Automatic Definition of MOISE Organizations for Adaptive Workflows

Massimo Cossentino[a], Salvatore Lopes[b] and Luca Sabatucci[c]
National Research Council of Italy (CNR), ICAR, Palermo, Italy
t

Keywords: Multi-Agent Systems, Business Process, BPMN Adaptation.

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 process definition to generate an agent organisation that can enact the workflow using different strategies. These are implemented as organisational schemes representing alternative goal decomposition trees. The availability of several equivalent solutions 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 definition.

1 INTRODUCTION

In the last two decades, web services enabled business process running over the Internet (Sawhney and Zabin, 2002). Business enterprises yield to redesign their information and process management systems to implement advanced features such as adaptation ability (Laukkanen and Helin, 2004; Gottschalk et al., 2008).

However, the objective to increase agility and flexibility of business processes conflicts with the current trend of over-specifying workflow details. Moreover, the activity-diagram style makes hard to re-arrange the order of service.

A promising direction is to use multi-agent systems. This is an alliance, very frequent in literature (Buhler and Vidal, 2005; Singh and Huhns, 1999; Ceri et al., 1997), in which the enactment of workflows takes great benefits from distinctive agent features like distribution, adaptation and smartness.

It remains the problem of coordinating heterogeneous, autonomous agents, whose internal designs could not be fully known a-priori.

Based on some recent results (Sabatucci and Cossentino, 2019), this paper presents a novel approach for automatically generating an agent-based adaptive workflow that can enact high-level business process. This paper is an extension of a previous work that has been presented in (Cossentino et al., 2020a).

[a] https://orcid.org/0000-0003-1258-9767
[b] https://orcid.org/0000-0003-1488-9755
[c] https://orcid.org/0000-0003-2852-9355

The proposed approach is based on three pillars:

1. The use of BPMN (Chinosi and Trombetta, 2012) for specifying the workflow; BPMN is a high-level language that focuses on analysis activity, differently from other execution languages such as BPEL4WS (Andrews et al., 2003) (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 (Sabatucci and Cossentino, 2019). 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 (Sabatucci and Cossentino, 2017).
3. Workflow is a cooperative problem-solving application. 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 (Sabatucci and Cossentino, 2019) 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 a social organisation for enacting the workflow. The organization is composed of several alternate solutions (goal decomposition trees) in form of schemes.

To support this claim, we selected MOISE (Hannoun et al., 2000), an organisational model for agents

Cossentino, M., Lopes, S. and Sabatucci, L.
Automatic Definition of MOISE Organizations for Adaptive Workflows.
DOI: 10.5220/0010319201250136
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 1*, pages 125-136
ISBN: 978-989-758-484-8

for demonstrating the automatic generation.

Therefore, we map concepts from (Sabatucci and Cossentino, 2019) and from (Sabatucci and Cossentino, 2017) into the MOISE metamodel.

An automated tool derived from the approach, that has been discussed in (Cossentino et al., 2020b).

The paper is structured as follows: Section 2 presents the problem and an overview of the approach. Section 3 summarizes the theoretical background at the basis of this work. The main contribution is discussed in Section 4, where Section 5 provides a justification scenario.Finally, conclusions and future works are sketched in Section 6.

2 PROPOSED APPROACH

This section provides motivations for converting BPMN into MOISE organizations. Furthermore, it collocates the contribution of the paper into an extended framework for enacting dynamic workflows. Despite discussing the whole framework is out of the scope of this paper, we provide an overview for the reader to get the idea behind goal-driven agent-based adaptive workflows.

A BPMN diagram defines a set of activities, documents and messages and, above all, a set of procedural rules that specify the order in which activities must be executed to achieve a specific business goal. Since the beginning, intelligent agents have been often associated to workflow management to simplify service provisioning, execution and exception (Judge et al., 1998; Singh and Huhns, 1999). An agent is suitable for these operations because it is an autonomous, co-operative and intelligent entity able to collaborate with other agents to process the tasks (Wooldridge, 1999; Ferber and Gutknecht, 1998). One of the motivations because agents have been used in workflows is their stateful and non deterministic nature. Whereas a service is similar to a stateless method invocation, an agent observes its environment, maintains an internal state, learns from the experience if necessary, and its response may change along time (Buhler and Vidal, 2005).

Moreover, agents promise to solve one of the most interesting open challenge in this research areas: managing changes (van der Aalst and Jablonski, 2000) when changes of operating conditions cannot be avoided and cannot be anticipated.

The use of agents carries out the advantage of providing workflow management systems with the ability to face dynamic changes in resource levels and task availability, as well as redistributing workload when required (Judge et al., 1998). So far, agent organizations play a central role in workflow management and one of the open challenge (Maia and Sichman, 2020) is to identify interaction mechanisms enabling workflow adaptation.

The approach we propose uses agents, organizations and goals. Agents encapsulate workflow tasks, providing the advantage of autonomous decision making, easiness in distribution, and, optionally, the ability to maintain a state and learn from previous cases. Organizations represent the interaction mechanism for composing the workflow as sum of services with the additional advantage of acting under a customizable normative background. Goals are the instrument for binding organizational rules (for formation and adaptation) to the initial BPMN definition. These goals are automatically derived from the BPMN diagram and capture the nature of dependencies between states of the system. Providing the workflow as goals (rather than BPMN diagrams) allows relaxing some of the strong relationships imposed by sequence-flows, supporting the workflow management engine with a higher degree of freedom in defining (at run-time) the flow of activities.

2.1 A Motivating Example

This paper uses the email-voting business process, available in (Object Management Group (OMG), 2010). It is a high-level description of the process for mediating and coordinating voting members in resolving issues. For reasons of readability, a compact version of the workflow is shown in Figure 1: a manager prepares the issue list for a discussion; all participants propose solutions via email. After one week, the discussion is closed with a voting session and the manager communicates the results. If the issue has not been solved, then a new discussion starts.

Let us suppose that, given high-level specification of this process, developers create a set of web services for enacting the workflow.

According to the available services, there could be several workflows because: 1) they use different services, 2) their control flow structure is different to overcome different set of input/output parameters.

In any case, the workflow will present several potential points of failure. A web-service could be temporarily unavailable, or it may produce wrong/incomplete results. Moreover, given the presence of humans in the loop, the process may fail because a human's delay/failure. To cope with situations like these, the most promising approach is changing behaviour at run-time.

The use of a dynamic workflow management system allows run-time modification of the original busi-

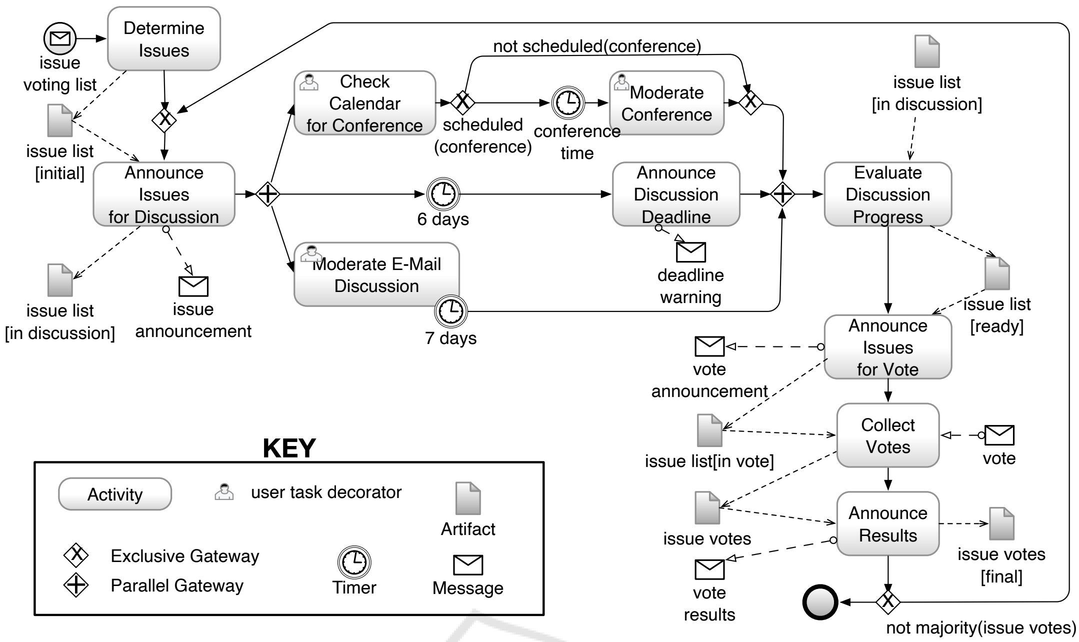

Figure 1: A compact version of the voting-by-email process, ri-elaborated from (Object Management Group (OMG), 2010).

ness process. However, despite the vast literature in dynamic worklfow, modifying at run-time the workflow structure still poses many challenges.

In our vision, if an agent organization encapsulates the workflow, then the use of agents in this context could provide significant benefits:

1. agent organizations are flexible structures: agents can enter/exit the organization at run-time; this allows repairing the portion of a malfunctioning workflow (local failure) by merely replacing the agent who is responsible of a failing service (or a block of services);
2. more complex changing strategies may be applied by exploiting agent interaction mechanisms: a distributed decision making could apply structural changes, i.e. modifying the order of activities for overcoming global problems;
3. agents may exploit learning activities for anticipating the need of adaptation, for example by associating an environmental condition with the probability that a failure happen.

This paper focuses on presenting the approach for automatically generating various organizational configurations for handling the same BPMN. Many aspects of self-adaptation are out the scope of this paper: 1) the algorithm for automatically extracting goals from BPMN is summarized in Section 3, but more details are in (Sabatucci and Cossentino, 2019); 2) the procedures for binding available services to the extracted goals are in (Sabatucci and Cossentino, 2015; Sabatucci et al., 2017).

2.2 Overview of the Proposed Approach

This work relies on the use of BPMN (or similar notations) for specifying the standard flow of tasks, at a high level of abstraction. It is suggested to model diagram as an information-centric workflow (Kumaran et al., 2008) i.e. a process with specific attention to the life cycles of information entities (documents, business objects, and artifacts). The workflow is abstract because tasks are not directly bounded to service.

This condition is necessary for enabling the algorithm to properly generate significant milestones and constraints to rule process execution. In general, this formal way to model workflows, including ontologies for specifying input and output data objects, is often encountered in literature (Francescomarino et al., 2011).

The overall framework is shown in Figure 2. After the workflow definition, an offline *pre-processing* phase precedes a run-time phase of *self-adaptive workflow* enactment. Two specific tools support the offline phase:

- the BPMN2Goal is a tool for automatically generating goals from BPMN workflow (Sabatucci and Cossentino, 2019);
- the Proactive Means-End Reasoning (PMR) (Sabatucci and Cossentino, 2015) is

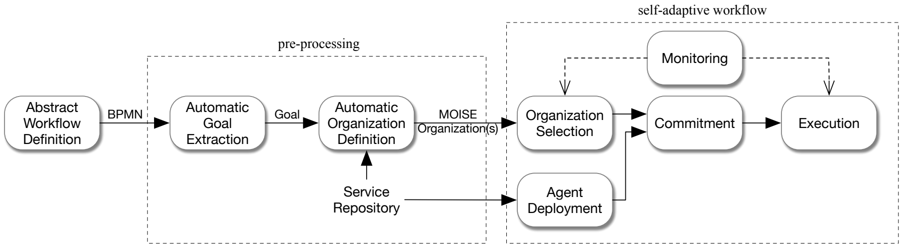

Figure 2: Overview of the proposed approach.

a service composition planner able of composing workflows by combining non-deterministic actions (i.e. actions with many possible results) to address linear-temporal logic goals (Sabatucci et al., 2017).

Here, the novel contribution is the Automatic Organization Definition, which elaborates goals and available services for producing one or more executable MOISE specification(s).

The *self-adaptive workflow* compartment of Figure 2 is implemented as a multi-agent system responsible for forming the organisation (by making each agent to play a role in it).

When an agent is not able to execute its service, or when a goal is violated, the current organization dismiss and another organization schema is selected, if available. In practice, self-adaptation ability is based on the *online switching between an agent organisation configuration to another one*. The Organization Selection phase acts as an organization manager: it evaluates all the available organisational options (previously generated and sorted according to some metric), and it is responsible for selecting the alternative one.

3 PRELIMINARY CONCEPTS

The first part of this section briefly introduces the main concepts used in this paper: BPMN, implicit goals and states of the workflow. The second part of this section provides an intuition of the BPMN2Goal procedure.

3.1 Background

The **Business Process Model and Notation** (BPMN) (Chinosi and Trombetta, 2012; Object Management Group (OMG), 2010) is a very expressive graphical notation for representing business processes of diverse natures. A graphical notation is the de-facto standard choice to express a representation of a process. The BPMN notation is very rich and allows several modelling perspectives (Chinosi and Trombetta, 2012; Object Management Group (OMG), 2010). This paper focuses on collaboration diagrams (similar to activity diagrams), in which a process is described as a collection of participants (each in a Swimlane) exchanging messages via message flows. Therefore these diagrams are composed of five categories of objects: activities, events, messages, data objects, and many kinds of gateways. Figure 1 reports the perspective of the Issue Manager role in the voting-by-email process.

Typically a designer encodes individual business goals to be accomplished in the model, after that the system is responsible to achieve these goals. The relationships between BPMN and goals is greatly studied (De la Vara González and Diaz, 2007; De la Vara González and Diaz, 2007; Adamo et al., 2018). Indeed, the OMG also offers alternative approaches for representing a BPMN in terms of goals, rather than activities. An example is the Case Management Model and Notation (CMMN) (Marin et al., 2012)in which designers describe what is allowed and disallowed in the process rather than how to actualize it. In (Adamo et al., 2018), authors present an investigation and a categorisation of the notion of goal in the context of business process.

This work focuses on **Implicit Goals**, declarative entities embedded in the structure of a workflow that define functional aspects of the process. They are operative, i.e. they aim at explaining why a workflow can evolve in a given way. Given this nature, implicit goals are necessarily formally expressed according to a precise formalism. Consequently, the definition of implicit goal, inspired by Zambonelli et al. (Abeywickrama et al., 2012) and aligned with most of the literature about goal models (Yu, 2011; Bresciani et al., 2004; Morandini et al., 2008) is:

Definition 3.1 (Implicit Goal)**.** An implicit goal is a pair: $\langle tc, fs\rangle$ where *tc* and *fs* are state-conditions. Respectively, the *trigger condition* (tc) describes when the goal becomes active, and it may be pursued; the *final state* (fs) represents the desired state to be addressed.

We describe the workflow by observing its evolution from the initial to the final state. The logic of possible worlds allows studying state dynamics along with a temporal line as a finite sequence of states. In order to support the analysis of **States of a Workflow**, in (Sabatucci and Cossentino, 2019) we introduced a specific semantic (based on predicate logics) for describing relevant states. For instance, the availability of a given artifact is expressed through a predicate like $available(\langle Data \rangle)$, whereas the predicate $\langle state \rangle(\langle Data \rangle)$ specifies the artifact assumes a given state. Similarly, $received$, $sent$, $caught$, $thrown$, at, $after$, $done$, $error$ are examples of predicates for other relevant states of the system.

For instance, a task with issue_list[initial] as input and issue_list[in_discussion] as output and an outgoing issue_management message is described as follows:

$$\begin{cases} data\_in(e) & = initial(issue\_list) \\ data\_out(e) & = in\_discussion(issue\_votes) \\ mess\_in(e) & = sent(issue\_announc, member) \end{cases}$$

3.2 The BPMN2Goal Procedure

The idea underlying converting BPMN into goals is: *if we were able to decompose the overall state evolution into a finite set of evolution steps, then each of these is one of the implicit goals we are looking for.*

At a first analysis, activities, events, and gateways contribute in different ways to the state evolution of a workflow. In particular activities are directly responsible for state changes, event elements capture either external events or produce internal signals, and gateway combines states according different strategies (exclusive choice, inclusive choice, parallel, and so on).

The idea, expressed in details in (Sabatucci and Cossentino, 2019), is that each BPMN element is ruled by internal conditions, i.e. conditions that holds when analyzing the element as isolated by the rest of of workflow (internal factors). However, when an element is connected to other elements, it also undergoes to external conditions that are generated by predecessors and successors. We must solve a 'balance of states': the internal factors must balance the external factors. By opportunely combining these forces, it is possible to estimate the state transition, and formulating the corresponding goal.

We report an example of the analysis of states for the Announce Issues for Discussion. Figure 3 shows relationships with predecessors and successors of the Announce Issues for Discussion. Two predecessors tasks are joined through some exclusive gateways. Briefly, we can assert that Announce Issues for Discussion

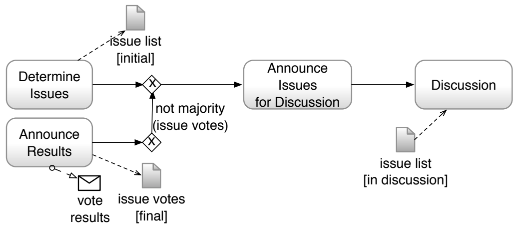

Figure 3: Predecessors and successors of the Announce Issues for Discussion task.

is ready for execution when the following condition holds: $initial(issue\_list) \oplus (sent(vote\_results) \wedge final(issue\_votes) \wedge \neg majority(issue\_votes))$ where the $\oplus$ derives from the exclusive gateways.

After its execution, to allow the workflow proceeds, the state must include $in\_discussion(issue\_list)$. Combining internal and external factors, we conclude the implicit goal is:

```
GOAL: Announce Issues for Discussion

WHEN:
(((final(issue_votes) and sent(vote_results))
and not majority(issue_votes))
xor initial(issue_list))

THE SYSTEM SHALL:
in_discussion(issue_list)
and sent(issue_announcement)
```

The BPMN2Goals tool is currently available as a web service[1]; it accepts BPMN files in the XMI format, according to the OMG specification. In the next section, two metamodels will be introduced to discuss how BPMN goals are mapped to MOISE organisations.

4 MAPPING GOALS TO ORGANIZATIONS

For realizing the agent organizations, we selected Jason (Bordini et al., 2007) for implementing BDI agents (Rao et al., 1995) and MOISE (Hannoun et al., 2000) for defining functional/structural/deontic aspects of the organization. The use of BDI agents amplifies the agent ability of taking decisions also considering environmental conditions. MOISE is the natural choice when working with Jason: it is supported by a well defined meta-model and by a strong theory behind. Agents encapsulate services

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

and make them available to the organization by registering them into a shared Yellow Page. A workflow is ready to be enacted when, selected the MOISE organization schema, interested agents commit to every missions. The architecture also comprises an internal-monitoring agent that estimates goal violations (Cossentino et al., 2018), thus raising adaptation signals when necessary.

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 (Sabatucci and Cossentino, 2019) to the corresponding MOISE concepts. The metamodel implicitly adopted in (Sabatucci and Cossentino, 2019) is represented in Fig. 4, 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 solution is composed by activities and flow relationships (relationships are used for organizing the activities control flows). In this work we are now omitting some elements of a classical BPMN workflow (sequence flow relationships, data objects, events, messages, ...).
- 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.
- Implicit 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, and therefore a 'Goal Extraction' procedural relationship is drawn from the BPMN Activity element to the Goal element.
- Service: it can be a cloud-, web-service or any other abstraction of an action that can be done in the agent's world.
- 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 (Sabatucci and Cossentino, 2019). The plan may be produced by the Proactive Means End Reasoning algorithm (Sabatucci and Cossentino, 2015) that composes a set of capabilities for satisfying the goal. A plan is made of a sequence 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.

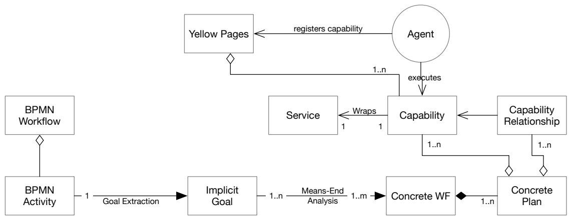

Figure 4: The metamodel implicitly adopted in extracting goal-oriented solutions from BPMN workflows.

It is significant to note that we can identify three main zones in this metamodel: first, the input part (on the left in Fig. 4). 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. 4: implicit goal, the key abstraction of the approach. Implicit goals are obtained by processing BPMN elements, mainly activities, as reported in Sect. 3. Finally, the rightmost part of Fig. 4 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 a different set of capabilities (and therefore by several plans), it is possible to create several Concrete Workflows that can pursue the set of (implicit) Goals extracted from the same BPMN workflow.

As discussed in sect. 2, a suitable agent organization is to be defined to enact the concrete workflows deduced from implicit goals extracted from the input BPMN workflow and, for discussing that in details, we will now refer to the metamodel of the MOISE organizational framework reported in Fig. 5.

MOISE models organizations from three different perspectives: structural, functional, and normative (Hannoun et al., 2000). The metamodel proposed in Fig. 5 only reports MOISE elements that are significant for the proposed approach. Indeed it also includes two elements (Collective Goal, Individual Goal) we introduced to better explicate the proposed approach; such elements are both coded as MOISE goals.

Below a description of the MOISE metamodel elements based on definitions proposed by Hubner in (Hannoun et al., 2000):

- 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 (Hubner et al., 2010)).
- 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. Not part of MOISE specifications.
- 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. Not part of MOISE specifications.
- 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 a global goal decomposition tree, it is the collective specification of the goals to be pursued by an organization. It also includes Missions allowing to pursue the goals defined in the scheme.

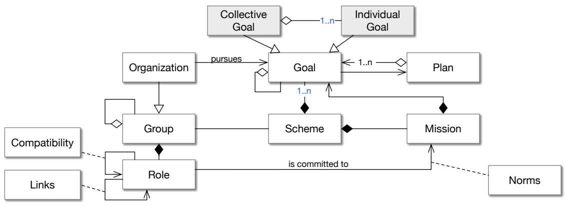

Figure 5: The portion of MOISE metamodel adopted in the proposed approach for workflow enactment.

- Mission: "is a set of goals for an agent commitment in the context of a scheme execution" (Hubner et al., 2010). It is 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.

In order to define the organization that could enact the 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. 5 starting from an instance of the metamodel reported in Fig. 4. The process for generating the MOISE organisation will be introduced in the following by referring to two main phases: the first consists in the definition of a set of solutions by employing the PMR algorithm as detailed before, the second phase consists in the instantiation of the MOISE metamodel elements.

4.1 BPMN Solutions Definition with PMR Algorithm

Let us suppose the BPMN process reported in Fig. 6 is the input and the extracted implicit goals are G1, G2, G3. They are all to be achieved in order to reach the objectives of the BPMN process. Therefore we could represent this situation with the implicit goal tree of Fig. 6. For an easier tracing of the details about goals extraction that are omitted in this paper, we are

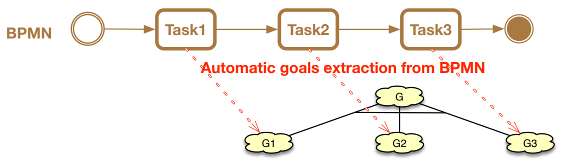

Figure 6: An example of implicit goal tree extracted from a simple BPMN process.

referring to the same example used in (Sabatucci and Cossentino, 2019).

Now, let us suppose the solutions found using the Proactive Means-end Reasoning (PMR) algorithm are w1, w2, w3, w4, w5 as reported in the leftmost part of Fig. 7. They are represented in the rightmost part of the same figure using this notation: rectangles represent services that can be used to achieve the goals. They are organized in plans using PMR solution control nodes like sequence (–>) parallel (||) or ExOR (×). Agents exhibit the behaviour prescribed by these services employing their capabilities (a capability often is purely a wrapping of a service). For instance, solution w1 is the sequence of three services: s1, s2, s3. Each of these services is connected to the goal it fulfils by a dashed line ending with a not filled arrow. The first level plan operator defining the outermost configuration of the solution (a sequence or parallel of other nodes) is the PMR solution plan (in black in Fig. 7). This is identified with the solution itself and therefore named after that. For the sake of simplicity, only solutions w1 and w5 are reported in Fig. 7.

Solution w4 represents an interesting situation: two services (s1", s1"') are to be executed in a sequence in order to fulfil goal G1, this is represented by using a PMR Solution Control Node (sequence type) and it is represented by introducing a blue hexagon with the sequence symbol (–>). The correct order is prescribed in numbers at the tail of the aggregation relationship between the hexagon and the rectangles representing the services.

Similarly a parallel is prescribed by solution w5 and this is represented by a plan icon with the parallel symbol ||. Solution w3 employs a service (s1') that can fulfil the two goals G1 and G2 at the same time. In Fig. 7 this is represented by introducing a grey coloured G1_2 goal. The same happens for the already cited parallel of s4', s4" that fulfils goals G2 and G3.

4.2 MOISE Organisation Definition

The definition of the MOISE organisation starts with the identification of its goals. One **goal** is defined per each service (see Fig. 8). Goal type is *achievement* and the pursued final state derives from the exit condition of the corresponding service. The plan for achieving each goal is straight since it just implies executing the corresponding service. Goals are composed into plans inside schemes according to the PMR solution plan (see later on).

One scheme is defined per each PMR solution. The plan for the scheme is represented by using a magenta hexagon in Fig. 8, and it is composed of two kind of elements: (MOISE) goals and (MOISE) plans (both in magenta). The root level goal is assumed to be the BPMN root level implicit goal (G in Fig. 8). Goal decomposition is reconstructed by referring to the PMR solution plan (black hexagons in Fig. 8) that have a 1:1 correspondence with the MOISE plan of each scheme (magenta hexagons). Goals in the scheme are those related to the services employed in the corresponding PMR solution.

Therefore, PMR solution w1 generates a MOISE scheme (label "Scheme_sol1") whose root goal is the same as the original BPMN goal tree (G in Fig. 8). The plan for achieving G is Plan1, and that is a sequence of goals g1, g2, g3. One mission is defined per each goal in the plan. Fig. 9 reports the schemes and groups defined from solutions w1 and w5.

One group is created per each scheme. Indeed we are here talking of subgroups of a root group containing a general role labelled Worker. All roles in the subgroups are extensions of the Worker role.

Inside each group, one role is defined per each service listed in the corresponding PMR solution. Roles are related by communication links when the plan in the scheme is a sequence of goals. In this way the first role may communicate to the second one when its work is done, and so on. Roles to be executed in a parallel are related by a communication link to roles responsible for tasks following the parallel (i.e. tasks positioned after the Join element in the BPMN) (not represented in this example).

One mission is defined per each goal in the scheme. It is worth saying that as prescribed by MOISE syntax (see (Hubner et al., 2010), pag. 16) plans inside the first level one are to be replace by auxiliary goals but this is a syntactic expedient that does not effect the proposed approach.

Roles are related to their corresponding mission by means of an Obligation norm.

Summarizing, we define one goal per service, one role per service (wrapped into a capability in agents), one mission per goal and we oblige each role to commit to the mission with the goal springing from its service.

One group is defined per scheme, it collects the roles involved in all the missions defined within the scheme.

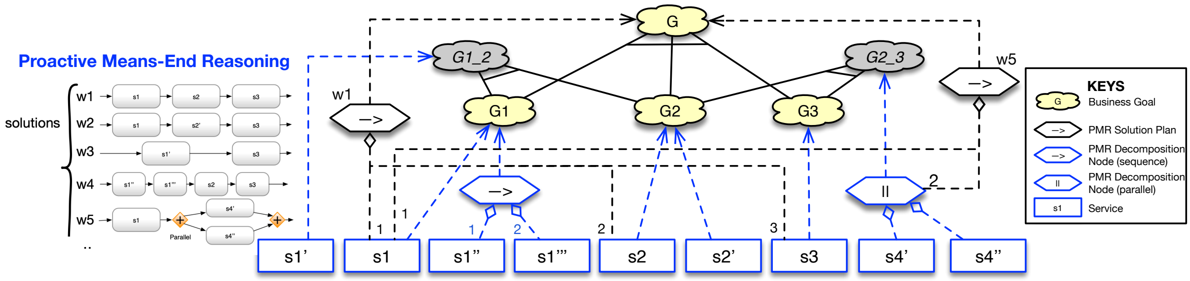

Figure 7: The solutions found by the PMR algorithm for fulfilling the goal tree.

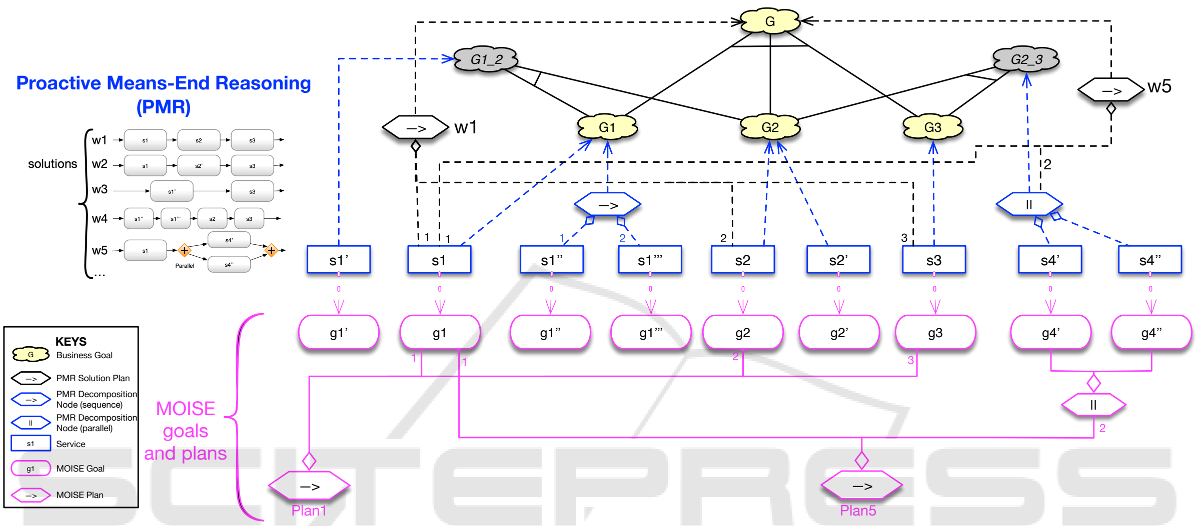

Figure 8: Goals and Plans for the MOISE organisation.

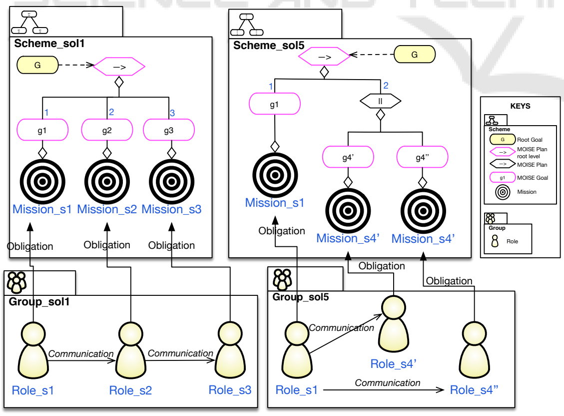

Figure 9: MOISE organisation schemes and groups.

We are aware this process could be enhanced, for instance, by clustering more goals in one mission like the alternate solution proposed in Fig. 10. In this solution, one mission is defined per each element in the second level of the scheme (the first level is represented by the plan connected to G in Fig. 10). Indeed such a solution would need some modification in the MOISE framework. If more than one role is obliged to commit to the mission, the framework requires that each role must be able to perform all the goals listed in the mission. This is against the way we construct roles (one role per service from the PMR solution). Building roles that are able to perform all the goals in a mission raises the need for more complex agents that have more capabilities (i.e. the ability to perform all the required services).

In more straightforward approach we propose in Fig. 9, each agent needs to have only one of the capabilities required for satisfying the mission's goals. This assumption is sufficient for it to participate in the collaborative effort towards the solution.

Future extensions of this work will explore enhancement strategies, also including the adoption of subgroups.

Fig. 11 reports the mapping between the elements of the metamodels proposed in Fig. 4 and Fig. 5 and a list of some elements introduced during the transformation process detailed in this section.

An automated tool for the generation of MOISE organisations was presented in (Cossentino et al., 2020b).

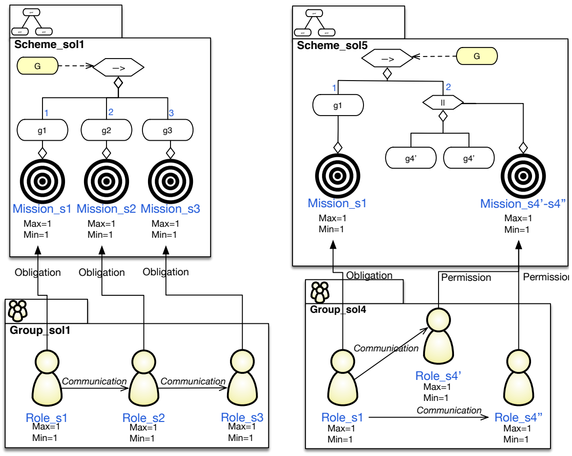

Figure 10: MOISE organisation with several goals inside one mission.

MOISE Metamodel	PMR Metamodel	Heuristics
G Moise Root Goal	G Implicit Goal	The organization root level goal is the same of the implicit goal
g1 Goal	s1 Service	One goal is defined per each service. Name: "g_" + name of the service
Scheme	PMR Solution	One Scheme per each PMR Solution. Name: "Scheme_"+ name of solution
1st level Plan	PMR Solution	One plan per solution (first decomposition of root goal in the scheme)
Plan	PMR Decomp. Node (seq. or par.)	One plan per each decomposition node in the PMR solution in successive plan levels (from second on)
Group	PMR Solution	One group per scheme (or PMR solution that is the same). Name: "Group_"+ name of solution
Role	s1 Service	One role per each service involved in the PMR solution. Name: "Role_"+ name of service
Communication	Solution/ Decomp. Node (sequence)	One communication link per each couple in a sequence plan or to relate all parallel tasks to the following sequence ones (if any)
Mission	g1 Goal	One mission per each goal in the scheme

Figure 11: The proposed mapping among PMR solutions and MOISE organisation metamodel elements.

5 ORGANIZATIONS AT WORK

In this section, we will provide a justification scenario for the proposed approach.

Suppose a business analyst designs (using BPMN) a workflow for dealing with some process. Despite the great attention and effort adopted in the design task, the analyst is well aware that a fixed process cannot cope with runtime changes in the execution environment like: entering new agents or disappearing existing agents, variations in the performance or cost of the services offered by agents, exogenous factors affecting the real world where the system operates.

For these reasons, the analyst prefers to adopt an adaptive execution environment for her workflow. Using an online utility (see sect. 3.2) she gets a set of goals from her BPMN file.The PMR algorithm computes a set of alternative solutions for achieving these goals. Each solution is composed of a set of capabilities (corresponding to those available in the yellow pages).

Behind each capability there is a service that can be invoked by the agent or some action the agent can perform. For each couple capability-agent, the yellow pages also list some quality of service (QoS) value. This latter is domain dependent and for each process the designer should provide the system with a formula for calculating the QoS of the overall solution.

Now, for the sake of conciseness, we will refer to the same example reported in the previous section. Therefore, let us suppose the solutions are w1...w5 and the previously described organization is adopted to execute the solutions.

It is now relevant to remind that the organization encompasses 5 schemes (one per each solution w1...w5), one different group of roles is defined for each scheme. The different set of roles in each group implies that a different set of agents is necessary to deploy the capabilities that can satisfy the goals in the scheme. It may be useful to remark that sometimes one agent offers more than one capability and therefore it may play several roles in the same group or it may participate in different groups. We label each set of agents that forms one group and can execute the corresponding scheme as *grounding configuration*. Generically speaking for the 5 groups of the example, we can suppose more than 5 grounding configurations may exist (it depends on the number of agents and the redundancy in their capabilities). For each grounding configuration, the system calculates the (expected) QoS, and the best one defines which configuration will be instantiated.
Hence, the best grounding configuration from the non functional point of view (expressed by a QoS) will be deployed to execute the process.

Process execution may incur in unexpected failures, for instance, because a remote service is no more reachable or the code of an agent is bugged. In that case process adaptation is needed and our approach offers an advantage. If the selected grounding configuration has not achieved all the goals, others (with a lower QoS) are promptly available to be deployed and continue the work. The organization remains the same, a new set of agents will commit to the missions of the same scheme (alternative grounding configuration for the same scheme) or to another one.

We are aware that adopting solutions computed to execute the whole process may not be the optimal choice when starting from a partially executed one. It could be anyway a reasonable compromise with the time and resources needed to compute a new set of solutions for the current state of the world, extracting goals, defining the supporting organization and injecting that in the running system (not trivial with the selected JaCaMo framework). If no grounding configuration will be able to conclude the execution, a new iteration will start from the calculation of a new set of solutions with the PMR algorithm.

Indeed, there is also another optimisation possibility we did not consider in this paper. The PMR algorithm is usually perpetually run if new capabilities are registered in the yellow pages or new agents enter the system and could produce new solutions. The new solutions could be better than the previous ones. Further development of the proposed approach will consider them, thus allowing their introduction during the adaptation phase.

6 CONCLUSIONS AND FUTURE WORKS

This paper presented an approach for a runtime goal-driven adaptation of MOISE organizations in the execution of BPMN processes. The approach is based on: 1) automatic generation of goals from BPMN and 2) mapping goals and service-oriented solutions into different schemes of the agent organization that can be selected according to performance criteria or to overcome a failure. Goals may relax BPMN constraints, and the proposed approach has the advantage of automatically defining alternative organizational solutions (several schemes inside one organization) for pursuing the goals underlying the input workflow (manually created by a business analyst). As a matter of fact, the availability of different organization's schemes allows selecting the scheme (goal decomposition tree and set of missions) that provides the best performance, according to the quality attributes registered in the yellow pages. It also allows switching from the current scheme to another one, in case of agent/service failures (runtime adaptation feature).

Although the proposed approach produces effective results, it could be improved in many ways. For instance, agent roles 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. Future works may propose some improvement on that.

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. 4 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 (Cossentino et al., 2010).

REFERENCES

Abeywickrama, D. B., Bicocchi, N., and Zambonelli, F. (2012). Sota: Towards a general model for self-adaptive systems. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2012 IEEE 21st International Workshop on*, pages 48–53. IEEE.

Adamo, G., Borgo, S., Di Francescomarino, C., Ghidini, C., and Guarino, N. (2018). On the notion of goal in business process models. In *International Conference of the Italian Association for Artificial Intelligence*, pages 139–151. Springer.

Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., et al. (2003). Business process execution language for web services.

Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming multi-agent systems in AgentSpeak using Jason*, volume 8. John Wiley & Sons.

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., and Mylopoulos, J. (2004). Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3):203–236.

Buhler, P. A. and Vidal, J. M. (2005). Towards adaptive workflow enactment using multiagent systems. *Information technology and management*, 6(1):61–87.

Ceri, S., Grefen, P., and Sanchez, G. (1997). Wide-a distributed architecture for workflow management. In *Research Issues in Data Engineering, 1997. Proceedings. Seventh International Workshop on*, pages 76–79. IEEE.

Chinosi, M. and Trombetta, A. (2012). Bpmn: An introduction to the standard. *Computer Standards & Interfaces*, 34(1):124–134.

Cossentino, M., Gaud, N., Hilaire, V., Galland, S., and Koukam, A. (2010). Aspecs: an agent-oriented software process for engineering complex systems. *Autonomous Agents and Multi-Agent Systems*, 20(2):260–304.

Cossentino, M., Lopes, S., and Sabatucci, L. (2020a). Goal-driven adaptation of moise organizations for workflow enactment. In *Proc. of the 8th International Workshop on Engineering Multi-Agent Systems (EMAS 2020), National Research Council of Italy*.

Cossentino, M., Lopes, S., and Sabatucci, L. (2020b). A tool for the automatic generation of moise organisations from bpmn. In *CEUR Workshop Proceedings of Workshop "From Objects to Agents", September 14–16, 2020, Bologna, Italy*.

Cossentino, M., Sabatucci, L., and Lopes, S. (2018). Partial and full goal satisfaction in the musa middleware. In *European Conference on Multi-Agent Systems*, pages 15–29. Springer.

De la Vara González, J. L. and Diaz, J. S. (2007). Business process-driven requirements engineering: a goal-based approach. In *Proceedings of the 8th Workshop on Business Process Modeling*. Citeseer.

Ferber, J. and Gutknecht, O. (1998). A meta-model for the analysis and design of organizations in multi-agent systems. In *Proceedings International Conference on Multi Agent Systems (Cat. No. 98EX160)*, pages 128–135. IEEE.

Francescomarino, C. D., Ghidini, C., Rospocher, M., Serafini, L., and Tonella, P. (2011). A framework for the collaborative specification of semantically annotated business processes. *Journal of Software Maintenance and Evolution: Research and Practice*, 23(4):261–295.

Gottschalk, F., Van Der Aalst, W. M., Jansen-Vullers, M. H., and La Rosa, M. (2008). Configurable workflow models. *International Journal of Cooperative Information Systems*, 17(02):177–221.

Hannoun, M., Boissier, O., Sichman, J. S., and Sayettat, C. (2000). Moise: An organizational model for multi-agent systems. In *Advances in Artificial Intelligence*, pages 156–165. Springer.

Hubner, J. F., Sichman, J. S., and Boissier, O. (2010). *Moise Tutorial (for Moise 0.7)*. http://moise.sourceforge.net.

Judge, D., Odgers, B., Shepherdson, J., and Cui, Z. (1998). Agent-enhanced workflow. *BT Technology Journal*, 16(3):79–85.

Kumaran, S., Liu, R., and Wu, F. Y. (2008). On the duality of information-centric and activity-centric models of business processes. In *International Conference on Advanced Information Systems Engineering*, pages 32–47. Springer.

Laukkanen, M. and Helin, H. (2004). Composing workflows of semantic web services. *Extending Web Services Technologies*, pages 209–228.

Maia, A. V. and Sichman, J. S. (2020). Representing planning autonomy in agent organizational models. *Theoretical Computer Science*, 805:92–108.

Marin, M., Hull, R., and Vaculín, R. (2012). Data centric bpm and the emerging case management standard: A short survey. In *International Conference on Business Process Management*, pages 24–30. Springer.

Morandini, M., Penserini, L., and Perini, A. (2008). Towards goal-oriented development of self-adaptive systems. *Proceedings of the 2008 international workshop on Software engineering for adaptive and self-managing systems*, pages 9–16.

Object Management Group (OMG) (2010). 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.

Rao, A. S., Georgeff, M. P., et al. (1995). Bdi agents: from theory to practice. In *ICMAS*, volume 95, pages 312–319.

Sabatucci, L. and Cossentino, M. (2015). 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*, pages 2–12. IEEE Press.

Sabatucci, L. and Cossentino, M. (2017). Self-adaptive smart spaces by proactive means–end reasoning. *Journal of Reliable Intelligent Environments*, 3(3):159–175.

Sabatucci, L. and Cossentino, M. (2019). Supporting dynamic workflows with automatic extraction of goals from bpmn. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 14(2):1–38.

Sabatucci, L., Lopes, S., and Cossentino, M. (2017). Self-configuring cloud application mashup with goals and capabilities. *Cluster Computing*, pages 1–17.

Sawhney, M. and Zabin, J. (2002). *The seven steps to nirvana: Strategic insights into ebusiness transformation*. McGraw-Hill, Inc.

Singh, M. P. and Huhns, M. N. (1999). Multiagent systems for workflow. *Intelligent Systems in Accounting, Finance & Management*, 8(2):105–117.

van der Aalst, W. M. and Jablonski, S. (2000). Dealing with workflow change: identification of issues and solutions. *Computer systems science and engineering*, 15(5):267–276.

Wooldridge, M. (1999). Intelligent agents. *Multiagent systems*, 6.

Yu, E. (2011). Modelling strategic relationships for process reengineering. *Social Modeling for Requirements Engineering*, 11:2011.