This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Metrics for BPEL Process Reusability Analysis in a Workflow System

Alireza Khoshkbarforoushha, Pooyan Jamshidi, Mahdi Fahmideh Gholami, Lizhe Wang, *Senior Member, IEEE*, and Rajiv Ranjan

***Abstract*—This paper proposes a quantitative metric to analyze potential reusability of a Business Process Execution Language (BPEL) process. The approach is based on the description and logic mismatch probability of a BPEL process that will be reused within potential contexts. The mismatch probabilities have been consolidated to a metric formula for quantifying the probability of potential reuse of BPEL processes. An initial empirical evaluation suggests that the proposed metric properly predicts potential reusability of BPEL processes. According to the experiment, there exists a significant statistical correlation between the results of the metric and the experts' judgments. This indicates predictive dependence between the proposed metric and the potential reusability of BPEL processes as a measuring stick for this phenomenon. If future studies ascertain these findings by replicating this experiment, the practical implications of such a metric are early detection of the design flaws and aiding architects to judge various design alternatives.**

***Index Terms*—Business process execution language (BPEL) process reusability, composite service, service-oriented architecture (SOA) metric, service reusability analysis.**

I. Introduction

Web service composition is a well-established approach to integrate applications and produce intraorganizational business processes. In service-oriented computing (SOC), a business process is a coarse-grained composite web service executing a control flow [i.e., service logic (SL)] to meet a business goal. Among various technologies, Business Process Execution Language (BPEL) is a *de facto* standard that is utilized to realize required orchestration and choreography between diverse web services. In fact, BPEL is a workflow-oriented composition model that brings a central piece in the heavily modularized SOC model. Throughout this paper, composite service and BPEL process terms are used interchangeably.

It has been widely recognized that reusability is inherent to service-oriented solutions, and studies demonstrate that organizations regard reuse as a top driver for service-oriented architecture (SOA) adoption [1]. In fact, reusability is acknowledged as the main purpose in the design of services [2].

According to research work reported in the literature [3], [4], there is a strong demand for reusing process models, particularly BPEL processes. In this regard, researchers end up in different reuse strategies, including process template [5], [6], reference process [7], [8], ad hoc modifications to existing process models, reusing parts of business processes with the aid of BPEL fragments [3], and recently by applying service supervision patterns [9].

A BPEL process would not be reused in potential contexts if there were any mismatches between potential requirements and those the service expected to realize. The context is categorized to current solution that a service being reused and potential future solutions that it might be reused. Therefore, service designers should keep in mind that any service they produce can potentially become a reusable asset [10]. Process designers should not exclusively focus on the requirements of the initial consumers of a service but rather should adopt appropriate reuse strategies and also undertake more extensive business analysis in order to determine more general requirements to produce reusable assets. Taking this approach into account, software developers should support in their preparation of software for potential reuse [11]. In this regard, Frakes and Kang [12] emphasize that there is a clear need for future research to identify good ways to estimate the number of potential reuses. Therefore, potential reusability R_P analysis refers to predicting to what extent a software element can be reused in the future [13]. This means that, in this paper, we do not aim to measure BPEL process reusability. Instead, this paper is to propose a measure to predict the potential reusability of BPEL processes as service-centric implemented processes. This is called predictive validity [14].

Although there is a set of guidelines and approaches [15], [16] that helps architects to ensure a balance of proper logic encapsulation and adherence to standard description in software services, there is no quantitative metric to analyze the extent to which a designed BPEL process can be reused in the future. Previous research studies on service reusability, while

Manuscript received October 18, 2013; revised February 18, 2014; accepted April 3, 2014. The work of P. Jamshidi was supported in part by Science Foundation Ireland through Lero–The Irish Software Engineering Research Centre under Grant 10/CE/I1855. The work of L. Wang was supported by Hebei Natural Science Foundation under Project F2014203093. *(Corresponding author: L. Wang.)*

A. Khoshkbarforoushha is with the School of Computer Science, The Australian National University, Canberra, A.C.T. 0200, Australia, and also with the Commonwealth Scientific and Industrial Research Organisation (CSIRO) Computational Informatics, The Australian National University, Canberra, A.C.T. 2601, Australia.

P. Jamshidi is with Lero–The Irish Software Engineering Research Centre, School of Computing, Dublin City University, Dublin 9, Ireland.

M. F. Gholami is with the College of Engineering and Computer Science, The Australian National University, Canberra, Australia.

L. Wang is with the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China (e-mail: Lizhe.wang@gmail.com).

R. Ranjan is with the Commonwealth Scientific and Industrial Research Organisation (CSIRO) Computational Informatics, The Australian National University, Canberra, A.C.T. 2601, Australia.

Digital Object Identifier 10.1109/JSYST.2014.2317310

contributing to the field, have been subject to the following criticisms. These include being interface driven and neglecting logical constructs and being too complex and hard to collect. In contrast, our metrics not only consider logical activities within BPEL processes but also their calculations are based on BPEL process standard specifications; hence, all the computations are automated and do not require user intervention.

The key contributions of this paper are definition and empirical validation of a quantitative metric to analyze potential reusability of BPEL processes. The proposed approach is based on description and logic mismatch probability (MMP) estimation of composite web services that will be reused across potential solutions.

II. Related Work

Although there are significant metrics in object-oriented (OO) and component-based development (CBD) area [18], [19], accommodating these metrics in SOA context has significant challenges.

Different abstraction levels: Services are typically at higher level of abstraction and encapsulation than software components and classes [21]. OO metrics only measure reusability by considering source code analysis and class complexity. With the same manner, the CBD metrics merely consider a component, including a number of classes, whereas a BPEL process can be developed through adopting a number of software services. Perepletchikov *et al.* have evaluated the applicability of some of the well-established OO metrics [22] in the context of SOA, particularly BPEL-based services. To illustrate deficiency of applicability of existing OO metrics, a predefined software project has been developed through both OO- and BPEL-based approaches, respectively. The quantitative comparison of results shows that the existing metrics cannot be applied, and there is a need to tailor existing metrics or introduce new ones to SOA.

Different measurable aspects: OO metrics measure reusability by considering the critical aspects of classes such as number of methods in a class, depth of inheritance hierarchy, number of children, lack of cohesion in methods, coupling between classes, and number of responsibilities provided by the class [18]. These metrics are not applicable in BPEL context since BPEL constructs are totally different, meaning that there is no relationship between BPEL measurable factors and OO. Existing CBD metrics measure externally visible features of a component mainly interface methods [19] rather than internal logic of the component that is concerned in the context of BPEL services.

With respect to the aforementioned issues, we have to focus on the latest research work in reusability metrics in the SOA domain. In [23], the authors suggest measuring reusability based on the use of the service by service consumers. Specifically, the number of existing consumers of a service indicates the reusability of the service or service reuse index. Similarly, they defined the number of consumers of that operation across services and business processes as the extent of operation reusability or operation reuse index. Although their proposed metrics are technology neutral and can be also applied to both atomic and composite services, they neglect a salient point. That is, their proposed metrics do not contemplate any features that help architects to determine if the designed services can be reused less or more in the future. Apparently, current reusability of a service cannot be the only predictor of reusability.

In [24], the authors presented a quality model for evaluating reusability of services. Their quantitative model is based on the key quality attributes of reusability, including business commonality, modularity, adaptability, standard conformance, and discoverability that are derived from the key features of services in SOA. The authors' work is more complement and comprehensive compared with the work in [23], but it suffers from two shortcomings. First, although their metrics can be applied to both atomic and composite services, they do not consider the logic or control flow of composite services. Second, none of their proposed metrics have been empirically validated.

There are also some metrics that measure cohesion and coupling of services through which reusability of a service can be inferred implicitly [25], [26]. For instance, in [26], Perepletchikov *et al.* proposed a number of quantitative metrics for measuring cohesion of service. The proposed cohesion metrics can be accommodated during design time based on the operations exposed in service interface. However, the research does not consider structures that most services are constructed via standard XML-based business process definition languages. The approach is mainly limited to service consumer concerns during utilization of the service, and the service provider concerns such as logic or control flow of services have been neglected. Moreover, achieving a proper assessment of reusability of a service based on coupling or cohesion is not straightforward and needs to be demonstrated by various replications with different settings.

III. Metric Rationales

A. Reusability in SOA Context

BPEL process reusability is defined as the extent to which a BPEL process (i.e., composite service) can be reused in other contexts, organizations, or SOA solutions with minimal effort and change (minimal change will be specified in more detail throughout Section IV). Reusability in a service-oriented area is not an isolated concept, and architects should decide about this while other contexts and projects are contemplated. This means that an architect must analyze whether the given composite service can be reused in other business processes or domains. The term context, in this paper, is categorized to current and potential ones. Reusability in the current and potential contexts makes sense if we consider the point that every identified, specified, realized, and finally implemented BPEL process should be reused in different possible service-oriented solutions. Other possible assumptions could be those cases where services are going to be published and utilized in a broader range and new business ventures in which they could take advantage of the reuse capabilities and reusable assets. According to this assumption, the service reusability must be calculated with respect to the current and potential reusability.

B. Impact of Business Rule Changes on BPEL Process Reusability

By adopting the provided reusability definition, we aim to justify the rationales behind our metrics through proposing a research question that is the inspiring source of our metric research work: "*In which conditions a BPEL process cannot be reused?*" Apparently a service cannot be reused where it cannot be matched with new or potential requirements in other contexts or solutions in the future. Now, this question may arise: "*What are these requirements?*" These requirements are in fact business rules. To be more specific, we have to emphasize that although they are not the same concept, they do have close relationship with each other. One of the distinctions between them is that, unlike requirements (particularly software requirement), business rules convey and enforce the policy of an organization [27] and they are usually flexible due to the nature of organizational change and often consist of very granular steps. Subsequently, business rules are often subject to frequent changes. On the other hand, the most important sources of requirement changes are business rules [28].

The smallest unit of business change is a business rule change. Regarding business rules driving the business process [29], business rules and business processes are interrelated; whenever the rules change, processes may change accordingly. Wagner [30] classified business rules in four basic types, namely, integrity constraints, derivation rules, reaction rules, and deontic assignment rules.

According to the taxonomy of flexibility, there are five perspectives in business processes, including functional, organizational, behavioral, informational, and operational. However, as specified in [31], the aforementioned business rules primarily influence the informational and the behavioral perspectives of a business process.

The informational perspective defines the information that is exchanged between activities, whereas the behavioral perspective describes under which preconditions activities are executed. Integrity constraints specify preconditions for particular activity types in the business process model. For instance, "*After year 2012, patients without health card cannot place a request*" is an integrity constraint that primarily influences on the informational perspective. Derivation rules are statements that define new business concepts or facts based on the existing ones. For example, "*Old persons receives 15% discount for medical care*" is a derivation rule that makes an impact on the informational perspective.

Reaction rules and deontic assignment rules state which activities need to be executed and in what order. Reaction rules define behavior in the first person that participates in business process interactions, whereas deontic assignment rules specify control flows of a process in terms of a third-person viewpoint. These rules primarily influence the behavioral perspective of a business process.

Services are realized at a higher level of abstraction directly supporting business processes [32], on the condition that they are composed and orchestrated in accordance with business rules [33]. This means that the structure and arrangement of services within a composite one is imposed by business rules.

Therefore, any changes in the mentioned business rules lead to some changes in informational and behavioral perspectives of a business process. Since informational perspective refers to the information that is exchanged between the activities, in case of any change, its main impact can be traced in web service interfaces. This is why every web service has an interface and the messages are exchanged through web service interfaces. If the change occurs on behavioral perspective, the orchestration of execution control flow, which represents behavioral perspective of a business process, may change accordingly.

Any changes in business rules may provoke some kinds of mismatches for either interface or control flow of a composite service in foreseen solutions. Consequently, a composite service cannot be reused in the future when it cannot be matched with the business rules of other contexts or solutions.

C. Basic Concepts of Potential Reusability Estimation

Frakes and Kang in [12] state that "*A key element in the success of reuse is the ability to predict needed variabilities in future assets.*" In fact, predicting potential reusability is the function of estimating to what extent a BPEL process can be adapted with future requirements and business rules. A BPEL process can be reused in the future if it can be matched with minimal effort to the expected one in foreseen contexts. Such a hypothesis is also supported on the basis of the research work such as [34] that proposes a process matching approach for flexible workflow process reuse.

To sum up, we formalize some definitions.

Definition 1 (*Mismatch*): Mismatch refers to the point that a BPEL process cannot be matched with other context requirements. In this regard, there are two kinds of mismatches, including description and logic mismatch.

Definition 2 (*Description Mismatch*): Description mismatch denotes to the mismatch between the requirements and the description of the given composite service, i.e., Web Services Description Language (WSDL).When a service consumer, either another service or an SOA architect, wants to use a given service based on their requirements, it may not be reused due to mismatch in WSDL, including data types and messages of service operations. Apparently, some techniques could help an architect to utilize a service with mismatch in description; hence, based on our reusability definition, some modifications would be tolerable. Therefore, description MMP calculation of a composite service can provide new insight about its reusability.

Definition 3 (*Logic Mismatch*): Logic mismatch is defined as mismatch between the requirements and the logic of a given composite service. Composite SL is utilized through the control flow of basic and structured activities within it. Regarding different contexts may enjoy dissimilar business rules, consequently, a service cannot be reused in potential contexts for the sake of disparate business rule, which impose disparate structure and control flow, i.e., logic mismatch.

Definition 4 (*MMP*): MMP refers to the probability that a given composite service cannot be matched with other context requirements. Probability is a numerical measure of the likelihood of an event relative to a set of alternative events.

IV. BPEL Process Reusability

A. Description Mismatch Analysis

Each BPEL process is a web service and needs a WSDL document. A WSDL document describes a web service using some elements, including ⟨`types`⟩, ⟨`message`⟩, ⟨`portType`⟩, and ⟨`binding`⟩.

A client will usually invoke an operation on the BPEL process to start it. With the BPEL process WSDL, we specify the interface for this operation. We also specify all message types, operations, and port types a BPEL process offers to other partners. However, the source of description mismatches come from operation message number and their data types.

The ⟨`types`⟩ element encloses data-type definitions that are relevant for the exchanged messages. For maximum interoperability and platform neutrality, WSDL uses XML schema syntax to define data types. The XML schema distinguishes between primitive, derived, and complex data types [36].

Primitive data types: Primitive data types are those that are not defined in terms of other data types. Currently, XML schema defines 19 primitive data types [37].

Derived data types: Derived data types are those that are defined in terms of other data types. There are three kinds of derivation, namely, by restriction, by list, and by union. The XML schema has 25 derived built-in types [37].

Complex Definition 4 types: XML schema provides a flexible and powerful mechanism for building complex data structures from its simple data types. Data structures can be created via making a sequence of elements and attributes. Additionally, user-defined types can be extended in order to create even more complex types. XML schema has three compositor elements that allow constructing complex data types from simpler ones: sequence, choice, and all. The behavior of these compositor elements is defined as follows.

1) Sequence: All the complex type fields must be present in the exact order they are specified in the type definition.
2) All: All of the complex types' fields must be present but can be in any order.
3) Choice: Only one of the elements in the structure can be placed in the message.

Data types are different from each other on the basis of their usage and mismatch proneness. This is due to the fact that some types could be covered or cast by some other data types. In practice, for example, strings could be used much frequently across an application than many other types. A BPEL workflow matching only against strings may be much more reusable than the one matching against an application-specific data type. More exactly, string covers all data types (i.e., 19 primitive data types plus 25 derived types) and its w is the value of 44; hence, its coverage weight becomes the value of 1 (i.e., 44/44). In a similar approach, the Boolean and Float data types cover 1 and 15 data types; hence, their coverage weight becomes the value of 0.02 (i.e., 1/44) and 0.34 (i.e., 15/44), respectively.

Based on the mentioned hypothesis, description MMP calculation is done as follows:

MP	match probability (MP);
MMP	MMP;
MP_P	MP of input/output parameters data types;
MP_{Pk}	MP of input/output parameters data types of operation k;
MP_{CD}	MP of a complex data type;
MMP_{CD}	MMP of a complex data type;
MP_{SD}	MP of service description;
MMP_{SD}	MMP of service description;
d	total number of defined or built-in primitive and derived type in XML schema. Thus, d has the constant value of 44;
w_j	number of data types that are covered by each data type;
l	number of primitive, derived, or complex data types of an operation;
l_k	number of primitive, derived, or complex data types of operation k parameters;
n	number of simple or even another complex types within sequence, choice, or all element;
O_k	service operation k;
m	number of service operations.

Equation (1) calculates MP of both input and output parameter data types

$$MP_P = \begin{cases} \prod_{j=1}^{l} \left(\frac{w_j}{d}\right)_j, & \text{primitive type} \\ \prod_{j=1}^{l} \left(\frac{w_j}{d}\right)_j, & \text{derived type} \\ \prod_{j=1}^{l} (MP_{CD_j})_j, & \text{complex type.} \end{cases} \quad (1)$$

As specified earlier, 25 derived types are defined within the specification itself, and further derived types can be defined by users in their own schemas. We refer to those derived types out of the 25 built-in ones as complex types. As complex data types include some primitive data types with *sequence* or *choice* behavior, for example, hence, their match/MMP must be calculated in terms of these behaviors as follows:

$$MP_{\mathrm{CD}} = \begin{cases} \left(\frac{1}{n!}\right) \times \left(\prod_{i=1}^{n} \left(\frac{w_j}{d}\right)_i\right), & \text{sequence} \\ \left(\frac{1}{2^n}\right) \times \left(\prod_{i=1}^{n} \left(\frac{w_j}{d}\right)_i\right), & \text{choice} \\ \left(\prod_{i=1}^{n} \left(\frac{w_j}{d}\right)_i\right), & \text{all.} \end{cases} \quad (2)$$

The events of occurring one of the possible data types for each of the input/output parameters are *independent events*. Based on probability theory, assuming independence (i.e., the mismatch of either parameter data type is not influenced by the success or failure of the other parameter data type), the service description MMP can be calculated through probability calculation of the description not matching, or the matching of the service description [see (3)], and then the probability of service description mismatch is simply 1 (or 100%) minus the MP as follows:

$$MP_{\text{SD}} = \left(\prod_{k=1}^{m} \left(\frac{1}{l_k} \times MP_{P_k} \right)_{O_k} \right) \tag{3}$$

$$MMP_{\text{SD}} = \left(1 - \left(\prod_{k=1}^{m} \left(\frac{1}{l_k} \times MP_{P_k} \right)_{O_k} \right) \right). \tag{4}$$

This question may arise: "*Did the metric take the number of parameters as an important source of mismatch into account?*" The answer is positive. We accept that the more parameters lead to the more description MMP and the $1/l_k$ coefficient covers this issue.

B. Logic Mismatch Analysis

In addition to the description mismatch concept, a BPEL process cannot be reused in potential contexts for the sake of disparate business rule, which lead to disparate structure and logic, i.e., logic mismatch. For the purpose of logic MMP calculation, first, we have to calculate MP of each structured activity within a composite service. Each BPEL process consists of steps. Each step is called an activity. BPEL activities divided into basic and structured categories [38]. Basic activities, including ⟨`invoke`⟩, ⟨`receive`⟩, ⟨`reply`⟩, and ⟨`assign`⟩, are used for common tasks such as invoking a web service or manipulating data. Structured activities are used for arranging the structure of BPEL process. Structured activities can contain both basic and structured activities in order to implement complex business processes. The important and common use structured constructs are ⟨`sequence`⟩, ⟨`flow`⟩, ⟨`switch`⟩, ⟨`pick`⟩, and ⟨`while`⟩. The following sections explore how MP of BPEL process constructs is calculated.

1) ⟨`Sequence`⟩ MP: A ⟨`sequence`⟩ activity is used to define activities that need to be performed in a sequential order. The MP of ⟨`sequence`⟩ activity is calculated as follows:

$$MP_i = \left(\frac{2^n - 1}{n! + 2^n - 2} \right) \tag{5}$$

where n is the number of activities within a sequence, and i refers to ⟨`sequence`⟩ construct.

In sequence construct, first, we have to define when a composite service can be matched with potential solution without tough efforts. In our perspective, this can be achieved when the arrangement of activities does not change.

Therefore, the MP is directly dependent on the different ways of arranging the activities in a sequence, i.e., permutations. Thus, in case of n activities, we had $n!$ in addition to the number of omission cases (i.e., $2^n - 1$) as a total number of sample space. Since there is one permutation in both $n!$ and $2^n - 1$, we have to subtract them by the value of 1.

The numerator of (5) is the number of matches that are omission cases in addition to the unchanged case that is $2^n - 1$. Based on the probability theory, the MP of a service with one *sequence* activity is computed via (5). However, there is an exception when considering ⟨`sequence`⟩ activity in a BPEL code. It is expected that the ⟨`sequence`⟩ activity, which is used upon for synchronous invocation of web service operations, is not taken into account since permutation is not possible.

2) ⟨`Switch`⟩ MP: The ⟨`switch`⟩ activity is used to express conditional behavior. It consists of one or more conditional branches defined by ⟨`case`⟩ elements, followed by an optional ⟨`otherwise`⟩ element. The case branches of the switch are considered in alphabetical order. The MP of ⟨`switch`⟩ activity is calculated as follows:

$$MP_i = \left(\frac{1}{2^n} \right) \tag{6}$$

where n is the number of conditions, and i refers to ⟨`switch`⟩ construct.

The ⟨`switch`⟩ activity consists of an ordered list of conditions specified by a ⟨`case`⟩ element followed by one optional ⟨`otherwise`⟩ element. *Switch* activity can be changed based on its conditions. That is, each condition in a *switch* activity can or cannot be changed according to business rule. We compute the contingency of this behavior via the power set of the number of conditions, i.e., 2^n. Therefore, the MP involved in a *switch* construct is insignificant since the match case is just the one in which none of the conditions have been changed.

3) ⟨`Pick`⟩ MP: The ⟨`pick`⟩ activity is used to wait for the occurrence of an event and then perform an activity associated with the event. The MP of the ⟨`pick`⟩ activity is computed as follows:

$$MP_i = \left(\frac{2^n - 1}{2^{n+1} - 2} \right) = \frac{1}{2} \tag{7}$$

where n is the number of events, which are cached through ⟨`onAlarm`⟩ and ⟨`onMessage`⟩ activities, and i refers to ⟨`pick`⟩ construct.

For the purpose of calculating the MP of ⟨`pick`⟩ activity, we have to enumerate the number of match and mismatch cases. In *pick* construct, the numbers of match cases are those in which the event conditions such as messages have not been changed. Moreover, deleted cases in which some of events removed regarding the business rules can be also contemplated as match cases. Composite service with such modifications can be reused without tough efforts. In this matter, the total number of both match and mismatch cases is twice the match cases. Thus, the result gets the value of 1/2.

4) ⟨`Flow`⟩ MP: The ⟨`flow`⟩ activity provides concurrent execution of enclosed activities. The *flow* activity also allows the synchronization of activities within the *flow*. To do so, a *link* construct (⟨`Link`⟩) specifies dependence between a source and target activity.

The MP of *flow* activity is one when the activities within a *flow* are processed on the condition that the order of execution is not defined. In this matter, the match cases are just the omission cases that are $2^n - 1$. Since the numbers of unmatched cases are zero, the MP takes the value of one. This means that different versions of the same business rule do not affect the flow control of ⟨`flow`⟩ construct.

However, if we define any dependence between activities using ⟨`Link`⟩ construct, the MP is dependent upon the different ways of arranging the activities in a *flow* construct. Since in dependence behavior activities rely on each other, hence, unlike *sequence* construct, omission cases are not acceptable. Therefore, the MP of *flow* is calculated as follows:

$$MP_i = \begin{cases} \frac{2^n-1}{2^n-1} = 1, & \text{The flow construct with concurrency behavior} \\ \frac{1}{n!}, & \text{The flow construct with dependence behavior} \end{cases} \quad (8)$$

where i refers to ⟨`flow`⟩ construct, and n is the number of activities within a flow.

5) ⟨`While`⟩ MP: A ⟨`while`⟩ construct is used to define an iterative activity. The iterative activity is performed until the specified Boolean condition no longer holds true. The MP of ⟨`while`⟩ construct has the value of 0.5 as the match can be occurred at Boolean condition of the activity

$$MP_i = \frac{1}{2} \quad (9)$$

where i refers to ⟨`while`⟩ construct.

6) Logic Mismatch Analysis: In the previous sections, the MP of all kinds of structured constructs was calculated. Since a BPEL process may have more than one of them at the same time, it is expected to compute the MP in case of more than one structured activity in a BPEL process. Thus, logic MP of a BPEL process with any number of structured constructs is calculated through the following equation:

$$MP_{\mathrm{SL}} = \left(\prod_{i}^{n} MP_i\right) \quad (10)$$

where i refers to specific structured construct, n refers to the total number of structured constructs within a certain BPEL process, and MP_{SL} refers to the SL MP.

Logic MMP of a composite service with any number of structured constructs is calculated via the following equation:

$$MMP_{\mathrm{SL}} = \left(1 - \left(\prod_{i}^{n} MP_i\right)\right) \quad (11)$$

where MMP_{SL} refers to SL MMP.

C. BPEL Process Reusability Analysis

Now, we are able to analyze the BPEL process reusability. In this regard, first, we have to calculate a BPEL process total MMP that is both description and logic MMPs. A BPEL process total MMP is as follows:

$$MMP_{\mathrm{S}} = (1 - [(MP_{\mathrm{SD}}) \times (MP_{\mathrm{SL}})]) \quad (12)$$

where MMP_{S} refers to MMP of a BPEL process, MP_{SL} refers to the SL MP, and MP_{SD} refers to MP of service description.

Based on the calculation of BPEL process total MMP, BPEL process reusability can be computed as follows:

$$R_P = R_c \times (1 - MMP_{\mathrm{S}}) \quad (13)$$

where R_c refers to the numbers a BPEL process is reused in current context, and R_P refers to the potential reusability of a BPEL process in potential contexts.

Equation (13) computes the potential reusability of a given BPEL process with respect to the MMP of a BPEL process in the prospect contexts.

V. Validation of the Reusability Metric

A. Theoretical Validation

In this section, we aim to theoretically validate the proposed metrics using the reusability properties that are proposed in [23], which themselves have been inspired from the property-based framework of Briand *et al.* [39]. This framework is a mathematical generic framework that introduces some intuitive properties for salient concepts of software engineering such as complexity, coupling, cohesion, and size, through which researchers and practitioners could analyze and validate the theoretical grounds of their measures irrespective of a specific development paradigm. Therefore, we examine the proposed reusability metrics against four reusability properties [23], including nonnegativity, null value, monotonicity, and merging of modules.

Property 1 (Nonnegativity): This property is satisfied since, for a given BPEL process, the value of reusability metric is equal to zero when the service does not have any consumer or some positive value representing the potential reusability value of a BPEL process. As a result, it will never be negative under any circumstances.

Property 2 (Null value): This property is also satisfied since, in case of no consumers, R_c gets the value of zero; hence, the reusability of a BPEL process (i.e., R_P) gets zero too. Therefore, the reusability will be null in case there are no consumers.

Property 3 (Monotonicity): This property is satisfied because the reusability value of a BPEL process cannot be decreased when the number of consumers increases. To be more specific, by increasing the number of consumers, R_c becomes greater, and subsequently, R_P gets the higher value.

Property 4 (Merging of modules): This property is also satisfied since the reusability of a BPEL process obtained by merging two processes is not greater than the sum of reusability of the two original ones. This is due to the fact that the obtained process is more granular, hence has less potential reusability.

As proposed potential reusability metrics adhere to all the prescribed properties for the corresponding attribute, hence, it can be considered as a valid characterization of potential reusability of a BPEL process.

B. Empirical Validation

For the experiment to be successful, it needs to be wisely constructed and executed. Therefore, we have followed some suggestions, provided by Perry *et al.* [39] and Mendonca and Basili [41], about the structure and the components of a suitable empirical study. To perform an experiment, several steps have to be taken in a certain order. An experiment can be divided into the following main activities [41]: goals of the study, hypotheses, experimental protocol, threats to validity, data analysis and presentation, results, and conclusions. In the

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

remainder of this section, we will explain how these activities have been performed.

1) Goal of the Study: The main goal of this study is "*Analyzing the proposed reusability metrics to evaluate their predictive capability with respect to the design-level estimation of the potential reusability of the BPEL processes*."

2) Hypothesis Formulation: Hypotheses are essential as they state the research questions in a semiformal way. We present our hypothesis in two levels of abstraction.

Abstract Hypothesis: "The potential reusability R_P metric is a good predictor and accurate metric to evaluate the potential reusability of composite services."

Concrete Hypothesis: "There is a significant correlation between the potential reusability R_P metric values and the expert's rating of the potential reusability of a set of processes as our experimental material."

3) Experimental Protocol: Having formulated the hypotheses, the design of the experiment occurred precisely according to the guidelines. An experimental design is a detailed plan for data collection and the other experimental tasks that will be used to test the hypotheses. This phase also explains how the experiment was conducted and has several components that are fully described in order to provide useful information for future replications.

Variable selection: Typically, in empirical studies, there are two kinds of variables, including dependent and independent, that their cause and effect should be evaluated by testing the hypothesis with appropriate techniques. In this paper, they are as follows: 1) The independent variable is the structure of BPEL processes; 2) the dependent variable is the potential reusability of processes, which varies when the structure of BPEL processes changes.

Expert selection: The experts we selected were students of the Faculty of Electrical and Computer Engineering enrolled in the Master's and Ph.D. programs in computer engineering at Shahid Beheshti University, Tehran, Iran. Twenty experts were selected according to the evaluations of their lecturers. Half of the participants were male in the 24–50 age groups, and the rest were female in the 23–29 age groups. Some of them had extensive industrial experience in several areas, but none had experience with business process management systems. By the time the experiment was done, all the students had taken a 50-h course on ultralarge-scale systems with emphasis on service-oriented systems, business processes modeling, and service composition, therefore gaining experience in the design and development of services particularly business services. To enhance their knowledge about service modeling, a group-based training session was carried out before doing the experiment. This session consisted of an introduction to BPEL, its constructs, and the quality attributes of a BPEL process.

Experimental design: The objects supposed to be rated by experts were process-based services graphically designed with the Eclipse BPEL designer. The material consisted of 70 professionally designed composite services of the different universe of discourses such as university, core banking, and travel agency with different structural characteristics and degrees of complexity. The participants were told how to carry out the experiment with exactly the same set of 70 predesigned composite services. In order to make the experience and knowledge of the participants more comparable, we made 10 groups out of the 20 participants.

The participants could use unlimited time to make a consensus based on their judgments. Our key aim to give them unlimited time was based on the fact that we did not intend to rush their consensus. During the workshop, they felt free to discuss enough to make a consensus. However, since the group's experiences were comparable, they finished rating after 3 h. In order to gather more precise rating, authors decided to design a professional questionnaire consisting the following questions that each of which investigates one aspect of service reusability. The first two questions are related to description, whereas the others are concerned with SL.

1) To what extent is the composite service interface granular?
2) To what extent is the composite service interface complex?
3) To what extent is the composite SL granular?
4) To what extent is the composite SL complex?
5) To what extent is the composite service abstract?
6) To what extent is the composite service context independent (decouple from context)?

These aspects, including complexity, coupling (context-dependence), granularity, abstraction, etc., are in fact the inherent indicators of service reusability [16], [42]. Although the rating score does not directly indicate the reusability of a service, the aforementioned characteristics that were considered in the review tend to affect the service reusability [15]. Therefore, the rating score reflects the reusability of a service and also the quality factors that have impact on the reusability. The reason of gathering information in an indirect manner is according to statistics since a direct questionnaire may lead to bias results. Moreover, we believe that the respondents could not rate reusability attribute intuitively.

The independent variable was measured using the R_P metric formulated in Section IV. The dependent variable was measured according to average of the expert's ratings for each question that has been rated by the participants from 1 to 10. These values are contemplated as to be on an interval scale for the analysis since the difference between two values is meaningful. Additionally, the questions are related to the reusability in the opposite manner, except the last one that has positive effect on it. It means that, when the participants rate this question, the small numbers indicate worse situation in the viewpoint of them.

4) Analysis of the Results: Since the experts rated services using a numerical scale, we have selected quantitative analysis to draw conclusions from the data. The qualitative analysis was done in conjunction with a number of statistical analyses.

Descriptive statistics: The descriptive statistics are presented in Table I for each independent and dependent variable.

Hypothesis testing: In this section, it is investigated if any correlation exists between experts' ratings and the proposed R_P metric value. The first step in correlation analysis is to ascertain whether the distribution of the data is normal; hence, the Kolmogorov–Smirnov test was applied. As it is obtained that the distribution was not normal, authors decided to use a

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
DESCRIPTIVE STATISTICS

		Mean	Std. Deviation	Variance	N
Group Index	1	.71	.15	.02	70
	2	.69	.15	.02	70
	3	.72	.14	.02	70
	4	.78	.13	.02	70
	5	.60	.16	.03	70
	6	.73	.13	.02	70
	7	.71	.14	.02	70
	8	.65	.10	.10	70
	9	.73	.13	.02	70
	10	.70	.13	.02	70
R_P		.30	.34	.12	70

TABLE II
SPEARMAN CORRELATION COEFFICIENT BETWEEN EXPERTS' RATINGS (THE ACADEMICS) AND THE VALUES GIVEN BY THE R_P METRIC

		r_s	$\alpha 1$	N
Group Index	1	.530	Reject H0	70
	2	.585	Reject H0	70
	3	.580	Reject H0	70
	4	.610	Reject H0	70
	5	.538	Reject H0	70
	6	.561	Reject H0	70
	7	.521	Reject H0	70
	8	.479	Reject H0	70
	9	.589	Reject H0	70
	10	.600	Reject H0	70
Average		.635	Reject H0	70

nonparametrical statistical test, i.e., the Spearman correlation [43] with the level of significance of $\alpha 1 = 0.01$, which indicates the probability of rejecting the null hypothesis when it is certain (type-I error). The Spearman r_S is a nonparametric statistic used to show the predictive relationship between the two variables, which are expressed as ranks (the ordinal level of measurement). The correlation coefficient is a measure of the ability of one variable to predict the value of another one. This does not imply the existence of any causal relationship between the two variables. Indeed, it just indicates a predictive relationship. Using Spearman's correlation coefficient, the R_P metric was correlated separately to the different experts' rates of reusability. In the experiment, the null hypothesis was:

H_0 : "there is no significant correlation between the R_P metric and the experts' rating of potential reusability."

H_1 : "there is a significant correlation between the R_P metric and the experts' rating of potential reusability."

The probability that the null hypothesis would be erroneously rejected was controlled with confidence level $\alpha 1 = 0.01$. An analysis performed on the collected data and Table II denotes the summary of results describing the Spearman correlation coefficient between experts' ratings and the rates given by the R_P metric in the experiment. The absolute value of the correlation coefficient indicates the strength, with larger absolute values indicating stronger relationships. The significance level (or p-value) is the probability of obtaining results as extreme as the one observed. Based on the results, which are denoted in Table II and taking $\alpha 1$ into consideration, correlation is significant at the 0.01 level for 100% of the experiment; therefore, the null hypothesis was rejected with 99% confidence.

Power analysis of the hypothesis test: A test without sufficient statistical power will not produce enough information to convince the researcher regarding the acceptance or rejection of the null hypothesis [44]. Since we are using relatively medium samples, the authors provide details of power analysis in this section to interpret the results. The power of the nonparametric tests is determined by appropriate parametric tests [44]. Therefore, we have used Pearson's r [45] for the Spearman rank correlation.

Given: 1) observation size of 70 from 10 groups of participants that produce 700 data points; 2) level of $\alpha = 0.01$; and 3) the effect size of 0.6; the achieved power is 0.81 that is considerably above the accepted norms [44] in software engineering studies. However, by the effect sizes lower than 0.6, the power would dramatically shrink, thereby suggesting that the statistical tests could produce a type-II error [44] (i.e., fail to reject a null hypothesis when it is in fact false) when the test does not exhibit large effect sizes.

Estimation of the R_P confidence interval: Metrics without any thresholds or intervals for measurement values may not be effectively used [19]. Therefore, we decided to calculate the confidence intervals of the metrics. We calculated the confidence intervals of R_P in terms of the reusable services using the rating scores. First, we assumed that the reusability of services that satisfy average rating score higher than 0.75 is high. Second, by an interval estimation based on the Wilcoxon signed-rank (non-parametric) test statistic [46], we calculated confidence intervals with confidence coefficient of 95% for R_P using all services that satisfy the condition among all samples. The number of such high-rated services was 29. The average of measurement values of all samples is 0.29, and the number of services whose measurement values are in the confidence interval is 8. The lower confidence limit is 0.375, and the upper confidence limit is 0.625.

Among all samples, the percentage of services whose values of R_P are in the confidence interval is only 11%. This means that the confidence interval of R_P is not a general range where every service corresponds. Therefore, it is evident that R_P with its confidence interval can be effectively used for measuring the potential reusability of services.

From the obtained confidence interval, users of the proposed metric can judge the potential reusability of a service at hand. From statistical point of view, if the potential reusability of a service were in the confidence interval, then by 95%, it is possible that the experts will give it a rate higher than 0.75. Moreover, it might be possible that the potential reusability was not in the confidence interval or even higher than the upper limit, then it should be evaluated what causes undermining of the other quality factors for the sake of reusability.

5) Threats to Validity: Threats to validity may cast several effects on the interpretation of the experimental data. Thus, in this section, we discussed various threats to validity in which they were categorized according to [39] and the way we attempted to alleviate them is as the way that Cardoso has conducted it in [47].

Construct validity: The independent variable that estimates the potential reusability of BPEL processes can be considered constructively valid because they are defined in a formal manner and also theoretically validated.

Internal validity: We have considered different aspects that could threaten the internal validity of the study such as differences among experts, precision of experts' ratings, learning effects, fatigue effects, and experts' incentive.

1) Differences among experts: We grouped the participant as previously described; therefore, error variance due to differences among participants was reduced. Experts involved in this experiment had medium experience in service design, and they attain required knowledge of BPEL process design; their ratings can be considered significant.
2) Learning and fatigue effects: Since the experimental materials were in the different universe of discourses with different structural characteristics and degrees of complexity and the participants were in the small groups discussing on the issues to reach an appropriate consensus, it is very unlikely that any potential learning and fatigue affects the data.
3) Anticipation effects: The participants were not told about the hypothesis that we wanted to test in order to ensure that expectations about specific levels of treatment did not influence their rates.

External validity: One threat to external validity is the number of participants that is limited to 20. This threat can limit the ability to generalize the results to settings outside the study.

Experimental materials and environment: The materials of our study are represented with BPEL and WSDL, which both of them were standard and are utilized in an industrial environment. Additionally, they were representatives of real-world BPEL processes in terms of size and complexity.

VI. Discussion and Future Direction

In the experiment explained in Section V, authors came across with a number of cases in which the R_P values were far from the experts' rating. More investigation shows by determining the weights for MP_{SD} and MP_{SL} the results will be more accurate and reliable. In fact, MP_{SD} and MP_{SL} seem to have different weights in constituting R_P. In this regard, the correlation between R_P, MP_{SD}, MP_{SL}, Experiments Average (ExpAve), Experiments Average value for SL (ExpLogic), and Experiments Average value for service description (ExpDes) were calculated. According to the obtained results, the correlation between ExpAve and ExpLogic (i.e., 0.968) is considerably more than the correlation between ExpAve and ExpDes (i.e., 0.508). In this matter, the behavior of our metrics is a little bit different in which the correlation between R_P and MP_{SL} (i.e., 0.500) is close and a little less than the correlation between R_P and MP_{SD} (i.e., 0.628). Therefore, we have to define some weights for MP_{SL} and MP_{SD} by means of some methods particularly regression, which is part of our future work.

Additionally, some interesting results were deduced. For instance, the correlation between ExpDes and ExpLogic is insignificant (i.e., 0.323), which is somewhat in accompany with the correlation between MP_{SD} and MP_{SL}. This means that the description and logic mismatch are the independent variables; thereby, it is possible to simply multiply one by the other as it is applied in (12). Moreover, the correlation between MP_{SL} and ExpLogic is noticeably significant (i.e., 0.800) and also the correlation value between MP_{SD} and ExpDes is high (i.e., 0.574). These two correlations further confirm that MP_{SL} and MP_{SD}, respectively, are the correct indicators of composite SL and description MP.

VII. Conclusion

In this paper, we have proposed a metric for a BPEL process potential reusability analysis. The approach is based on description and logic MMP of a BPEL process that will be reused within potential contexts. The mismatch probabilities were combined to a metric formula for quantifying the probability of potential reuse of BPEL processes. We also reported results from an experiment designed to investigate its validity. The obtained results reveal that there exists a reliable statistical correlation between the proposed metric and the experts' judgments.

References

[1] J. Hurwitz, R. Bloor, and C. Baroudi, Thinking from Reuse SOA for Renewable Business, 2006. [Online]. Available: ftp://ftp.software.ibm.com/software/soa/pdf/IBMThinkingfromReuse.pdf

[2] M. Herr, U. Bath, and A. Koschel, "Implementation of a service oriented architecture at Deutsche post mail," in *Proc. ECOW LNCS*, L.-J. Zhang and M. Jeckle, Eds., Berlin, Germany, 2004, vol. 3250, pp. 227–238, Springer-Verlag.

[3] Z. Ma and F. Leymann, "BPEL fragments for modularized reuse in modeling BPEL processes," in *Proc. 5th ICNS*, Washington, DC, USA, 2009, pp. 63–68.

[4] M. Rosemann, "22 potential pitfalls of process modeling," *Bus. Process Manage. J.*, vol. 11/12, pp. 1–13, 2006.

[5] R. Khalaf, "From RosettaNet PIPs to BPEL processes: A three level approach for business protocols," *Data Knowl. Eng.*, vol. 61, no. 1, pp. 23–38, Apr. 2007.

[6] W. Lam, "Process reuse using a template approach: A case study from avionics," *ACM SIGSOFT Softw. Eng. Notes*, vol. 22, no. 2, pp. 35–38, Mar. 1997.

[7] P. Fettke and P. Loos, "Using reference models for business engineering—State-of-the-art and future developments," in *Proc. Innov. Inf. Technol.*, 2006, pp. 1–5.

[8] P. Fettke, P. Loos, and J. Zwicker, "Business process reference models: Survey and classification," in *Proc. Bus. Process Manage. Workshops*, 2006, vol. 3812, pp. 469–483, Springer-Verlag.

[9] M. Tanaka1, T. Ishida, Y. Murakami, and D. Lin, "Service supervision patterns: Reusable adaptation of composite services," in *Proc. Cloudcomp*, vol. 34, *LNICST*, D. R. Avresky, M. Diaz, A. Bode, and E. Dekel, Eds., 2010, pp. 147–163.

[10] D. J. Artus, "SOA realization: Service design principles," IBM Developer Works, 2006. [Online]. Available: http://www-128.ibm.com/developerworks/webservices/library/ws-soadesign/

[11] J. J. Marshall and R. R. Downs, "Reuse readiness levels as a measure of software reusability," in *Proc. IEEE IGARSS*, Jul. 7–11, 2008, vol. 3, pp. III-1414–III-1417.

[12] W. B. Frakes and K. Kang, "Software reuse research: Status and future," *IEEE Trans. Softw. Eng.*, vol. 31, no. 7, pp. 529–536, Jul. 2005.

[13] A. Khoshkbarforoushha, P. Jamshidi, and F. Shams, "A metric for composite service reusability analysis," in *Proc. Int. WETSoM ICSE*, Cape Town, South Africa, 2010, pp. 67–74.

[14] L. J. Cronbach and P. E. Meehl, "Construct validity for psychological tests," *Psychol. Bull.*, vol. 52, no. 4, pp. 281–302, 1955.

[15] M. P. Papazoglou and W. J. van den Heuvel, "Service oriented design and development methodology," *Int. J. Web Eng. Technol.*, vol. 2, no. 4, pp. 412–442, Jul. 2006.

[16] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*. Upper Saddle River, NJ, USA: Prentice-Hall, 2005.

[17] G. Feuerlicht and J. Lozina, "Understanding service reusability," in *Proc. 15th Int. Conf. Syst. Integr.*, Prague, Czech Republic, Jun. 10–12, 2007, pp. 144–150.

[18] S. R. Chidamber and C. F. Kemerer, "A metrics suite for object oriented design," *IEEE Trans. Softw. Eng.*, vol. 20, no. 6, pp. 476–493, Jun. 1994.

[19] H. Washizaki, H. Yamamoto, and Y. Fukazawa, "A metrics suite for measuring reusability of software components," in *Proc. 9th Int. Symp. Softw. METRICS*, Washington, DC, USA, Sep. 3–5, 2003, pp. 211–223.

[20] P. Vitharana, "Risks and challenges of component-based software development," *Commun. ACM*, vol. 46, no. 8, pp. 67–72, Aug. 2003.

[21] M. Perepletchikov, C. Ryan, and Z. Tari, "The impact of service cohesion on the analyzability of service-oriented software," *IEEE Trans. Serv. Comput.*, vol. 3, no. 2, pp. 89–103, Apr.–Jun. 2010.

[22] M. Perepletchikov, C. Ryan, and K. Frampton, "Comparing the impact of service-oriented and object-oriented paradigms on the structural properties of software," in *Proc. 2nd Int. Workshop MIOS OTM*, Oct. 2005, pp. 431–441.

[23] R. Sindhgatta, B. Sengupta, and K. Ponnalagu, "Measuring the quality of service oriented design," in *Proc. ICSOC-ServiceWave*, vol. 5900, *LNCS*, Berlin, Germany, 2009, pp. 485–499, Springer-Verlag.

[24] S. W. Choi and S. D. Kim, "A quality model for evaluating reusability of services in SOA," in *Proc. 10th IEEE Conf. E-Comm. Technol., 5th IEEE Conf. Enterprise Comput., E-Comm. E-Serv.*, Jul. 2008, pp. 293–298.

[25] M. Perepletchikov, C. Ryan, K. Frampton, and Z. Tari, "Coupling metrics for predicting maintainability in service-oriented designs," in *Proc. ASWEC*, 2007, pp. 329–340.

[26] M. Perepletchikov, C. Ryan, and K. Frampton, "Cohesion metrics for predicting maintainability of service-oriented software," in *Proc. QSIC*, 2007, pp. 328–335.

[27] J. Krebs, Dissecting Business From Software Requirements. [Online]. Available: http://www.ibm.com/developerworks/rational/library/aug05/krebs/index.html

[28] W. M. N. Wan-Kadir and P. Loucopoulos, "Relating evolving business rules to software design," *J. Syst. Architect.*, vol. 50, no. 7, pp. 367–382, Jul. 2004.

[29] K. Geminiuc, "A services-oriented approach to business rules development," SOA Best Practices: The BPEL Cookbook.

[30] G. Wagner, "The agent–object–relationship meta-model: Towards a unified view of state and behavior," *Inf. Syst.*, vol. 28, no. 5, pp. 475–504, Jul. 2003.

[31] S. Goedertier and J. Vanthienen, "Compliant and flexible business processes with business rules," in *Proc. 7th Workshop Bus. Process Model.*, 2006, pp. 94–104.

[32] O. Zimmermann, P. Krogdahl, and C. Gee, Elements of Service-Oriented Analysis and Design, 2004. [Online]. Available: http://kacit.ru/upload/iblock/853/8537b4f35143ad7f263b1092faa34a31.pdf

[33] J. Yu *et al.*, "Synthesizing service composition models on the basis of temporal business rules," *J. Comput. Sci. Technol.*, vol. 23, no. 6, pp. 885–894, Nov. 2008.

[34] H. Zhuge, "A process matching approach for flexible workflow process reuse," *Inf. Softw. Technol.*, vol. 44, no. 8, pp. 445–450, Jun. 1, 2002.

[35] J. Wang, Develop SOA Solutions for Healthcare Organizations Using Business-Driven Development. [Online]. Available: http://www.ibm.com/developerworks/webservices/library/ws-soa-bddhealth/

[36] W3C: XML Schema Part 2: Datatypes Second Edition. [Online]. Available: http://www.w3.org/TR/xmlschema-2/

[37] XML built in data type hierarchy. [Online]. Available: http://msdn.microsoft.com/en-us/library/ms256131.aspx

[38] M. Juric, B. Mathew, and P. Sarang, *Business Proess Execution Language for Web Services*, 2nd ed. Birmingham, U.K.: Packt Publishing, 2006.

[39] L. C. Briand, S. Morasca, and V. R. Basili, "Property-based software engineering measurement," *IEEE Trans. Softw. Eng.*, vol. 22, no. 1, pp. 68–86, Jan. 1996.

[40] D. E. Perry, A. A. Porter, and L. G. Votta, *Empirical Studies of Software Engineering: A Roadmap, in the Future of Software Engineering*, A. Finkelstein, Ed. New York, NY, USA: ACM Press, 2000.

[41] M. G. Mendonca and V. R. Basili, "Validation of an approach for improving existing measurement frameworks," *IEEE Trans. Softw. Eng.*, vol. 26, no. 6, pp. 484–499, Jun. 2000.

[42] C. Steghuis, "Service granularity in SOA projects: A trade-off analysis," M.S. thesis, Bus. Inf. Technol., Univ. Twente, Enschede, The Netherlands, 2006.

[43] S. Siegel and N. J. Castellan, Jr., *Nonparametric Statistics for the Behavioural Sciences*. New York, NY, USA: McGraw-Hill, 1988.

[44] T. Dyba, V. B. Kampenes, and D. I. K. Sjoberg, "A systematic review of statistical power in software engineering experiments," *Inf. Softw. Technol.*, vol. 48, no. 8, pp. 745–755, Aug. 2006.

[45] W. L. Hays, *Statistics*, 5th ed. New York, NY, USA: Harcourt Brace, 1994.

[46] E. L. Lehmann, *Non-Parametric: Statistical Methods Based on Ranks*. San Francisco, CA, USA: Holden-Day, 1975.

[47] J. Cardoso, "Process control-flow complexity metric: An empirical validation," in *Proc. IEEE Int. Conf. SCC*, 2006, pp. 167–173.

Alireza Khoshkbarforoushha received the M.Sc. degree in information technology engineering from Tarbiat Modares University, Tehran, Iran. He is currently a Ph.D. candidate in the College of Engineering and Computer Science, The Australian National University, Canberra, Australia.

He is also a Graduate Researcher with the Commonwealth Scientific and Industrial Research Organisation (CSIRO), Canberra. His research interests include data-intensive computing and cloud computing.

Pooyan Jamshidi received the B.S. and M.S. degrees in computer science and systems engineering from Amirkabir University of Technology (formerly, Tehran Polytechnic), Tehran, Iran, in 2003 and 2006, respectively. He is currently a Ph.D. candidate in the School of Computing, Faculty of Engineering and Computing, Dublin City University, Dublin, Ireland.

He is currently with The Irish Centre for Cloud Computing and Commerce (IC4), Dublin City University. His general research interests are in the field of software engineering, and his focus lies predominantly in the areas of cloud computing, software architecture, model-based self-adaptation, and self-adaptive software. He has served as a reviewer for six software engineering journals and conferences.

Mahdi Fahmideh Gholami is currently working toward the Ph.D. degree in the School of Information Systems, Technology and Management, University of New South Wales, Sydney, Australia.

His research interests include cloud migration, service-oriented software engineering, software development methodologies, and method engineering.

Lizhe Wang (M'06–SM'09) is a Professor with the Chinese Academy of Sciences, Beijing, China. His research interests include data-intensive computing, high-performance computing, and grid/cloud computing.

Prof. Wang is a Fellow of The Institution of Engineering and Technology and the British Computer Society.

Rajiv Ranjan received the Ph.D. degree in engineering from the University of Melbourne, Melbourne, Australia, in 2009.

He is a Senior Research Scientist and a Julius Fellow with the Commonwealth Scientific and Industrial Research Organisation (CSIRO), Canberra, Australia. He is an expert in datacenter cloud computing, application provisioning, and quality-of-service optimization in distributed systems. He has published 81 scientific peer-reviewed papers (i.e., seven books, 35 journals, 31 conferences, and five book chapters).