Measuring Expected Integration Effort
in Service Composition

Tomer Sagi, Avigdor Gal
Technion – Israel Institute of Technology
Haifa, Israel
{stomers7@tx, avigal@ie}.technion.ac.il

Matthias Weidlich
Imperial College
London, United Kingdom
m.weidlich@imperial.ac.uk

Abstract—Evaluating alternative solutions for service compositions is done by various properties, each requiring an associated evaluation measure. In this paper, we propose a new measure, namely integration effort, to capture the expected effort a human programmer is expected to invest in integrating composed services into a functioning process. We present several integration effort evaluation measures, which were adapted from the related research areas of schema and ontology matching. These measures are embedded in an extendible framework, allowing application in different levels of refinement. Our measures are empirically validated to be effective proxies of integration effort.

Index Terms—Evaluation; Service Composition; Semantic Web

I. INTRODUCTION

Service-oriented engineering of process-aware information systems (PAIS) builds on ready-made, re-usable services. To implement a business process (the target process) in a PAIS, candidate services are assembled into a composition. A typical service composition scenario is composed of two steps. First, a composition skeleton is created by selecting candidate services, aligning their parameters with each other and with the tasks of the business process. As a second step, this composition skeleton is prepared for enactment by inserting mediating code that implements data transformations or completes missing parameters for the established alignments and links.

Given a description of a business process, there is typically a large search space of possible compositions due to duplication and overlap between services. Numerous automatic and semi-automatic composition methods have been suggested to search for an optimal solution within such a (very large) search space. To assist search and to evaluate the relative value of disparate solutions, different quality functions have been devised and grouped into two major categories, namely functional such as semantic quality [1] and non-functional, also named quality of service or QoS, see [2] for a detailed overview. These approaches evaluate the composition skeleton, thereby assessing the quality of the selected services. To the best of our knowledge, none of the functions presented to date estimate the integration effort. However, we observe that this effort (and its associated cost) may be prohibitive, overshadowing the cost of using a service with slightly worse semantic quality or one that comes with marginally worse QoS.

In this paper, we formally introduce the notion of integration effort and suggest measures to assess it in the context of service composition. More specifically, the paper makes the following contributions:

1. We provide a framework for the definition of measures for integration effort in service composition.
2. We instantiate several measures in this framework.
3. We empirically validate that the proposed measures are well correlated with the expected effort.

The rest of the paper is structured as follows. We position our work within the service composition literature in Section II and present a formal model for service composition in Section III. Section IV presents a framework for the definition of measures for integration effort. Section V empirically validates the effectiveness of the measures as proxies to integration effort. Finally, we review related work (Section VI) and conclude the paper (Section VII).

II. BACKGROUND

A multitude of service composition approaches has been presented over the years. The sheer breadth of work on this subject has brought Li et al. [3] to state that ‘examining all of the related work in this area becomes a mission next to impossible.’ Undeterred, they continued to provide a classification matrix for service composition, a subset of which is presented in Table I, limiting the discussion to those dimensions pertinent to the positioning of our work.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model-driven</td>
<td>X</td>
</tr>
<tr>
<td>AI-Planning</td>
<td>X</td>
</tr>
</tbody>
</table>

We discuss here three technologies, namely workflow-based, model-driven, and AI-planning. Measures for integration effort can be used as fitness measures for workflow-based approaches, verification criteria for model-driven approaches, and search guidance criteria in AI-planning. In the pattern dimension, we assume the orchestration of services by a central entity. In the semiotics dimension, we assume
semantic annotation is available (commonly referred to as Semantic Web Services) in order to judge on the quality of the assignments and links that are part of the composition skeleton.

In the design-time dimension, which refers to the extent of user involvement, our work is placed in the semi-automatic category. The presented measures are aimed at measuring expected manual labour incurred by a user completing the skeleton of a service composition.

Given an approach to service composition that falls into the aforementioned classes, the following example highlights the importance of integration effort.

**Example 1:** The upper part of Fig. 1 specifies a business process by means of process tasks and control flow (right-hand side), and an excerpt of an ontology for service parameters (left-hand side). The bottom part shows five services and, again, an ontology for the respective parameters. Using the semantic annotations, the skeleton of a service composition has been derived: semantic assignments (sa1 to sa3) relate process tasks to candidate services and semantic links (sl1 to sl3) connect different candidate services. Comparing the ontology concepts and their relations for the parameter types, it is obvious that some integration work is required: A ‘User Record’ consists of an ‘ID’ that needs to be extracted to match the type of the input parameter ‘Integer’ (link sl3). Further, certain parameters of process tasks and candidate services have not been aligned or linked, e.g., ‘Time Period’ and ‘Domain Code’.

Such differences in semantics and data types require mediation code and incur additional integration costs that should be considered when evaluating the quality of service composition and when selecting among different alternative solutions. Although existing semantic distances can be used to assess the quality of a single assignment or link, the question of how to use such measures to comprehensively evaluate the overall integration effort, including efforts to complete missing assignment and links, has not been addressed before. We fill this gap by presenting a framework for the definition of such measures and offer several concrete measures to estimate the integration effort.

**III. PRELIMINARIES**

A service composition task is described by a process model and a repository of candidate services. To solve a composition task, candidate services are selected, assigned to process tasks, and linked to other services. The question of how to solve a composition task is not our focus. Therefore, our model is based on the two essential building blocks of such a solution, namely, process tasks and candidate services with their respective input and output interfaces. These interfaces are sets of parameters (inputs and outputs for short), where inputs may be mandatory or optional.

**Definition 1 (Process Task):** A process task $T$ of a process model is a triple $T = (I^T_m, I^T_o, O^T)$, where $I^T_m$ is a finite set of mandatory inputs, $I^T_o$ is a finite set of optional inputs, and $O^T$ is a finite set of outputs.

**Definition 2 (Candidate Service):** A candidate service $W$ in a service repository is a triple $W = (I^W_m, I^W_o, O^W)$, where $I^W_m$ is a finite set of mandatory inputs, $I^W_o$ is a finite set of optional inputs, and $O^W$ is a finite set of outputs.

For a process task $T$, $I^T = I^T_m \cup I^T_o$ is the set of all inputs. Similarly, $I^W = I^W_m \cup I^W_o$ is the set of inputs of a candidate service $W$. The introduced notions allow us to define a composition task as follows.
Definition 3 (Composition Task): A composition task $C$ is defined as a pair $\langle T, R \rangle$ where $T$ is a finite set of process tasks that should be implemented by a selection of services from $R$, a finite set of candidate services (a repository). Figure 2 summarizes the relations between the concepts.

Example 2: Consider the process model in Fig. 3(a), which requires implementation of a single process task $T_1$ with two inputs of types Name and String, the latter is declared as optional, and a single output of type Integer. Hence, we have $T_{in}^{T_1} = \{\text{Employee}\}$, $T_{out}^{T_1} = \{\text{dept.}\}$, and $O_{in}^{T_1} = \{\text{extension no.}\}$ as the input and output sets. Figure 3(b), in turn, shows three candidate services $R = \{W_1, W_2, W_3\}$. For example, the parameter sets for service $W_1$ are defined as $T_{in}^{W_1} = \{\text{Building,Room}\}$ and $O_{in}^{W_1} = \{\text{extension no.}\}$.

Next, we define the components of a solution to a composition task. First, the parameters of a process task are assigned to parameters of candidate services. This is typically done based on semantic annotations of the parameters, see Sycara et al. [5] for details on the ontological constructs used in semantic alignment. For our work, however, it suffices to abstract from these annotations and begin by assuming each assignment to be evaluated as either exact or imprecise, where imprecise links include plug-in, subsume, and intersection match types.

Definition 4 (Semantic Assignment): Let $T$ be a process task and let $W$ be a candidate service. Let $pt$ and $pw$ be either inputs of $T$ and $W$, $pt \in T^T$ and $pw \in T^W$, or outputs, $pt \in O^T$ and $pw \in O^W$. Then, $sa = (pt, pw, e)$ is a semantic assignment between the parameters with evaluation $e \in \{\text{exact, imprecise}\}$.

A semantic assignment is defined over parameters, but also induces a relation between a process task acting as the source and a candidate service acting as the target of the assignment.

The parameters of different candidate services are aligned by semantic links, cf., Sycara et al. [5].

Definition 5 (Semantic Link): Let $W_1, W_2$ be candidate services. Let $ow_1 \in O^{W_1}$ and $ow_2 \in O^{W_2}$ be an output and an input of the two services, respectively. Then $sl = \langle ow_1, ow_2, e \rangle$ is a semantic link between the parameters with evaluation $e \in \{\text{exact, imprecise}\}$.

Semantic assignments and links give rise to the notion of a composition task solution.

Definition 6 (Composition Task Solution): Let $C$ be a composition task. Let $A$ be the set of semantic assignments over the process tasks and candidate services of $C$ and let $L$ be a set of semantic links defined over the services of $C$. Then $S_C = \langle A, L \rangle$ is a composition task solution.

As illustrated in Fig. 4, semantic assignments are defined between parameters of a process task and a candidate service and, thus, induce a relation between tasks and services. Similarly, semantic links defined between input and output parameters induce a relation between two candidate services.

Example 3: Returning to our running example, let us assume that there are two solutions to the composition task. In the first, $W_2$ is assigned to $T_1$ (Fig. 5(a)). In the second a composition of $W_3$ and $W_1$ is assigned to $T_1$ (Fig. 5(b)). For this solution, Table II also lists the respective assignments and links.

IV. Measures for Integration Effort

This section first elaborates on the atomic operations required to prepare a composition skeleton for enactment. Then, we discuss our general approach before we turn to the definition of the actual measures.
Atomic Operations. Given a set of semantic assignments and semantic links that have been derived by one of the existing approaches to service composition, a series of coding operations is required to transform a set of candidate services to a functioning computerized process. The need to perform these operations stems from two sources. First, there may be (partial) incompatibility of outputs and inputs due to data type or semantic heterogeneity. Second, there may be parameters of process tasks (input and output) or services (input) that are not assigned or linked, but are mandatory.

To facilitate the discussion of integration effort, we define two classes of atomic operations that are applied when transforming a skeleton of a service composition into a functioning process. An atomic transformation operation is a piece of mediation code written to convert a single input/output parameter from the original format to the required format. An atomic completion operation entails some search for compatible services within the repository and the writing of mediation code to fit them within the existing solution in order to complete a missing mandatory parameter. In what follows, we require an estimate of the severity of a particular atomic operation, which may involve, e.g., assessment of data type compatibility or semantic distances between the parameter types. This information is used to evaluate the amount of integration effort related to transformations and completions.

General Approach. In related realms such as schema and process matching, integration effort is approached by precision and recall, each measuring a different aspect of a solution. The former measures to what extent the proposed solution is correct and does not require transformation effort. The latter measures the completeness of the solution, or how much of the task has been successfully completed. For illustration, consider the example of pair-wise matching of two data schemas. The atomic unit of a solution to a schema matching task is a pair of matched attributes, so that a solution \( M \) contains a set of such pairs that is evaluated against an exact match \( E \), which is the expected set of matched attribute pairs. Then, two measures are defined, precision \( f_{Pr} \) and recall \( f_{Re} \):

\[
f_{Pr} = \frac{|M \cap E|}{|M|} \quad f_{Re} = \frac{|M \cap E|}{|E|}
\]

Defining similar measures for the area of service composition requires definition of a minimal solution element. The smallest unit of consideration in the model presented in Section III is an input or output parameter of a candidate service or process task. Specifically, we examine mandatory input parameters. The assignment between an optional input parameter of a process task and a parameter of a candidate service is typically not a necessary condition for successful implementation of the process model. The same holds true for semantic links between optional parameters. In addition, output parameters of process tasks are considered since they represent the expected process model results. However, when it comes to semantic links, unlinked service output parameters do not hinder achieving a functioning process.

Below, we formally define the sets of relevant parameters with the notions of a task parameter set and a composition parameter set.

Definition 7 (Task Parameter Set): Let \( C = \langle T, R \rangle \) be a composition task. Then, the task parameter set \( P^T = \bigcup_{T \in T} \{ p \mid p \in T_m \cup O_T \} \) is the union of all mandatory input parameters and output parameters.

Definition 8 (Composition Parameter Set): Let \( S_C = \langle A, L \rangle \) be a solution for composition task \( C = \langle T, R \rangle \). The set of services participating in the solution is denoted by \( W \subseteq R \) and contains a service \( W \), iff

- \( pw \in T_m \cup O_T \) (there is a semantic assignment \( \langle pt, pw, e \rangle \in A \) related to parameters of \( W \)), or
- \( ow \in O_W \) or \( iw \in T_m \) (there is a semantic link \( \langle ow, iw, e \rangle \in L \) related to parameters of \( W \)).

Then, \( P_W = \bigcup_{W \in W} \{ p \mid p \in T_m \} \) is the composition parameter set.

Example 4: For our running example (see Example 3), the task parameter set for the process task \( T_1 \) is \( P_{T_1} = \{ \text{Employee, extension no.} \} \). For Solution 1, the composition parameter set is \( P^{W_2} = \{ \text{Person Name} \} \) and for Solution 2, \( P^{W_1} = \{ \text{Building, Room, Person Name} \} \).
We now define the structure of functions used to evaluate the quality of assignments and links relative to the considered parameters. Informally, evaluation functions take a parameter and the semantic assignments or links related to the parameter and evaluate them to a measure in the $[0, 1]$ range. Below, we use the notation $2^S$ to denote the power set of some set $S$.

**Definition 9 (Task Evaluation Function):** Let $S_C = \langle A, L \rangle$ be a solution for composition task $C = \langle T, R \rangle$ and let $P_T$ be the task parameter set. A *task evaluation function* takes the form:

$$f_T : P_T \times 2^A \rightarrow [0, 1]$$

**Definition 10 (Composition Evaluation Function):** Let $S_C = \langle A, L \rangle$ be a solution for composition task $C = \langle T, R \rangle$ and let $P_W$ be the composition parameter set. A *composition evaluation function* takes the form:

$$f_W : P_W \times 2^{\mathcal{L} \cup A} \rightarrow [0, 1]$$

Below, we suggest several measures for integration effort that are based on different instantiations of these functions. We present a simple binary instantiation that can be applied with no additional information on service composition tasks. We then extend the functions to incorporate semantic distance and data-type scoring functions. Finally, we allow for varied parameter importance.

**Binary Precision and Recall.** Our first instantiation assigns a value of 1 to parameters that are *exactly* matched according to the evaluation value of the semantic assignment or link; and 0 to all other parameter. This approach can be applied to evaluate semantic assignments yielding a binary task evaluation function $f_T^b$ or to evaluate semantic links leading to a binary composition evaluation function $f_W^b$. Both functions are defined identical, but they are applied to different domains as presented in Definitions 9 and 10.

$$f_T^b(x, Y) = f_W^b(x, Y) = \begin{cases} 1 & \forall (p, p', e) \in Y : x \in \{p, p'\} \Rightarrow e = \text{exact} \\ 0 & \text{otherwise} \end{cases}$$

(1)

Based on these two specific evaluation functions, we present notions of precision and recall adapted for service composition. As outlined above, these measures share an evaluation function in the numerator and differ by the normalization factor used in the denominator.

First, binary task precision and recall proxy the integration effort stemming from semantic assignments as follows.

**Definition 11 (Binary Task Precision and Recall):** Let $S_C = \langle A, L \rangle$ be a solution for composition task $C = \langle T, R \rangle$ and let $P_T$ be the task parameter set. For parameter $p \in P_T$, the set of related semantic assignments is defined as $A_p = \{ \langle pt, pw, e \rangle \in A \mid pt = p \}$. Then, *binary task precision* is defined as

$$P_{T_b} = \frac{\sum_{p \in P_T} f_T^b(p, A_p)}{|A|}$$

(2)

and *binary task recall* is defined as

$$R_{T_b} = \frac{\sum_{p \in P_T} f_T^b(p, A_p)}{|P_T|}.$$  

(3)

Second, we adapt the same approach to define binary composition precision and recall to proxy integration effort that stems from semantic links.

**Definition 12 (Binary Composition Precision and Recall):** Let $S_C = \langle A, L \rangle$ be a solution for composition task $C = \langle T, R \rangle$ and let $P_W$ be the composition parameter set. For parameter $p \in P_W$, let $A_p$ be the set of related semantic assignments (Definition 11) and let $L_p = \{ \langle ow, iw, e \rangle \in L \mid p \in \{ow, iw\} \}$ be the set of related semantic links. Then, *binary composition precision* is defined as

$$P_{W_b} = \frac{\sum_{p \in P_W} f_W^b(p, L_p)}{|L|}$$

(4)

and *binary composition recall* is defined as

$$R_{W_b} = \frac{\sum_{p \in P_W} f_W^b(p, A_p \cup L_p)}{|P_W|}.$$  

(5)

**Example 5:** Calculation of these measures for Solution 2 (Fig. 5(b) and Table II) of our example yields the following results. Out of the two semantic assignments, one is evaluated to be exact whereas the other is imprecise. Thus, binary task precision evaluates to $1/2 = 0.5$. Since process task $T_1$ has one mandatory input and one output, but only one assignment is exact, binary task recall is $1/2 = 0.5$. Since both semantic links of Solution 2 are exact, binary composition precision evaluates to $2/2 = 1$. Among the solution services, $W_{S_1}$ has two input parameters and $W_{S_3}$ has one. Both input parameters of $W_{S_1}$ are covered by semantic links that are evaluated to be exact. However, input parameter Person Name of $W_{S_3}$ is covered by a semantic assignment that is imprecise. Therefore, binary composition recall evaluates to $2/3 \approx 0.66$.

Before continuing, we should point out that, in related realms, the expected result is a real result generated by a human domain expert. In our case, the expected result is a theoretical perfect composition, that may not be available given a set of service composition algorithms. However, we retain the similarity to related realms in the fact that post-algorithmic effort emanates from converting the obtained result to the expected one.

**Non-Binary Precision and Recall.** In previous work [6], we have shown binary measures to be inferior compared to non-binary measures due to their lack of sensitivity and smoothness. We therefore suggest non-binary versions of precision and recall for service composition by introducing a scoring function. This function is defined over $[0, 1]$ rather than $\{0, 1\}$ as in the binary case (Equation 1). This function needs to be well correlated with the amount of transformation and completion effort required, which suggests to rely on a combination of measures for data type compatibility (e.g., an effort estimation matrix for basic XML-Schema types or a tree-edit distance function for complex types) and semantic distances for the parameter types.
Definition 13 (Scoring Function): Let $S_C = \langle A, L \rangle$ be a solution for composition task $C = \langle T, R \rangle$ and let $P^T$ and $P^W$ be the task parameter set and the composition parameter set, respectively. Then, a scoring function $f_s$ takes the following form $f_s : P^T \cup P^W \times P^T \cup P^W \rightarrow [0,1]$.

Based on this scoring function, we are able to define non-binary variants $f_T^{nb}$ and $f_W^{nb}$ of the two evaluation functions $f_T$ and $f_W$ given in Equation 1 as follows:

$$f_T^{nb}(x, Y) = f_W^{nb}(x, Y) = f_s(x, y_1) \cdot \ldots \cdot f_s(x, y_n) \quad (6)$$

for $Y = \{y_1, \ldots, y_n\}$ \quad (7)

Then, the non-binary versions of precision and recall are obtained by replacing the binary evaluation functions in Definitions 11 and 12 with their non-binary counterparts, which we omit for space considerations.

Non-binary Parameter Importance So far, we distinguished parameters that are declared to be mandatory from those that are optional. In practice, many standards used for the description of service interfaces, e.g., the Web Services Description Language (WSDL) [7], support the definition of mandatory and optional parameters. As such, these languages provide the basis for a binary interpretation of parameter importance. In some cases, however, parameter importance may not be explicitly stated, which requires white-box evaluation of service code to conclude on parameter optionality.

In addition, parameter importance may be evaluated with a non-binary measure to come to a more fine-grained judgement. Different approaches can be taken to instantiate such a function. For instance, the order of parameters in service interfaces often carries implicit hints on their importance, listing the most important parameters first, whereas less important parameters, that may be applicable only in rare cases, are listed last. We accommodate for such a fine-grained notion of importance with an explicit function.

Definition 14 (Parameter Importance Function): Let $S_C = \langle A, L \rangle$ be a solution for composition task $C = \langle T, R \rangle$ and let $P^T$ and $P^W$ be the task parameter set and the composition parameter set, respectively. Then, a parameter importance function $f_i$ takes the following form $f_i : P^T \cup P^W \rightarrow [0,1]$. Modifying the non-binary precision and recall measures to account for parameter importance entails multiplication of the respective (task or composition) evaluation function with its importance, and the replacement of a parameter $l$ assignment count in the denominator with a sum over importance. Due to space limitations, we outline the adapted definition solely for task recall:

$$Re_T^{nb} = \frac{\sum_{p \in P^T} f_i(p) \cdot f_T^{nb}(p, A_p)}{\sum_{p \in P^T} f_i(p)} \quad (8)$$

V. EMPIRICAL EVALUATION

In this section, we demonstrate the effectiveness of the proposed concepts in measuring expected integration effort.

Setup. The basis of our evaluation is the Web Service Challenge\textsuperscript{1}. This challenge describes five composition tasks and 30 solutions for these tasks, that are ranked by QoS properties such as latency. A composition task is described in WSDL [7] and semantically annotated in OWL [8]. The solutions are built from a service repository of 4180 semantically annotated service descriptions and presented in WS-BPEL [4]. All parameters, semantic annotations and data types are opaque and represented by random numbers. For example, consider Task 1 of the challenge (upper right part of Fig. 6), which has 10 input parameters and four output parameters. The first place solution is a composition of seven services providing all four output parameters and requiring all 10 input parameters to execute. In the composition in Fig. 6, lines represent semantic links between the composed services and rectangular outlines mark output parameters semantically assigned to the process task.

We establish the ground truth for our experiments, i.e., the amount of integration effort induced by a solution, based on the aforementioned atomic operations.

Transformation effort. Quality of each assignment and link is available as a semantic path between the respective ontological concepts. Here, a zero-length path represents an exact semantic match. A path of length one represents parameters which are two instances of the same semantic concept. Paths of length two or more represent subsumption of the concepts describing the respective parameters. Hence, the length of the semantic path provides us with a measure of the first dimension of integration effort, i.e., the number of required atomic transformations.

Completion effort. The second dimension of integration effort, i.e., the number of atomic completions, cannot be measured directly since the challenge required complete coverage of task parameters and does not include interim solutions. Thus, we created synthetic solutions by removing links and assignments, which is justified as follows. When a solution contains effort-intensive assignments or links, users may consider only those for which the quality is higher than some threshold. To simulate this scenario, we used a decreasingly strict threshold for assignments and links to be included in the solution. Then, the number of completions required to obtain a running solution is our measure for the second dimension of integration effort.

To investigate to which extent these types of integration effort are mirrored by the proposed measures, we select the non-binary versions of task/composition precision and recall (see [6] for a discussion on the benefits of non-binary measures). We instantiate the scoring function (Definition 13) for two parameters as the relative length of their semantic path, defined as the difference between the maximal path length observed in the solution and their path length, divided by the maximal path length. The actual evaluation functions are then derived by composing the results according to Equation 6 using the arithmetic average.

\textsuperscript{1}see http://ws-challenge.georgetown.edu/wsc10/ [accessed 2014-02-19]
Results. We first investigated the behaviour of our measures in relation to the incompleteness of a solution, i.e., regarding the integration effort induced by atomic completion operations. Focusing on the evaluation related to the composition of services, i.e., the effort induced by semantic links between service candidates, Fig. 7 outlines the values of our precision and recall measures when including only those links, for which the semantic path is shorter or equal to the threshold (averaged over all composition tasks of the dataset). We observe that both measures show the expected behaviour: As the threshold becomes less strict (from left to right), recall increases (less integration effort related to completions is required) and precision decreases (more integration effort related to transformations is required).

To validate our hypothesis that precision will be a valid proxy for integration effort stemming from transformations, we plotted task precision against the number of semantic assignments requiring such an operation and composition precision against the number of semantic links requiring an atomic transformation operation (Fig. 8). Here, the results are shown for task 5 of the Web Service Challenge, which is the largest one in the dataset and is comparably similar to the other tasks. In both cases, i.e., integration effort related to assignments as well as related to links, there is a clear correlation between our measures and the actual integration effort. Also, the respective linear least squares regressions
explain a large share of the observed variability, around 70% for task precision and 95% for composition precision.

Our second hypothesis, claiming recall to be a good proxy for integration effort stemming from completions, is validated as well: the integration effort required to complete the solution is clearly correlated with the non-binary recall measures (Fig. 9) and the respective linear least squares regressions largely explain the observed variability.

VI. RELATED WORK

Detailed semantic quality measures are rare in service composition literature. Notably, Lécué and Mehandjiev [1] identified semantic quality of links as a proxy to integration effort by stating: ‘selecting links with the best functional quality will ensure easy end-to-end integration between services’. Our work follows this line, but overcomes several issues of the two quality functions defined in [1] over semantic links. That is, we address aggregation of these measures for composition solutions and also consider unmatched inputs, which are neglected in [1] despite being a major source of integration effort. Unlike [1], we do not assume that all output-input pairs of a certain semantic match type are assigned the same score, but consider the differences in transformation effort related to concrete types of parameters.

Related measures for data complexity were considered by Basci and Misra [9] in the context of estimating future maintenance costs. Their work was extended by Mao [10] to include the complexity created by the composition itself under a business process specified in BPEL. Of the nine metrics considered, the data complexity metric shows most promise for our purposes as it may be conceptually correlated with the integration effort. Highly complex links should incur higher effort. However, by focusing on the data transmitted (output parameter) it fails to consider differences in complexity between the output and input parameters of the same link and the effort required to resolve these differences. Technical difficulties using this measure as a metric may arise due to the fact that it is defined over $[0, \infty)$ and, thus, numerical comparison of results for different composition plans may prove difficult. Classic Information Retrieval versions of precision and recall were considered by works such as Toch et al. [11]. However, their focus is on the selection of services for a composition skeleton and is limited to binary relevance judgement of a ranked list of candidate services. Therefore, it is not suitable for our scenario. In earlier work, Toch et. al [12] suggested a ranking function for services w.r.t a process task based upon parameter similarity. The proposed function resembles some of the evaluation functions presented in Section IV by virtue of being based upon parameters and an evaluation of the extent of parameter similarity and importance. However, the function ignores the effects of composing several services to answer a single task and does not evaluate a complete process task w.r.t a proposed composition.

VII. CONCLUSIONS

We addressed the evaluation of service composition for the dimension of integration effort, i.e., the expected effort by a human programmer to integrate composed services into a functioning process. By proposing a framework that allows defining coarse and fine-grained measures, we support comparison of alternative compositions during a solution search process, or during design-time evaluation of algorithms. Our empirical evaluation shows that the proposed measures are indeed effective proxies for integration effort.

Despite its importance for realizing semi-automatic service composition, integration effort had been largely neglected as an evaluation dimension with our work being the first to present appropriate measures. In future work, we intend to investigate the correlation of effort based measures with actual effort encountered by domain experts during composition work. Furthermore, we intend to expand upon the notion of variable parameter importance and explore proxies for its estimation and representation.

ACKNOWLEDGEMENT

This work was partially supported by the D.F.G. German Research Foundation.

REFERENCES