Semi-automatic Process Composition via Semantics-enabled Sub-process Selection and Ranking

Fabrizio Smith1, Devis Bianchini2

[1] National Research Council, IASI “Antonio Ruberti”, Viale Manzoni 30, 00185, Rome, Italy
fabrizio.smith@iasi.cnr.it,

[2] University of Brescia, Dept. of Information Engineering, via Branze 38, 25123, Brescia, Italy
bianchini@ing.unibs.it

1 Introduction

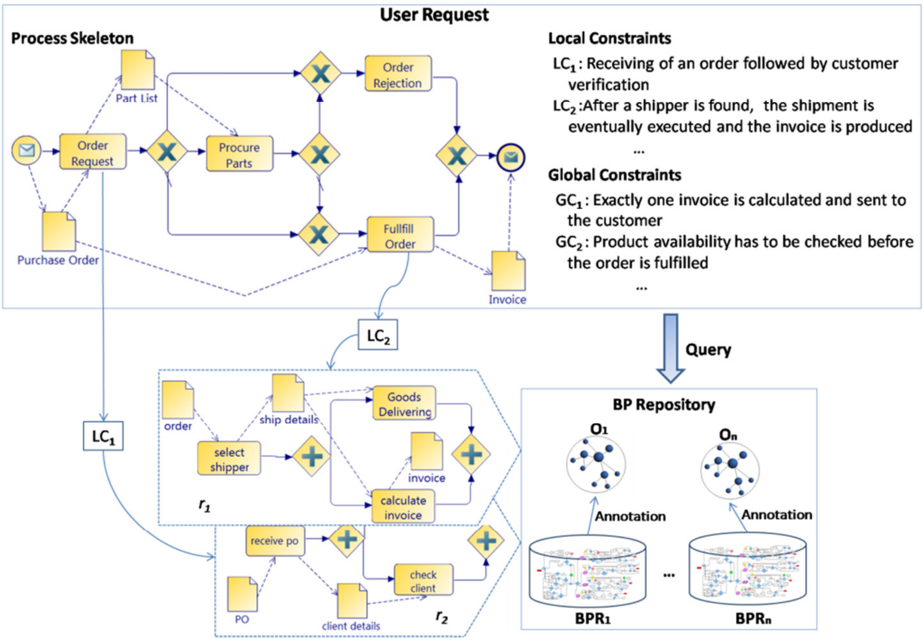
The strong acceleration towards new forms of enterprises, such as virtual and networked enterprises, requires efficient tools for the integration and composition of Business Processes (BPs) designed and deployed by different organizations. In this context, the notion of a Service Oriented Architecture (SOA) plays a key role, since Web Services allow packaged functionalities to be offered as a suite of interoperable services, platform-independent and autonomously implemented. In a Service Oriented Architecture [1] an orchestration is described as a Business Process schema, i.e., a workflow graph that specifies the planned order of operations execution. A BP is hence built from an orchestration of component sub-processes, often deployed and exposed as platform-independent Web services, each of which performs a well-defined activity within the process and can be shared across different enterprises to incentivate its reuse. Each Web service can be in turn defined as an orchestration of other Web services. Service-oriented computing offers then the opportunity of defining cross-enterprise, global BPs, to virtually form a single logical system.

In this paper we propose a methodological framework and a suite of tools that support the design of a BP, though the orchestration of sub-processes, exposed as Web services in a dynamic and distributed environment. Each organization publishes its Web services on its own repository. A business expert queries the repositories in order to retrieve fragments to be used in the design of a new global BP, specifying features and properties that the retrieved artefacts must exhibit.

In our proposal we make an extensive use of semantic techniques for the enrichment of business process modeling, in order to i) overcome heterogeneities deriving from the adoption of different semantic models; ii) take advantage of reasoning capabilities, in particular for querying process repositories; iii) enabling advanced matching and analysis techniques that require domain knowledge provided by computational ontologies. The starting point is not a black-box goal to be satisfied, but a BP specification that is made by an expert designer, thus partially reducing performance limitations and complexities of traditional Web service composition approaches.

2 Motivating Scenario

As a motivating example, we consider the scenario sketched in Figure 1. In order to implement a process for handling a purchase order, a business designer has to identify components (atomic or composite) from a Business Process Repository (BPR), assembling such fragments according to the given requirements. Such requirements derive from the functionalities that the resulting process must provide, but also from additional criteria that should be considered in order to enforce the quality of the designed process (e.g., appropriate granularity level, components cohesion/coupling).



Local Constraints

LC1: Receiving of an order followed by customer verification

LC2: After a shipper is found, the shipment is eventually executed and the invoice is produced

...

Global Constraints

GC1: Exactly one invoice is calculated and sent to the customer

GC2: Product availability has to be checked before the order is fulfilled

...

Fig. 1. Motivating scenario

As shown in the picture, the user's request is composed of the process skeleton, the local constraints and the global constraints. The process skeleton constitutes a high level definition of the process to be assembled, where tasks represent placeholders for the components to be retrieved. Local constraints are associated to each task of the process skeleton (e.g., LC1 to the *OrderRequest* task) to describe the properties that the corresponding retrieved component must exhibit. Finally, global constraints capture requirements of the resulting composite process as a whole. Once the user's request has been formulated, it is matched against the BPR to select the appropriate components (e.g., r_1, r_2 in the picture) and associate them to the tasks of the process skeleton. We assume a service-oriented perspective, where components are exposed as services and the final global BP will be implemented as a service orchestration.

In our framework, domain ontologies are used to semantically characterize BP elements. However, constraining the adoption of a unique ontology is not feasible and, hence, we propose techniques to handle situations where BPs are annotated with distinct ones, as in the scenario of Figure 1, where the BPR is constituted by a collection of process repositories ($BPR_1,\ldots,BPR_n$) which refer to different ontologies ($O_1,\ldots,O_n$).

3 Background Tools

We provide a set of tools within the framework for advanced querying of semantically annotated BPs from repositories, organization of BPs across different repositories regardless the reference domain ontology used for semantic annotation, ranking of candidate compositions.

BPAL Framework. BPAL (Business Process Abstract Language) [2] is a logic-based language for representing and reasoning with process knowledge. It is essentially a process ontology, strongly inspired from the BPMN [3] notation, which provides an explicit formalization of the meta-model and of the execution semantics thus allowing advanced BP query facilities that take into account both the structure (i.e., the workflow graph underlying the BPs) and the behavior (i.e., the possible executions) of BPs. Thanks to its grounding into logic programming, BPAL can be easily adopted in conjunction with rule-based ontology languages (e.g., OWL-RL [4]) for the annotation of BP schemas w.r.t. domain ontologies. Within the framework presented here, the BPAL reasoning framework [5] is exploited for the management of the BP repositories, specifically: (1) the possibility of importing BPMN processes (in their XPDL linear form) through a direct translation into BPAL; (2) support for the semantic annotation and the reasoning over the ontologies (in particular, OWL-RL is fully supported); (3) a query language, QuBPAL, based on the SELECT-FROM-WHERE paradigm, to express queries over repositories of semantically annotated BPs also in terms of local and global constraints; (4) a query engine, based on Prolog, that provides efficient, sound and complete evaluation of QuBPAL queries.

FC-MATCH Tool Suite. FC-MATCH (Functional Compatibility service hybrid MATCHmaker) is a tool suite for comparison of semantically annotated Web Service interfaces. Basically, given a Web service request WS_R and a set of available Web services WS_A, the matchmaker ranks the Web services in WS_A according to their functional compatibility, obtained as a combination of: (1) similarity-based Web service matchmaking, to quantitatively compute how much two semantically annotated Web service interfaces are similar; (2) logic-based Web service comparison, based on equivalence/subsumption checking of ontological concepts which annotate service inputs/outputs and operations. Moreover, the FC-MATCH tool suite facilities are based on the evaluation of the "semantic affinity" between concepts regardless the ontologies they belong to. Concept affinity evaluation between two concepts C_1 and C_2, denoted with $Caff(C_1,C_2)$, is an hybrid approach that combines a domain-dependent matching based on ontologies (intra-ontology concept affinity) and a terminological (domain-independent) matching based on WordNet (inter-ontology concept affinity). Such facilities will be exploited in order to allow semantic annotation and composition of BPs across different ontologies, as shown in Section 4.

P2S Tool Suite. The P2S (Process-to-Services) tool suite is a set of facilities for the identification of component sub-processes within a semantically annotated business process to be exposed as Web services, being characterized by key properties such as loose coupling, high internal cohesion, high reuse within the same business process [6]. To this aim, the P2S tool suite relies on advanced, semantic-enriched metrics for the identification of similar/reusable components and provides additional functionalities for the identification of the best business process decomposition, in particular for the evaluation of overall cohesion/coupling between component sub-processes, that are included within the framework presented in this paper.

4 Methodology

Semantic Annotation. In this phase, domain ontologies are used to annotate the BPs that, exposed as services, will be available in the BPR. The semantic annotation of process elements (task names, inputs and outputs) enables the automatic detection of correspondences between different BPs exposed as services, through the computation of the metrics used in the other phases of the methodology. Essentially, a semantic annotation is a relation $\sigma \subseteq BpsEl \times Concept$, defining a correspondence between elements of a BP and concepts of an ontology, in order to describe the meaning of the former in terms of a suitable conceptualization of the domain of interest provided by the latter. As previously stated the semantics-enriched characterization of the components in the BPR relies on the BPAL framework, that allows the annotation of a BP both as a black box (i.e., considering the exposed interfaces and the capabilities only), and at a fine-grained level, where the components of its internal structure (represented as a workflow) are considered.

The intra-ontology and inter-ontology concept affinity evaluation methods provided by FC-MATCH are exploited here in order to set up mapping relations between concepts belonging to different ontologies. In particular, the annotation of a BP, in terms of a given ontology, can be extended according to semantic relationships within the same ontology or according to concept affinity across different ontologies. We distinguish among two cases. Within the same ontology, the semantic annotation is preserved by the subsumption (ISA) relation, i.e., $\sigma(X,C_2) \leftarrow \sigma(X,C_1) \wedge subClassOf(C_1,C_2)$. If we consider two concepts C_1 and C_2 belonging to two distinct ontologies, a mapping relation is set between C_1 and C_2, denoted with C_1~>C_2, if $Caff(C_1,C_2) \geq \delta$, where δ is a threshold experimentally set (see [7]). Then, the semantic annotations defined for C_1 are propagated to C_2, i.e., $\sigma(X,C_2) \leftarrow \sigma(X,C_1) \wedge C_1$~>$C_2$.

User's request formulation. The input of the methodology is constituted by the user's request, i.e., the process skeleton and the associated constraints. Through a graphical workflow-based notation (in particular, we commit to the BPMN [3] platform-independent notation) the user defines the main building blocks of the process to be composed (*skeleton tasks*), together with the associated control flow dependencies and the expected interfaces. For instance, in the example in Figure 1, four main tasks are identified: the purchase order is received from a customer, the requested items are collected, the order is rejected or, in case the order is accepted, it is fulfilled (e.g., goods are delivered and invoice is produced). Furthermore, data objects are used to describe the inputs and outputs of the skeleton tasks (e.g., *FulfillOrder* takes a *PurchaseOrder* as input and produces an *Invoice* as output).

Skeleton tasks are then associated to local constraints that specify, in a declarative fashion, the criteria for the retrieval of the components to implement them. Local constraints are essentially interpreted as queries on the BPR, and are formalized according to the QuBPAL language. We do not go here into the details of the language, but we only introduce it through an example, showing the QuBPAL formulation of the constraint LC_2 associated to the skeleton task *FulfillOrder:*

```
SELECT <?p,?s,?e>
WHERE belongs(?a::o1:Requesting,?p,?s,?e) AND
output(?a,?r::o1:ShipperResponse,?p) AND response(?a,?a1,?p,?s,?e) AND
output(?a1,?inv::o1:Invoice,?p) AND response(?a,?a2::o1:Shipment,?p,?s,?e)
```

We prefix variable names with a question mark (e.g., *?x*) and we use the notation *?x::Conc* to indicate the semantic typing of a variable, i.e., σ(*?x,Conc*). The above query searches for Single-Entry-Single-Exit sub-processes (identified by the triple <*?p,?s,?e*>*,* where *?p* is a BP identifier, *?s* is the starting element and *?e* is the ending element) such that: *i)* they contain (*belongs* predicate) an activity *?a* annotated with the concept o_1*:Requesting*, whose output is a o_1*:ShipperResponse*, *ii)* in every execution of the sub-process, an activity *?a1* with output o_1*:Invoice* and an activity *?a2* annotated with o_1*:Shipment* are eventually executed after *?a* (*response*).

Similarly, global constraints are formulated as (boolean) QuBPAL queries too. They express constraints over the whole composed process skeleton, as in the case of CG_2, which states that, at the moment of executing the *FulfillOrder* task, an activity where the product availability is verified has been previously executed.

Process Composition. In the process composition, the reasoning engine is fed with the process skeleton constraints in order to assemble BP schemas that satisfy both the local and the global constraints. It is worth noting that, while the user's request is formulated according to the terminology provided by one particular ontology included into the BPR (e.g., O_1), the process composition makes use the ontology mappings previously computed to expand the queries in order to exploit also the BPs annotated w.r.t. other ontologies, in a way that is transparent to the user. The process composition is conducted according to the following steps:

1) the local constraints are used to retrieve process fragments to be used as building blocks for the process composition; for each skeleton task *K:*
   a) the QuBPAL query corresponding to the local constraint LC_k is evaluated, and the results are collected in a set RLC_k of BPs (or process fragments);
   b) every element r_i of RLC_k is checked for compatibility with respect to the input/output defined for *K*, that is performed by applying the FC-MATCH facilities, which compare the interfaces (in terms of input/output data objects) of r_i and *K*, classifying them among exact, plug-in and overlapping (to denote that r_i and *K* provide the same functionalities, r_i adds functionalities to those provided by *K* or r_i and *K* presents partially overlapping functionalities, respectively), together with the similarity value between them;

2) the fragments identified in the step 1.a that are compatible with the skeleton according to the step 1.b are organized in a collection *S* of BP schemas implementing the process skeleton;
3) finally, the composed BP schemas that are not compliant with the global constraints are discarded; the QuBPAL queries corresponding to each global constraint GC_k are evaluated against each BP in *S*, in order to select the set of candidate BP schemas to be proposed to the user for their final validation.

Ranking. The candidate BP schemas identified in the previous step should be ranked to enable the user to choose the best solution. The ranking is computed according to the evaluation of sub-process cohesion/coupling, which is performed to ensure the best sub-process granularity. Homogeneous granularity is a strong requirement for effective collaboration. This step enables the process decomposition into a set of self-contained, decoupled units of work. Specifically, identified sub-processes must ensure high internal cohesion and low coupling. The adopted cohesion/coupling metrics have been inspired by their well-known application in software engineering field [8] and have been implemented within the P2S Tool Suite [6]. These metrics are used to evaluate the degree of similarity correspondence between I/O flows among tasks within identified sub-processes. The cohesion coefficient evaluates how much tasks within a single sub-process contribute to obtain a process output. The coupling coefficient evaluates how much tasks belonging to different sub-processes need to interact. The ratio between coupling and cohesion coefficients must be minimized and it is used to rank the proposed composition.

5 Conclusions

In this paper, we proposed a semantics-enabled framework to support BP design through the orchestration of semantically annotated sub-processes that are composed according to: (i) an high level definition of the target BP orchestration, in the form of a process skeleton; (ii) fine-grained querying of available BPs, based on their structure and behavior, to retrieve component sub-processes to implement the target skeleton; (ii) high-level compatibility checking of retrieved component sub-processes based on the analysis of their interfaces; (iii) ranking of the proposed solutions on the basis of cohesion/coupling metrics. Semantic annotation of sub-processes is performed without constraining the adoption of a unique domain ontology and the sub-process selection relies on an hybrid approach to identify mapping between concepts across different ontologies. The scalability of the framework is partially ensured since: 1) ontology matching is performed off-line only when new ontologies are added to the system; 2) high-level compatibility checking and cohesion/coupling metrics evaluation present good response time [6,7]; 3) fine-grained queries are polynomial and the query engine presents considerable scalability [5]. Additional tools must be added to the framework to support the annotation of component sub-processes and the specification of local and global constraints and to provide enhanced, user-friendly interfaces for business process designers. Nevertheless, since sub-process providers are not constrained to adhere to the same ontology, the burden deriving from the semantic annotation is reduced. The set of heuristics adopted to speed up the composition must be properly extended to make the system further scalable. Finally, real test cases on which the methodology can be tested are needed.

References

[1] Papazoglou, M.P.: Web Services: Principles and Technology. Prentice Hall, 2007.

[2] De Nicola, A., Missikoff, M., Proietti, M., Smith, F.: An Open Platform for Business Process Modeling and Verification. Proc. DEXA 2010. LNCS 6261, pp. 66-90, Springer, 2010.

[3] OMG: Business Process Model and Notation, http://www.omg.org/spec/BPMN/2.0.

[4] OWL 2: Profiles, http://www.w3.org/TR/owl2-profiles.

[5] Missikoff, M., Proietti, M., Smith, F.: Querying Semantically Enriched Business Processes. Proc. DEXA 2011. LNCS 6861, pp. 294-103, Springer, 2011.

[6] Bianchini D., Cappiello C., De Antonellis V., Pernici B.: P2S: A Methodology to Enable Inter-organizational Process Design through Web Services. Proc. CAiSE 2009, pp. 334-348, Amsterdam, The Netherlands, 2009.

[7] Bianchini D., De Antonellis V., Melchiori M.: Flexible Semantic-based Service Matchmaking and Discovery, World Wide Web Journal, 11(2):227-251, 2008.

[8] Vanderfeesten I., Reijers H.A., van der Aalst W.M.P.: Evaluating workflow process designs using cohesion and coupling metrics. Computer in Industry, 59(5):420-437, 2008.