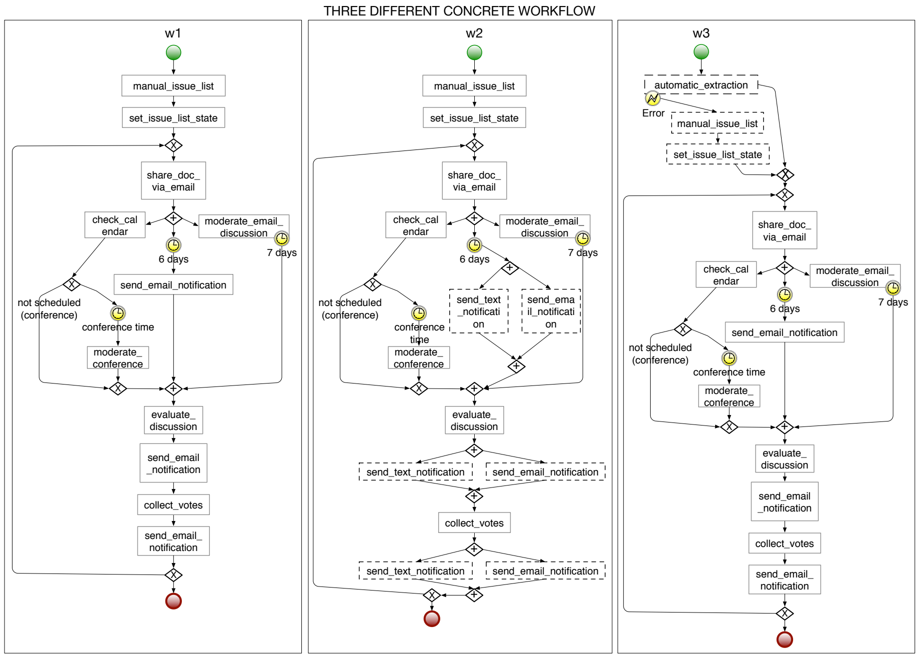

Fig. 6. These three concrete workflows are the obtained by observing the output of three different agent organisations.

agents owning these capabilities are involved in the solution organization, and are called to play one role for each of their capabilities needed for fulfilling the goals. Such an approach enforces the collaboration among agents and enables a large distribution of the work to be done, in an automatic way.

The proposed approach has the advantage of automatically defining alternative organizations for pursuing the goals underlying an input workflow (manually created by a business analyst).

For instance, Figure 6 shows of the activity logs produced by executing three different MOISE organisations, generated from the same initial set of goals and services.

The availability of a set of organizations that could pursue the same set of goals in different ways is a strong advantage because, if some service (or the owning agent) is no more available at the exact moment of the process execution, alternative solutions are promptly available and the problem may be overpassed. As a matter of fact, the availability of different organizations allows for:

– selecting the organization that provides the best performance, according to the quality attributes registered in the yellow pages, for each service and owning agent.
– switching the current organization, in case of agent/service failures (runtime adaptation feature).

If the failure occurs at the moment of recruiting the agents for the selected organization, an alternative one may be promptly chosen, allowing the enactment of the process despite of the initial failure.

Of course, if a failure occurs while the process is already running, it may be the case some goal had already been achieved and anyway the state of the world has been changed. Therefore, a new running of the PMR algorithm is advisable because from the current state of the world a different solution (in terms of goals and organizations) may be more appropriate.

Limits of the Proposed Approach. Despite the proposed approach produces effective results, it could be improved in many ways. For instance, the roles of the organization may be optimized in number and specialization. So far, different roles are defined for different capabilities even when they have the same pre- and post-conditions but different name. Although, future works may propose some improvement on that.

Moreover, workflow loops are out of the mapping proposed in Section 4 because of a lack of the corresponding structure in the MOISE specification. They are supported, in the proposed approach, at the agent level, who is directly responsible of checking loop conditions and decides to repeat or to leave the cycle. Although, as a future work, this should be integrated at level of functional specification of the organization. The implementation strategy will consider the loop as a *maintain* goal with some exit-conditions allowing to repeat the corresponding portion of process every time that is necessary. We think this limitation does not inhibit the overall validity of the approach since such limit is not theoretical but it is actually only in the implementation.

As part of the future works (and limits of the current work), we also would like to note that a few elements of the BPMN metamodel have been omitted in Fig. 3 for simplicity. While this is not relevant for most of them, we consider the sub-process a significant element that could lead to interesting extensions to the proposed approach. In fact, dealing with that as a kind of process in the process (as it is indeed), the result could bring to the design of organizations conceived to act within other higher-level (or lower-level) ones in a kind of hierarchy that may resemble a holarchy and some methodological issue may arise from that [8].

7 Conclusions and Future Works

This paper presented dynamic workflows based on 1) automatic generation of goals from BPMN and 2) mapping goals and services into agent organizations. Goals may relax BPMN constraints, allowing agents to re-arrange service's order and, in case of failure, to adapt the whole organization to overcome new operative context. Despite a broad literature on agent organizations, it still remains the problem of coordinating heterogeneous, autonomous agents, whose internal designs may not be fully known. And even more interesting, it is necessary to define re-organization rules and models for adapting the collective behaviour to unexpected events.

References

1. Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., et al.: Business process execution language for web services (2003)
2. Boissier, O., Bordini, R.H., Hübner, J.F., Ricci, A., Santi, A.: Multi-agent oriented programming with jacamo. Science of Computer Programming **78**(6), 747–761 (2013)
3. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming multi-agent systems in AgentSpeak using Jason, vol. 8. John Wiley & Sons (2007)
4. Buhler, P.A., Vidal, J.M.: Towards adaptive workflow enactment using multiagent systems. Information technology and management **6**(1), 61–87 (2005)
5. Ceri, S., Grefen, P., Sanchez, G.: Wide-a distributed architecture for workflow management. In: Research Issues in Data Engineering, 1997. Proceedings. Seventh International Workshop on. pp. 76–79. IEEE (1997)
6. Chinosi, M., Trombetta, A.: Bpmn: An introduction to the standard. Computer Standards & Interfaces **34**(1), 124–134 (2012)
7. Coalition, D.S., Ankolekar, A., Burstein, M., Hobbs, J.R., Lassila, O., Martin, D., McDermott, D., McIlraith, S.A., Narayanan, S., Paolucci, M., et al.: Daml-s: Web service description for the semantic web. In: The Semantic Web-ISWC. pp. 348–363. Springer (2002)
8. Cossentino, M., Gaud, N., Hilaire, V., Galland, S., Koukam, A.: Aspecs: an agent-oriented software process for engineering complex systems. Autonomous Agents and Multi-Agent Systems **20**(2), 260–304 (2010)
9. Cossentino, M., Sabatucci, L., Lopes, S.: Partial and full goal satisfaction in the musa middleware. In: European Conference on Multi-Agent Systems. pp. 15–29. Springer (2018)
10. Gottschalk, F., Van Der Aalst, W.M., Jansen-Vullers, M.H., La Rosa, M.: Configurable workflow models. International Journal of Cooperative Information Systems **17**(02), 177–221 (2008)
11. Han, Y., Sheth, A., Bussler, C.: A taxonomy of adaptive workflow management. In: Workshop of the 1998 ACM Conference on Computer Supported Cooperative Work. pp. 1–11 (1998)
12. Hannoun, M., Boissier, O., Sichman, J.S., Sayettat, C.: Moise: An organizational model for multi-agent systems. In: Advances in Artificial Intelligence, pp. 156–165. Springer (2000)
13. Hermosillo, G., Seinturier, L., Duchien, L.: Creating context-adaptive business processes. In: International Conference on Service-Oriented Computing. pp. 228–242. Springer (2010)
14. Hildebrandt, T.T., Mukkamala, R.R.: Declarative event-based workflow as distributed dynamic condition response graphs. arXiv preprint arXiv:1110.4161 (2011)
15. Hubner, J.F., Sichman, J.S., Boissier, O.: Moise Tutorial (for Moise 0.7). http://moise.sourceforge.net (2010)
16. Laukkanen, M., Helin, H.: Composing workflows of semantic web services. Extending Web Services Technologies pp. 209–228 (2004)
17. Milanovic, M., Gasevic, D., Rocha, L.: Modeling flexible business processes with business rule patterns. In: Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International. pp. 65–74. IEEE (2011)

18. Object Management Group (OMG): Business Process Model and Notation (BPMN 2.0) by Example. Available online at https://www.omg.org/cgi-bin/doc?dtc/10-06-02.pdf (2010)
19. Rao, A.S., Georgeff, M.P., et al.: Bdi agents: from theory to practice. In: ICMAS. vol. 95, pp. 312–319 (1995)
20. Reichert, M., Dadam, P.: Adept flex—supporting dynamic changes of workflows without losing control. Journal of Intelligent Information Systems **10**(2), 93–129 (1998)
21. Ricci, A., Viroli, M., Omicini, A.: Programming mas with artifacts. In: International Workshop on Programming Multi-Agent Systems. pp. 206–221. Springer (2005)
22. Sabatucci, L., Cossentino, M.: From means-end analysis to proactive means-end reasoning. In: Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems. pp. 2–12. IEEE Press (2015)
23. Sabatucci, L., Cossentino, M.: Self-adaptive smart spaces by proactive means–end reasoning. Journal of Reliable Intelligent Environments **3**(3), 159–175 (2017)
24. Sabatucci, L., Cossentino, M.: Supporting dynamic workflows with automatic extraction of goals from bpmn. ACM Transactions on Autonomous and Adaptive Systems (TAAS) **14**(2), 1–38 (2019)
25. Sawhney, M., Zabin, J.: The seven steps to nirvana: Strategic insights into ebusiness transformation. McGraw-Hill, Inc. (2002)
26. Singh, M.P., Huhns, M.N.: Multiagent systems for workflow. Intelligent Systems in Accounting, Finance & Management **8**(2), 105–117 (1999)
27. Van Eijndhoven, T., Iacob, M.E., Ponisio, M.L.: Achieving business process flexibility with business rules. In: Enterprise Distributed Object Computing Conference, 2008. EDOC'08. 12th International IEEE. pp. 95–104. IEEE (2008)