Automated deduction of cross-organizational collaborative business processes

Aurelie Montarnal [a,*], Wenxin Mu [b,*], Frederick Benaben [a], Jacques Lamothe [a], Matthieu Lauras [a], Nicolas Salatge [a]

[a] *IMT Mines Albi, University of Toulouse, Albi, France*
[b] *Department of Information Management, Jiao Tong University, Beijing, China*

ARTICLE INFO

ABSTRACT

Being able to implement efficient cross-organizational collaborations has become a key factor for enterprises to respond to emerging market opportunities. The business process management approach is commonly used to design cross-organizational collaborations. This type of business process aims at achieving specific collaborative objectives by addressing three main steps according to a top-down approach: (i) defining the business services that have to be performed to reach the objectives, (ii) finding the best set of partners to provide them and (iii) ordering the business services in an optimized way. While the resulting business processes are a cornerstone to support the interoperability among the partners of a collaboration, their design step remains often humanly-conducted and laborious. Moreover, seeking the "best" set of partners involves non-additive criteria such as the delivery time (i.e. business services can be performed in sequence or in parallel within the process). In this context, this paper presents a decision support system based on an Ant Colony Optimization algorithm to exploit collaborative knowledge gathered from companies on a dedicated platform (companies' profile models registered to the platform and collaborative opportunity models) and deduce quasi-optimal collaborative business processes. A prototype that supports this system is also presented.

1. Introduction

In the current economic context, surviving as an isolated company has become unrealistic. The necessity to quickly respond to the emerging market opportunities leads to two consequences: (i) the companies have to react quickly and (ii) must be ready to establish new collaborations dedicated to the specific opportunities. While the recent advances in information technologies have enabled the creation of useful tools to support the coordination between the partners of a collaboration, there is still a lack when it comes to help companies in their collaborations (i.e. share and achieve common goals with other heterogeneous enterprises). In most of the cases designing cross-organizational collaborations relies on empirical abilities, informally held by specific people in the company. As a result, the scope of potential partners remains narrow and there is a lack of flexibility when mounting on-the-fly collaborations dedicated to emerging opportunities.

* Corresponding author.
 E-mail addresses: aurelie.montarnal@mines-albi.fr (A. Montarnal), wxmu@bjtu.edu.cn (W. Mu).
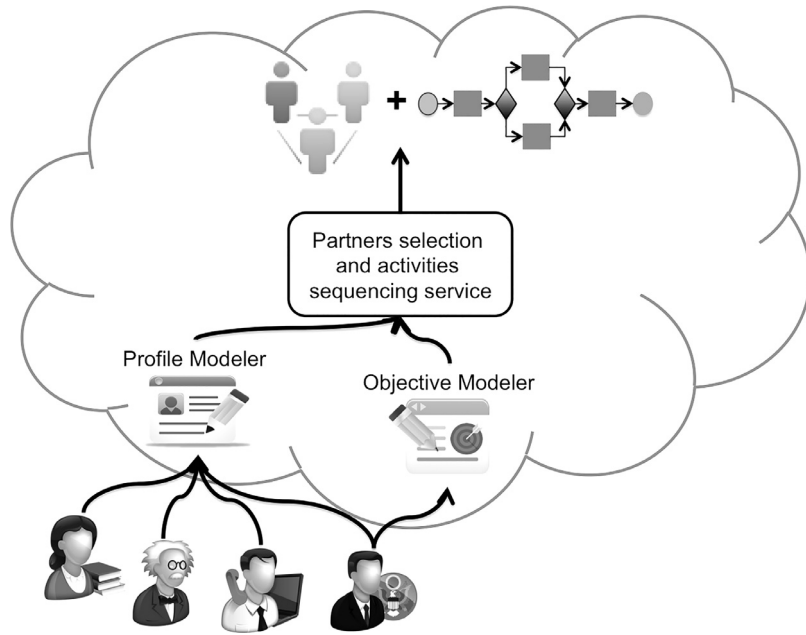
Fig. 1. Functional big picture of the collaborative process deduction service of OpenPaaS.

In this sense, the notion of interoperability of organizations has emerged as the ability of heterogeneous systems to build harmonious and intentional collaborative behaviors without needing deep changes in their structure [34]. Among many methods, Mu et al. [27] claim that setting up an ontology-based mediation information system appears to be the best way to ensure interoperability because, on the contrary to other methods, it doesn't rely on any standardization often laborious to integrate and update on a long term. Moreover, interoperability concerns the business level as well as the technical level of a collaboration, which, in information systems, can be transposed as business process and workflows respectively. A Mediation Information System (MIS) [46] is a suitable solution to bridge the gap between both business and technical levels by dealing with process orchestration and technical service selection. However, before the collaborative process can be orchestrated, it has to be designed.

This research is part of a bigger project called MISE (Mediation Information System Engineering) [2] that proposes to automatically set up MIS to fit the specific needs of inter-organizational collaborations. It is based on an Model-Driven Architecture and presents a knowledge-based system that can be decomposed into the following steps: (i) gather collaborative knowledge via business-oriented models, (ii) elaborate a suitable collaborative business process to reach the latter gathered collaborative objectives, (iii) turn the collaborative business process into a technical workflow and (iv) orchestrate the workflow. The design-time of the system consists in the three first steps, while the run-time concerns the final orchestration of the workflow. In continuation with the works presented by Mu et al. [27] (Section 2.1 highlights the novelties of our research), this paper focuses then on the two first parts and aims to address the question of the automated elaboration of collaborative business processes in response to specific collaborative objectives. It is also part of the OpenPaaS[1] project that aims to implement a collaborative Platform as a Service that offers companies helpful services in order to improve their cross-organizational collaborations. Enterprises can describe themselves through a Profile Modeler (cf. Fig. 1). Then, an Opportunity Modeler (cf. Fig. 1) allows enterprises to propose new collaborative opportunities. Limited to these two inputs, it is proposed to infer a suitable collaborative processes to set up specific MIS dedicated to address each opportunity. As it is expected to have numerous organizations able to provide the same capabilities to reach a specific collaborative opportunity, we aim to find an optimal process according to non-functional criteria (e.g. cost, delivery time...). However, because of some criteria like the entire process execution time, the whole set of partners and their ordered business services must be known before assessing a candidate solution.

Hence, this research can be decomposed into three main problems that have to be addressed simultaneously:

- Which business services should be performed in order to fulfill the objectives? (*What?*)
- Who could provide these business services to achieve the best results (in terms of criteria such as time of completion, or cost)? (*Who?*)
- When should each organization execute its business services? (*When?*)

---

[1] http://open-paas.org/.

最终选择，bpmn2.0作为最终流程的格式

It should to be noted that the Business Process Modeling Notation 2.0 (BPMN 2.0) has been chosen as the expected format of the final process, since it provides orchestrable processes without requiring any model transformation [35].

This paper is articulated in four main parts. Section 2 proposes a state of the art focused on how such systems could be implemented to support collaborations, as well as on the different algorithms typically used within the service composition topic. Then, Section 3 explains the knowledge representation adopted within our system and focuses particularly on its specific structure, which could be depicted as a knowledge graph with AND and OR nodes. Based on that, Section 4 is entirely dedicated to the exploitation of this knowledge base, with the implementation of an original and particularly suitable Ant Colony Optimization algorithm, and the evaluation of its performance. Section 5 proposes a use-case to illustrate the benefits and limits of the implemented system. Finally the ideas outlined in this paper will be discussed and some perspectives will be highlighted in the conclusion.

2. Literature review and research positioning

目的：改善企业之间的协作

Since the beginning of the rise of the IT technologies and with the implementation of web-based platforms, the applications for improving inter-enterprises collaborations have become a priority. First, with this growing interest, standards like RosettaNet [18] have been set up to provide more efficient collaborations. But due to their lack of flexibility and because they only focus on a technical level, other works have been led, rather oriented toward design-time automation: in particular, the service composition research field concerns a large part of the literature review, both in Management Science and Computer Science. In this Section, the first part provides an overview on the few already existing methods to deduce business processes in response to collaborative opportunities. Then, the second part focuses on the service composition topic and brings the fact that these kind of problems can be considered as NP-hard problems. That is why the metaheuristics domain is studied and in particular the use of ACO in service composition context.

建立了标准
提供更有效的协作
缺乏灵活性，只关注技术层面

2.1. Existing solutions for business process deduction

In [26], Mu et al. presents a metamodel and the corresponding knowledge exploitation for the deduction of business processes following the Business Process Modeling Notation (BPMN) [30]. [26] is centered on the use of a collaborative ontology via first order logical and inferences rules to obtain a BPMN process from (i) a set of partner wishing to work together but without knowing how exactly to do this and (ii) business objectives of collaboration. The research conducted by Mu et al. [26] sets the foundations of this paper.

To the authors' knowledge, the research of Ko et al. [21] faced the closest issues. Ko et al. [21] have based their approach on a Hierarchical Task Network (HTN) ontology called the Business-OWL (BOWL). The strength of their work relies on the implementation of an ontology not only as a knowledge base but that also embeds the resulting business task decomposition of the HTN. At the time of the beginning of these works, in 2007, Cloud Computing was just a new research topic and was not yet one of the highest concerns in Computer Science and the idea of social networks were also at a very early stage. Since then, both have revolutionized IT habits, particularly with the growing need for more flexible approaches. To the authors' belief one of the main key factor of current process deduction system for industrial applications is the ability to offer this flexibility. For instance, a simpler decomposition of the central collaborative ontology would allow users to bring their own knowledge, either by creating their own already structured ontology, or by an alignment of their existing ontologies with the collaborative ontology of the system. We propose to use a collaborative ontology that does not include the intelligence of the deduction but a flat and 'accessible-to-newcomers' description, by answering the following questions 'What are the activities usually needed to fulfill this collaborative objective?' and 'Is it possible to divide this objective into lower-level sub-objectives?'. Moreover, the explosion of the cloud computing technologies now allows for the users to access to a Platform as a Service (PaaS) on which the business process deduction service can be deployed, and also to bring their own Softwares as a Service (SaaS) and link them to their business services: in such a way the tasks of the processes can be human (e.g. production tasks), automated (e.g. customized forms creation) or also directly linked to a SaaS.

2.2. Research positioning regarding service composition

Many research works have been led in the field of service composition with the guiding thread of the establishment of business processes in order to model new cross-organizational collaborations. These works usually deal with two main topics: (i) Management Science and most of all (ii) Computer Science. Thus, those are the two investigated topics for this section.

2.2.1. Service composition in management science

In Management Science, many works focus on the establishment of supply chain networks and more generally on the creation of Virtual Enterprises (VEs). Sha and Che in [40] propose a Genetic Algorithm to find an optimal set of partners providing the required business services in the context of a supply chain. Based on a 'macro-process' also called a collaborative pattern, they discover candidate partners and achieve a global optimization on various non-functional criteria as cost, capacity... This method is interesting because of the multi-objective context and the necessity of finding a whole optimal set of partners, which is not the same as finding each single best partner. Even if in our case this algorithm cannot be applied,

because of the hypothesis of an already known 'macro-process', it leads to an interesting use of a metaheuristics. In [8], Crispim and Sousa propose a way to select partners in the context of a VE. Making the hypothesis of a star-like collaboration (i.e. a central decision maker and a network of linked partners all around), and project(s) whose activities are known, Crispim and Sousa use a Tabu Search algorithm in order to find the best set of partners to achieve the project(s), according to various global objectives and constraints. One can also note that the planning discipline has been widely discussed in Management Science, as shown in [24]. Tang and Shen [41], for example, describe a HTN approach to provide on-the-fly and specifically suitable crisis response processes. However, such planning techniques assume at least that the partners are already known.

As a result, the relative works led in Management Science on the service composition and partners discovery topics illustrate (i) the methods usually applied in this field (i.e. metaheuristics) and (ii) the strong assumptions made (i.e. either the main steps of the collaborative process or the partners of the collaboration are already known).

2.2.2. Service composition in Computer Science

Concerning Computer Science, the literature is prolific on service composition, since it is one of the main breakthroughs made possible particularly with the emergence of the SOA (Service Oriented Architecture) paradigm as a way to conceive services as the fundamental elements of an application [32]. In [13], Dustdar and Schneider have proposed a survey on web services composition solutions, which are not so far from the scope of this paper, since web services composition approaches can generally be adapted and conducted on a business level (and vice versa). They divide the topic into two approaches: (i) the static composition composes services in design-time, and builds on the hypothesis that partners are fixed; (ii) the dynamic one allows the discovery of services during the run-time and leads to evolving processes. Because of obvious reasons linked to inter-enterprise business agreements, and in line with the idea of automated one shot bids, the dynamic approach is clearly discarded. According to them, the service composition is closely linked to business workflows, since each business task provides information for finding corresponding technical services: a condition to fit the operational needs, the linkage among the web services with message flows, events, its provider. These last information should also be deduced in the context of business process deduction. However, one can observe that the technical and business approaches are conducted on two completely different abstraction levels and consequently with different hypothesis with regard to the input. For instance, in [1], Benaben et al. propose a semantic reconciliation from n business services to m web services, in order to obtain a BPEL file from a BPMN process. The transformation concerns the use of ontologies as knowledge bases for the matching between business tasks information and web services annotations: what is known about a business task is (i) its business role, (ii) its flows (input, output) and optionally (iii) its provider (i.e. one can think of 'generic' business partners to be found and whose role is only made clear on a technical level). When compared, the deduction of a business process is however only based on information about 'what the broker wants to do', which could be only compared to a very high-level 'business role' of the process, that should be decomposed. In [17], Garcia-Crespo et al. highlight the importance and the contribution of the Semantic Web and the numerous derived knowledge-driven approaches dedicated to the business process management, in particular web services composition.

After workflow based composition, Rao and Su [36] present Artificial Intelligence (AI) planning composition defined by a five-tuple $\{S, S_0, G, A, \Gamma\}$, with S the set of all possible state of the world, S_0 the initial state, G the goal state of the world, A the set of actions available and Γ the presets and effects of each action if executed. Besides, Wang et al. [44] dedicates their survey on bio-inspired algorithm for web services composition. According to them, with the current need Quality of Service (QoS) evaluation, the issues of services composition have evolved: and based on the explanation of Canfora et al. [6], it has become NP-hard problems, and their solving algorithms have evolved consequently. Section 2.3 is therefore entirely dedicated to the service composition through current promising metaheuristic approaches.

As a result, Fig. 2 illustrates the usual service composition approaches at a functional level. One can note that usually, two types of knowledge are always known initially: (i) the conditions of the collaboration (e.g. objective of the collaboration or objective of a business task for web service composition) and (ii) a repository of available services, which are intended to be composed. Then, supplementary information are required, either on (iii) the partners who already know they want to work together but don't know to formalize the corresponding process or (iii bis) the main sequence of activities not already assigned to any partner. In this world of research, OpenPaaS takes both (i) and (ii) as input, however there are no information about the (iii) or the (iii bis), and this is where the contribution of the process deduction service is developed in this paper.

An overview of the literature on service composition topic now shows the current limitations and the emergence of new relevant approach based on metaheuristics.

2.3. NP-complete problems, towards ant colony algorithm

As evoked in Wang et al. [44], new methods have emerged in the service composition field in order to optimize the deduced final result either based on local optimization, when the performances of each task is assessed (e.g. cost optimization, the costs of all the tasks can indeed be added to obtain the global cost), or based on global optimization (e.g. global delivery time of the business process, time assessment should take task parallelism or sequencing into account, and consequently the process should be entirely deduced before assessment). In the context of OpenPaaS, the second category of optimization is the most interesting, since the non-functional preferences of the broker can be based on very heterogeneous
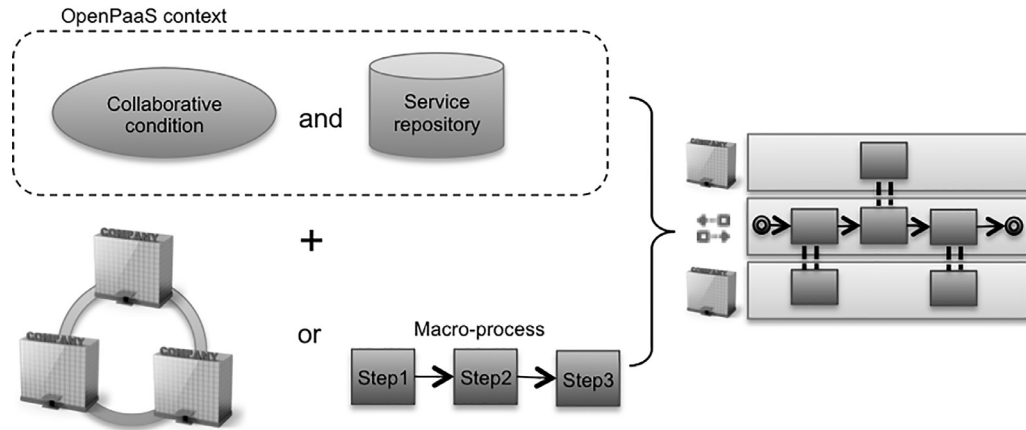
Fig. 2. Contributions of the OpenPaaS project in service composition field.

criteria including time. Moreover, in [37], Rosenberg at al. claimed that a metaheuristic is an "iterative generation process which guides a subordinate heuristic by combining different concepts for exploring and exploiting the search space" . They are particularly suitable for large combinatorial problems and allow to find quasi-optimal solution.

A brief overview on metaheuristics frequently applied to service composition positions our system and explains the authors' choice of a metaheuristics.

The Simulated Annealing (SA) and the Tabu Search (TS) are described as single-state methods by Luke [22]. The SA is inspired from annealing in metallurgy, which consists in controlling materials heating and cooling to obtain a better quality by increasing the size of the crystals (*i.e.* in terms of thermodynamics, lower the energy of the material to obtain a more stable state) [38]. According to Dréo et al. [12], the SA is known as a demanding method in terms of adjustments (*e.g.* to control the cooling), and can become very time-consuming, usually leading to parallel implementation. There is no notion of *memory*, which means that solutions cannot be based on previous simulations. In the literature, the SA can be associated with a Genetic Algorithm to obtain more efficient algorithms in service composition like in the works of Gao et al. [16] or with a TS in the works of Ko et al. [20]. The TS is based on human memory mechanisms. Schematically, it consists in exploring the neighborhood of an initial candidate solution, to find better candidates. All along an iteration, a solution can be chosen even if it is worse than the previous, in order to avoid local minima. Then, Dréo et al. [12] explain that a memory of the last explored solutions is kept, so that the mechanism is able to forbid them: it avoids studying an already retained solution (and local minima by the same time).

The Genetic Algorithm (GA) is considered as population-based and is a type of evolutionary algorithm [22]. Jaeger and Mühl [19] consider the GA as characterized through a loop within four phases as applied in QoS (Quality of Service) aware web service composition: the generation of a base population composed of various solutions from random combinations, the selection of an even number of solutions, the crossing of these chosen solutions to obtain new children solutions in the population and finally the mutation of individuals. This algorithm is applied both in Computer and Management Sciences, and for instance, Sha and Che [40] use it to select partners in a supply chain network design work. The limitations of this algorithm come from the fact that the crossing step only works if all the solutions (*i.e.* chromosomes) have the same number of individuals (*i.e.* genes). Consequently, the hypothesis that a macro-process exists enables to encode the chromosome structure from the process pattern. However, in our case, two business processes could have different number of tasks (depending on their granularity levels), and still fulfill the same collaborative objectives, thus they cannot be crossed together.

The Particle Swarm Optimization (PSO), firstly implemented for continuous problems, has then been adapted to discrete ones. Nayak et al. [28] and Zhao et al. [47] have for example used this kind of metaheuristics in VE paradigm, for the selection of partners. In both cases, the PSO is used on fixed graphs, in which the steps of the processes are already known. In addition, while the PSO seems to require strong adaptation for specific discrete graph, its particular adaptation to constrained AND/OR nodes constrained graphs seems quite laborious.

Ant colony algorithms (ACO) are particularly suitable for combinatorial optimization in a graph structure. Dorigo and Birattari [10] explain that, in theory, this type of metaheuristics can be applied to any discrete optimization problem for which some solution construction mechanism can be conceived. In the case of the composition of business process services composition, this is ideal.

2.4. Ant Colony Optimization's benefits and need of adaptation

Because of the ability of the ants to travel over a graph structure, the ACO seems quite suitable to exploit our knowledge base AND/OR nodes tree structure (cf. Section 3). The agents (*i.e.* ants) can thus be constrained in their "paths" so that they
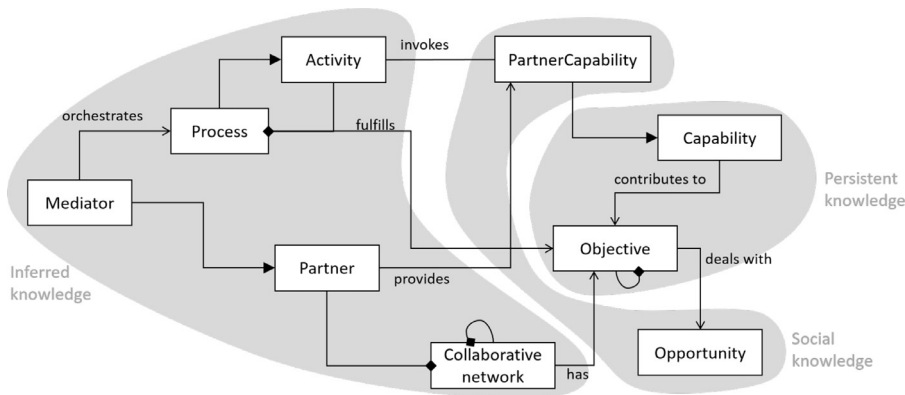
Fig. 3. Proposition of a metamodel to structure the knowledge-based system.

can explore the solutions space in a proper way. This brings a real strength when deducing collaborative processes, since it can be adapted to exploit the AND and OR nodes of the implemented collaborative knowledge base. It also allows obtaining candidate solutions by any completion criteria, and not only a fixed number of individual as in the GAs.

The ACO possibilities have been explored in various fields, such as decision trees exploitation [31] or also data-mining advances with, for example, the Ant-Miner algorithm, whose goal is to discover classification rules from gathered data [33]. However, the complex structure (i.e. a tree with AND and OR nodes) brings an unusual use of ACO. Wang et al. [43] detail such ACO, adapted to a directed acyclic graph. In that paper, the graph is actually a workflow and the goal of the ACO is to find a quasi-optimal services composition for that workflow (*i.e.* finding the best set of service candidates that fulfill each of the abstract services of the workflow). For this purpose, Wang et al. [43] defines the workflow as an AND/OR graph according to the AND/OR relationships between services and their successors. Ants begin their travel on the start event of the workflow, and then follow the workflow: when they meet AND nodes, they are replicated and when they meet OR nodes, they chose one of the path. Functionally speaking, such services composition makes the assumption that the workflow is already known. Hence, the selection is made "one-by-one" for each service of the workflow, which means that all the nodes (considering each node is a service) are always studied by any ants to find a candidate solution. However, in our context, candidate processes size cannot be known in advance: the tree knowledge graph depicted in Section 3 shows indeed that a collaborative opportunity can be answered at different levels of granularity (i.e. either at a high-level with turnkey solutions, or at a very low-level of granularity by for example dealing with each of the suppliers one-by-one), which leads to knowledge graphs to exploit, with variable complexity (i.e. more or less sequences of adjacent OR and AND nodes).

3. Knowledge representation and acquisition

3.1. Structure of the handled knowledge via a metamodel

In a first step, in order to structure the proposed knowledge-based system, a metamodel is proposed, as illustrated in Fig. 3. This metamodel is basically divided into three main packages:

- The Social Knowledge concerns the knowledge acquired by the system, thanks to modelers. It is composed of the emerging market Opportunity that users would like to answer and the different Partner Capabilities that are made available by the companies using the platform.
- The Persistent Knowledge is initially injected into the system, and composed of a decomposition of collaborative Objectives into Sub-Objectives and, for each of them, the set of Capabilities that should performed to achieve them.
- The Inferred Knowledge is about the knowledge that has to be ultimately deduced from the two previous kinds of knowledge. It thus focuses on the concepts related to the building of the final collaborative process. In this sense, on the one hand, it describes the collaborative network - composed of the partners of the collaboration - that will fulfill the Objectives, and on the other hand, how these partners' capabilities are turned into activities, that together shape the collaborative process that will be orchestrated by the Mediator.

As a consequence, it can be noted that the three types of knowledge rely on one another following this rule: the Social Knowledge is humanly brought inside the system, and builds on what is already known by the system thanks to the Persistent Knowledge and the Inferred Knowledge only can be deduced from both the Social and Persistent Knowledges. For this specific reason, the following parts are articulated as follows: (i) description of the Persistent Knowledge, (ii) description of the acquisition of Social Knowledge and (iii) description of the deduction of the Inferred Knowledge.
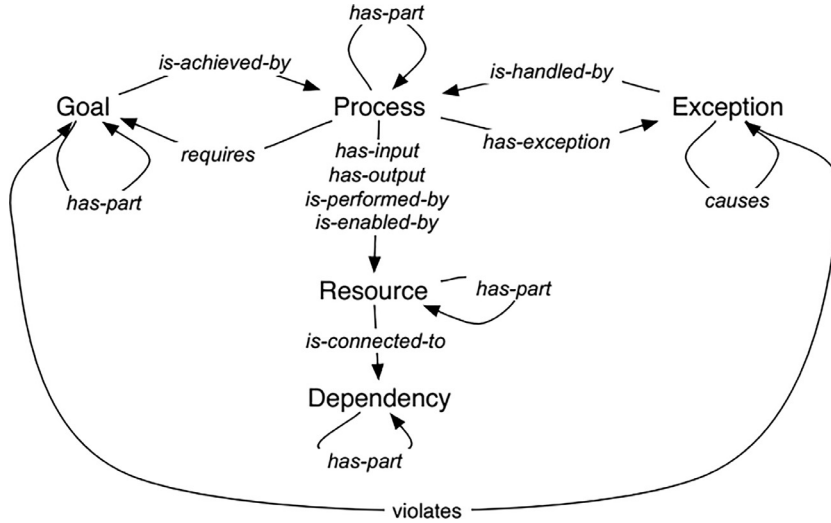
Fig. 4. MIT Process Handbook schema, [3].

Fig. 5. OpenPaaSs CO structure.

3.2. Persistent Knowledge

实现一个协作工作流，有两种方式，自上而下，或者自下而上

The choice of the structure of the Persistent Knowledge is based the recommendations found in the literature. In the field of cross-organizational collaborations, top-down and bottom-up approaches are both commonly used: they provide two different ways of achieving a collaborative workflow. In the first case business objectives should be decomposed into sub-objectives and/or business services, which are decomposed into technical services that could then be ordered into a sequence. In [39], Schulz and Orlowska recommend to use the two methods as follows: the top-down approach allows the coalition of organizations to build its common workflows by describing their interactions, while the bottom-up approach is rather suitable for each organization to specialize its tasks with its own private workflows. According to this recommendation and as preconized also by Ko et al. [21] the top-down approach seems highly suitable in the case of business objectives and capabilities decomposition as required for the business process deduction, that is why it has been chosen.

自上而下的方法允许组织联盟通过描述他们的交互来构建其共同的工作流程，而自下而上的方法更适合每个组织使用自己的私有工作流专门化其任务。根据此建议和由Ko等人预先确定。[21]自上而下的方法似乎非常适用于业务目标和功能分解的情况，这是业务流程扣除所必需的，这就是为什么选择它的原因。

3.2.1. Collaborative Ontology

The CO consists thus in Objectives of Collaboration that are decomposed into a set of required capabilities or into a set of sub-objectives, which is itself decomposed the same way. Now that the structure of the OpenPaaS Collaborative Ontology (CO) is defined it has been populated with adequate individuals. The CO has been entirely populated with the MIT Process Handbook [23]. In Fig. 4, it is interesting to note that the handbook incorporates *Goals* achieved by *Processes* that rely on *Resources* and also a decomposition of the *Goals* with the *has-part* reflective relationship. A very simple 1-to-1 transformation has been made from the *Goals* of the MIT Process Handbook to the *Objectives* of the CO, and from the *Processes* to the *Capabilities*. The associations between these classes have also been transformed as illustrated in Fig. 5: *is-achieved-by* becomes *isFulfilledBy* and *has-part* becomes *hasSubObjective*. The transformation allowed obtaining all the individuals of the OWL version of the MIT Process Handbook into individuals that fit the COs needs. As a last step, the simple inverse logic axioms (1) and (2) were propagated into the CO in order to obtain practical relationships for the further knowledge exploitation step (cf. Fig. 5).

$$isSubobjectiveOf = hasSubObjective^{-} \tag{1}$$

$$contributesTo = isFulfilledBy^{-} \tag{2}$$

3.2.2. Business Field Ontology

A Business Field Ontology (BFO) has been implemented in order to specify the COâs knowledge. The research was oriented toward large sources of information about business activities. Several classifications exist in this area, including the
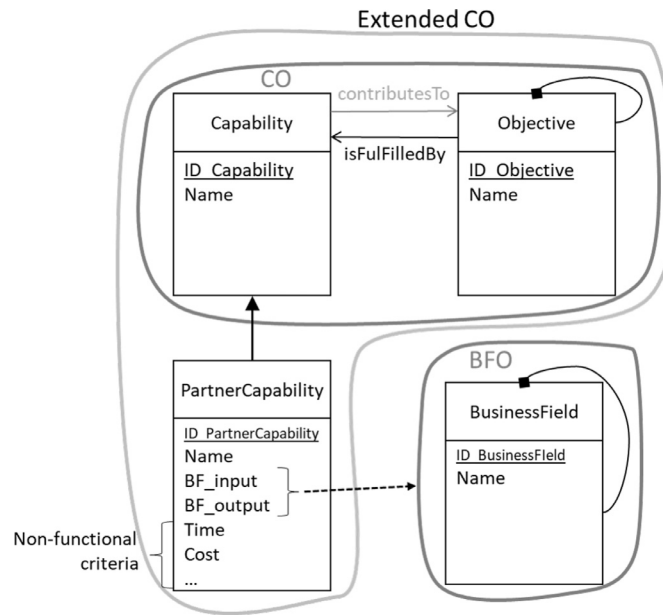
Fig. 6. Class diagrams of both the CO and the BFO and their use by the concept PartnerCapability.

Nomenclature of Economical Activities [14], the North American Industry Classification Systems [5] or the International Standard Industrial Classification of All Economic Activities (ISIC) established by the Statistics Division of the United Nations [42]. The ISIC decomposes business activities through 21 top-level activities, on four hierarchical levels and provides the largest scope of business activities since it provides a worldwide point of view, and not only a constrained geographical vision.

3.3. Social Knowledge

As an ESN, the first step for the user of OpenPaaS, is to describe himself via a Profile Modeler. The profile of an organization can be considered as a repository of all the capabilities it wants to share during collaborations supported by the PaaS.

It consists thus of a graphical interface called 'Profile Modeler' that allows the user to create instances of capabilities and link them to the IT knowledge (i.e. the ontologies). The description of each capabilities is based on the IDEF0 standard [15] with the central capability and the definition of its inputs and outputs. Concretely, the user creates semantic links between each capability and the capabilities available in the CO, and annotates its inputs and outputs by linking them with the BFO.

Then, an 'Opportunity Modeler' comes as a second graphical interface that helps the users characterizing their collaborative opportunities. It is based on the same semantic principle: the user creates an instance of objective of collaboration and provides a semantic link between it and an objective of the CO. Then the user specifies the opportunity by providing the business domains concerned by the collaboration with semantic links with the BFO.

Finally, Fig. 6 describes both the Class Diagrams for the CO and the BFO, and how they are used by when a user brings a new PartnerCapability thanks to the Profile Modeler. One can note that the Business Fields included in the BFO are only used as properties for the input and output of the PartnerCapability (and more broadly as a property of the Opportunity described in the Opportunity Modeler).

3.4. Logical structure of the knowledge, toward the deduction of collaborative process

At this point, the users have described the collaborative context by bringing (i) a repository of capabilities they all want to bring within collaborations and (ii) collaborative opportunities. This acquired knowledge is directly linked to the one injected initially in the CO and the BFO. As a result, a constrained knowledge graph is obtained. These constraints are due to the decomposition of the *Objectives* and *Capabilities* and can be expressed by simple logic rules, as follows:

- The decomposition from an *Objective* to its complementary *Capabilities* is modeled with an AND-node, since all the *Capabilities* should be performed to fulfill the *Objective* of the collaboration.
- The decomposition from an *Objective* to its complementary *SubObjectives* is modeled with an AND-node, since all the *SubObjectives* supersede together the high-level *Objective*.
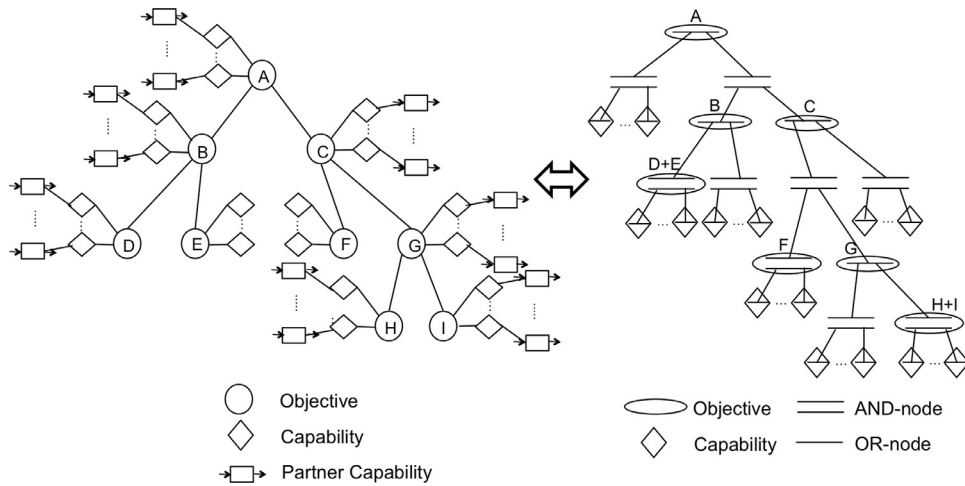
Fig. 7. Transformation from CO to a logical graph.

- The decomposition from a *Capability* of CO to its linked *PartnerCapabilities* is modeled with an OR-node, since only one organization should be chosen to execute the *Capability* in the process (in Fig. 7, the *Capabilities* have been depicted as OR-nodes, the *PartnerCapabilities* have not been represented in the logical graph to keep it readable).
- Each *Objective* that has a *SubObjective* is modeled as an OR-node, since the choice should be made between the set of *PartnerCapabilities* to directly fulfill it, or the set of *SubObjectives* that decompose it.
- Each *Objective* that has no *SubObjective* is modeled as an AND-node, since the only children consist in the set of the corresponding *Capabilities*.

Fig. 7 illustrates the transformation from CO to a logical graph, which is in fact the graph used by the ants of the ACO.

For instance, the *Objective* A is transformed into an OR-node that leads either to a final AND-node for all the *Capabilities*, either to an AND-node of two *SubObjectives* B and C. Since D and E are two final *SubObjectives* of the edge they are merged together into an AND-node of all their corresponding *Capabilities*.

In the remaining of this paper, the term of 'extended CO' refers to the CO plus the capabilities of organizations that have been linked to the CO. Therefore, the extended CO contains both permanent knowledge (the CO) and the 'living' knowledge arising from enterprises.

4. Exploitation of the knowledge graph with an ACO

The deduction of an inter-organizational collaborative process remains the critical part of the whole system. The challenge of the deduction algorithm described in this sub-section is to provide an 'optimal' cross-organizational business process. Here optimal is used to mention a business process that offers good compromise according to non-functional criteria as cost, delivery time quality, etc. The selection of all the cheapest complementary capabilities for an objective will definitely provide the cheapest process. However, it is obviously not the case for the time criterion for example because of the parallelism or the sequencing possibilities during the service composition (i.e. when the capabilities are ordered into a collaborative process).

That is why the ambition of this deduction algorithm is to exploit the minimal knowledge it holds from the two previous modelers, in order to simultaneously provide (i) a relevant set of partners, (ii) their quasi-optimal set of capabilities and (iii) the appropriate sequence of capabilities.

According to the famous maxim 'divide and conquer', the following parts allow to first describe the optimization problem and provide a main ACO algorithm, and then detail its three sub-algorithms (i.e. three steps of the ACO): the Exploration phase, the Construction phase and the Evaluation phase. Finally an improved version of the Exploration is proposed.

4.1. Problem description

A process can be described as a sequenced set of capabilities of organizations. Thus, this optimization has two main goals: (i)find a 'good' set of capabilities and (ii) deduce the sequences of capabilities to be able to assess the corresponding process.

In the following section, the variable *capa* is associated with the generic capabilities contained in the CO, whereas *oCapa* designates capabilities that have been proposed by organizations on the Profile Modeler.

```
 1: procedure MAIN-ACO(objective)
 2:     Initialize N the number of cycles, A the number of ants per cycle
 3:     Initialize pheromone on all the nodes of the extended CO with τ₀
 4:     repeat
 5:         Initialize S_p^k
 6:         for Ant=1 to A do
 7:             oCapaSet ← Exploration(objective)
 8:             Update local pheromone on each oCapa in oCapaSet, according
           to Equation 3
 9:             process ← Construction(capaSet, objectiveDomains)
10:             if process is feasible then
11:                 for all non-functional objectives k do
12:                     S_process ← Evaluation(process,nonFunctionalObjectives
           ,nonFunctionalObjectivesWeights )
13:                 end for
14:             end if
15:         end for
16:         Determine the best and second-best solutions
17:         Update global pheromone on the corresponding capaSet according to
           Equation4
18:     until non-functional objectives are fulfilled or N is reached
19: end procedure
```

Fig. 8. OpenPaaS's ACO main algorithm.

A set of capabilities is modeled as a vector of complementary capabilities of organizations $oCapaSet = (oCapa_1, \ldots, oCapa_n)$. The variable *objective* also corresponds to the collaborative objective proposed by a user on the Opportunity Modeler. The variable *process* is used to refer to the deduced process corresponding to the sequencing of *oCapaSet*.

Initially, the number of cycles *N* and the number of ants per cycle *A* are given. Each ant generated begins the algorithm with an empty *oCapaSet = null*. All the paths of the extended CO are also initialized with the same amount of pheromone τ_0. One ant goes through three parts:

- Exploration: the ant explores the extended CO, from the collaborative objective *objective* to the capabilities of organizations, according to specific constraints due to the structure of the extended CO. At the end of the Exploration, the local pheromone is updated (i.e. decreased for the diversification of further candidate solutions) for each parent branch of each *oCapa* of the chosen *oCapaSet*, according to the Eq. (3).
- Construction and feasibility: the previous set of capabilities of organizations is ordered. According to the input and output of each capability (i.e. the business domains), they are linked to each other in order to obtain a process. If this process is effectively deduced, then it shows that this process is feasible: it is a candidate solution.
- Evaluation: this potential solution is evaluated according to *k* non-functional objectives given in the Opportunity Modeler. For each potential *process* previously deduced, a vector $S_{process}^k$ is generated. Then the *k* evaluation are aggregated and finally $S_{process}$ represents the global evaluation for this potential solution. The pheromone on the visited branches of the two best solutions of each cycle evolves according to the Eq. (4) ((i.e. increases for the intensification of the visited paths).

Fig. 8 presents the corresponding main algorithm.
The pheromone evolves according to two steps:

- Local pheromone: each time a path of the graph is visited, its pheromone decreases according to:

$$\tau_{branch} = (1 - \rho) \cdot \tau_{branch} + \rho \cdot \tau 0 \qquad (3)$$

- Global pheromone: at the end of each cycle, the two best solutions are intensified, with an increasing of the pheromone of all its parent branches according to:

$$\tau_{branch} = (1 - \rho) \cdot \tau_{branch} + \rho \cdot \Delta\tau_{branch} \qquad (4)$$

with τ_{branch} the amount of pheromone on the *branch*, ρ the pheromone evaporation rate ($\rho \in [0; 1]$) and $\Delta\tau_{branch}$ as follows:

$$\Delta\tau_{branch} = \begin{cases} 10 & \text{if} & branch \in bestsolution \\ 5 & \text{if} & branch \in secondbestsolution \\ 0 & \text{if} & otherwise \end{cases} \qquad (5)$$

The three modules Exploration, Construction and Evaluation are the fundamental cornerstone of this ACO and each of the following sub sections are dedicated to them.

```
 1: procedure EXPLORATION(objective)
 2:     Initialize objectiveList with objective
 3:     Initialize listpCapa null
 4:     while objectiveList not empy do
 5:         for all currentObj in objectiveList do
 6:             Initialize listCapa null
 7:             children ← children nodes of currentObj
 8:             child ← chosen node in children, according to Equation6
 9:             if Type of child = 'Capability' then
10:                 Add listCapa ← child and its complementary capabilities
11:             else if Type of child = 'Objective' then
12:                 Add objectiveList ← child and its complementary objectives
13:             end if
14:             Remove currentObj from objectiveList
15:             for all capa in listCapa do
16:                 pCapa ← chosen node in children, according to Equation6
17:                 Add pCapa to listpCapa
18:             end for
19:         end for
20:     end while
21: end procedure
```

Fig. 9. OpenPaaS's ACO Exploration sub-algorithm.

4.2. Exploration

The Exploration algorithm (cf. Fig. 9) implementation is based on the logical graph of the extended CO (cf Fig. 7). *objectiveList* is initiated with *objective*, the *Objective* of the collaboration. All along the algorithm this list evolves with the decomposition into *SubObjectives*. The list *listpCapa* is null at the beginning and grows every time an ant chooses a *Partner-Capability*. First, the ant is dropped on *objective*, then it chooses a child node in CO, according to the Eq. (6): it can be an *Objective* or a *Capability*.

- If it is an *Objective*, this *SubObjective* and its complementary *Objectives* are kept and added to *objectiveList*, and as the parent *Objective* has been processed it is removed from *objectiveList*. The ant re-do this loop.
- In the case of a *Capability*, the child and its complementary *Capabilities* are kept in *listCapa* for further decomposition, and as the parent *Objective* has been processed it is removed from the *objectiveList*. Once *listCapa* contains *Capabilities* to execute, the ant needs to find partners able to provide each *Capability*. For one *Capability* of CO, many children *Partner-Capabilities* are able to provide it, but with different non-functional criteria. The ant chooses one of them, according to the usual Eq. (6).

The choice of the children node depends usually on two parameters η_{edge} the attractiveness of each edge going to each children, and τ_{edge} its amount of pheromone. The variable *edges* represents the whole set of children edges available, and *edge* one of them.

$$edge = \begin{cases} argmax_{u\in edges}[(\tau_u)^\alpha \cdot (\eta_u)^\beta] & \text{if} \quad q \le q_0 \\ J & \text{if} \quad q > q_0 \end{cases} \tag{6}$$

q is a random variable chosen between 0 and 1, and J is chosen according to the following probability distribution:

$$p_{edge} = \frac{(\tau_{edge})^\alpha \cdot (\eta_{edge})^\beta}{\sum_{h\in edges}[(\tau_h)^\alpha \cdot (\eta_h)^\beta]} \tag{7}$$

The attractiveness of an edge η_{edge} is found by applying the further Evaluation assessment explained in Section 4.4, applied to only one *Capability*. Given the non-functional *Objectives* $P = p_1,\ldots,p_n$ expressed and weighted by $W = w_1,\ldots,w_n$ the broker, the attractiveness of one edge is $\eta_{edge} = \sum_{i=1}^n (w_i \cdot p_i)$.

Obviously, this Formula can only work when the edge is positioned between a *Capability* and a *PartnerCapability* (case 1 in Eq. (8)) . For these reasons, α and β have been fixed as follows:

$$edge = \begin{cases} \alpha = 1 \text{and} \beta = 2 & \text{if} \quad case1 \\ \alpha = 1 \text{and} \beta = 0 & \text{if} \quad otherwise \end{cases} \tag{8}$$

Finally, the whole set of *PartnerCapabilities* is set up. The next step is to build the corresponding process.

4.3. Construction

The process to be deduced here must respect the BPMN 2.0 specification [30]. As stated in the literature review, the benefit of the second version of the BPMN Standard relies on its ability to be orchestrated. In such a perspective, the deduced process must contain a pool for each partner of the collaboration but also a MIS (Mediation Information System) pool whose goal is to interact via message flows with all the partners during the orchestration. The creation of the pool is indeed very easy and this is the very first step of the construction. Furthermore, the assumption is made that the parallel gateway is the unique gateway deduced, since the deduced process is 'the potential solution', meaning that the inclusive gateways is understood here as another process i.e. another potential solution.

Thus, the main idea here is to deduce the sequence of activities in the MIS pool, since the tasks in partner the pools only appear as atomic activities invoked all along the orchestration. In the Opportunity Modeler, when a user provides a new objective of collaboration, he/she always links it to one or more business domains (i.e. individuals of the BFO). As the first step, the Exploration ensures that the set of organizations capabilities can fulfill the objective, the remaining doubt relies on the fact that these capabilities can effectively work together to fit the domain(s) of the collaboration.

The authors partly addressed the construction step in [25] by an intuitive 'right-to-left' and 'left-to-right' processing. It is easily explained by the inputs of the algorithm the *objectiveDomains* and the set of organizations capabilities *capaSet*, which have to be ordered into the process: only the result of the process, which matches the objective of the collaboration, is known here. First of all, an EndEvent is created and added to *process*, the final process. For each *domain* in *objectiveDomains*, all the *capa* of *capaSet* are examined. As soon as an output of *capa* has the same domain as *domain, capa* is (i)added as a MIS task in the MIS pool of *process*, (ii) added as a partner Task in the corresponding partner pool, (iii) message flows are created between the MIS task and the partner task and (iv) the MIS task is linked with the EndEvent. *capa* is removed from *capaSet*, since it has now been used in the process. Then, this matching is made again in a loop, between the input of each new tasks in *process* and the output of remaining *capa* in *capaSet*.

Everytime a task is linked with several tasks, a gateway is created and added to *process*, and the corresponding tasks are consequently linked to it, instead of been linked to one another:

- Either a task can be linked with several previous tasks, then a 'closing' parallel gateway is created and linked with the subsequent task and all the antecedent tasks.
- Or a several tasks can be linked with several subsequent task which would provide output(s) for each of them. Then an 'opening' parallel gateway is created and linked with all the subsequent tasks and the antecedent task.

The process finishes when *capaSet* is empty: all the organizations capabilities have been used and ordered. However if *capaSet* remains non-empty, this means that the process is not feasible, and according to the main ACO algorithm (cf Section 4.1) it won't be part of the further Evaluation.

4.4. Evaluation

Now that the process has been built, and consequently fits the functional needs of the broker, it must now be evaluated according to the non-functional objectives $P = p_1, \ldots, p_n$ expressed and weighted by $W = w_1, \ldots, w_n$ in the Opportunity Modeler, and associated with *objective* the objective of the collaboration.

The method applied here is synthesized by Berrah et al. [4]. The expression of the performance of a system can be expressed in two basic ways, directly when the performance can be defined through a single expression and indirectly when the performance is the combination of several elementary expressions and especially in the case if these expressions do not have the same dimensions. The indirect expression of the global performance of a system is concretely applied by Cliville et al. [7], for instance.

Here the variable *process* represents the deduced process in the Construction step. The performances of *process* are evaluated according to the n non-functional dimensions and are stored in the vector $v = v_1, \ldots, v_n$. Then, these scores are normalized according to their corresponding non-functional objective and are calculated as follows: $s = \frac{v_1}{p_1}, \ldots, \frac{v_n}{p_n}$. Finally all these unitary elementary performances are aggregated according to the corresponding weights and the final global score is $S_{process} = \sum_{i=1}^{n} (s_i \cdot p_i)$.

This approach corresponds to Algorithm in Fig. 10.

4.5. Improved Exploration

It is to be noticed that as proposed in its first shape, the Exploration algorithm tends to make the ants exploring the 'high-level' paths: the probability for an ant is higher to choose a *Capability* of a high-level *Objective* than of a low-level *SubObjective*. For instance, on Fig. 11 the probability for an ant to choose a *Capability* of A is p, and the probability to choose a *Capability* of (H + I) is $w \cdot t \cdot q$. In the simple algorithm of Fig. 9, the weights $p, \ldots, w$ are all equal to 0.5. Thus the comparative chance to choose a *Capability* of A is 0.5 against 0.125 for (H + I).

We propose to balance the weights of all the edges established from OR-nodes (*i.e.* between *Objectives* and their *Capabilities* or their *SubObjectives*), except if the *Objective* is at the lowest-level decomposition in order to ensure a better

```
1:  procedure EVALUATION(process, P, W)
2:      Initialize S_process = 0
3:      n the size of P
4:      Initialize v and s
5:      for i = 1 to n do
6:          v_i ← Performance of process according to P_i
7:          s_i ← v_i/p_i
8:          S_process ← S_process + (s_i · p_i)
9:      end for
        return S_process
10: end procedure
```
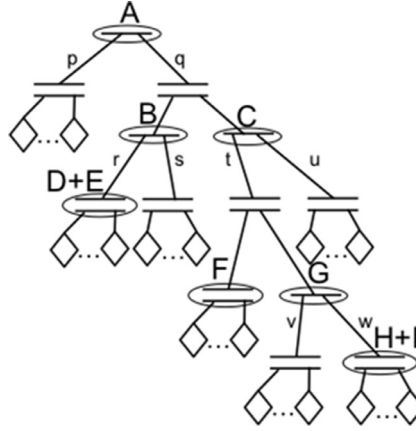
Fig. 10. OpenPaaS's ACO Evaluation algorithm.

Fig. 11. Weighted CO's logical graph.

equi-probability. As for each OR-node, there are always two edges possible: (i) the AND-node of a decomposition into *Capabilities* or (ii) the AND-node of a decomposition into *SubObjectives*, then, for any OR-node, exactly two children edges exist. The probability to take one or the other is 1, which means that the two children edges are complementary (for instance, $p + q = 1$).

As it comes to the rest of the equations to state, the approach begins with the lowest-level of *Objective i.e.* (H + I). The closest parent OR-node is G, from which the two children can be similarly chosen: $v + w = 1$. Then, in order to ensure the equi-probability, the two longest paths starting from each of the children OR-node edges should be also similarly chosen. Here, the second closest parent OR-node is C. Each of the children longest edges of C should be similarly chosen, which means that $u = t \cdot v$. In the same way, for the A OR-node, it can be stated that $p = q \cdot t \cdot v$. Finally, for the B OR-node, only two single-level edges exist, therefore $r = s$

This leads to the following equation system:

$$
\begin{cases}
p + q = 1 \\
r + s = 1 \\
t + u = 1 \\
v + w = 1 \\
r = s \\
v = w \\
u = t \cdot v \\
p = q \cdot t \cdot v
\end{cases}
\tag{9}
$$

The result, for this example is: $p = 1/4, q = 3/4, r = 1/2, s = 1/2, t = 2/3, u = 1/3, v = 1/2, w = 1/2$.

This approach has been generally applied to the whole ontology. Assuming that the *rank* of an *Objective* determines its maximum number of decomposition (for instance, $rank_G = 1$ since it can only be decomposed one time into *SubObjectives*, $rank_C = 2$ since it can be at most decomposed two times into *subObjectives* and $rank_B = 1$), then it is easily verifiable that:

- The probability coefficient of an edge to be chosen, between an *Objective* OR-node and its *SubObjectives* decomposition is $coefficient_{edge} = \frac{rank_{Objective}}{1 + rank_{Objective}}$.
- The probability of an edge to be chosen, between an *Objective* OR-node and its *Capabilities* decomposition is $coefficient_{edge} = \frac{1}{1 + rank_{Objective}}$.

Finally, these equi-probability coefficients are applied to the amount of pheromone of the corresponding edge. Thus the new pheromone amount of an edge $new\tau_{edge}$ used in Eq. (6) becomes $new\tau_{edge} = \tau_{edge} \cdot coefficient_{edge}$, with τ_{edge} the "basic" amount of pheromone and $coefficient_{edge}$ the equi-probability coefficient of the *edge*, in Eq. (6).

4.6. Performances analysis

In order to evaluate the performances of this ACO, it has been decided to assess the global score all along the iterations (*i.e.* ants cycles). For this purpose, different scenarios were set up by varying some of the parameters. First, the four chosen scenarios are described, and then the performance of the ACO on each of them are detailed.

4.6.1. Test scenarios

The scenarios have been chosen so that the behavior of the ants could be assessed, depending on two main parameters: the number of decomposition of *Objectives* into *SubObjectives*, which will be called *nbObjLevels* and the number of *PartnerCapabilities* available for each *Capability* of the extended CO (ECO), which corresponding variable is *nbPartnerCap*. The number of *Capabilities* per *Objective* has been set to 4, and the number of *SubObjectives* per *Objective* is also 3.

Hence, four ECOs have been generated along these parameters:

- ECO#1: *nbObjLevels* = 3 and *nbPartnerCap* = 10.
- ECO#2: *nbObjLevels* = 3 and *nbPartnerCap* = 20.
- ECO#3: *nbObjLevels* = 5 and *nbPartnerCap* = 10.
- ECO#4: *nbObjLevels* = 5 and *nbPartnerCap* = 20.

As a non-functional *Objective*, the top-level *Objective* of each ECO was always chosen - which is consistent with the choice of *nbObjLevels*. As it comes to the non-functional objectives, they depend on two non-functional criteria, which are the *global cost* and the *global time of delivery* of the solution. Both objective values have been selected so that they could not be reached, in order to evaluate the convergence of the ACO, with a weight of 0.8 for the cost and 0.2 for the time.

All the *PartnerCapabilities* have been generated with the same input and output flows, as such, even if the calculation of the global time of delivery of a process corresponds to the maximum time of the *PartnerCapabilities*, the time assessment algorithm is still the same: it has no impact on the evaluation of each candidate solution, but it is quite convenient to generate viable (and consequently assessable) processes.

Based on the simulations on different cases and also on the observations from ACO precursors Dorigo et al. [11], the related works of Dréo et al. in [12] and the results of Doerner et al. [9], the various parameters have been fixed:

$$\begin{cases} number\,of\,cycles\,N = 100 \\ number\,of\,ants\,per\,cycle\,A = 10 \\ \rho = 0.1 \\ \tau_0 = 0.1 \\ q_0 = 0.7 \end{cases} \tag{10}$$

4.6.2. Results

As a result, for each of the four previously stated scenarios, 10 simulations have been done on the same native collaborative ontology, and the average fitness values of the best process deduced all along the cycles have been calculated. In parallel, it has been compared to the best known solution, found along the corresponding mathematical model, in order to assess the quality of the process found by the ACO.

It should be noted that if all possibilities had been evaluated, the evaluation and construction algorithms of the ACO should have been invoked more than 10^{36}, 20^{36}, 10^{60}, 20^{60} times respectively for ECO#1, ECO#2, ECO#3 and ECO#4.

The whole system has been implemented using Java, whilst the ontologies have been developed as graph databases, using Neo4j [29] and exploited with its corresponding Java APIs. The evaluation tests have been executed on a computer embedding a i5 processor, 8Gb of memory.

Fig. 12 depicts the average evolution of the best candidate solutions found for each cycle, on 100 cycles along the 10 simulations that have been led on the same ECO. As a result, a decent convergence is observed from the 80th cycle in each of the evaluated cases. Nevertheless, due to the exploding combinatory of ECO#3 and ECO#4, it can be noted that logically the convergence occurs more slowly than for ECO#1, ECO#2, along the 100 cycles.

Table 1 summarizes the results observed all along the simulation. In average, the time to perform 100 cycles to explore the ECOs are the following: 17 min for ECO#1, 18 min for ECO#2, 23 min for ECO#3 and 26 min for ECO#4 with respectively average reached accuracies of 107%, 103%, 114% and 109% with regard to the best respective solutions.

4.6.3. Complexity analysis

More interesting, because independent of the hardware and software settings, a short analysis of the complexity of the whole algorithm can be performed. Overall, the Eq. (11) corresponds to the main algorithm complexity according to N the number of cycles, A the number of ants per cycle and the complexity of the sub-algorithms.

$$Complexity = N * A * (C_{Exploration} + C_{Construction} + C_{Evaluation}) \tag{11}$$
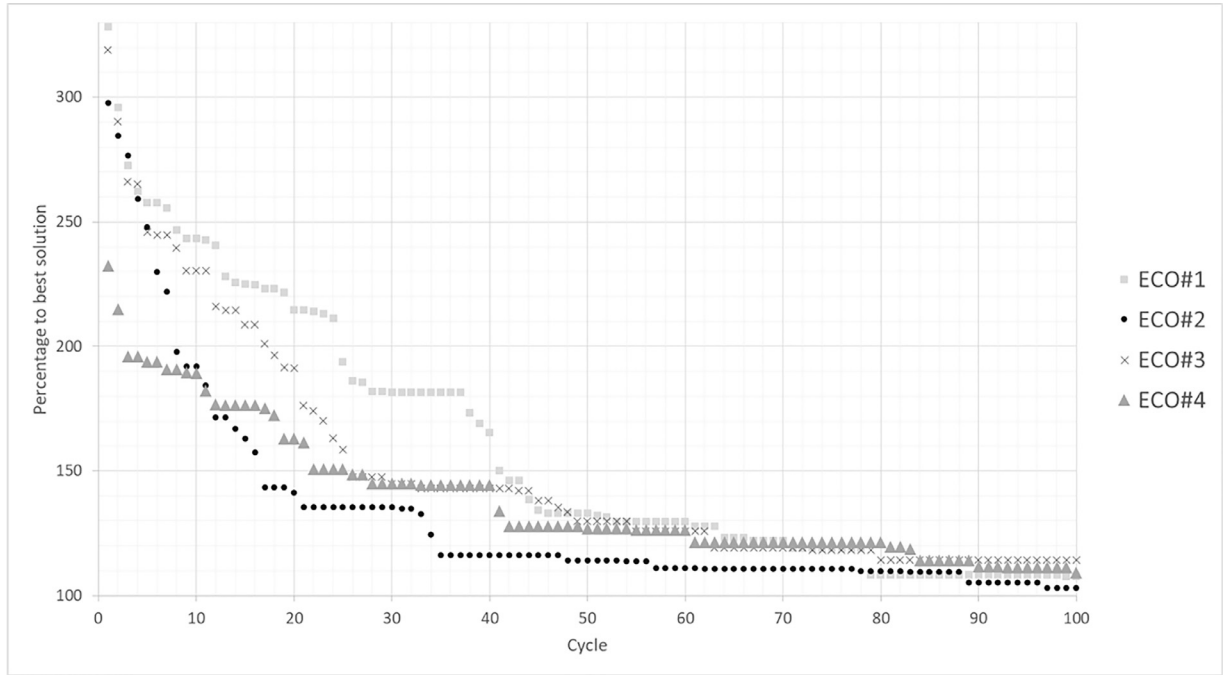
Fig. 12. Average evolution of the best candidate solutions found at each cycles, on 100 cycles, 10 simulations.

Table 1
Summary of the results observed for the three use cases.

	Accuracy (%)				Time at 100 cycles (min)
	25 cycles	50 cycles	75 cycles	100 cycles	
ECO#1	193	133	119	107	17
ECO#2	135	114	110	103	18
ECO#3	158	129	118	114	23
ECO#4	150	127	121	109	26

The complexity of the Exploration sub-algorithm is the highest among the three sub-algorithm, and thus is the single one analyzed further. In the worst case possible (i.e. the Exploration reaches the "bottom" of the ECO), Eq. (12) corresponds to the estimation of the complexity for this sub-algorithm. The following variables are used: *nbObjLevels* the number of levels of decomposition of objectives, *nbObjPerLevel* the number of objectives per level of decomposition and *nbCapaPerObj* the number of Capability per Objective.

$$C_{Exploration} = nbObjPerLevel^{nbObjLevels} * nbCapaPerObj \tag{12}$$

Finally a good estimation of the complexity (in the worst case) is:

$$Complexity = N * A * nbObjPerLevel^{nbObjLevels} * nbCapaPerObj \tag{13}$$

As an order of comparison, the algorithm to find the exact best solutions has a much higher complexity, which estimation is:

$$Complexity = nbPartnerCap^{nbObjPerLevel^{nbObjLevels}*nbCapaPerObj} \tag{14}$$

As explained in Section 2, very few works have addressed the question of simultaneous partners selection and process composition. The problem being built on an unusual structure of ECO, no other metaheuristic seems to be applicable directly, without thinking about such adaptation (just as proposed for the ACO). It will then be interesting in further works to build other suitable algorithms and compare their complexity and ultimately find the best alternatives depending on the initial complexity of the ECO to be exploited.

5. Illustrative case

The OpenPaaS' ACO has been implemented as described in the solution proposal of this paper. This Section introduces the human experience on OpenPaaS' modelers and shows a concrete result of the business service deduction, all along a
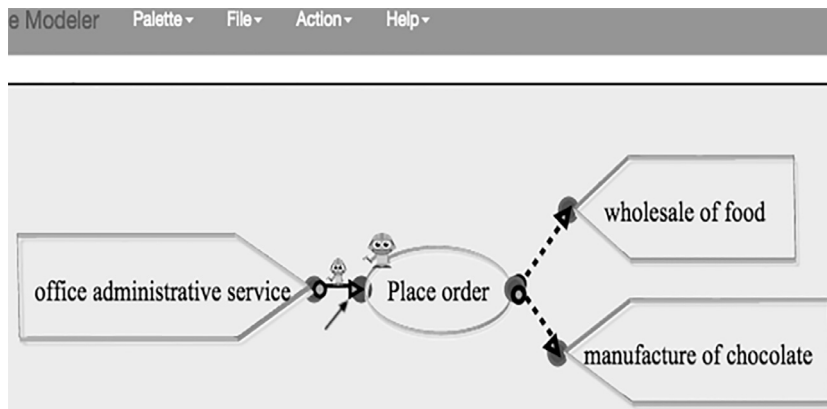
Fig. 13. Profile Modeler, modeling of a capability.

Fig. 14. Profile Modeler, details of the capability description.

short use case. Thus as a first part, the collaboration is defined via the use of the previously evoked Profile Modeler and Opportunity Modeler. The user-transparent ACO algorithm then deduces the process, which is written in a file that can be opened in a Collaborative Process.

Brake France is a French food wholesaler company for enterprises and this use case focuses on a particular bid for chocolate products.

5.1. Profiles of organizations modeling

As a first step, each of the organizations of the platform uses the Profile Modeler in order to create their own profile by detailing the capabilities they are able to provide and want to share in collaborative contexts. Fig. 13 shows the main interface of the Profile Modeler. Since it is graphical, the Profile Modeler is very intuitive for the user. Here, Brake France enterprise creates a new capability that is 'Place Order', since it is one of its main capabilities as a wholesaler. Then the input and output are detailed. The input concerns the '8211 - Combined office administrative service activities' class of the ISIC classification, whereas two domains have been chosen for the output: '1073 - Manufacture of cocoa, chocolate and sugar confectionery' and '4630 - Wholesale of food, beverages and tobacco'. Thus the first output specifies the fact that the 'Place Order' capability concerns the ability of buying from factories of chocolate, and the second one explains that the orders actually correspond to wholesale.

These links between the model and the CO and BFO are done by double-clicking on each instance. Then box opens with all the possible details, as in Fig. 14. Here the 'Place Order' capability is linked with the corresponding individual in the CO, then the price and delivery time are defined (several other non-functional criteria are also available but cannot appear here because of a lack of space).
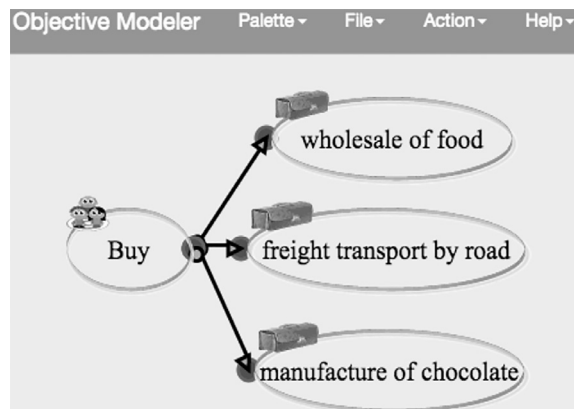
Fig. 15. Opportunity Modeler, modeling of a collaborative objective.

Fig. 16. Opportunity Modeler, details of the objective description.

5.2. Collaborative objective description

Similarly, the Opportunity Modeler provides a graphical interface for the users to express their opportunities of collaboration. Here, in Fig. 15, Brake France would like to obtain a collaborative process that fulfills a 'Buy' objective. The users can link as many domains as needed to specify the business fields of the collaboration. For instance, the 'Buy' objective is linked with three domains: '4630 - Wholesale of food, beverages and tobacco','1073 - Manufacture of cocoa, chocolate and sugar confectionery' and '4923 - Freight transport by road'. The first two domains specify the context of the buying objective, whereas the third domain concerning transport expresses another domain that Brake France would like to take into account. This means that they also want a delivery 'flow' in the process.

As in the Profile Modeler, when double-clicking on the instances they can be detailed. In Fig. 16, Brake France links the 'Buy' objective with the corresponding individual of the CO, and express its non-functional objectives.

5.3. Deduction of the collaborative process

As it can be seen in Fig. 15, a main menu at the top of the Modeler provides a 'Action' list. In this list, there is a 'Deduce process' button. When clicked, this launches the ACO: ants are dropped on the collaborative objective of the objective model just designed. Within less than ten minutes, they are able to provide a corresponding cross-organizational business process. A third Collaborative Process Modeler has been implemented to easily observe this process, as illustrated in Fig. 17.
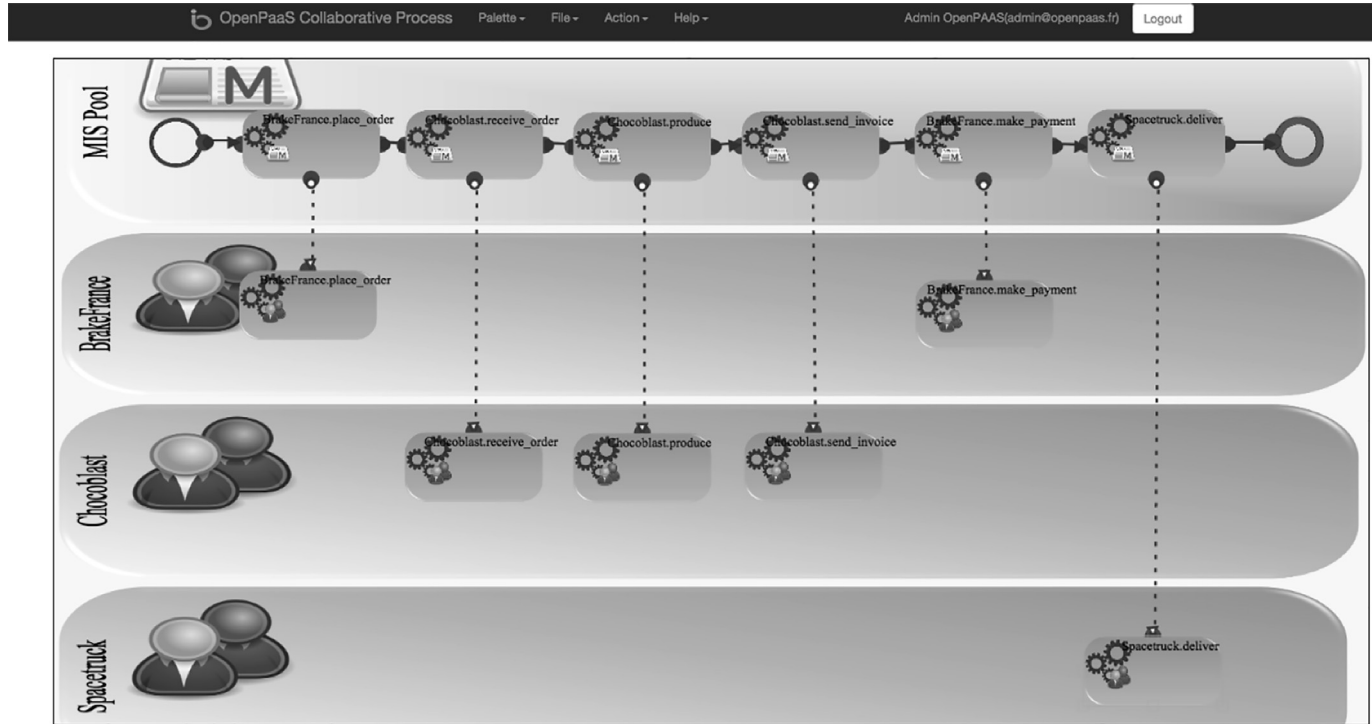
Fig. 17. Collaborative Process Modeler, result of the process deduction service.

As explained in Section 4.3, a central pool is created: the MIS Pool. It corresponds to the pool of the Mediation Information System. This pool holds the sequenced part of the process, corresponding to the result of the ordering of all the organizations capabilities that have to be executed. They are linked with sequence flow. Then message flows between the MIS pool and the partners' pools illustrate the invocation of each partner task.

This process shows that the final 'best' process found by the ACO concerns a collaboration of Brake France with the two companies 'Chocoblast', which is involved in all the tasks concerning chocolate manufacturing and 'Spacetruck' that offers transport services.

6. Discussions and conclusion

This paper presents a decision support system based on an ACO algorithm to automatically elaborate collaborative business processes based on only two inputs: (i) a repository of enterprises' profiles and their business services and (ii) a collaborative opportunity. This system is based on two parts: (i) 'flat' knowledge bases that gather the information about collaborations and (ii) an ACO algorithm to exploit this knowledge. The knowledge bases have actually been implemented as ontologies: a Collaborative Ontology brings the generic knowledge about collaborations and a Business Field Ontology allows to specify them. These ontologies are first used for describing the collaborative contexts through two modelers: (i) the Profile Modeler helps the companies to define their capabilities and (ii) the Opportunity Modeler is an interface to propose new opportunities of collaboration.

From profiles and objectives of collaboration, an innovative ACO can exploit the knowledge contained in the ontologies and deduce a non-functionally (e.g. according to cost, delivery time...) optimized collaborative business process. Three gaps are crossed in one time: (i) finding the required capabilities to fulfill the objective of collaboration, (ii) finding the 'best' set of partners able to provide them and (iii) order these capabilities into a collaborative business process.

This process deduction service addresses the current lack of flexibility of such systems through the establishment of different innovative parts of the solutions proposed in this paper:

- Knowledge bases have been structured and implemented as datagraph using Neo4J technologies [29] in the system. The Business Field Ontology presents a quite large knowledge base that is intended to fit any business activities. The Collaborative Ontology can be noticed for its top-down structure that allows a very intuitive decomposition of collaborative objectives into sub-objectives and capabilities. These ontologies are used as knowledge bases and provide individuals linked together according to their meta-models. The knowledge bases do not embed any 'intelligence', this means that the exploitation algorithm is only based on the structure of the ontology but has been implemented independently. As a consequence, any ontologies respecting the right meta-model can be used in the collaborative process deduction system.
- An ACO has been applied to exploit these ontologies and deduce collaborative process. The strength of this algorithm is to be able to provide in the same time (i) the required capabilities to answer the collaborative objectives, (ii) a set of corresponding partners able to provide these capabilities according to expected non-functional criteria and (iii) order these capabilities into a collaborative process. This ability is what distinguishes this ACO from most of the deduction of collaborations in the literature. Moreover the proposed ACO proposes the originality to fit 'hierarchical tree graph' structures with AND and OR nodes.

On the one hand, some improvements could be brought to the ACO itself. In further research, ideas to improve the quality of the output solutions could be based on works such as those of Zhong et al. [48] who propose an hybrid algorithm integrating GA and ACO in partner selection problems in VEs contexts and claim to obtain better solutions than a GA. Nevertheless, the results brought in our approach seem promising because of the original adaptation of the ACO to a particular problem that has not been addressed a lot in the literature (i.e. business process deduction without knowing neither the partners nor the steps to be executed). Such hybridization should be carried out with care because of the unusual structure of this problem.

On the other hand, a work could be done on the user experience with such platform. In this perspective, two aspects could offer a better experience and fit a larger scope of collaborations: (i) the linkage between the user's information and (ii) the transformation of any structured collaborative ontologies to the CO's meta-model. The research works of Wang et al. in [45] focus on the creation of a syntactico-semantic reconciliation that allows first to deduce common meanings between several words, and based on that to align ontologies. At a higher level, with this ontology-alignment promise, any ontologies could be uploaded and used for the deduction of collaborative processes. Even more interesting, the new added ontologies could be easily linked and merged to the already existing so that the system would finally holds a growing Collaborative Ontology. With such an agile and increasing knowledge base, the platform could support an always wider scope of types of collaborations, which is basically impossible due to the complexity of gathering all this knowledge from scratch.

Acknowledgments

References

[1] F. Benaben, N. Boissel-Dallier, H. Pingaud, J.-P. Lorre, Semantic issues in model-driven management of information system interoperability, Int. J. Computer Integr. Manuf. 26 (11) (2013) 1042–1053.

[2] F. Benaben, W. Mu, N. Boissel-Dallier, A.-M. Barthe-Delanoe, S. Zribi, H. Pingaud, Supporting interoperability of collaborative networks through engineering of a service-based Mediation Information System (MISE 2.0), Enterp. Inf. Syst. 9 (5-6) (2015) 556–582.

[3] A. Bernstein, M. Klein, H.J. Lee, C. Kiefer, Mit Process Handbook Ontology Schema, 2008 (https://files.ifi.uzh.ch/ddis/oldweb/ddis/fileadmin/ph/rocessHandbook-Schema-16-10-06.pdf). (Accessed on 04/05/2016).

[4] L. Berrah, G. Mauris*, F. Vernadat, Information aggregation in industrial performance measurement: rationales, issues and definitions, Int. J. Prod. Res. 42 (20) (2004) 4271–4293.

[5] U.S.C. Bureau, North american industry classification system (naics), 2015 (http://www.census.gov/eos/www/naics/).

[6] G. Canfora, M. Di Penta, R. Esposito, M.L. Villani, A lightweight approach for qos-aware service composition, in: Proceedings of 2nd International Conference on Service Oriented Computing (ICSOC04), 2004.

[7] V. Cliville, L. Berrah, G. Mauris, Quantitative expression and aggregation of performance measurements based on the macbeth multi-criteria method, Int. J. Prod. Econ. 105 (1) (2007) 171–189.

[8] J.A. Crispim, J.P. de Sousa, Partner selection in virtual enterprises, Int. J. Prod. Res. 48 (3) (2010) 683–707.

[9] K. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, C. Stummer, Pareto ant colony optimization: a metaheuristic approach to multiobjective portfolio selection, Ann. Oper. Res. 131 (1–4) (2004) 79–99.

[10] M. Dorigo, M. Birattari, Ant Colony Optimization, in: Encyclopedia of Machine Learning, Springer, 2010, pp. 36–39.

[11] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, Evol. Comput. IEEE Trans. 1 (1) (1997) 53–66.

[12] J. Dréo, A. Petrowski, P. Siarry, E. Taillard, Metaheuristics for Hard Optimization: Methods and Case Studies, Springer Science & Business Media, 2006.

[13] S. Dustdar, W. Schreiner, A survey on web services composition, Int. J. Web Grid Serv. 1 (1) (2005) 1–30.

[14] Europische KommissionStatistisches Amt, NACE Rev. 2 Statistical Classification of Economic Activities in the European Community, 2008. Luxemburg.

[15] Federal Information Processing Standards Publications, Integration Definition for Function Modelling (IDEF0), 1993.

[16] Z.-P. Gao, C. Jian, X.-S. Qiu, L.-M. Meng, Qoe/qos driven simulated annealing-based genetic algorithm for web services selection, J. China Univ. Posts Telecommun. 16 (2009) 102–107.

[17] A. Garcia-Crespo, B. Ruiz-Mezcua, J. Lopez-Cuadrado, I. Gonzalez-Carrasco, Semantic model for knowledge representation in e-business, Knowl. Based Syst. 24 (2) (2011) 282–296.

[18] GS1U., Rosettanet Standard, 2015.

[19] M.C. Jaeger, G. Mühl, Qos-based selection of services: the implementation of a genetic algorithm, in: Communication in Distributed Systems (KiVS), 2007 ITG-GI Conference, VDE, 2007, pp. 1–12.

[20] J.M. Ko, C.O. Kim, I.-H. Kwon, Quality-of-service oriented web service composition algorithm and planning architecture, J. Syst. Software 81 (11) (2008) 2079–2090.

[21] R.K.L. Ko, E.W. Lee, S.G. Lee, Business-OWL (BOWL)-a hierarchical task network ontology for dynamic business process decomposition and formulation, IEEE Trans. Serv. Comput. 5 (2) (2012) 246–259, doi:10.1109/TSC.2011.48.

[22] S. Luke, Essentials of Metaheuristics, Second, Lulu, 2013. Available for free at http://cs.gmu.edu/~sean/book/metaheuristics/.

[23] T.W. Malone, K. Crowston, G.A. Herman, Organizing Business Knowledge: The MIT Process Handbook, MIT press, 2003.

[24] M.D.R.-Moreno, P. Kearney, Integrating {AI} planning techniques with workflow management system, Knowl. Based Syst. 15 (56) (2002) 285–291. https://doi.org/10.1016/S0950-7051(01)00167-8.

[25] A. Montarnal, A.M.B. Delanoe, F. Bénaben, M. Lauras, J. Lamothe, A paas to support collaborations through service composition, in: Services Computing (SCC), 2014 IEEE International Conference on, IEEE, 2014, pp. 677–684.

[26] W. Mu, F. Bénaben, H. Pingaud, A methodology proposal for collaborative business process elaboration using a model-driven approach, Enterp. Inf. Syst. 9 (4) (2015) 349–383.

[27] W. Mu, F. Bénaben, H. Pingaud, Collaborative process cartography deduction based on collaborative ontology and model transformation, Inf. Sci. 334 (2016) 83–102.

[28] N. Nayak, K. Prasanna, S. Datta, S. Mahapatra, S. Sahu, A novel swarm optimization technique for partner selection in virtual enterprise, in: Industrial Engineering and Engineering Management (IEEM), 2010 IEEE International Conference on, IEEE, 2010, pp. 1118–1122.

[29] Neo4J, Neo4j: The World's Leading Graph Database, 2016 (http://neo4j.com/). (Accessed on 04/13/2016).

[30] OMG, Business Process Model and Notation (BPMN) Version 2.0, 2011.

[31] F.E. Otero, A.A. Freitas, C.G. Johnson, Inducing decision trees with an ant colony optimization algorithm, Appl. Soft Comput. 12 (11) (2012) 3615–3626.

[32] M.P. Papazoglou, Service-oriented computing: concepts, characteristics and directions, in: Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on, IEEE, 2003, pp. 3–12.

[33] R.S. Parpinelli, H.S. Lopes, A. Freitas, et al., Data mining with an ant colony optimization algorithm, Evol. Comput. IEEE Trans. 6 (4) (2002) 321–332.

[34] H. Pingaud, Rationalité du Développement de Linteropérabilité dans les Organisations, Management des Technologies Organisationnelles, 2009, pp. 19–30.

[35] P. Poizat, G. Salaün, Checking the realizability of bpmn 2.0 choreographies, in: Proceedings of the 27th Annual ACM Symposium on Applied Computing, ACM, 2012, pp. 1927–1934.

[36] J. Rao, X. Su, A survey of automated web service composition methods, in: Semantic Web Services and Web Process Composition, Springer, 2005, pp. 43–54.

[37] F. Rosenberg, M. Muller, P. Leitner, A. Michlmayr, A. Bouguettaya, S. Dustdar, Metaheuristic optimization of large-scale qos-aware service compositions, in: Services Computing (SCC), 2010 IEEE International Conference on, IEEE, 2010, pp. 97–104.

[38] R. Rutenbar, et al., Simulated annealing algorithms: an overview, Circuits Devices Mag. IEEE 5 (1) (1989) 19–26.

[39] K.A. Schulz, M.E. Orlowska, Facilitating cross-organisational workflows with a workflow view approach, Data Knowl. Eng. 51 (1) (2004) 109–147.

[40] D. Sha, Z. Che, Supply chain network design: partner selection and production/distribution planning using a systematic model, J. Oper. Res. Soc. 57 (1) (2006) 52–62.

[41] P. Tang, G.Q. Shen, Decision-making model to generate novel emergency response plans for improving coordination during large-scale emergencies, Knowl Based Syst 90 (2015) 111–128. https://doi.org/10.1016/j.knosys.2015.09.027.

[42] United NationsStatistical Division, International Standard Industrial Classification of All Economic Activities (ISIC), United Nations, New York, 2008.

[43] L. Wang, J. Shen, G. Beydoun, Enhanced ant colony algorithm for cost-aware data-intensive service provision, in: Services (SERVICES), 2013 IEEE Ninth World Congress on, IEEE, 2013, pp. 227–234.

[44] L. Wang, J. Shen, J. Yong, A survey on bio-inspired algorithms for web service composition, in: Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on, IEEE, 2012, pp. 569–574.

[45] T. Wang, S. Truptil, F. Benaben, Semantic approach to automatically defined model transformation, in: Model-Driven Engineering and Software Development (MODELSWARD), 2014 2nd International Conference on, IEEE, 2014, pp. 340–347.

[46] G. Wiederhold, M. Genesereth, The conceptual basis for mediation services, IEEE Expert 12 (5) (1997) 38–47.

[47] Q. Zhao, X. Zhang, R. Xiao, Particle swarm optimization algorithm for partner selection in virtual enterprise, Progr. Nat. Sci. 18 (11) (2008) 1445–1452.

[48] Y. Zhong, L. Jian, W. Zijun, An integrated optimization algorithm of ga and aca-based approaches for modeling virtual enterprise partner selection, ACM SIGMIS Database 40 (2) (2009) 37–56.