Prioritization of Services for SOA Performance Improvement Using Multidigraphs to Represent BPEL Business Processes

Rodrigo Mendes Leme*1, Marcelo Novaes de Rezende2

*1,2Computer Engineering, Institute for Technological Research (IPT), São Paulo, Brazil
2Information Systems, College of Engineering and Management (ESEG), São Paulo, Brazil
*1rmleme@gmail.com; 2mrezende@eseg.edu.br

Abstract- **Due to properties such as interoperability and reuse, there is an increasing adoption of Service-Oriented Architecture (SOA) in organizations. The combined use of SOA and Business Process Execution Language (BPEL) allows automating business processes. Analyzing potential performance issues in a SOA can be a complex task, especially when there is a high number of business processes and services. In this context, an alternative is to prioritize: select the most relevant services for performance improvement. To facilitate this prioritization, it is proposed a definition of service relevance, based on a representation of the business processes of a SOA in the form of a multidigraph, along with performance metrics collected from the logs of the BPEL orchestrator. A tool that automatically calculates the relevance of each service was also developed, which was tested in an experimental SOA. The results of that experiment showed the prioritization done according to the index of service relevance allowed an improvement of the average performance of the SOA.**

Keywords- SOA; BPEL; Prioritization of Services; Performance Improvement; Multidigraph

I. INTRODUCTION

SOA (Service Oriented Architecture) has emerged as an alternative to the development of enterprise systems, due to properties as greater interoperability, reuse and organizational agility [1]. Such properties contribute to its increasing use in organizations [2, 3].

According to [4], SOA, through service composition, can be seen as a mechanism for implementation of business processes (the activities performed by them to generate results to a client). That is relevant when one considers the current organizations are increasingly a great set of these processes [5].

Through the concept of orchestration, and with BPEL specification (Business Process Execution Language) [6], organizations can automate their business processes in a SOA, representing the business logic in a standardized and service-based way [1]. In fact, BPEL has become the standard language for service orchestration [2, 7, 8].

The greater interoperability and flexibility of SOA [1] have as a possible trade-off the performance quality attribute, due to factors such as distributed processing, calls to directory services and the existence of intermediaries between services [9]. In SOA implementations that use web services with SOAP (Simple Object Access Protocol) [10], the data format of the protocol, XML (Extensible Markup Language) [11], also influences the performance. This occurs due to the size of the messages and the operations of parsing, validation and transformation of XML documents. These three operations require more memory and processing [1, 9]. Performance is therefore an issue in SOA implementations with web services, and strategies to mitigate this issue are desirable [1].

There are methods to minimize the loss of performance caused by SOA, such as [2], [9] and [12]. Another possibility, when it is not feasible to improve the performance of all services of a SOA, is to prioritize a subset of them (so the selected subset of services may be refactored, resulting in a better overall performance improvement). This leads to a possible question prior to the improvement effort: given a set of BPEL business processes, how to select the most relevant services for performance improvement? To answer this, it is necessary to define in advance "service relevance" in the context of this work.

As will be seen ahead, the variables considered in this work for the relevance of a service are:

a) Number of business processes in which the service is involved: it is logically reasonable to assume the greater the number of business processes of which the service is part of, the greater its relevance.

b) Number of service invocations over time: performance metrics of the service are also needed to determine its relevance. One such metric is the number of service invocations over time. Collecting the logs from the BPEL orchestrator allows to obtain this metric.

c) Service response time over time: another performance metric required to calculate the relevance of a service is its response time[1], which is also collected from the logs of the BPEL orchestrator.

d) Expected response time for the service: the service response time alone is only a metric. It is the comparative analysis of this metric with the expected response time that allows to come to conclusions regarding the relevance of the service for performance improvement. For example, for a given service, if the expected response time is 5 s and the response time is 3 s, its relevance would be lower; but if the expected response time is 1 s, its relevance for the improvement would be greater. The expected response time of a service can be found in documents like SLA (Service Level Agreement) [13].

Particularly, to generate variable (a), it is proposed a representation of SOA and its multiple business processes in the form of a multidigraph. This abstraction allows representing the control flow of BPEL and the sharing (reuse) of services between business processes [14]. Furthermore, by labeling the vertices and arcs of the multidigraph, it is also possible to represent the other three variables.

How to prioritarily select, through a numerical index of service relevance, a subset of the services of a SOA whose response times will be decreased, aiming at the performance improvement of SOAs with BPEL business processes? As basis for the index, it is considered a representation of all these processes in a multidigraph [14]. Such questioning constitutes the problem which is subject of research in this paper, organized as follows. Section II contains related work. The index of service relevance is shown in Section III. Section IV elaborates on the experiment done to evaluate the index. Finally, Section V concludes the article.

II. RELATED WORK

There are approaches in the state of the art to mitigate the performance issues of SOA. Among those approaches studied, three were considered. One approach uses representations, in the form of graphs, of the business processes of a SOA. The other uses operations research techniques. Finally, the last one relies on architectural analysis.

OptBPEL [2] is a tool to improve the performance of business processes based on BPEL. It uses two methods of analysis (synchronization and concurrency) to perform optimizations. Both use graphs as basis: in the synchronization analysis, the BSG (BPEL Segment Graph), applied to the link elements of the business process; in the concurrency analysis, the TCFG (Threaded Control Flow Graph) and PDG (Program Dependence Graph).

Reference [12] proposed a method for performance improvement of a SOA based on the redirection of the messages exchanged between services. To this end, this method relies on a SIG (Service Invocation Graph). The SIG, through successive refactorings [15] in the source code of the services, is transformed into a new SIG. From this refactored version is extracted a SRG (Service Redirection Graph), so that invocations between services are replaced by others with a lesser communication overhead.

In the framework SPEWS [16] it is proposed an architecture for transforming a business process specified in BPEL into a GSPN (Generalized Stochastic Petri Net). The GSPN allows evaluating the performance of the business process during design phase yet, before its execution.

Operations research techniques were used in [17] to develop a performance analysis model and make predictions regarding BPEL business processes. With such model available, a SOA administrator could estimate the expected durations of new processes, looking for possible performance degradations and SLA violations.

Reference [9] discusses the architectural factors which negatively affect the performance of a SOA (already mentioned in Section I). Beyond a careful design previous to the implementation of the SOA, the architect should consider strategies to attenuate such factors, like caching of the service location, or deploying the service in multiple nodes (thus allowing load-balancing).

Although these studies show how to improve the performance of a SOA [2, 9, 12], or how to evaluate its performance earlier in a project [16, 17], they do not define which subset of the services of a SOA to prioritarily select in a performance improvement effort.

III. INDEX OF SERVICE RELEVANCE

In Section I, it mentioned the variables to consider for the calculation of the relevance of a service: number of business processes in which it is involved, number of invocations, response time and expected response time. The composition of these four variables determines the relevance of a service[2]. To this end, it is quantitatively defined how each one of them influences the calculation of the relevance, according to the following rationale:

[1] In this paper, the performance of a service is measured by its response time, according to the definition of [4], which includes both the transmission and processing times.

[2] The proposed formulation for the index of service relevance, using these four variables, is also part of the research hypothesis, being tested in the experiment.

a) The greater the number of business processes of which the service is part of, the greater its relevance. Such value can be obtained from the analysis of the representation of business processes in the multidigraph. It is defined as the number of incoming arcs of the vertex representing service i (id i). This number is normalized by the average number of incoming arcs of all other vertices of the SOA under analysis.

DEFINITION 3.1 Static relevance of service i: $r_{si} = \frac{id\ i}{\frac{1}{n}\sum_{j=1}^{n} id\ j}$, where id i is the indegree of vertex i.

b) The more the service is invoked, the greater its relevance in the SOA. This value is directly taken from the execution logs of the SOA under analysis. It is normalized by the average number of invocations of all other SOA's services.

DEFINITION 3.2 Dynamic relevance of service i: $r_{di} = \frac{inv(i)}{\frac{1}{n}\sum_{j=1}^{n} inv(j)}$, where inv(i) is the number of invocations of service i.

c) The average response time of the service, when seen as a quotient of its expected response time, provides a measure of how much that service is performing above (quotient > 1), equal to (quotient = 1) or below (quotient < 1) what is defined in its SLA, as shown in the example cited in Section I. The greater this quotient, the more the service is relevant for performance improvement.

DEFINITION 3.3 Performance relevance of service i: $r_{pi} = \frac{\Delta t_i}{SLA_i}$, where Δt_i is the average response time of service i and SLA_i is the expected response time of service i.

According to the objective of this work, for purposes of performance improvement it was considered a value that was a composition of the three definitions, in order to reach a unique index.

This composition is made by multiplying the three components. As the three are ratio scales [18] and are dimensionless, one can multiply/divide a ratio scale by another, thus arriving at a third ratio scale, which is a consolidated view of these three other components: $r_i = r_{si} \times r_{di} \times r_{pi} = \frac{id\ i}{\frac{1}{n}\sum_{j=1}^{n} id\ j} \times \frac{inv(i)}{\frac{1}{n}\sum_{j=1}^{n} inv(j)} \times \frac{\Delta t_i}{SLA_i}$, where r_i is the relevance of service i.

It is important to emphasize that the index of relevance only provides a consolidated view of r_{si}, r_{di} and r_{pi}. Each individual component allows the SOA administrator to make separate analyses. With r_{si} he can, e.g., verify which services are the most critical, for being part of multiple business processes. With r_{di} it is possible to know the most accessed services, allowing preventive actions to be taken so they always have connectivity. And r_{pi} provides visibility of services with potential performance issues.

Particularly on r_{pi}, there may be services where Δt_i is much greater than SLA_i. In that case, the services in question need specific attention, as it could mean a sign of major faults in their execution, and not only a mere tuning for performance improvement. Before starting any performance improvement effort based on the index of relevance, the architect should do a separate analysis of these services.

To exemplify the use of the index of service relevance, it was applied to an arbitrary SOA, containing three business processes, as can be seen in Fig. 1. The SOA is already represented as a multidigraph.

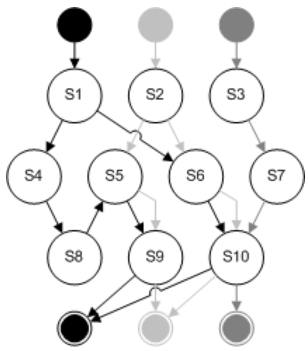

Fig. 1 Multidigraph representing an arbitrary SOA

As this is an example, the values for the number of invocations, average response time and expected response time of the services were arbitrarily filled. The number of incoming arcs of the vertices is obtained by analyzing the multidigraph. Table I presents these values and the index of relevance calculated for each service, in addition to its three components.

TABLE I INDEX OF SERVICE RELEVANCE OF FIG. 1 SERVICES

Service	id i	r_{si}	inv(i)	r_{di}	Δt_i (s)	SLA_i (s)	r_{pi}	r_i
S1	1	0.67	50	0.97	3	5	0.6	0.39
S2	1	0.67	60	1.17	3.5	3	1.17	0.92
S3	1	0.67	80	1.55	5.5	5	1.1	1.14
S4	1	0.67	25	0.49	4	4	1	0.33
S5	2	1.33	50	0.97	5	4	1.25	1.61
S6	2	1.33	55	1.07	3	4	0.75	1.07
S7	1	0.67	70	1.36	1.5	2	0.75	0.68
S8	1	0.67	20	0.39	4.5	5	0.9	0.24
S9	2	1.33	35	0.68	2	2	1	0.9
S10	3	2	70	1.36	2.5	2	1.25	3.4

The index of relevance of each service is graphically displayed in Fig. 2. From Fig. 2 it can be concluded that, for purposes of performance improvement prioritizing a subset of the services, the most relevant are S10, S5 and S3, and the less relevant are S8, S4 and S1.

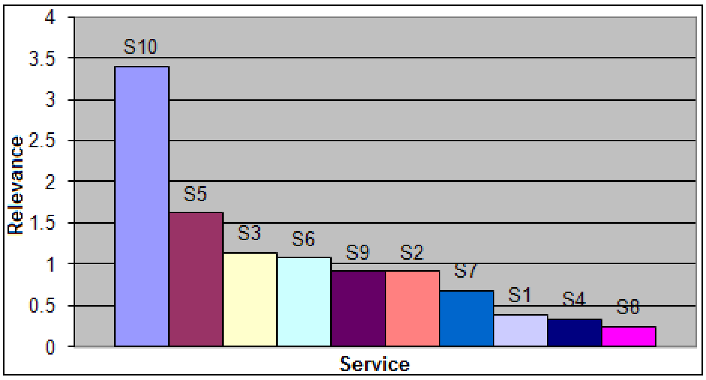

Fig. 2 Graphic of relevance of Fig.1 services

Even in the case of simpler SOAs, to manually retrieve the index of service relevance is a tedious and error-prone process. For the calculation of the index could be done automatically, a tool was developed. Having as input all the artifacts used by the calculation, it provides the multidigraph representing the SOA under analysis, as well as the index of relevance for each service of that SOA. The architecture of the tool, layered, was based on the architecture of the framework SPEWS [16]. The platform adopted for its implementation was Java SE 6 (Java Platform, Standard Edition) [19].

IV. EXPERIMENT AND ANALYSIS OF RESULTS

In order to evaluate the prioritization of services performed according to the index of service relevance, a quantitative experiment [20] was done. Fig. 3 shows an overview of the steps that constitute this experiment.

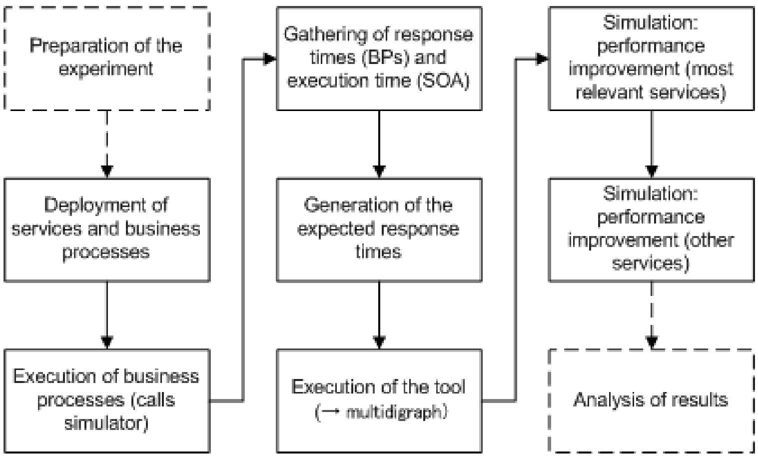

Fig. 3 Overview of the experiment

The goal of the experiment is to verify, in the context of performance improvement, if the priority selection of services, made with the index of relevance, effectively reduces the response time of the business processes and the execution time of the SOA. Besides checking whether this reduction occurs, a comparative analysis is also done, with a subset of the SOA's services. The selected services are used to simulate a performance improvement effort, and it is determined how much the performance improvement made in accordance to the index of service relevance classifies in relation to the other possibilities of improvement.

A. *Preparation of the Experiment*

In order to carry out the experiment, two activities were done previously: definition of the execution environment and creation of an experimental SOA.

1) Execution Environment:

a) Hardware

- A single box.
- Processor: Intel Pentium D 32 bits 3.2 GHz.
- RAM memory: 4 GB.

b) Software

- Operating system Microsoft Windows XP Professional Service Pack 3 32 bits.
- Virtual machine Java SE 6 [19], version 1.6.0_17 32 bits.
- Application server Java EE 5 (Java Platform, Enterprise Edition) [21] JBoss 5.1.0.GA [22].
- Database Oracle Express Edition, version 10.2.0.1.0.
- Oracle BPEL Process Manager 10.1.2.0.2 [23].

2) Creation of the Experimental SOA:

The SOA target of the experiment consists of five BPEL business processes with eight invoke activities by business process. Since this is an experiment, simulated services were used instead of real services, as the simulated services provide a means to execute the experiment without the additional complexity of a SOA with real services. Besides the invoke activities, the business processes also consist of sequence, switch, while and flow activities, among others.

The services of the experimental SOA, 30 in total, just interrupt its execution during a random time interval, and have predefined expected response times.

B. *Execution of the Experiment*

The input consisted of the BPEL specifications of the experimental SOA, with its 5 business processes. Table II shows how the 30 operations of the 5 services are distributed among the invoke activities of the business processes.

TABLE II INVOKE ACTIVITIES EXISTING IN EACH BUSINESS PROCESS

BusinessProcess1	**BusinessProcess2**	**BusinessProcess3**	**BusinessProcess4**	**BusinessProcess5**
Service1.operation3	Service1.operation3	Service1.operation1	Service1.operation2	Service1.operation6
Service2.operation8	Service1.operation5	Service2.operation9	Service1.operation4	Service2.operation10
Service2.operation12	Service2.operation10	Service2.operation12	Service2.operation7	Service2.operation12
Service3.operation15	Service3.operation15	Service3.operation13	Service2.operation11	Service3.operation14
Service3.operation16	Service3.operation16	Service3.operation15	Service3.operation18	Service3.operation17
Service4.operation22	Service3.operation18	Service3.operation16	Service4.operation20	Service4.operation20
Service5.operation25	Service4.operation24	Service4.operation19	Service4.operation23	Service4.operation21
Service5.operation27	Service5.operation30	Service5.operation29	Service5.operation28	Service5.operation26

1) Deployment of Services and Business Processes:

After starting the application server (JBoss [22]) and the BPEL orchestrator (Oracle BPEL [23]), the services and the business processes were deployed, respectively, in these two servers.

In the case of the services, they were packaged in a Java EE [21] standard file and deployed in JBoss, thus being available to receive SOAP calls.

In relation to the business processes, they were deployed in Oracle BPEL using its management application.

2) Execution of Business Processes:

The business processes were triggered by a SOAP calls simulator. For each business process, the simulator started a thread.

Each thread, in turn, recurrently performed the following algorithm, through 5 minutes:

- Wait a random time interval (between 5 s and 30 s).
- Randomly generate the total number of SOAP calls to be made to the business process (between 1 and 5).
- For each SOAP call to be made to the business process, schedule a thread of execution starting after a random period of time (between 5 s and 30 s).

The simulator was executed 30 times. Table III summarizes the total number of executions of each business process, obtained through a query to the database of the BPEL orchestrator. They are used to calculate r_{di} (Definition 3.2 of Section III).

TABLE III NUMBER OF EXECUTIONS OF EACH BUSINESS PROCESS

Business Process	Number of executions
BusinessProcess1	197
BusinessProcess2	283
BusinessProcess3	94
BusinessProcess4	167
BusinessProcess5	208
Total	**949**

3) Gathering of the Response Times of the BPs and of the Execution Time of the SOA:

The average response time of the business processes and the total execution time[3] of the SOA are gathered from the database. The results of these queries are presented in Table IV.

TABLE IV RESPONSE TIMES OF THE BUSINESS PROCESSES AND EXECUTION TIME OF THE SOA

Business Process	Average response time (s)
BusinessProcess1	47.1
BusinessProcess2	32.29
BusinessProcess3	112.97
BusinessProcess4	55.66
BusinessProcess5	43.39
SOA	**Total execution time (s)**
	47359.9

4) Generation of the Expected Response Times:

To generate the expected response times of the experimental SOA's services, it was executed a routine to randomly generate these numbers, ranging between 2 s and 20 s. These numbers were chosen ad hoc, merely as a way to illustrate the proposed approach. The result of the execution of this routine can be seen in Table V.

TABLE V EXPECTED RESPONSE TIMES OF THE EXPERIMENT SERVICES (S)

Service1	Service2	Service3	Service4	Service5
operation1: 6	operation7: 14	operation13: 18	operation19: 16	operation25: 10
operation2: 4	operation8: 16	operation14: 6	operation20: 4	operation26: 5
operation3: 2	operation9: 2	operation15: 13	operation21: 19	operation27: 4
operation4: 2	operation10: 13	operation16: 18	operation22: 18	operation28: 9
operation5: 6	operation11: 17	operation17: 2	operation23: 19	operation29: 17
operation6: 3	operation12: 12	operation18: 15	operation24: 14	operation30: 19

5) Execution of the Tool for Calculation of Service Relevance:

In order to calculate the index of service relevance of the experimental SOA's services, the tool, available as a Java SE [19] library, was initiated and the following steps were performed:

- Import the 5 business processes of the experimental SOA.
- Generate the multidigraph that represents these 5 business processes.
- Register the expected response time of the services, using the values generated in Section IV-B-4.
- Collect the performance metrics from the BPEL orchestrator.

[3] The total execution time of a SOA, in this paper, is measured as the summation of the response times of each execution instance of every business process pertaining to that SOA, as if they had run sequentially.

- Calculate the index of service relevance of the 30 services, making them available in a list sorted by the index.

In Fig. 4 the result of the execution of the tool can be seen. The vertices of the multidigraph represent calls to services (activities invoke of the business processes). The numbers of service invocations are arc labels. Below the multidigraph, there is a list of services sorted by relevance. Particularly, the description below vertex "Service4.operation21" shows the average response time of the service (8.99 s) and its expected response time (19 s).

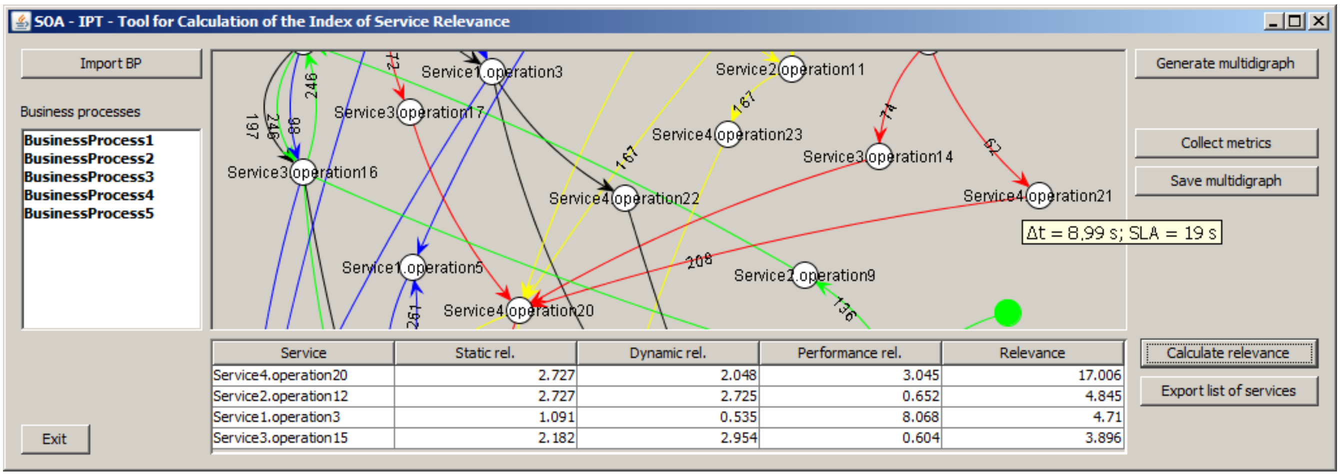

Fig. 4 Execution of the tool for the experimental SOA

The complete list of the index of service relevance of the 30 services of the experimental SOA can be seen in Table VI.

TABLE VI INDEX OF SERVICE RELEVANCE OF THE EXPERIMENTAL SOA'S SERVICES

Service	r_{si}	r_{di}	r_{pi}	r_i
Service4.operation20	2.73	2.05	3.05	17.07
Service2.operation12	2.73	2.72	0.65	4.83
Service1.operation3	1.09	0.54	8.07	4.75
Service3.operation15	2.18	2.95	0.6	3.86
Service3.operation16	1.64	2.95	0.5	2.42
Service1.operation4	0.55	0.46	8.48	2.15
Service1.operation5	1.09	1.43	1.23	1.92
Service1.operation6	0.55	1.14	2.82	1.77
Service2.operation9	0.55	0.74	4.26	1.73
Service3.operation18	1.09	1.87	0.84	1.71
Service4.operation19	2.18	1.54	0.48	1.61
Service5.operation25	1.64	1.08	0.88	1.56
Service5.operation28	1.64	0.91	0.9	1.34
Service5.operation27	0.55	1.08	1.93	1.15
Service1.operation2	0.55	0.91	1.88	0.94
Service5.operation26	0.55	1.14	1.44	0.9
Service3.operation17	0.55	0.39	3.96	0.85
Service2.operation10	1.09	0.93	0.6	0.61
Service2.operation11	1.09	0.91	0.51	0.51
Service4.operation23	0.55	0.91	0.9	0.45
Service3.operation14	0.55	0.4	1.5	0.33
Service1.operation1	0.55	0.34	1.34	0.25
Service3.operation13	0.55	0.8	0.47	0.21
Service4.operation24	0.55	0.54	0.55	0.16
Service5.operation29	0.55	0.46	0.49	0.12
Service5.operation30	0.55	0.48	0.43	0.11
Service4.operation21	0.55	0.34	0.47	0.09
Service4.operation22	0.55	0	0	0
Service2.operation7	0.55	0	0	0
Service2.operation8	0.55	0	0	0

6) Simulation: Performance Improvement of the Most Relevant Services:

To simulate the performance improvement of the services, the approach adopted was to recalculate the response times of these services, as well as the response times of their respective business processes[4]. It was assumed as ad hoc premises a reduction of 50% in their response times, and a selection of 20% of the SOA's services (totalling 6 services). The performance metrics of the services were already available in the database of the BPEL orchestrator.

With the simulation approach defined, it was used to simulate the performance improvement provided by the selection of the 6 services with the greatest index of service relevance: Service4.operation20, Service2.operation12, Service1.operation3, Service3.operation15, Service3.operation16 and Service1.operation4. For each execution instance of each business process (in a total of 949 instances, as shown in Table II), it was applied the 50% performance improvement factor to the 6 selected services. Their response times and of their respective business processes were recalculated next, according to the approach previously explained. Table VII shows the new average response times of the business processes and execution time of the SOA.

TABLE VII RESPONSE TIMES OF THE BPS AND EXECUTION TIME OF THE SOA AFTER PERFORMANCE IMPROVEMENT SIMULATION WITH INDEX OF RELEVANCE

Business Process	**Average response time (s)**
BusinessProcess1	32.95
BusinessProcess2	27.12
BusinessProcess3	89.94
BusinessProcess4	46.24
BusinessProcess5	35.24
SOA	**Total execution time (s)**
	37672.12

7) Simulation: Performance Improvement of the Other Services:

The last step of the experiment is to simulate the other possibilities of performance improvement: a combination of 30 services taken 6 at a time, excluding the combination of the previous step, resulting in a total of 593774 different possibilities.

Due to the large volume of data manipulated and generated by this simulation, it was necessary to create a routine that could apply the performance improvement factor (50%) to the selected services and recalculate the response times. The input was the metrics collected from the database of the BPEL orchestrator. For each of the 593774 possible combinations, the routine recalculated the response times of the 949 execution instances of the business processes, thereby generating numbers which could be used in a comparative analysis.

C. Analysis of Results

The experiment described in Section IV-B provided a large dataset that, when analyzed, allows verifying whether the research problem was solved and, if so, how it was solved. This verification is done, initially, by comparing the average response times of the business processes and the total execution time of the SOA before and after the performance improvement simulation using the index of service relevance; and, then, classifying each one of the performance improvement possibilities according to the execution time of the SOA, thus allowing comparisons.

The first part of the verification is summarized in Table VIII. It allows quantifying the effect of the performance improvement with the index of service relevance in the experimental SOA. The data had already been collected during the experiment, just being necessary to compare them.

TABLE VIII EFFECT OF THE INDEX OF RELEVANCE IN THE EXPERIMENTAL SOA

Business Process	**Response time (before) (s)**	**Response time (according to index of relevance) (s)**	**Performance improvement (%)**
BusinessProcess1	47.1	32.95	**30.04**
BusinessProcess2	32.29	27.12	**16.01**
BusinessProcess3	112.97	89.94	**20.39**
BusinessProcess4	55.66	46.24	**16.92**
BusinessProcess5	43.39	35.24	**18.78**
SOA	**Execution time (before) (s)**	**Execution time (according to index of relevance) (s)**	**20.46**
	47359.9	37672.12	

The second part of the verification was done by consulting the temporary table created during the experiment. The 593775 possibilities of performance improvement were decreasingly sorted by the average response time (business processes) and the

[4] The response time of a BP is, in part, the summation of the response times of its services - i.e. activities invoke.

total execution time (SOA). Fig. 5 shows the percentile distribution of the total execution times of the experimental SOA.

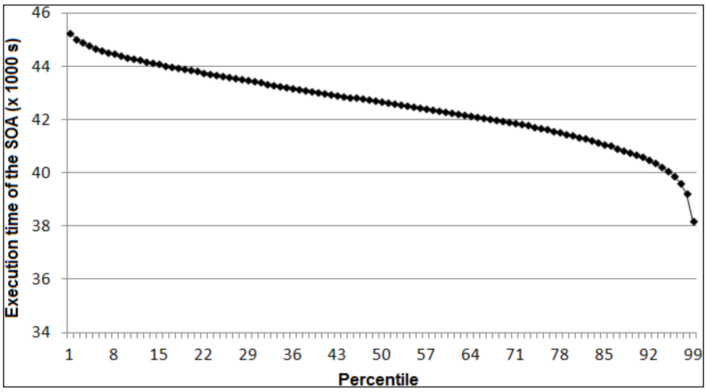

Fig. 5 Percentiles of the total execution times of the experimental SOA

Fig. 5 is important by allowing comparing the effect of all possibilities of performance improvement in the total execution time of the SOA. Since there were 593775 different combinations of services to be optimized in the experiment, a percentile chart makes the comparison easier, by grouping in each of the percentiles the execution times of the SOA resulting from these optimizations.

In Table IX are summarized some collected metrics.

TABLE IX COMPARATIVE OF THE PERFORMANCE IMPROVEMENTS

Business Process	**Minimum response time (after) (s)**	**Maximum response time (after) (s)**	**Response time (according to index of relevance) (s)**	**Percentile (according to index of relevance)**
BusinessProcess1	23.55	47.1	32.95	**98**
BusinessProcess2	18.21	32.29	27.12	**81**
BusinessProcess3	66.38	112.97	89.94	**93**
BusinessProcess4	30.67	55.66	46.24	**85**
BusinessProcess5	24.41	43.39	35.24	**87**
SOA	**Minimum execution time (after) (s)**	**Maximum execution time (after) (s)**	**Execution time (according to index of relevance) (s)**	**99**
	7581.28	47359.9	37672.12	

The last column of Table IX illustrates that, for all business processes, the performance improvement made according to the index of service relevance resulted in an improvement always located in the fourth best quartile. When analyzing the execution time of the SOA as a whole, the results are even more significant: the selection of services suggested by the index of service relevance made the performance improvement stay in the 99th best percentile.

D. Limitations and Restrictions

During the execution of the experiment, some limitations and restrictions of the proposed approach in this work could be raised:

- Approach costs: in a scenario where an organization has already decided on a performance improvement effort of a SOA, calculating the service relevance, assuming through the tool, turns out to be an additional step, with the consequent need to estimate, plan and execute it. Another cost to an organization is to ensure the SOA performance metrics are recorded by the BPEL orchestrator and made available to the tool, so the index of relevance can be calculated. Depending on the maturity level of the organization, to obtain such data may require an extra effort.

- Throughout its existence, a SOA is dynamic: as services are added, modified or removed, the relevance of these services to the SOA also changes, making the results generated by the tool obsolete over time. Changes in the usage profile of the service (e.g. increasing demand) or refactorings of the SOA also contribute to the obsolescence of the results. In summary: the index of relevance is just a "snapshot" of the service relevance in a given moment of time. The index of service relevance needs to be periodically recalculated to remain up-to-date.

- Typically, the expected response time of a service is expressed in SLA documents as a percentage of calls within a time frame (e.g. 99% within 2 s). The proposed index of service relevance, in its performance relevance component (Definition 3.3 of Section III), does not take such percentages in consideration, relying on the expected time frame only. However, the index of service relevance could be refined in order to include the percentages in the calculation.

- Regarding the service relevance, this work does not consider functional aspects which although are also important [24], are beyond its scope. Therefore, it was provided only a criterion for the analysis of a SOA, which can be supplemented by others, such as service provider reputation, price or business value [24].

V. CONCLUSIONS

Taking as starting point the research problem mentioned in Section I, a series of works related to performance improvement in SOA was studied. Most of them were based on a representation of business processes in the form of a graph, and how information useful to that purpose could be obtained from this type of representation.

From a rationale constituent of the research hypothesis, an analysis of a specific type of graph (the multidigraph [14]) and performance metrics collected from the SOA (number of invocations, average response time and expected response time), it was proposed an index of service relevance, for purposes of performance improvement of a SOA.

In sequence, an experiment to evaluate the index of service relevance was done, assisted by a tool specifically created for this purpose. A simulation of a performance improvement effort was made, and from such experiment could be concluded that the performance improvement made according to the index of relevance proved to be effective in terms of the execution time of the SOA.

As for possible future works, it is suggested using the index of service relevance in a case study involving a real SOA; and the creation of indexes of relevance for other quality attributes, such as reliability or availability.

REFERENCES

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

[2] S. Chen, L. Bao and P. Chen, "OptBPEL: a tool for performance optimization of BPEL process", in *Proc. 7th SC*, pp. 141-148, 2008.

[3] A. A. Mohammed, "Towards a model-driven approach for planning a standard-based migration of enterprise applications to SOA", in *Proc. 1st SERVICES*, pp. 471-474, 2009.

[4] J. Her, S. Choi, S. H. Oh and S. D. Kim, "A framework for measuring performance in service-oriented architecture", in *Proc. 3rd NWeSP*, pp. 55-60, 2007.

[5] J. E. L. Gonçalves, "As empresas são grandes coleções de processos", *Revista de Administração de Empresas*, vol. 40, n. º1, pp. 6-19, jan./mar. 2000.

[6] (2003) Business Process Execution Language for Web Services Version 1.1. [Online]. Available: http://public.dhe.ibm.com/software/dw/specs/ws-bpel/ws-bpel.pdf.

[7] Z. Ma, W. Lu and F. Leymann, "Query structural information of BPEL processes", in *Proc. 4th ICIW*, pp. 401-406, 2009.

[8] C. Mao, "Slicing web service-based software", in *Proc. 2nd SOCA*, pp. 91-98, 2009.

[9] L. O'Brien, P. Merson and L. Bass, "Quality Attributes for Service-Oriented Architectures", in *Proc. 1st. SDSOA*, pp. 3, 2007.

[10] (2000) SOAP: Simple Object Access Protocol 1.1. [Online]. Available: http://www.w3.org/TR/2000/NOTE-SOAP-20000508.

[11] (2008) XML: Extensible Markup Language 1.0. [Online]. Available: http://www.w3.org/TR/xml.

[12] J. S. Jwo and Y. C. Cheng, "Performance Improvement for an SOA Based Enterprise Application Using Service Redirection", in *Proc. 4th SOSE*, pp. 287-290, 2008.

[13] T. Erl, *Principles of Service Design*, Ed. Upper Saddle River, USA: Prentice Hall, 2007.

[14] R. M. Leme and M. N. Rezende, "An approach with multidigraphs for representing multiple business processes of a SOA", in *Proc. 1st CITS*, pp. 1-6, 2012.

[15] M. Fowler, K. Beck, J. Brant, W. Opdyke and D. Roberts, *Refactoring: Improving the Design of Existing Code*, Ed. Boston, USA: Addison-Wesley, Jul. 1999.

[16] H. Holanda, G. C. Barroso and A. Serra, "SPEWS: a framework for the performance analysis of web services orchestrated with BPEL4WS", in *Proc. 4th ICIW*, pp. 363-369, 2009.

[17] D. Rud, A. Schmietendorf and R. Dumke, "Performance Modeling of WS-BPEL-Based Web Service Compositions", in *Proc. 1st SCW*, pp. 140-147, 2006.

[18] S. S. Stevens, "On the Theory of Scales of Measurement", *Science: New Series*, vol. 103, iss. 2684, pp. 677-680, Jun. 1946.

[19] (2006) Java Platform, Standard Edition 6. [Online]. Available: http://java.sun.com/javase/.

[20] B. Kitchenham, S. Linkman and D. Law, "DESMET: a methodology for evaluating software engineering methods and tools", *Computing & Control Engineering Journal*, vol. 8, iss. 3, pp. 120-126, Jun. 1997.

[21] (2006) Java Platform, Enterprise Edition 5. [Online]. Available: http://java.sun.com/javaee.

[22] (2009) JBoss Application Server 5.1.0.GA. [Online]. Available: http://www.jboss.org/jbossas.

[23] (2006) Oracle BPEL Process Manager 10.1.2.0.2. [Online]. Available: http://www.oracle.com/technetwork/middleware/bpel/overview/index.html.

[24] Y. Lee, "Quality factors of business value and service level measurement for SOA", in *Proc. 2nd PACCS*, pp. 473-478, 2010.