# Translations of Service Level Agreement in Systems Based on Service Oriented Architecture

Adam Grzech and Piotr Rygielski

Institute of Computer Science
Wroclaw University of Technology
Wybrzeze Wyspianskiego 27
50-370 Wroclaw, Poland
{adam.grzech,piotr.rygielski}@pwr.wroc.pl

**Abstract.** The gain of the paper is to introduce and to discuss a formal specification of computer system's Service Level Agreement (SLA) and its translation into structure of complex services (composed of atomic services) delivering required functionalities and non-functionalities in distributed environment. The SLA is composed of two parts specifying quantitative and qualitative requirements. The former requirements define structure of the adequate complex services as a directed graph, where potential parallelism of atomic services performance may be taken into account. The qualitative requirements are applied to select the optimal complex service realization scenario; it is based on assumption that various atomic services distinguished in the complex services structure are available in the environment in different versions and locations. Different versions of atomic services are delivering the required functionalities and satisfy non-functionalities at various levels. Various locations of atomic services means that the time and cost of atomic services delivery (communication and calculation) may vary. Proposed model of SLA translation into complex services causes that a scenario variants may be applied — among others — to calculate upper and lower complex services' delivering times and to estimate validity of possible parallelism in complex services.

**Keywords:** SOA, Quality of Service, graph parallelism.

## 1 Introduction

In systems based on SOA (Service Oriented Architecture) paradigm, complex services are delivered by a distributed computer communication system as a composition of selected atomic services [3,5]. Each atomic service provides certain and well defined functionality and is characterized by parameters describing quality of required and delivered service. Functionality of requested complex service is a sum of functionalities of ordered atomic services; the order is a consequence of complex service decomposition into simple tasks ruled by same time and data interdependencies. Besides requested functionality, service requester may specify certain non- functional properties of complex service formulated in terms of security, availability, cost, response time, quality measures, etc. Values of the non-functional parameters — given in the particular SLA (Service Level Agreement) [4,7] — of the requested complex service may be delivered without changes or should be negotiated before acceptance [1]. In order to deliver complex service with requested functional and non-functional properties, appropriate atomic services must be chosen in the process of complex service composition.

One of the most important non-functional properties of services, which is a commonly used measure of the quality of service (QoS), is service response time. Service response time is mainly influenced by three factors: execution time of atomic services, load of the system (number of requests present in the system) and communication delays on links connecting atomic services. In order to guarantee stated in SLA service response time all these factors must be taken into account in the process of service composition.

## 1.1 SLA Translation Model

It is assumed that new incoming $i$-th complex service request is characterized by proper Service Level Agreement description denoted by $SLA(i)$. The $SLA(i)$ is composed of two parts describing functional and nonfunctional requirements, respectively $SLA_f(i)$ and $SLA_{nf}(i)$. The first part characterizes functionalities that have to be performed, while the second contains values of parameters representing various quality of service aspects (delivery times, guarantees, etc.).

The $SLA_f(i)$ is a set of distinguished, separated and ordered functionalities subsets:

$$SLA_f(i) = \left\{\Gamma_{i1}, \Gamma_{i2}, \ldots, \Gamma_{ij}, \ldots, \Gamma_{in_i}\right\} \tag{1}$$

where:

- $\Gamma(i) = \Gamma_{i1} \cup \Gamma_{i2} \cup \ldots \cup \Gamma_{ij} \cup \ldots \cup \Gamma_{in_i}$ is a set of all functionalities necessary to complete the $i$-th complex service request,
- $\Gamma_{i1} \prec \Gamma_{i2} \prec \ldots \prec \Gamma_{ij} \prec \ldots \prec \Gamma_{in_i}$ ordered subset of distinguished functionalities subsets required by $i$-th complex service request; $\Gamma_{ij} \prec \Gamma_{ij+1}$ (for $j = 1, 2, \ldots, n_i - 1$) denotes that delivery of functionalities from the subset $\Gamma_{ij+1}$ cannot start before completing functionalities from the $\Gamma_{ij}$ subset.
- $\Gamma_{ij} = \{\varphi_{ij1}, \varphi_{ij2}, \ldots, \varphi_{ijm_j}\}$ (for $i = 1, 2, \ldots, n_i$) is a subset of functionalities $\varphi_{ijk}$ ($k = 1, 2, \ldots, m_j$) that may be delivered in parallel manner (within $\Gamma_{ij}$ subset); the formerly mentioned feature of particular functionalities are denoted by $\varphi_{ijk} \parallel \varphi_{ijl}$ ($\varphi_{ijk}, \varphi_{ijl} \in \Gamma_{ij}$ for $k, l = 1, 2, \ldots, m + j$ and $k \neq l$).

The proposed scheme covers all possible cases; $n_i = 1$ means that all required functionalities may be delivered in parallel manner, while $m_j = 1$ ($j = 1, 2, \ldots, n_i$) means that required functionalities have to be delivered in sequence.

It is also assumed that the $\varphi_{ijk}$ ($j = 1, 2, \ldots, n_i$ and $k = 1, 2, \ldots, m_j$) functionalities are delivered by atomic services available at the computer system in several versions. It is assumed that the various versions of the distinguished atomic service offer exactly the same functionality and different values of non-functional parameters.

The nonfunctional requirements may be decomposed in a manner, similar to that introduced for functional requirements, i.e.;

$$SLA_{nf}(i) = \{H_{i1}, H_{i2}, \ldots, H_{ij}, \ldots, H_{in_i}\} \tag{2}$$

where $H_{ij} = \{\gamma_{ij1}, \gamma_{ij2}, \ldots, \gamma_{ijm_j}\}$ is a subset of nonfunctional requirements related respectively to the $\Gamma_{ij} = \{\varphi_{ij1}, \varphi_{ij2}, \ldots, \varphi_{ijm_j}\}$ subset of functionalities.

According to the above assumption the $SLA_f(i)$ of the $i$-th complex service request may be translated into ordered subsets of atomic services:

$$SLA_f(i) = \{\Gamma_{i1}, \Gamma_{i2}, \ldots, \Gamma_{in_i}\} \Rightarrow \{AS_{i1}, AS_{i2}, \ldots, AS_{in_i}\}, \tag{3}$$

where $\{AS_{i1}, AS_{i2}, \ldots, AS_{ij}, \ldots AS_{in_i}\}$ is a sequence of atomic services subsets satisfying an order $(AS_{i1} \prec AS_{i2} \prec \ldots \prec AS_{ij} \prec \ldots \prec AS_{in_i})$ predefined by order in the functionalities subsets $\Gamma_{i1} \prec \Gamma_{i2} \prec \ldots \prec \Gamma_{ij} \prec \ldots \prec \Gamma_{in_i}$ (Fig. 1).

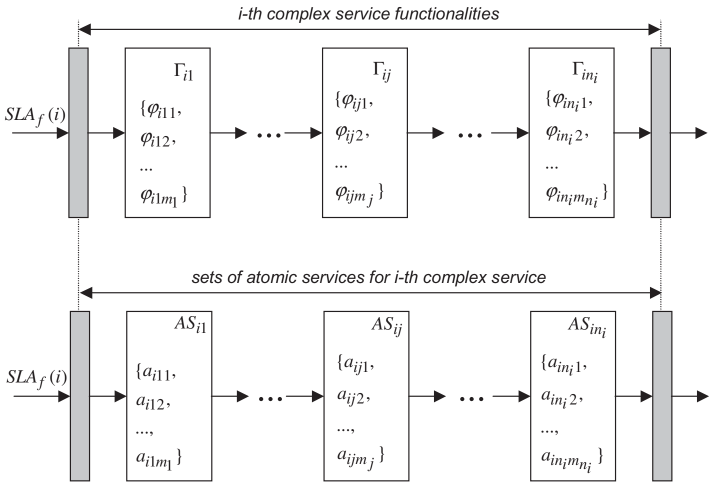
i-th complex service functionalities
$SLA_f(i)$
$\Gamma_{i1}$
$\{\varphi_{i11}, \varphi_{i12}, \ldots \varphi_{i1m_1}\}$
$\Gamma_{ij}$
$\{\varphi_{ij1}, \varphi_{ij2}, \ldots \varphi_{ijm_j}\}$
$\Gamma_{in_i}$
$\{\varphi_{in_i1}, \varphi_{in_i2}, \ldots \varphi_{in_im_{n_i}}\}$
sets of atomic services for i-th complex service
$SLA_f(i)$
$AS_{i1}$
$\{a_{i11}, a_{i12}, \ldots, a_{i1m_1}\}$
$AS_{ij}$
$\{a_{ij1}, a_{ij2}, \ldots, a_{ijm_j}\}$
$AS_{in_i}$
$\{a_{in_i1}, a_{in_i2}, \ldots, a_{in_im_{n_i}}\}$

**Fig. 1.** Transformation of functionalities into sets of atomic services

The order in sequence of atomic services is interpreted as the order in functionalities subsets: $AS_{ij} \prec AS_{i,j+1}$ (for $j = 1, 2, \ldots, n_i - 1$) states that atomic services from subsets $AS_{i,j+1}$ cannot be started before all services from the $AS_{ij}$ subset are completed.

Each subset of atomic services $AS_{ij}$ $(j = 1, 2, \ldots, n_i)$ contains $a_{ijk}$ atomic services $(k = 1, 2, \ldots, m_j)$ available at the computer system in several versions $a_{ijkl}$ $(l = 1, 2, \ldots, l(k))$; $AS_{ij} = \{a_{ij1}, a_{ij2}, \ldots, a_{ijn_i}\}$. Moreover, it is assumed that any version $a_{ijkl}$ $(l = 1, 2, \ldots, l(k))$ of the particular $a_{ijk}$ $(a_{ijk} = \{a_{ijk1}, a_{ijk2}, \ldots, a_{ijkl(k)}\})$ atomic service $(k = 1, 2, \ldots, m_j)$ assures the same required functionality $\varphi_{ijk}$ and satisfies quality requirements at various levels.

The above assumption means that — if $fun(a_{ijkl})$ and $nfun(a_{ijkl})$ denote, respectively, functionality and level of nonfunctional requirements satisfaction delivered by $l$-th version of $k$-th atomic service ($a_{ijkl} \in AS_{ij}$) — the following conditions are satisfied:

- $fun(a_{ijkl}) = \varphi_{ijk}$ for $l = 1, 2, \ldots, m_k$,
- $nfun(a_{ijkl}) \neq nfun(a_{ijkr})$ for $l, r = 1, 2, \ldots, m_k$ and $l \neq r$.

The ordered functionalities subsets $SLA_f(i)$ determines possible level of parallelism at the $i$-th requested complex service performance (in the particular environment). The parallelism level $l_p(i)$ for $i$-th requested complex service is defined by the maximal number of atomic services that may be performed in parallel manner at distinguished subsets of functionalities $\big(SLA_f(i)\big)$, i.e.,

$$l_p(i) = \max\{m_1, m_2, \ldots, m_j, \ldots, m_{n_i}\}. \tag{4}$$

The possible level of parallelism may be utilized or not in processing of the $i$-th requested complex service [6]. Based on the above notations and definitions two extreme compositions exist. The first one utilizes possible parallelism (available due to computation resources parallelism), while the second extreme composition means that the requested functionalities are delivered one-by-one (no computation and communication resources parallelism).

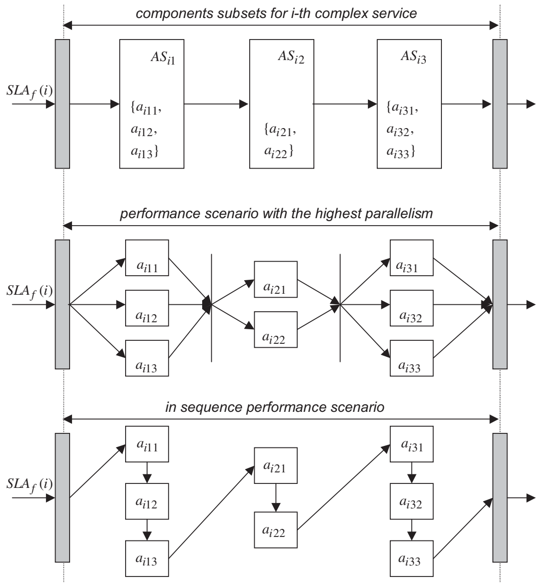
components subsets for i-th complex service
$SLA_f(i)$
$AS_{i1}$
$\{a_{i11}, a_{i12}, a_{i13}\}$
$AS_{i2}$
$\{a_{i21}, a_{i22}\}$
$AS_{i3}$
$\{a_{i31}, a_{i32}, a_{i33}\}$
performance scenario with the highest parallelism
$a_{i11}$
$a_{i12}$
$a_{i13}$
$a_{i21}$
$a_{i22}$
$a_{i31}$
$a_{i32}$
$a_{i33}$
in sequence performance scenario

**Fig. 2.** Two extreme performance scenarios for required complex service

The above presented discussion may be summarized as follows (example — Fig. 2). The known functional requirements $SLA_f(i)$ may be presented as a sequence of subsets of functionalities, where the size of the mentioned latter subsets depends on possible level of parallelism. The available level of parallelism defines a set of possible performance scenarios according to which the requested complex service may be delivered. The space of possible solutions is limited, from one side, by the highest possible level parallelism and, from another side, by natural scenario, where all requested atomic services are performed in sequence.

The mentioned above extreme compositions determines some set of possible $i$-th requested complex service delivery scenarios. The possible scenario can be represented by a set of graphs $G(i)$. Nodes of graph represent particular atomic services assuring requested complex service functionalities, while edges represent an order according to which atomic services functionalities have to be delivered.

The set of all possible graphs $G(i)$ $\Big(G(i) = \{G_{i1}, G_{i2}, \ldots, G_{is}\}\Big)$ assures the requested functionality, but fulfill nonfunctional requirements various levels. The latter may be obtained (and optimized) assuming that at least one node of the particular graph contains at least two versions of the requested atomic service. Two listed above extreme compositions are denoted by graph $G_{par}(i)$ and set of equivalent graphs $G_{seq}(i)$ ($G_{par}(i) \in G(i)$ and $G_{seq}(i) \in G(i)$) for parallel and sequenced realization of the $i$-th required complex service, respectively.

It is worth to note that the higher possible parallelism of delivering atomic services is represented only by one graph $G_{par}(i)$, while — in general case — possible parallelism of delivering atomic services belonging to the distinguished $AS_{ij}$ groups means that there is set of equivalent compositions (set of equivalent graphs $G_{seq}(i)$) according to which the required complex service may be delivered. The set of graphs $G_{seq}(i)$ is in practice a set of equal-length paths, where the lengths of paths (measured by number of hops) are equal to number of all atomic services increased be one $\big((m_1 + m_2 + \ldots + m_j + \ldots + m_{n_i}) + 1\big)$. In the above presented example (Figure 2) the paths lengths are equal to 9 and the number of all equivalent realizations of complex service by sequences of atomic services is equal to 72.

## 2 Quantitative Analysis of Complex Services Realizations

In general, the optimization task may be formulated (under assumption that all scenarios from the set $G(i)$ deliver at least functional requirements) as follows:

$$SLA^*_{nf}(i) \leftarrow \max_{G_{i1}, G_{i2}, \ldots, G_{is}} \left\{ \max_{a_{ijkl} \in a_{ijk} \in AS_{ij}} \{H_{i1}, H_{i2}, \ldots, H_{in_i}\} \right\}. \tag{5}$$

The latter task may be reduced where the particular $i$-th requested complex service composition (i.e., an graph equivalent to particular $i$-th requested complex service processing scheme) is assumed. In such a case the optimization task can be formulated as follows:

$$SLA^*_{nf}(i, G_i) \leftarrow \max_{a_{ijkl} \in a_{ijk} \in AS_{ij}} \{H_{i1}, H_{i2}, \ldots, H_{in_i}\}, \tag{6}$$

where $G_i$ represents the selected $i$-th required complex service's scenario.

The above formulated task means that the optimal versions of atomic services, determined by the selected $i$-th requested complex service performance scenario, should be selected. Such optimization problems can be found in [2].

Any graph $G_{ip}$ ($G_{ip} \in G(i)$ and $p = 1, 2, \ldots, s$) ($p$-th possible performance scenario for the required $i$-th complex service) may be interpreted as a root-graph for a set of realization graphs $\{G_{ipr}\}$ for $r = 1, 2, \ldots, R$, where $R$ is a number of all different combinations of atomic services versions. For example — assuming that there are three distinguished groups of atomic services ($n_i = 3$) containing, respectively, 3, 2 and 3 atomic services ($m_1 = 3$, $m_2 = 2$ and $m_3 = 3$) with two versions of each, the example of root-graph is presented on Figure 3; in the considered example the required complex service root-graph is a sources of 256 ($R = 256$) different $i$-th complex service realization graphs with possible level of parallelism equal to 3 ($l_p(i) = 3$). The illustrative calculations simply show how size of the complex services (measured by number of atomic services necessary to complete the required complex service) influences increase of the number of possible solutions.

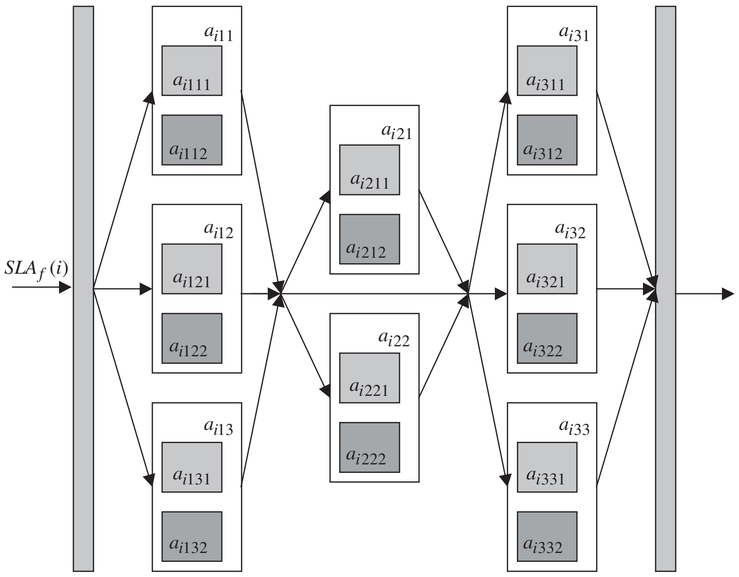

**Fig. 3.** Root-graph for the required complex service realization graphs

Any two realization graphs (determined by a particular root-graph defined by a particular complex service functionalities) int the set $\{G_{ipr}\}$ differ at least by one version of the set of atomic services $a_{ijk}$ standing for the graph $G_{ip}$ nodes. Two possible realization graphs ($G_{ip1}$ and $G_{ip2}$) for the above required complex service root-graph are presented at the Figure 4.

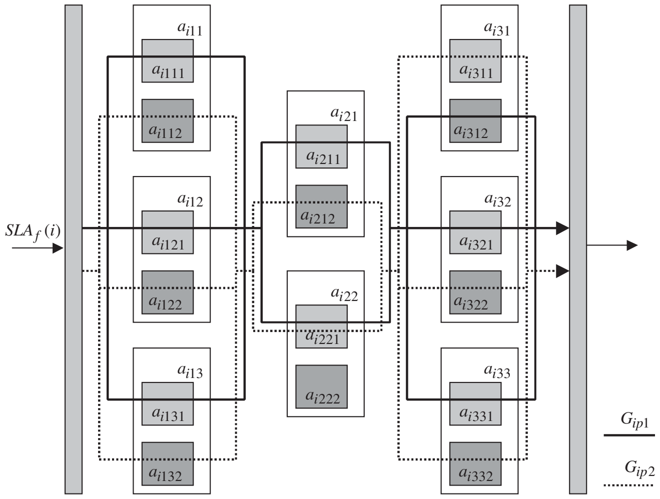

**Fig. 4.** Two possible complex service realization graphs for root-graph (Figure 3)

## 3 Complex Service Delivery Times and Resources Utilization

The two distinguished extreme compositions of the $i$-th required complex service, i.e., based on the highest possible parallelism of atomic services (represented by the $G_{par}(i)$ graph) and based on sequence of atomic services (represented by the $G_{seq}(i)$ set of graphs) can be used to estimate the limits of the $i$-th required complex service completing times.

Let one denote by $d\big(G_{par}(i)\big)$ and $d\big(G_{seq}(i)\big)$ the completing time for the two mentioned extreme compositions for the $i$-th required complex service execution time. The introduced times (delays) satisfy the inequalities given below.

For the highest possible parallel composition of the $i$-th required complex service (under assumption that the computation and communication resources parallelism, available at the distributed system, does not introduce any complex service performance limits):

$$d(G_{par}(i)) \leq \max_{k=1,2,\ldots,m_1} com_{in}(a_{i1k}) \max_{k=1,2,\ldots,m_1} cal(a_{i1k}) +$$

$$+ \sum_{j=1}^{n_i-1} \left\{ \max_{k=1,\ldots,m_j;l=1,\ldots,m_{j+1}} com(a_{ijk}, a_{i,j+1,l}) + \max_{l=1,\ldots,m_{j+1}} cal(a_{i,j+1,l}) \right\} + \quad (7)$$

$$+ \max_{l=1,\ldots,m_{n_i}} com_{out}(a_{in_il}),$$

where:

- $com_{in}(a_{i1k})$ is a communication delay between the $i$-th complex service input interface and the $k$-th $(k = 1, 2, \ldots, m_1)$ atomic service from the $A_{i1}$ set
- $cal(a_{i1k})$ is a calculation time of $k$-th atomic service from the $A_{i1}$ set
- $com(a_{ijk}, a_{i,j+1,l})$ is a communication delay between the $k$-th $(k = 1, 2, \ldots, m_j)$ and $l$-th $(l = 1, 2, \ldots, m_{j+1})$ atomic services belonging to the two separate, neighbouring and ordered sets of atomic services $A_ij$ and $A_{i,j+1}$, respectively
- $cal(a_{ijk})$ is a calculation time of $k$-th atomic service from the $A_{ij}$ set
- $com_{out}(a_{in_ik})$ is a communication delay between the $k$-th atomic service from the $A_{in_i}$ set and the $i$-th complex service output interface.

For sequence composition of the $i$-th required complex service (under assumption that the resources parallelism does not exists and the complex service is composed with atomic services performed one-by-one):

$$d(G_{seq}(i)) \leq \max_{k=1,2,\ldots,m_1} com_{in}(a_{i1k}) \max_{k=1,2,\ldots,m_1} cal(a_{i1k}) +$$

$$+ \sum_{j=1}^{n_i} \left( \sum_{k=1}^{m_j} cal_{a_{ijk}} + \left( (m_j - 1) \max_{l,u \in \{1,2,\ldots,m_j\}; l \neq u} com(a_{ijl}, a_{iju}) \right) \right) + \tag{8}$$

$$+ \max_{k=1,\ldots,m_{n_i}} com_{out}(a_{in_ik}),$$

where $com(a_{ijl}, a_{iju})$ is a communication delay between successive atomic services ordered according to the given $G_{seq}(i)$.

It is rather obvious that above given expressions for $d\big(G_{par}(i)\big)$ and $d\big(G_{seq}(i)\big)$ present, respectively, lower and upper $i$-th complex service's completing (delivering) times and moreover, that the compared times always satisfy the following inequality: $d\big(G_{par}(i)\big) \leq d\big(G_{seq}(i)\big)$.

The value $d\big(G_{seq}(i)\big) - d\big(G_{par}(i)\big)$, which is always non-negative, may be considered as a simple measure of parallelism attractiveness at the $i$-th complex service's completing procedures. If the distance between the two considered completing times is relatively high, it means that parallel processing of the selected atomic services is worth to be organized and deployed. If the distance between the two compared values is small it means that all efforts necessary to organize and perform atomic services in parallel manner are not compensated by noticeable reducing of the $i$-th complex service's completing time.

The value $d\big(G_{seq}(i)\big) - d\big(G_{par}(i)\big)$, i.e., parallelism attractiveness, strongly depends on possible parallelism in $i$-th complex service's completing procedures, i.e., level of resources parallelism (the latter is measured by highest number of atomic services that can be computed in parallel). Assurance of the highest possible level in computing atomic services in a parallel manner is possible if, and only if a required level of resources distribution is obtainable. It further means that the possible parallelism attractiveness should be compared with the efficiency of the required resources utilization.

The above discussed $d\big(G_{seq}(i)\big) - d\big(G_{par}(i)\big)$ value may be applied both as a simple measure of resources utilization in distributed environment as well as for distributed environment design purposes. The gain of the former application is to evaluate resources utilization factors reflecting validity of the distributed environment (especially available parallelism) for required complex services realization purposes. The aim of the latter is to predict required level of resources parallelism for known set of complex services (given by known set of complex services SLA).

## 4 Experimental Example

In former analysis a lower and upper bound for service completing time (represented as graph) were introduced (equation (7) for parallel and (8) for serial). Moreover, there has been a parallelism measure for graph $G(i)$ introduced (equation (4)) having its upper and lower bound dependent on number of vertices used in graph, respectively, $l_p(G_{sec}(i)) = \sum_{k=1}^{n_i} m_k$ and $l_p(G_{par}(i)) = 1$. Of course one can use other measures to describe the possible grpah parallelism level.

Having such parallelism measure allows to estimate differences in service completing times with respect to various values of mentioned measure. This can be used to estimate the benefit (expressed as service completing time in this example) of using scenario graphs with various parallelism level values.

Experimental example shows lower bound of service completing time (shortest obtainable completing time at given level of parallelism) for exemplary graph with 7 vertices (two of vertices represent service input and output interface). In example there has been another parallelism measure ($PM1$) used for comparison which is expressed as total number of paths connecting input and output interface divided by vertices count. Measure $PM2$ is a measure mentioned formerly (equation 4) expressed as maximum number of vertices that can be used simultaneously in processing one request.

The experiment consisted of generating all possible directed acyclic graphs with 7 vertices and with distinguished starting and ending vertex. Each vertex of such generated graph has been represented as computational node with proper atomic service version installed. Edges of mentioned graph were modeled as communication channels linking computational nodes. To model the system and execute the experiment an OMNeT++ simulation environment has been used.

For each of generated scenario graph there was single complex service request executed and completion time of such composed complex service with graph parallelism measure values has been stored. From the collected data lower bounds have been chosen (lowest possible completion time for given parallelism measure value) and presented on a chart 5.

Proper delays have been modeled as follows: single atomic service completing time was set to 1, transport delay for each edge was set to 0.1. Noteworthy is fact that parallelism measures used in experiment have different domains and best and worst possible completing time — in this example $d\big(G_{par}(i)\big) = 1.2$ and $d\big(G_{seq}(i)\big) = 5.6$ — are reached for various parallelism level values.

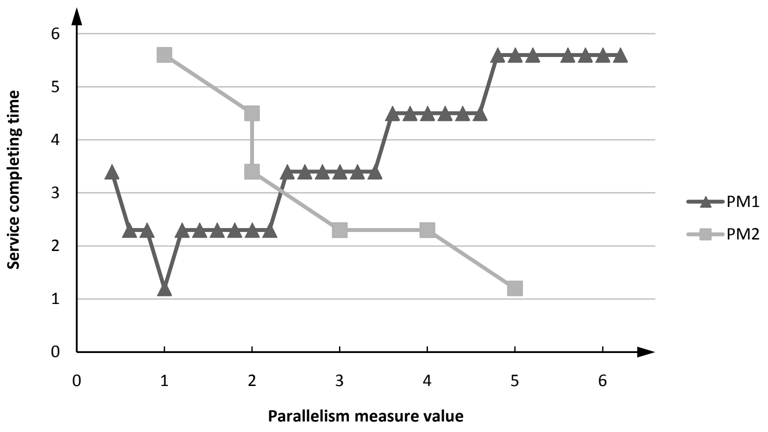

**Fig. 5.** Best obtainable service completing time with respect to constraints put on parallelism level value using two different parallelism measures proposed

Introduced approach can be useful in various optimization tasks concerning shaping of service execution graph with respect to quality of service constraints and can be implemented in real SOA system for using it in composition process to optimize quality of services delivered to users.

**Acknowledgments.** The research presented in this paper has been partially supported by the European Union within the European Regional Development Fund program no. POIG.01.03.01-00-008/08.

## References

1. Anderson, S., Grau, A., Hughes, C.: Specification and satisfaction of SLAs in service oriented architectures. In: 5th Annual DIRC Research Conference, pp. 141–150 (2005)
2. Grzech, A., Rygielski, P., Swiatek, P.: QoS-aware infrastructure resources allocation in systems based on service-oriented architecture paradigm. In: HET-NETs 2010, pp. 35–48 (2010), ISBN 978-83-926054-4-7
3. Johnson, R., Gamma, E., Helm, R., Vlisides, J.: Design patterns; elements of reusable object-oriented software. Addison-Wesley, Reading (1995)
4. Keller, A., Ludwig, H.: The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. Journal of Network and Systems Management 11(1), 57–81 (2003)
5. Milanovic, N., Malek, M.: Current Solutions for Web Service Composition. IEEE Internet Computing 8(6), 51–59 (2004)
6. Stefanovic, D., Martonosi, M.: Limits and Graph Structure of Available Instruction-Level Parallelism (Research Note). In: Bode, A., Ludwig, T., Karl, W.C., Wismüller, R. (eds.) Euro-Par 2000. LNCS, vol. 1900, pp. 1018–1022. Springer, Heidelberg (2000)
7. Li, X., et al.: Design of an SLA-Driven QoS Management Platform for Provisioning Multimedia Personalized Services. In: AINA Workshop, pp. 1405–1409 (2008)