5-15-2012

QUALITY OF PROCESS ? A BUSINESS PROCESS PERSPECTIVE ON QUALITY OF SERVICE

Rico Knapper
Research Center for Information Technology

Daniel Poodratchi
Research Center for Information Technology

Lennart Job
Research Center for Information Technology

QUALITY OF PROCESS – A BUSINESS PROCESS PERSPECTIVE ON QUALITY OF SERVICE

Knapper, Rico, Research Center for Information Technology, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany, knapper@fzi.de

Poodratchi, Daniel, Research Center for Information Technology, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany, poodra@fzi.de

Lennart, Job, Research Center for Information Technology, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany, job@fzi.de

The fundamental paradigm shift from a product- to a service-oriented economy implies novel technical and organizational challenges. The resulting dynamic of the technical infrastructure and the increasing development towards requesting external business services to be integrated into end-to-end business processes requires mechanisms ensuring the reliability of the organization's composed services, workflows and business processes. From a business perspective, QoS characteristics defined based on technical services within the infrastructural layer have to be aggregated to more business-relevant Key Performance Indicators on business process layer to express the Quality of Process. These KPIs represent quality that is highly related to the business's performance (e.g. processing time of a business service) and are crucial for achieving predefined goals in order to stay competitive in the market. The contribution of this paper is threefold: We (i) provide an in-depth requirements analysis for such a holistic quality management framework, we (ii) develop a holistic aggregation framework which enables service level aggregation incorporating the loosely coupled structure of business processes with invoked systems and services in an instance based manner. To demonstrate the expressive power of our framework we (iii) provide an exemplary industrial application scenario and illustrate the functioning and interplay of the designed artifacts.

Keywords: Quality of Service, Quality of Process, Business Process, Service Composition.

1 Introduction

The fundamental paradigm shift from a product- to a service-oriented economy implies novel technical and organizational challenges. The value generated by a service is mainly represented by intangible elements exposed at execution (Hill, 1977). Therefore, a service consumer expects a service to function reliably and to deliver a consistent outcome at a variety of levels, i.e. Quality of Service (QoS). To provide, control and assure QoS it is necessary to focus on functional properties of a service as well as on non-functional aspects. From an economic perspective, QoS is the most important characteristic that differentiates service offerings and leverages market advantage, as price competition is tough due to low variable costs of service provisioning. Thus, QoS is the key criterion to keep the business side competitive as it has serious implications on the provider and customer side (Papazoglou, 2008). The provision of services with a defined QoS over electronic networks such as the Web is challenging due to issues like infrastructure problems, unpredictable reliability, low performance of Web protocols and many more. In addition, the distributed nature of Web service environments and their high degree of complexity requires a comprehensive description of Web service quality characteristics, both functional and non-functional. For detailed information about the main aspects of QoS in a Web service context, the interested reader is referred to (Cardoso, Sheth, Miller, Arnold, and Kochut 2004; Liu, Ngu, and Zeng 2004; Mani and Nagarajan 2002; Papazoglou, 2008; Zeng, Benatallah, Dumas, Kalagnanam, and Sheng, 2003).

Companies tend to concentrate on their core competencies while requesting modularized business services from different service providers. Service-oriented architectures (SOAs) enable the seamless integration of distributed services into end-to-end business processes (BP)[1]. That is the BP host underlays various BP steps provided by functionality of appropriate services and diverse sources. Since the single service component does not provide value for the customer without being combined with other components it is important to compute the overall quality level of the BP – the Quality of Process (QoP). Hence, from a business perspective, QoS characteristics defined based on technical services within the infrastructural layer have to be aggregated to more business-relevant Key Performance Indicators (KPIs) to express the QoP (Jaeger, Rojec-Goldmann, and Mühl, 2004; Jaeger, Rojec-Goldmann, and Mühl, 2005; Cardoso, Sheth, Miller, Arnold, and Kochut, 2004; Knapper, Blau, Speiser, Conte, and Weinhardt, 2010; Canfora, Penta, Esposito, and Villani, 2008). These KPIs represent service quality that is highly related to the business's performance (e.g. processing time of a business service) and are crucial for achieving predefined goals in order to stay competitive in the market. The quality of the BP' output is essential for the corporation since it directly impacts the company's profit, the customers' satisfaction, and the company's reputation.

Coping with the described issues, our contribution is threefold: We (i) provide an in-depth requirements analysis and present related approaches and their shortcomings. Following a design science approach (Hevner, March, Park, and Ram, 2004), we (ii) develop a holistic aggregation framework which enables service level aggregation incorporating the loosely coupled structure of BPs with invoked systems and services in an instance based manner. The framework deals with aggregation issues along the whole lifecycle of a SOA system according to the PDCA (plan–do–check–act) cycle. To demonstrate the expressive power of our framework we (iii) provide an exemplary industrial application scenario and illustrate the functioning and interplay of the designed artifacts.

The remainder of this paper is structured as follows: In Section 2, requirements upon an approach to aggregate the Non-Functional Attributes (NFA) on a process level are identified. Based on these results, Section 3 analyses current work in this domain. Coping with the shortcomings of these

[1] In this paper we do only use the term “business process” without limiting the approach which can be applied just as well to complex services, composite services, workflows etc.

approaches based on the outlined requirements, we present the holistic process instance based aggregation framework in Section 4. To demonstrate the expressive power and applicability of our approach, Section 5 provides a numerical case study. Finally, Section 6 summarizes our contribution and outlines open questions and future work.

2 Scope & Requirements

In this section we analyze the fundamental requirements upon a Quality Management Framework (QMF) which forms the base for specification, estimation, monitoring and controlling (Cardoso, Penta, Esposito, and Villani, 2004) NFAs in SOAs. Business value on customer side is generated on BP level. Hence, considering quality on a service level (QoS) is not sufficient. Since the needed QoP depends on the underlying QoS of invoked services, a QMF, which allows for dealing with QoP in all BP lifecycle steps[2], is a fundamental need. In this section, following a design science approach (Hevner, March, Park, and Ram, 2004), requirements upon such a framework are derived based on current literature, which mainly focuses on a more workflow oriented view without considering correlations between service (QoS) and business (QoP) level (literature is specified in the section below).

Lifecycle

Deal with NFAs and their aggregation at (R01) design-time and (R02) run-time, including (R03) real-time NFA monitoring.

Based on the work of Cardoso, Sheth, Miller, Arnold, and Kochut (2004) we deliver the lifecycle requirements by generalizing them in the context of the Plan-Do-Control-Act (PDCA) cycle. In the planning-step (design-time of the BP system) of the PDCA cycle (Deming, 2000), the definition and specification of NFA metrics for atomic services (QoS) and for BPs (QoP) takes place. After NFA specification the do-step requires a QMF, providing the ability to deal with these metrics. In the check-step (run-time of the BP system), the need to monitor their compliance arises. Especially the context of B2B and B2C relationships and the trend of requesting business services or even BPs from external providers, clarifies the particular meaning of monitoring NFA to ensure the compliance with Service Level Agreements (SLAs) between suppliers and customers. Furthermore, monitoring QoP, in addition to monitoring SLA fulfillments on service level, is essential from a customer perspective. The interest of both parties to discover possible deviations in a very short time and to minimize the period between the check and act phase, motivates the need of monitoring NFA in real-time. This monitoring aspect and the fact, that we have to deal with the impact of QoS changes on QoP, leads to the requirement to be able to provide aggregation algorithms at run-time.

Attribute Representation

Support QoS aggregation of (R04) absolute NFAs values and (R05) probability distributions.

Corresponding to the work of Hwang, Wang, Tang, and Srivastava (2007), which argues that it's not realistic enough to treat NFAs as deterministic values, we support this thesis. On the one hand, the NFA aggregation of absolute values as well as probability distributions is grounded on the property of determinism and non-determinism. QoS attributes like security are deterministic and do not vary. In this case, it is sufficient to aggregate absolute NFA values. However, in the case of non-deterministic attributes (e.g. availability, response-time), which vary by different factors (e.g. time, instance, current load), the aggregation of NFA probability distributions is required. On the other hand, considering non-deterministic NFAs, we have to distinguish between design-time and run-time. At design-time

[2] In our work, the Plan-Do-Check-Act (PDCA) lifecycle is taken as the generic basis for a BP lifecycle model. However, our requirements as well as our model are not limited to PDCA but can also be transformed to other well-known lifecycle models.

(plan- and do-step), the QMF has to provide NFA estimations for BPs. In the case of services and processes integrated from external providers, only absolute values (e.g. delivered by SLAs or derived as expected value out of historical data) are usually available. Therefore, it is sufficient to aggregate these absolute values to estimate QoP. In the case of internal services, when probability distributions are available, the QoS aggregation of these distributions represents the estimated overall QoP best. However, in terms of NFA monitoring at run-time (cf. (R02), (R03)), it is sufficient to aggregate absolute values in real-time to then represent these values as probability distributions for BPs.

Process Representation

Provide a QMF which is able to deal with (R06) generic processes and supports the NFA aggregation by (R07) instance for the largest possible range of process definitions.

Regarding the wide range of workflow patterns (Van Der Aalst, Hofstede, Kiepuszewski, and Barros, 2003) in present Workflow Management Systems (WFMs) and the ongoing development in this field[3], it is a nearly unsolvable challenge to provide NFA aggregation for each pattern to fulfill a complete quality management of BPs. Instead of pattern oriented aggregation algorithms, which can deal only with a limited range of workflow patterns and need to get tight coupled with the process execution system, we consider instance based aggregation algorithms to deal with a larger range of processes and execution systems. A second aspect, which shows the advantages of instance based aggregation mechanisms, is the wide range of applications in heterogeneous SOAs and non- or partially-automated processes. A remarkable part of processes in organizations does not run in process execution systems and is not sufficiently formally defined. It is therefore essential to provide the ability to deal with these kinds of processes in a QMF. An instance based approach, which aggregates the NFA values and probability distributions for each instance to then generate overall QoP, uncouples the aggregation of NFA from the formal representation of processes.

Integration

Provide a (R08) loosely coupled approach to enable the largest possible compatibility in heterogeneous SOAs and to enable the required flexibility to (R09) vertically integrate a QMF from service to business level. Supporting (R08) and (R09) is the fundamental base for measuring cross-level dependencies (e.g. service and BP level) and for enabling (R10) an NFA management.

The requirement of vertical integration is needed to deal with BPs interacting throughout different abstraction layers in a service-oriented enterprise. Especially in the case of QoS deviations, the measurement of the impact on high level BPs requires a vertically integrated approach. Therefore, a holistic QMF, subjected to this requirement, has to meet the challenge to interact with heterogeneous and varying enterprise systems at different business levels in organizations. The most suitable way to overcome this challenge is to provide a loosely coupled organizational integration. Main purpose of this work is enabling an overall managing component, as part of a QMF, to monitor, analyze, and control NFA. In combination with an aggregating NFA approach, according (R01) – (R09), such a component allows for managing NFA throughout the enterprise.

3 Related Work

Current approaches do not fulfill all requirements discussed in our requirement analysis. They are mainly centered on QoS management in the context of B2B relationships and do not consider the meaning of general NFAs within an enterprise. This section presents the related work and classifies current approaches according to our requirement analysis (cf. Table 1).

[3] htp://www.workflowpatterns.com

Category	Requirements	Zeng, 2004	Cardoso, 2004	Jaeger, 2005	Hwang, 2007	Rosenberg, 2009	Dumas, 2010	Knapper, 2010	This work
Lifecycle	(R01) Design-time	●	●	●	●	●	●	●	●
	(R02) Run-time	●	●	●	●	●	●	○	●
	(R03) Real-time	○	○	○	○	○	○	○	○
Attribute Representation	(R04) Absolute Values	●	●	●	●	●	●	●	●
	(R05) Probability Distribution	○	○	○	○	●	○	○	○
Process Representation	(R06) Generic Process Coverage	○	◐	◐	◐	◐	◐	◐	●
	(R07) Instance Based Approach	○	○	○	○	○	○	○	●
Integration	(R08) Loose Coupling	○	◐	○	○	◐	○	●	●
	(R09) Vertical Integration	○	◐	◐	○	○	○	●	●
	(R10) Management	○	○	◐	○	◐	○	◐	●

Table 1 Related work classification according to the requirement analysis (cf. Section 2).

Current approaches (Jaeger, Rojec-Goldmann, and Mühl, 2004; Jaeger, Rojec-Goldmann, and Mühl, 2005; Cardoso, Sheth, Miller, Arnold, and Kochut, 2004; Knapper, Blau, Speiser, Conte, and Weinhardt, 2010; Canfora, Penta, Esposito, and Villani, 2008) address QoS *aggregation* by providing pattern based rules to aggregate overall QoP for well-structured service orchestration models (inspired by Van Der Aalst, Hofstede, Kiepuszewski, and Barros, 2003).

Dumas, García-Bañuelos, Polyvyanyy, Yang, and Zhang (2010) address the limitation of aggregating QoS only for well-structured orchestrations and provide a model that can deal with unstructured parts of control-flows. However, these approaches can deal only with a limited range of process definitions. Furthermore, they do not provide a loosely coupled way to aggregate QoS on QoP. Hence, they are limited to the application within process execution systems and do not take into account non- or partially-automated functions.

Regarding the application lifecycle, the QoP estimation of composed services during design-time is the focus of Dumas, García-Bañuelos, Polyvyanyy, Yang, and Zhang (2010); Jaeger, Rojec-Goldmann, and Mühl (2005); Cardoso, Sheth, Miller, Arnold, and Kochut (2004); Canfora, Penta, Esposito, and Villani, (2008).

Run-time support is provided by Zeng, Benatallah, Ngu, Dumas, and Kalagnanam (2004); Cardoso, Sheth, Miller, Arnold, and Kochut (2004); Jaeger, Rojec-Goldmann, and Mühl (2005); Rosenberg (2009); Dumas, García-Bañuelos, Polyvyanyy, Yang, and Zhang (2010). However, dealing with real-time NFA monitoring is still a lack in this domain.

The aspect, that QoS attributes like response-time or availability are not fix but vary by different factors (R05), has been ignored by the above approaches. Jaeger, Rojec-Goldmann, and Mühl (2004); Jaeger, Rojec-Goldmann, and Mühl (2005); Cardoso, Sheth, Miller, Arnold, and Kochut (2004); Knapper, Blau, Speiser, Conte, and Weinhardt (2010); Canfora, Penta, Esposito, and Villani, (2008); Dumas, García-Bañuelos, Polyvyanyy, Yang, and Zhang (2010) are limited to absolute values and cannot deal with probability distributions. Van Dinther, Knapper, Blau, Conte, Anandasivam (2010) overcome this shortcoming and provide a mathematical model dealing with probability distributions for the NFAs execution time, availability and cost. Hwang, Wang, Tang, and Srivastava (2007) also address this shortcoming by modeling QoS attributes as random variables and provide pattern based aggregation rules to estimate overall QoP for web-services-based workflows.

The lack of providing integrated NFA models throughout multiple enterprise layers is the main shortcoming in current literature. Zeng, Benatallah, Ngu, Dumas and Kalagnanam (2004) present a middleware platform focusing on the selection of the best service combination for composed services including mechanisms to register and automatically control services in terms of overall QoP. Cardoso, Sheth, Miller, Arnold, and Kochut (2004) focus on the importance of QoS in the context of workflows and therefore provide a QMF, that is able to specify, estimate, and monitor QoP for general workflows executed in WFMs. Rosenberg (2009) addresses the lack of an integrated QoS model and presents a holistic approach to enable a multi-layer QoS model to provide QoS-aware Service-Oriented Computing (SOC) throughout the application lifecycle. However, none of these approaches takes into account the impact of QoS attributes on multiple layers throughout the enterprise, but centers on dealing with composed services or workflows, especially in the context of B2B relationships. Regarding our integration requirements (R08) – (R10), there is a fundamental need for a holistic NFA monitoring approach that forms the base to analyze NFAs to determine the impact of QoS variation from service (composed services, workflows) to business layer (e.g. BPs). This allows for an overall perspective to proactively determine possible bottlenecks and to evaluate processes throughout the enterprise.

4 Aggregating Quality of Service

QoP in process-oriented and customer centric organizations has a direct impact on the success of the organization. Our approach forms the base to address the challenge of a) ensuring QoP and to b) identify the impact of exceptional deviations throughout the organization's BPs. The fact, that processes are embedded in complex, hierarchical structures and therefore depend among each other within and across (e.g. externally requested services/processes) the organization, requires a QoS aggregation approach to determine overall QoS for services and composed services on service layer and for QoP on process layer (cf. Figure 1). Figure 1 depicts a work in progress version of our NFA taxonomy representing the dependencies between KPIs on service-, process- and business layer. Evaluating this taxonomy can be regarded as important future work.

We provide a stepwise NFA aggregation approach based on every executed instance of a process model. This enables a customer centric overall perspective to provide an intelligent and integrated process monitoring which is able to interpret low-level QoS impacts on high-level BPs throughout the enterprise.

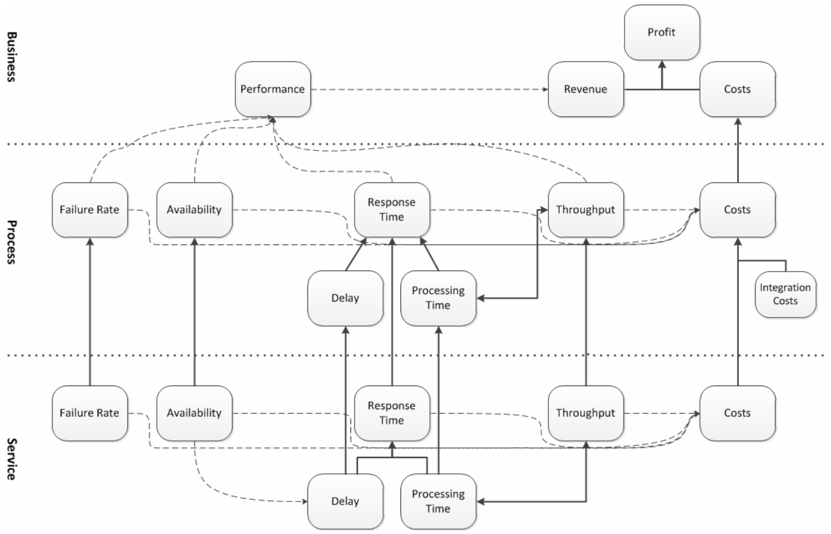

Figure 1 NFA Taxonomy.

4.1 Process Model Definition

For this paper, we use basic elements of the Event-driven Process Chain (EPC) as the underlying orchestration model. However, this is not a limitation, but our approach is adoptable to every process model. Figure 2 shows an exemplary EPC model.

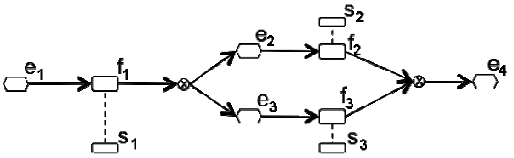

Figure 2 Process example.

We define a process model as a tuple $P=(E,F,C,S,V)$, with

$E=\{e_1,...,e_m\}\cup E_{\#}$: The events of P. $e_1\in E$: The starting event and $E_{\#}\subset E$: The set of ending events.

$F=\{f_1,...,f_n\}$: All functions of P.

$C\subseteq(E\times F)\cup(F\times E)$: The set of directed edges between functions and events.

$S=\{s_1,...,s_p\}$: The set of IT-services.

$V\subseteq(F\times S)$: The set of undirected edges between functions and services

The set $\Re(f_i)=\{s\,|\,(f_i,s)\in V\}$ contains all services, which are connected to the function f_i.

The process of gaining the values of NFAs for single services is beyond the scope of this research. We assume that NFA values are delivered by service or infrastructure monitoring systems. To formally define NFA metrics, we extend the process model using the sets M and Q_{m_i}. Therefore, a process is a tuple $P=(E,F,C,S,V,M,Q)$, with:

$M=\{m_1,...,m_l\}$: The set of NFA metrics (e.g. availability, response-time).

Q_{m_i}: The codomain of the NFA m_i (e.g. [0,1] for availability).

A NFA $m_i\in M$ is a function, $m:S\mapsto Q_{m_i}$ which maps each service to a value of Q_{m_i}.

4.2 Execution Paths

The concept of Execution Paths (EPs) splits a process model into a finite number of sub process models (called EP), so that each EP contains exactly the process elements executed for a certain instance (or in workflow language a case). Dealing with historical data, the possible number of EPs is finite because a loop is represented by a sequential control-flow within an EP. However, with regards to QoP estimation, the potential number of EPs for a process model containing loops is infinite. To deal with this problem, we remove the loops within an EP and instead assign them to the EP in form of an independent sub EP (called loop EP). Therefore, an instance is represented by an EP and optionally – when the instance is executing loops – by one or more loop EPs.

Consequently an EP contains only sequential and parallel control flows and therefore the probability of execution for every edge within the (loop) EP is 1.

The formal representation of a (loop) EP A_i is a tuple with $A_i=(E_i,F_i,C_i)$, with

$E_i=\{e_1,...,e_b\}$: The events of A_i.

For an EP, it holds that : $e_1 \in E$ is the starting event and $e_b \in E_\#$ is an ending event of A_i.

$F_i = \{f_c, ..., f_d\} \subseteq F$: The functions of A_i.

$C_i \subseteq ((E_i \times F_i) \cup (F_i \times E_i)) \subseteq C$: The set of directed edges between events and functions.

The tuple $A = (A_1^{EP}, ..., A_k^{EP})$ contains all possible EPs of a process model and $\Im(A_i^{EP}) = \{A_1^{lEP}, ..., A_z^{lEP}\}$ represents all loop EPs assigned to A_i.

4.3 The Layer Model

Our NFA aggregation approach is based on a multi-layer model, which allows for calculating overall QoP in a stepwise manner from a) service metrics to b) function metrics to c) EP metrics and finally to c) process metrics (cf. Figure 3). Generally, we assume that QoS attribute values are delivered by monitoring systems and the metrics in the upper layers are determined by aggregating QoS attribute values. For the layer model it makes no difference at what level the metrics are measured and therefore we consider only fully automated processes, where functions are provided by one or more services.

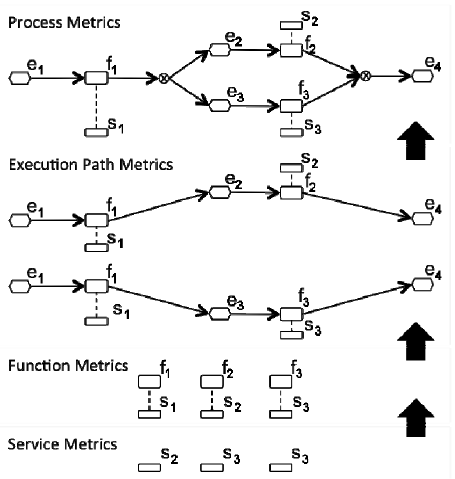

Figure 3 The Layer Model.

Having service metrics measured, the aggregation of these metrics to function metrics takes place. This layer contains only information about each function and it's providing services. The next step is to calculate metrics for the EPs (cf. section 4.2). The calculation is based on generic formulas explained below. Determining QoP is the last step and appears by aggregating the underlying EP metrics.

4.4 Layer Model Aggregation

The fact, that EPs contain only sequential and parallel control-flows leads to the generic aggregation formulas as follows:

1) Sequential aggregation for the NFA m_i: $f_{m_i}^{sequence} : Q_{m_i} \mapsto Q_{m_i}$

2) Parallel aggregation for the NFA m_i: $f_{m_i}^{parallel} : Q_{m_i} \mapsto Q_{m_i}$

4.4.1 Service Metrics

As described above, we assume that QoS attributes are available (delivered by monitoring systems).

4.4.2 Function Metrics

According to our process model, the execution of a function can consist of the execution of one or more services. In case of a single service providing the execution of a function, no aggregation is required and the service's QoS attribute value equals the function's QoS attribute value. In the case of multi services per function, we assume a sequential control-flow model. Since such complex control-flow types can be regarded as (sub) workflow models themselves, the recursive application of our layer model allows for dealing with such constellations as well. We define the formula for aggregating NFAs on function layer for the NFA attribute m_i as follows:

$$f_{m_i}^{function} : F \mapsto Q_{m_i}$$

$$f_{m_i}^{function}(f_i) = f^{sequence}(m_i(s_1),...,m_i(s_n)) \text{ with } \{s_1,...,s_n\} = \Re(f_i)$$

4.4.3 Execution Path Metrics

EPs contain only sequential and parallel control-flow types. The formula to aggregate the EP's overall quality for the NFA m_i is defined as $f_{m_i}^{EP} : A_i \mapsto Q_{m_i}$.
In case of an equal sequential and parallel aggregation formula for a NFA m_i, it follows that there is only one formula for the aggregation of QoS for EPs:

If $f_{m_i}^{sequence} = f_{m_i}^{parallel}$, it holds that
$f_{m_i}^{EP}(A_i) = f_{m_i}^{sequence}(f_{m_i}^{function}(f_1),...,f_{m_i}^{function}(f_n)) = f_{m_i}^{parallel}(f_{m_i}^{function}(f_1),...,f_{m_i}^{function}(f_n))$ with $\{f_1,...,f_n\} = F_i$

4.4.4 Process Metrics

Determining NFAs for the process layer is done by calculating the weighted arithmetic mean of all EPs. At design-time, the weight $\alpha_{A^{EP}}$ per EP A^{EP} and the weight $\beta_{A^{lEP}}$ per loop EP (lEP) has to be estimated. At run-time the estimated values can be adjusted based on the number the EPs and loop EPs executed. The process aggregation formula for the NFA attribute m_i is defined as $f_{m_i}^{process} : A \mapsto Q_i$:

$$f_{m_i}^{process}(A) = \left(\sum_{A^{EP} \in A} \left(\alpha_{A^{EP}} \cdot f_{m_i}^{EP}(A^{EP}) \cdot \prod_{A^{lEP} \in \Im(A^{EP})} \left(A^{lEP} \right)^{\beta_{A^{lEP}}} \right) \right)$$

4.5 Aggregation Formulas

The QoS model, presented in Section 4, provides generic formulas to aggregate overall NFA for functions, EPs and processes. In this section we assign each NFA, illustrated in our NFA taxonomy to its calculation type for sequential and parallel control-flows (cf. Table 3).

QoS Attribute m_i		**Sequential Aggregation** $f^{sequence}$			**Parallel Aggregation** $f^{parallel}$			
QoS Attribute Name	Q_{m_i}	Additive	Multiplicative	Minimum	Additive	Multiplicative	Minimum	Critical-Path
Failure Rate m_{fr}	[0,1]		✖			✖		
Availability m_{av}	[0,1]		✖			✖		
Delay m_{de}	□ +	✖						✖
Processing-time m_{pt}	□ +	✖						✖
Response-time m_{rt}	□ +	✖						✖
Throughput m_{th}	□ +			✖			✖	
Cost m_{co}	□ +	✖			✖			

Table 2 QoS attribute aggregation classification.

The EP's aggregation formulas are presented in Table 3. The calculation of the overall NFAs for processes is generic and is presented in section 4.4.4.

QoS Attribute m_i	**EP Aggregation** f^{EP}
Failure Rate m_{fr}	$f^{EP}_{m_{fr}}(A_i) = \prod_{f \in F_i} f^{function}_{m_{fr}}(f)$
Availability m_{av}	$f^{EP}_{m_{av}}(A_i) = \prod_{f \in F_i} f^{function}_{m_{av}}(f)$
Delay m_{de}	$f^{EP}_{m_{de}}(A_i) = \sum_{i=1}^{n} f^{F}_{m_{de}}(f_i)$ with $(f_1,\ldots,f_n) =$ critical (max) path of A_i and and $\{f_1,\ldots,f_n\} \subseteq F_i$
Processing-time m_{pt}	$f^{EP}_{m_{pt}}(A_i) = \sum_{i=1}^{n} f^{F}_{m_{pt}}(f_i)$ with $(f_1,\ldots,f_n) =$ critical (max) path of A_i and and $\{f_1,\ldots,f_n\} \subseteq F_i$
Response-time m_{rt}	$f^{EP}_{m_{rt}}(A_i) = \sum_{i=1}^{n} f^{F}_{m_{rt}}(f_i)$ with $(f_1,\ldots,f_n) =$ critical (max) path of A_i and and $\{f_1,\ldots,f_n\} \subseteq F_i$
Throughput m_{th}	$f^{EP}_{m_{th}}(A_i) = \min_{f \in F_i} f^{function}_{m_{th}}(f)$
Cost m_{co}	$f^{EP}_{m_{co}}(A_i) = \sum_{f \in F_i} f^{F}_{m_{co}}(f)$

Table 3 QoS aggregation formulas.

5 Application Scenario

In order to present our NFA aggregation approach, we determine the QoP attributes for an exemplary industrial process illustrated in Figure 4. The process describes the treatise on an incoming call at the technical support department. The employee determines if the caller is already registered or not to then solve the customer's problem.

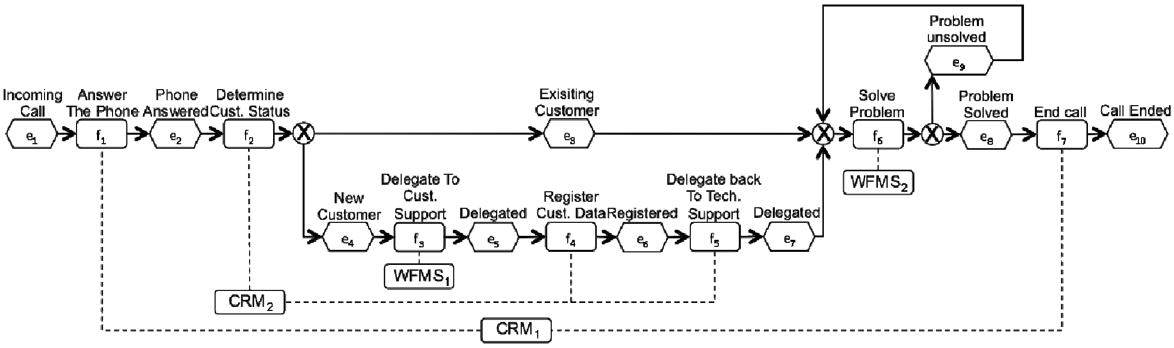

Figure 4 Application Scenario.

Presenting our NFA aggregation approach, we exemplary calculate the QoP attribute availability. We assume the following availability values: CRM_1: 0,98; CRM_2: 0,96; $WFMS_1$: 0,96; $WFMS_2$: 0,93. For the reader's convenience, we provide only the overall availability for the technical infrastructure and ignore the fact, that employees must be available to execute the process. Therefore, the service metrics are equal to the function metrics:

$f_{m_{av}}^{function}(f_1) = f_{m_{av}}^{function}(f_7) = 0,98$; $f_{m_{av}}^{function}(f_2) = f_{m_{av}}^{function}(f_4) = f_{m_{av}}^{function}(f_5) = 0,96$; $f_{m_{av}}^{function}(f_3) = 0,97$; $f_{m_{av}}^{function}(f_6) = 0,93$

The EPs of the process illustrated in Figure 4 are: (a) e_1-f_1-e_2-f_2-e_3-f_6-e_8-f_7-e_{10}; (b) e_1-f_1-e_2-f_2-e_3-f_6-$_{e9}$-f6-e_8-f_7-e_{10} with its associated loop EP: (b.1) e_9-f_6; (c) e_1-f_1-e_2-f_2-e_4-f_3-e_5-f_4-e_6-f_5-e_7-f_6-e_8-f_7-e_{10}; (d) e_1-f_1-e_2-f_2-e_4-f_3-e_5-f_4-e_6-f_5-e_7-f_6-e_9-f_6-e_8-f_7-e_{10} with its associated loop EP: (d.1) e_9-f_6. Based on this data, we lead to the availability for each (loop) EP by multiplying the invoked functions' availabilities (cf. Table 3):

$f_{m_{av}}^{EP}(a) \approx 0,857$; $f_{m_{av}}^{EP}(b) \approx 0,797$; $f_{m_{av}}^{EP}(b.1) \approx 0,93$; $f_{m_{av}}^{EP}(c) \approx 0,767$; $f_{m_{av}}^{EP}(d) \approx 0,713$; $f_{m_{av}}^{EP}(d.1) \approx 0,93$

Furthermore, we assume the following historical data illustrating how often the process, or more precise, its EPs were executed: a) 1 012; b) 10 239, b.1) 21 257; c) 5 982; d) 932, d.1) 2 494. This leads to the number of 18 165 executed instances. Last but not least we have to calculate the weights for the EPs and loop EPs. The EP weights α_i are calculated by dividing the number of EP executions by the overall number of instances. The loop EP weights β_i represent the relative number of EP executions and is therefore calculated by dividing the number of loop EP executions by the number of its associated EP executions.

$\alpha_a \approx 0,056$; $\alpha_b \approx 0,564$; $\alpha_c \approx 0,329$; $\alpha_d \approx 0,051$ and $\beta_{b.1} \approx 2,076$; $\beta_{d.1} \approx 2,676$

Now we are able to calculate the overall availability for the process:

$$f_{m_{av}}^{process}(a,b,c,d) = 0,857 \cdot 0,056 + 0,797 \cdot 0,564 \cdot (0,93)^{2,076} + 0,767 \cdot 0,329$$

$$+ 0,713 \cdot 0,051 \cdot (0,93)^{2,676} = 0,717$$

6 Conclusion and Future Work

The increasing development towards service-oriented, modularized paradigms throughout all enterprise layers (composed services, workflows, BPs) and the accompanying dynamic on the technical and BP infrastructure requires a QMF that is equally flexible but also highly integrated to enable a holistic, customer centric quality perspective. A generic concept of NFAs that can deal with this challenge and takes into account the complex, hierarchical structures within and across organizations' BPs design is perfectly suitable for such a fundamental approach.

Addressing these challenges, we consider NFAs as a least common denominator for a QMF allowing for provision, control and assure NFAs along the whole PDCA lifecycle of a SOA system in the context of an end-to-end BP integration. As a starting point, we provide an in-depth requirement analysis according to a QMF in the above-described manner. Based on the achieved requirements, we present the related work and point out the shortcomings of the one-sided technical, B2B-related view on QoS and QoP, which can be found in current literature. To address these issues, we present an aggregation approach, which forms the base to enable a QMF fulfilling our developed requirements. Contrary to current approaches our QMF is developed in a loosely coupled instance based manner, covering the whole PDCA lifecycle of SOAs. Section 5 illustrates our aggregation approach presenting an exemplary application scenario.

This work forms a base for future work in various fields. The presented aggregating formulas are limited to absolute NFA values. However, considering non-deterministic NFAs (e.g. availability) varying by various factors the aggregation of probability distributions represents the QoP best.

As an instance based framework, our approach is predestined for a customer centric quality management in the field of BP management (BPM) and business activity monitoring (BAM). In this context we plan to provide a proactively BP monitoring framework, which is adoptable to generic BPs based on the research presented in this paper.

References

Canfora, G., Di Penta, M., Esposito, R., Villani, M. (2008). A framework for qos-aware binding and re-binding of composite web services. Journal of Systems and Software, 81(10), 1754–1769.

Cardoso, J., Sheth, A., Miller, J., Arnold, J., Kochut, K. (2004). Quality of service for workflows and web service processes. Web Semantics: Science, Services and Agents on the World Wide Web, 1(3), 281–308.

Deming, W. (2000). Out of the Crisis. The MIT Press.

Van Der Aalst, W., Ter Hofstede, A., Kiepuszewski, B., Barros, A. (2003). Workflow patterns. Distributed and parallel databases, 14(1), 5–51.

Van Dinther, C., Knapper, R., Blau, B., Conte, T., Anandasivam, A. (2010). Managing the quality of modular services – a process-oriented aggregation of expected service levels based on probability distributions. Commerce and Enterprise Computing (CEC), 2010 IEEE 12th Conference, pp. 162–167. IEEE.

Dumas, M., García-Bañuelos, L., Polyvyanyy, A., Yang, Y., Zhang, L. (2010). Aggregate quality of service computation for composite services. Service-Oriented Computing, pp. 213–227.

Hevner, A., March, S., Park, J. and Ram, S. (2004). Design Science in Information Systems Research. Management Information Systems Quarterly, 28, 1, 75–106.

Hill, T. P. (1977). On Goods and Services. Review of Income and Wealth, 23 (4), 315-338.

Hwang, S., Wang, H., Tang, J., Srivastava, J. (2007). A probabilistic approach to modeling and estimating the qos of web-services-based workflows. Information Sciences 177(23), 5484–5503.

Jaeger, M., Rojec-Goldmann, G., Mühl, G. (2004) Qos aggregation for web service composition using workflow patterns. Proceedings of the Enterprise Distributed Object Computing Conference, Eighth IEEE International, p. 149 – 159.

Jaeger, M., Rojec-Goldmann, G., Mühl, G. (2005). Qos aggregation in web service compositions. Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05) on e-Technology, e-Commerce and e-Service, p. 181 – 185.

Knapper, R., Blau, B., Speiser, S., Conte, T., Weinhardt, C. (2010). Service Contract Automation. Proceedings of the 16th Americas Conference on Information Systems. Lima, Peru.

Liu, Y., Ngu, A. H., Zeng, L. Z. (2004). QoS Computation and Policing in Dynamic Web Service Selection. Proceedings of the 13th international World Wide Web conference on Alternate Track Papers & Posters, p. 66-73. New York, USA.

Mani, A., Nagarajan, A. (2002). Understanding quality of service for Web services. IBM Developer Works.

Papazoglou, P. (2008). Web Services: Principles and Technologies. Prentice Hall.

Rosenberg, F. (2009). Qos-aware composition of adaptive service-oriented systems. Dissertation.

Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J.,Sheng, Q. Z. (2003). Quality Driven Web Services Composition. Proceedings of the 12th international conference on World Wide Web ACM, p. 411-421. Budapest, Hungary.

Zeng, L., Benatallah, B., Ngu, A., Dumas, M., Kalagnanam, J., Chang, H. (2005). Qos-aware middleware for web services composition. Software Engineering, IEEE Transactions, 30(5), 311–327.