Measuring The Satisfaction Degree Of Quality Attributes Requirements For Services Orchestrations

Nabil Fakhfakh, Hervé Verjus, Frédéric Pourraz, and Patrice Moreaux
University of Savoie
LISTIC Laboratory
Annecy-Le-Vieux, France
Email: {nabil.fakhfakh, herve.verjus, frederic.pourraz, patrice.moreaux}@univ-savoie.fr

***Abstract*—Quality attributes are an important concern in critical business processes supported through services orchestration. Various works in the literature have addressed quality attributes values computation of services orchestration. This paper proposes a novel approach that combines workflow patterns aggregation rules and a multi-criteria decision making method named MACBETH. This approach allows us to measure the satisfaction degree of services orchestration to the quality attributes requirements as defined at design-time.**

***Keywords*-quality of service/quality attributes aggregation; SOA; satisfaction degree measurement.**

I. Introduction

Service Oriented Architectures (SOAs) is an emerging paradigm for the development of business applications supported through services orchestrations. Services are the central concept of SOAs. Multiple services with same functionalities are spread over Internet. This makes clients faced to a large choice between functionally equivalent services as well as services providers faced to increased competition. Therefore, both clients and services providers need to agree on some guarantees: clients require guarantees that satisfy their expectations and services providers need guarantees to prevent from hazardous use of their services.

Today, quality attributes for services and services orchestrations become an important issue. At design-time, clients choose services satisfying services orchestration requirements in term of quality attributes (also referred as Quality of Service in the literature). However, at execution time, some services involved in the orchestration may unpredictably fail or change their quality attributes values positively or negatively. This will result in possible deviations of quality attributes values of the orchestration. Therefore, clients need to know how much the orchestration meets their requirements.

Various works dealing with computation of services orchestration quality attributes values exist in the literature [1] [2] [3]. Currently, none approach supports different kinds of quality attributes simultaneously in order to provide a high level information for decision making. The satisfaction degree of services orchestration is a such high level information. Measuring the satisfaction degree of services orchestrations can be seen as a Multi-Criteria Decision Making (MCDM) problem. In this paper, we present a novel approach giving a measurement of the satisfaction degree of services orchestrations. This approach has the advantage of taking into account clients preferences from quality attributes point of view. It is based on workflow patterns aggregation rules [1] [2] [3] and a MCDM method named the Measuring Attractiveness by a Categorical-Based Evaluation TecHnique (MACBETH) [4].

The remainder of this paper is organized as follows. Section II introduces related work on quality attributes values aggregation and MCDM methods. Section III details our approach for measuring services orchestration satisfaction degree, while Section IV concludes the paper.

II. Related Work

Various approaches have been proposed to compute each quality attribute value independently for services orchestration (e.g., giving response time values for all services composing the orchestration, how to compute the overall response time of the orchestration?). These approaches can be classified in two categories: *probabilistic models-based approaches* [6][7][8][9] and *workflow patterns-based approaches* [1][2][3][5]. The first category of approaches that allow to compute quality attributes values of services orchestration is based on probabilistic models. It consists in transforming the services orchestration model into a probabilistic model (e.g., Continuous Time Markov Chain (CTMC) model [6][7] or Discrete Time Markov Chain (DTMC) [8] or Stochastic Petri Nets (SPN)[9]). Then, the probabilistic model is annotated with quality attributes values. Finally, these approaches use tools like PRISM [6] or SPNP [9] to compute each quality attribute value of services orchestration. The major drawback of these approaches is that they support only reliability and/or response time. The second category consists in defining aggregation rules of quality attributes values for each composition pattern. A composition pattern is a combination of pairs of workflow patterns [10]; it is composed from one split pattern

 ISBN: 978-1-61208-126-7

Table I: Workflow Pattern-Based Approaches and Related Works

Research work	composition patterns	quality attributes
Jaeger and al. [2]	Sequence, loop, AND-AND, XOR-XOR, AND-XOR, OR-OR, OR-XOR, AND-N/M, OR-N/M	throughput, response time, cost, availability, reputation, encryption grade
Rosenberg and al. [3]	Sequence, loop, AND-AND, XOR-XOR	throughput, response time, cost, availability, reputation, encryption grade, scalability, accuracy, robustness
Cardoso and al. [1]	sequence, loop, AND-AND, XOR-XOR, Fault-tolerant systems	response time, cost, reliability, fidelity
Coppolino and al. [5]	sequence, loop, AND-AND, XOR-XOR, AND-N/M	reliability

(e.g., AND-split) and one join (synchronisation) pattern (e.g., XOR-join) except the sequence and loop patterns, which are considered individually. Table I summarizes, for each research work, the composition patterns taken into account and the supported quality attributes. The advantage of workflow patterns-based approaches is that they support larger set of quality attributes. Moreover, they are extensible: *(i)* more composition patterns could be added and *(ii)* new quality attributes could be integrated. For that reason, in our proposal, we will exploit a workflow patterns-based approach.

However, when changes affect positively and/or negatively some quality attributes values, it becomes difficult to estimate how much the whole orchestration fits client' expectations and satisfaction. Thus, it would be useful to have a high-level information. This information represents the services orchestration satisfaction degree. This could be done by aggregating the quality attributes values to provide only one value that measures the satisfaction degree of the services orchestration. To this end, we propose to use a MCDM method.

The most common MCDM method used in industrial applications is Analytical Hierarchy Process (AHP) [11]. It is based on the Weighted Arithmetic Mean (WAM) operator. AHP allows the elementary performance expression and the WAM operator weights quantification using human expertise. However, this method suffers from a lack of consistency between the expression of elementary performances step and the determination of the weights step: the weights being expressed on a ratio scale are not consistent with the interval scale of the elementary performances.The MACBETH method [4], as the AHP method, defines quantitative performance expression and aggregation from qualitative pairwise comparisons of situations issued from the decision-maker. But unlike AHP, the MACBETH methodology satisfies the measurement theory requirements (i.e., with respect to the commensurability and signifiance [4]). That's why, we choose the MACBETH method to aggregate quality attributes values and to measure the satisfaction degree of the services orchestration.

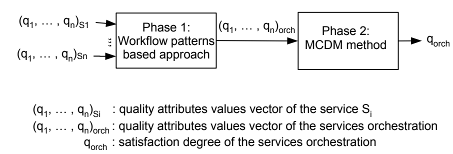


Figure 1: Principe of the Aggregation Approach

III. Aggregation proposal

In this section, we introduce our proposal for measuring the satisfaction degree of services orchestrations. We define the satisfaction degree as the percentage of respect to quality attributes requirements specified at design time (e.g., in SLAs). In this paper, we do not deal with requirements definition and we assume that these requirements are provided.

Each service S_i, involved in a given services orchestration, has a quality attributes values vector (see Fig. 1) denoted as: $(q_1, ..., q_n)_{Si}$, where $q_{1 \leq j \leq n}$ is the quality attribute value. Starting from these vectors, they are firstly aggregated in phase 1 using workflow patterns aggregation rules. This results in one vector of quality attributes values of the orchestration (see Fig. 1). Next, the quality attributes values of this vector will be aggregated in phase 2 using a MCDM method. That provides us only one value, which is the satisfaction degree of services orchestration. Before detailing these two phases, we present a simplified manufacturing process supported through a services orchestration described in Fig. 2. We will use this services orchestration model to illustrate our approach hereafter.

The process begins with launching a production request (service operation *LaunchProductionRequest*). Then, when the production terminates, the production planning is updated (service operation *UpdateProductionPlanning*); in parallel, a quality control is executed on the product (service operation *LaunchQualityControl*). If the control is positive, the product is moved to stock (service operation *MoveProductToStock*), otherwise it will be reported as wasted product (service operation *Report WasteProduct*).

Hereafter, we assume that quality attributes requirements

 ISBN: 978-1-61208-126-7

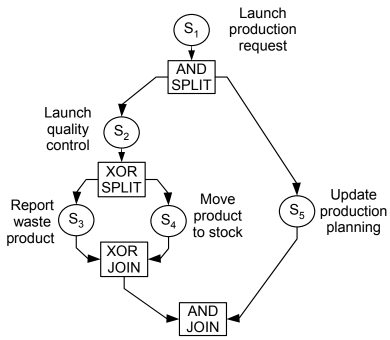


Figure 2: Example of Manufacturing Process

and measurements are given respectively by the client and a monitoring system (quality attributes measurement is out of the scope of this paper).

A. Phase 1: Aggregation Based on Workflow Patterns Rules

In the first phase (see Fig. 1), we use aggregation rules based on workflow patterns (see Section II) to compute each quality attribute value of the services orchestration. This consists in applying step-by-step rules in order to aggregate quality attributes values. The applied rules are those corresponding to the composition patterns used in the services orchestration model. Beginning from the most nested composition pattern, the orchestration model is parsed and aggregation rules for each value in the quality attributes values vector are progressively applied. This is done until reducing the whole orchestration model into a sole node (Fig. 3). The resulted quality attributes values of the end node form the quality attributes values vector of the services orchestration. This approach is relevant for each quality attribute value that has aggregation rules for the composition patterns. We will detail this phase through the services orchestration described in Fig. 2.

For simplification purpose, we will consider a set of three quality attributes values: *response time* (q_{rt}), *reliability* (q_{rel}) and *availability* (q_{av}). The aggregation rules for each of the composition patterns and for each of the quality attributes values are summarized in Table II. The first step consists in checking the most nested composition pattern, which is the XOR-XOR pattern between S_3 and S_4 in Fig. 3a. Then, we apply the respective aggregation rule from Table II. The quality attributes values computation of this composition pattern gives :

$$\begin{aligned} q_{rt}(S_{3,4}) &= p_1.q_{rt}(S_3) + p_2.q_{rt}(S_4) \\ q_{rel}(S_{3,4}) &= p_1.q_{rel}(S_3) + p_2.q_{rel}(S_4) \\ q_{av}(S_{3,4}) &= p_1.q_{av}(S_3) + p_2.q_{av}(S_4) \end{aligned}$$

Thus, the orchestration model is reduced to that given in Fig. 3b. Then, taking into account the reduced orchestration model, the next composition pattern to be considered is the sequence pattern of S_2 and $S_{3,4}$. The quality attributes values computation of this composition pattern gives :

$$\begin{aligned} q_{rt}(S_{2,3,4}) &= q_{rt}(S_2) + q_{rt}(S_{3,4}) \\ q_{rel}(S_{2,3,4}) &= q_{rel}(S_2).q_{rel}(S_{3,4}) \\ q_{av}(S_{2,3,4}) &= q_{av}(S_2).q_{av}(S_{3,4}) \end{aligned}$$

The following composition pattern identified in step 3 (Fig. 3c) is the parallel pattern associated to the synchronization pattern (AND-AND). The resulting quality attributes values after the reduction of the orchestration model are:

$$\begin{aligned} q_{rt}(S_{2,3,4,5}) &= \max(q_{rt}(S_{2,3,4}), q_{rt}(S_5)) \\ q_{rel}(S_{2,3,4,5}) &= q_{rel}(S_{2,3,4}).q_{rel}(S_5) \\ q_{av}(S_{2,3,4,5}) &= q_{av}(S_{2,3,4}).q_{av}(S_5) \end{aligned}$$

The obtained orchestration model from this step is composed of two nodes structured in sequence (Fig. 3d). By aggregating quality attributes values of these two nodes in sequence, the services orchestration is reduced to a sole node. The quality attributes values of this node form the quality attributes values vector of the whole services orchestration model. This vector is denoted as $(q_{rt}(orch), q_{rel}(orch), q_{av}(orch))$ where :

$$\begin{aligned} q_{rt}(orch) &= q_{rt}(S_{1,2,3,4,5}) = q_{rt}(S_1) + q_{rt}(S_{2,3,4,5}) \\ q_{rel}(orch) &= q_{rel}(S_{1,2,3,4,5}) = q_{rel}(S_1).q_{rel}(S_{2,3,4,5}) \\ q_{rel}(orch) &= q_{av}(S_{1,2,3,4,5}) = q_{av}(S_1).q_{av}(S_{2,3,4,5}) \end{aligned}$$

This resulting vector $(q_{rt}(orch), q_{rel}(orch), q_{av}(orch))$ or more simply $(q_{rt}, q_{rel}, q_{av})_{orch}$ will be the input of the phase 2 (Fig. 1).

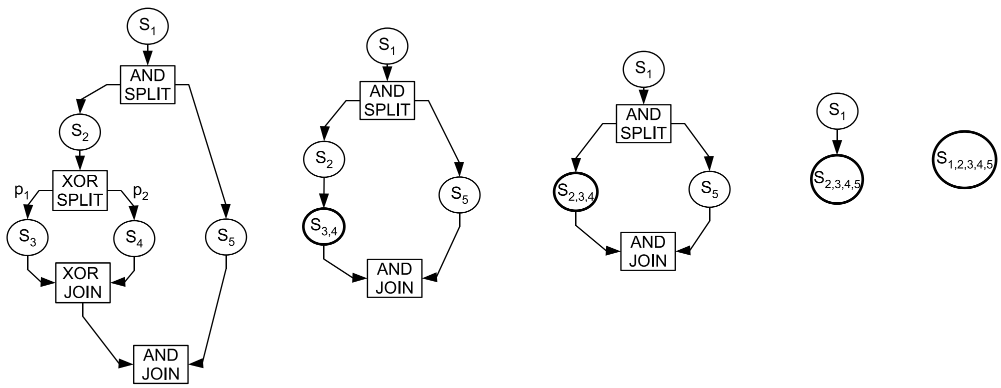
B. Phase 2: Aggregation Based on Weighted Mean Method (MACBETH)

The goal of this phase is to aggregate different values in the quality attributes values vector of the services orchestration (e.g., $(q_{rt}, q_{rel}, q_{av})_{orch}$) in order to obtain a measure of the satisfaction degree of the services orchestration (q_{orch}) (see Fig. 1). This measure allows us to interpret the positive or negative changes that affect the quality attributes values vector of the orchestration. For example, how to compare the vectors $(10, 0.95, 0.99)_{orch}$ and $(7, 0.90, 0.99)_{orch}$. A client who has strong time constraint may consider variation of the quality attributes values as an improvement of quality orchestration. An other one who is interested to reliability can report this variation as a degradation. For that purpose and in order to discriminate these alternatives, we use the MACBETH method [4]. MACBETH is a weighted mean

Table II: Aggregation Rules [1][3]

	Response Time	Reliability	Availability
Sequence	$\sum_{i=1}^{n} q_{rt}(s_i)$	$\prod_{i=1}^{n} q_{rel}(s_i)$	$\prod_{i=1}^{n} q_{av}(s_i)$
Loop	$q_{rt}(s_i) * c$	$q_{rel}(s_i)^c$	$q_{av}(s_i)^c$
AND-AND	$\max(q_{rt}(s_1), ..., q_{rt}(s_n))$	$\prod_{i=1}^{n} q_{rel}(s_i)$	$\prod_{i=1}^{n} q_{av}(s_i)$
XOR-XOR	$\sum_{i=1}^{n} p_i.q_{rt}(s_i)$	$\sum_{i=1}^{n} p_i.q_{rel}(s_i)$	$\sum_{i=1}^{n} p_i.q_{av}(s_i)$

[a] c denote the number of occurring loops
[b] p_i the probabilities of the outgoing branches for XOR-XOR



(a) Step 1 (b) Step 2 (c) Step 3 (d) Step 4 (e) Step 5

Figure 3: Workflow Pattern-Based Aggregation Steps

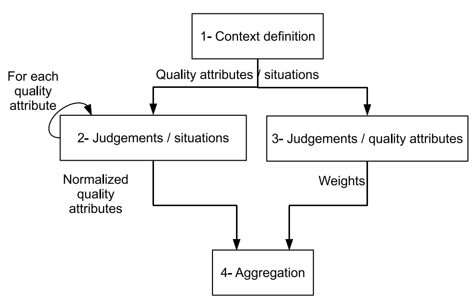


Figure 4: The Main Steps of MACBETH Method

method that allows us to translate quality attributes values to a satisfaction degree and thus thanks to :

- normalization of quality attributes values according to measurement theory (i.e., with respect to the commensurability and signifiance requirement [4]),
- determination of the weights of the weighted mean operator,
- aggregation of normalized quality attributes values.

It is based on pairwise comparisons of situations made by the client (the expert). The MACBETH method comprises four main steps (Fig. 4):

1) *Context definition step:* The first step consists in identifying the criteria, which are quality attributes in our case. Secondly, situations that will be compared are defined. In our context, situations are represented by vectors of quality attributes values that result from several instantiations of the services orchestration model. For example, we consider three instances of services orchestration represented by measured quality attributes values as follows:

$$S^1 = (q^1_{rt}, q^1_{rel}, q^1_{av}) = (10, 0.9, 1)$$
$$S^2 = (q^2_{rt}, q^2_{rel}, q^2_{av}) = (15, 0.6, 0.8)$$
$$S^3 = (q^3_{rt}, q^3_{rel}, q^3_{av}) = (5, 1, 0.6)$$

Two more reference situations are introduced denoted situation *good* and situation *neutral*. The *good* situation represents the total satisfaction of the requirements associated to the criteria (i.e., the best quality attributes values vector) while the *neutral* situation represents the minimum satisfaction of the requirements (i.e., the worst quality attributes values vector). These two reference situations could be retrieved for example from Service Level Agreement (SLA) contracts. For example :

$$S^{good} = (q_{rt}^{good}, q_{rel}^{good}, q_{av}^{good}) = (3, 1, 1)$$
$$S^{neutral} = (q_{rt}^{neutral}, q_{rel}^{neutral}, q_{av}^{neutral}) = (20, 0.5, 0.5)$$

Note that the *good* situation and the *neutral* situation are associated respectively to the vectors of elementary performance expression $(1, 1, 1)$ and $(0, 0, 0)$ (i.e., after normalization). Therefore, if we consider that these two situations are formed from upper and lower bounds of quality attributes values, all other situations will be classified between them.

2) *The elementary performance expression step:* In this step, the goal is to normalize quality attributes values. To this end, the client (the expert) uses his expertise to judge given situations and fulfill the matrix of judgements like the one given in Table III. Firstly, he is asked for each criterion (i.e., quality attribute) about his preferences between pairs of situations (including the two reference situations). If the client prefers situation S^i to S^j for a criterion k, this is noted as follows:

$$S^i \succ S^j$$

and means that for the normalized quality attributes values $q_k^i > q_k^j$. This is mapped in Table III into the classification of the situations by their order of preference.
Secondly, the client expresses his strengths of preference about the same situations. The strengths of preference are characterized with seven levels: *0=null, 1=very weak, 2=weak, 3=moderate, 4=strong, 5=very strong, 6=extreme* (see Table III). If the client cannot give his strengths of preference but only his preferences, this is noted by *positive* or more shortly *P*. The client prefers the situation S^i to S^j with a strength $h \in \{0, ..., 6\}$ for a criterion k i.e.,

$$S^i \succ^{\mathbf{h}} S^j$$

This is equivalent to :

$$q_k^i - q_k^j = \mathbf{h}\alpha$$

where α is a coefficient necessary to meet the condition q_k^i and $q_k^j \in [0, 1]$.
Example : We ask the client to compare the response time of the situations S^1 and S^2. He says that he prefers S^1 to S^2 i.e.,

$$S^1 \succ S^2 \Leftrightarrow q_{rt}^1 > q_{rt}^2$$

and that he prefers extremely $(h = 6)$ S^1 to S^2 i.e.,

$$S^1 \succ^6 S^2 \Leftrightarrow q_{rt}^1 - q_{rt}^2 = 6\alpha$$

Next, once all the strengths of preference between situations are provided. The matrix of judgements is

Table III: Preferences and Preferences Strengths for Response Time

Response Time	Good	S^3	S^1	S^2	Neutral
Good	No	strong [a]	P	P	P
S^3		No	very strong	P	P
S^1			No	extreme	P
S^2				No	extreme
Neutral					No

[a] 0=null, 1=very weak, 2=weak, 3=moderate, 4=strong, 5=very strong, 6=extreme

fulfilled (e.g., see Table III) and a system of equations can be extracted. It takes the form:

$$q_k^i - q_k^j = h\alpha$$

By solving this system of equations, elementary performance expressions (normalized quality attributes values) are quantified in the interval [0,1].
Example: for the strengths of preference expressed in Table III, the system of equations is the following :

$$\begin{aligned} (q_{rt}^{good} = 1)\ q_{rt}^{good} - q_{rt}^3 = 1 - q_{rt}^3 &= 4\alpha \\ q_{rt}^3 - q_{rt}^1 &= 5\alpha \\ q_{rt}^1 - q_{rt}^2 &= 6\alpha \\ (q_{rt}^{neutral} = 0)\ q_{rt}^2 - q_{rt}^{neutral} = q_{rt}^2 - 0 &= 6\alpha \end{aligned}$$

The above system is solvable and the solution is:

$$q_{rt}^1 = 0.5714,\ q_{rt}^2 = 0.2857,\ q_{rt}^3 = 0.8095$$

Note that the same procedure is established for each quality attribute (i.e., for reliability and availability too). In the same way, we get for reliability and availability:

$$q_{rel}^1 = 0.6364,\ q_{rel}^2 = 0.0909,\ q_{rel}^3 = 1$$
$$q_{av}^1 = 1,\ q_{av}^2 = 0.7333,\ q_{av}^3 = 0.4$$

3) *Weights determination step:* MACBETH is based on the weighted mean operator for the aggregation given as follows:

$$q_{ag} = \sum_{i=1}^{n} w_i.q_i \quad (1)$$

where w_i represents the relative importance of quality attribute q_i. Thus, to determine the weights w_i, the client expresses his judgements about the relative importance of quality attributes. To this end, he has to compare particular (possibly fictive) situations where only one normalized quality attribute value is set to 1 e.g., (0,...,0,1,0,...,0). So, the aggregated quality attribute value is reduced to $q_{ag} = w_i$, where q_{ag} is the aggregated value of the quality attributes values vector

Table IV: Expert Judgements for Weights Determination

	$(0,1,0)$	$(0,0,1)$	$(1,0,0)$	$(0,0,0)$
$(0,1,0)$	No	Strong	Positive	Positive
$(0,0,1)$		No	Very weak	Positive
$(1,0,0)$			No	Very weak
$(0,0,0)$				No

with $q_i = 1$ and $q_j = 0$ for $i \neq j$. To determine the *n* weights of the quality attributes, the client has to give at least *n* strengths of preference between considered situations (e.g., Table IV). This results in *n* equations:

$$q_{ag}^{i} - q_{ag}^{j} = h\alpha = w_i - w_j$$

with the condition $\sum_{i=1}^{n} w_i = 1$. Note that $h \in \{0,...6\}$ represents the strengths of preference.

Example: let us consider a quality attributes values vector with three quality attributes values $(q_{rt}, q_{rel}, q_{av})$. So we have to compute three weights w_{rt}, w_{rel} and w_{av}. Taking the matrix of judgements fulfilled by the client (Table IV), we can write the following equations :

$$\begin{aligned} q_{ag}^{(0,1,0)} - q_{ag}^{(0,0,1)} &= 4\alpha = w_{rel} - w_{av} \\ q_{ag}^{(0,0,1)} - q_{ag}^{(1,0,0)} &= \alpha = w_{av} - w_{rt} \\ q_{ag}^{(1,0,0)} - q_{ag}^{(0,0,0)} &= \alpha = w_{rt} - 0 \\ w_{rt} + w_{rel} + w_{av} &= 1 \end{aligned}$$

The solution of the system of equations is:

$$w_{rt} = 0.1111, \; w_{rel} = 0.6666, \; w_{av} = 0.2223$$

4) *Aggregation step:* In this step, as quality attributes values are normalized (step 2) and weights are computed (step 3), we have just to apply the formula (1) for every situation characterized by a quality attributes values vector $(q_1, ..., q_n)$.
Example: The aggregated values of situations S^1, S^2 and S^3 are :

$$q_{ag}^{S^1} = 0.71 \; , q_{ag}^{S^2} = 0.2553 \; , q_{ag}^{S^3} = 0.8455$$

The aggregation results show that the situation S_3 is better than S_1 and that S_1 is better than S_2.

IV. Conclusion and Future Work

In this paper, we have proposed a novel approach for measuring the satisfaction degree of services orchestrations. Our work aims to preserve services consumer satisfaction and to do the necessary adaptations whenever any violation of the services consumer requirements occurs. However, as seen in Section III-B, the reference situations (situations *Good* and *Neutral*) are defined respectively by the best quality attributes values and by the minimum ones agreed by the client. Therefore, the minimum satisfaction degree that can be measured (i.e., equal to ”0”) correspond to the minimum quality attributes values agreed by the client. So, the approach does not allow us to predict possible violation of client requirements. This will be the subject of our future work.

Acknowledgements

This work is partially funded by the FEDER MES project granted by the French Rhône-Alpes Region.

References

[1] J. Cardoso, J. Miller, A. Sheth, and J. Arnold, “Modeling quality of service for workflows and web service processes,” *Journal of Web Semantics*, vol. 1, pp. 281–308, 2002.

[2] M. C. Jaeger, “Optimising quality-of-service for the composition of electronic services,” Ph.D. dissertation, Berlin University, Germany, 2007.

[3] F. Rosenberg, “Qos-aware composition of adaptive service-oriented systems,” Ph.D. dissertation, Technical University Vienna, Austria, June 2009.

[4] V. Cliville, L. Berrah, and G. Mauris, “Quantitative expression and aggregation of performance measurements based on the macbeth multi-criteria method,” *International Journal of Production Economics*, vol. 105, pp. 171–189, 2007.

[5] L. Coppolino, L. Romano, N. Mazzocca, and S. Salvi, “Web services workflow reliability estimation through reliability patterns,” *Security and Privacy in Communications Networks and the Workshops,*, pp. 107–115, 2007.

[6] S. Gallotti, C. Ghezzi, R. Mirandola, and G. Tamburrelli, “Quality prediction of service compositions through probabilistic model checking,” in *QoSA ’08: Proceedings of the 4th International Conference on Quality of Software-Architectures*, Berlin, Heidelberg, 2008, pp. 119–134.

[7] N. Sato and K. S. Trivedi, “Stochastic modeling of composite web services for closed-form analysis of their performance and reliability bottlenecks,” in *ICSOC ’07: Proceedings of the 5th international conference on Service-Oriented Computing*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 107–118.

[8] V. Cortellessa and V. Grassi, “Reliability modeling and analysis of service-oriented architectures,” in *Test and Analysis of Web Services*, 2007, pp. 339–362.

[9] D. Zhong and Z. Qi, “A Petri net based approach for reliability prediction of web services,” in *OTM Workshops (1)*, 2006, pp. 116–125.

[10] W. van der Aalst, A. ter Hofstede, B. Kiepuszewski, and A. Barros, “Workflow patterns,” in *Distributed and Parallel Databases*, vol. 14 (1), 2003, pp. 5–51.

[11] E. H. Forman and M. A. Selly, *Decision by objectives: how to convince others that you are right*. World Scientific Pub Co Inc, 2001.