A Methodology to Define QoS and SLA Requirements in Service Choreographies

Victoriano P. Diaz and Daniel M. Batista
Computer Science Department
University of São Paulo (IME-USP)
Postal Code 166281 – Sao Paulo – SP – Brazil
Email: {alfonso7,batista}@ime.usp.br

***Abstract*—Service choreography allows the composition of services in a collaborative way, taking advantage of various benefits of P2P systems. The existing modeling approaches to evaluate choreographies are the interconnection and interaction models. However, these approaches don't evaluate choreographies focusing on QoS or in the earlier stages of development of the distributed application. This paper proposes an approach to assess the impact of QoS attributes specified in a choreography interaction model. With our proposal it is possible to establish requirements for QoS and SLA in early stages of development in order to plan the capacity of the network elements connecting the hosts involved in the enactment and deployment of the choreography.**

I. Introduction

The process to develop applications and complex systems evolved over time until converge to the approach know as Service Oriented Computing - SOC [1]. The service-based applications require compositions of services as a key factor to allow rapid development, loosely coupling and flexible integration with other systems. In this scenario, now, a single application is formed by several services spread over the network.

Among the various methods to compose services, Web services choreography is an efficient way to implement inter-organizational business processes, as the participants' business interactions are mutually independent (autonomous and heterogeneous) [2]. A service choreography is a description of peer to peer interactions among existing services, i.e., in this model there isn't the role of a central controller. The various services communicate with each other similarly to what occurs in a P2P application [3].

During the enactment of web services choreographies, the state of network elements (devices and links) plays a fundamental role. There must be guarantees of Quality of Service - QoS so that there are advantages of using a decentralized business model. A common method to define guarantees between a service provider and a client (which may also be a service) is by means of a Service Level Agreement - SLA. After the choreography be specified, constraints of QoS between each participant must be defined through SLAs [4]. To meet the SLA agreements there must be some mechanism for management at runtime. This mechanism must involve monitoring, control and decisions against violations or degradation of service quality. All the concern about guarantee of the QoS requirements of participants comes from the fact that the QoS of composite service, represented by the choreography, depends directly on the QoS of the separate services.

To reduce the number of SLA violations and the corrective measures to fulfill the QoS requirements during the enactment of choreographies, the detection of non-functional failures in early stages of the development of the choreography can be employed. For example, some non-functional failures (or undesirables properties) are communication and network errors such as high latency and low bandwidth.

This paper presents a new approach to detect non-functional failures of the participants in a choreography during the design stage. Our approach conducts a performance evaluation by means of generalized stochastic Petri nets (GSPN) as the intermediate representation of a choreography specified in BPMN2, a notational language to model business process, with the representation of the QoS model. Our work differs from those found in the literature because it is employed in stages before the development (for example, in the modeling and design). With our proposal it is possible to establish requirements for QoS and SLA in early stages in order to plan the capacity of the network connecting the hosts involved in the deployment of the application, reducing the costs and unexpected behaviors in later stages of the development.

This paper is organized as follows: Section II introduces the basic concepts necessary to understand the proposal and presents the example of a choreography specified in BPMN2. Section III presents the related work. Section IV describes the proposal, which includes the formalization, the QoS model and the mapping for GSPN. Section V presents the application of the proposal in a scenario with the objective to evaluate this effectiveness. Finally, Section VI presents the conclusions and future work.

II. Basic Concepts

The two main approaches to compose services are choreographies and orchestrations. A service choreography allow the composition of services in a collaborative manner. A choreography describes the P2P interactions of the externally observable behavior of its participants. Different from orchestrations, choreographies don't have a single point of control or coordination [3]. Choreography tend to be more efficient than

orchestrations because the probability to have a bottleneck in the system is reduced [5].

According to [6], languages to specify choreographies can be classified based on two criteria: implementation and approach used in modeling. In terms of approach, there are languages with interaction models and languages with interconnection models. To standardize the representation of choreographies, the BPMN graphical representation, currently in version 2.0, was proposed (http://www.bpmn.org/). It supports the modeling of interconnection through collaboration, and the modeling of interaction through choreography of processes. In this paper it is assumed that the choreography is represented by models of interaction because they incur less redundancies and errors [7].

Figure 1 shows the BPMN elements considered in this paper. We consider only this subset because they are enough to evaluate our proposal. To the reader interested, in [8] all the elements allowed in BPMN choreographies are presented.

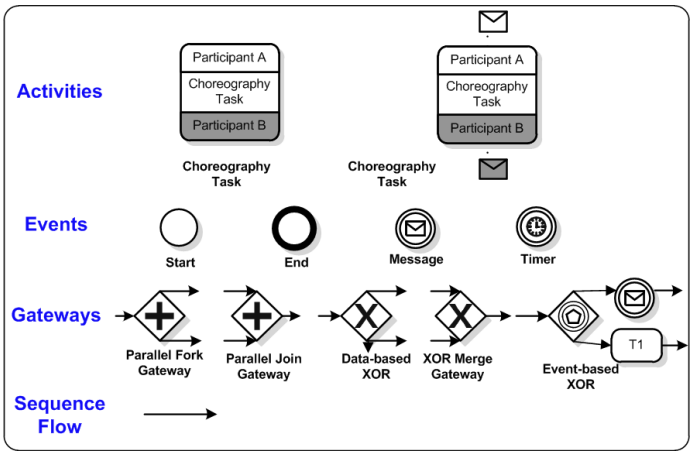

Figure 1: BPMN elements for modeling choreographies

Figure 2 presents an example of choreography modeled with elements presented in Figure 1. It is a choreography for the offer of investment in stock market presented in [9]. First, a financial advisor offers an investment to a customer and, after a time interval, the customer decides to accept or reject such a proposal, thereby ending the process.

In the scenario of Figure 2, basic elements of choreography processes are used, such as three choreography tasks, two events (*start* and *end*), six sequence flows and one gateway (*Data-based XOR*). A choreography task represents an atomic interaction between two or more participants performed by exchanging messages via the network.

III. Related Works

In [10], it is proposed an approach to predict the QoS of a Web service choreography specified in WSCI (*Web Service Interaction Choreography*). A GSPN is used as an intermediate model. An analytical evaluation is performed in the QoS metrics tree and transformation rules to map activities are specified. A problem with the proposal presented in [10] is that the WSCI language has been discontinued.

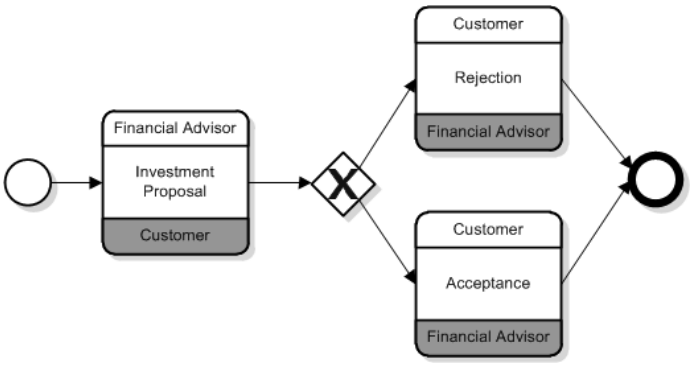

Figure 2: Example of choreography using BPMN2 elements.

Recently, [11] proposed an approach to predict the reliability of composite services specified in WS-CDL. It is also presented a complete translation of a choreography specified in WS-CDL to a GSPN. Similar to [11], we also use the formalism of Petri nets as an intermediate representation of a choreography. [11] is focused in reliability, while our proposal is focused in QoS attributes of communication network and we use the interaction model of BPMN2 as choreography language.

[12] and [13] present a proposal to evaluate the performance of choreographies represented in WS-CDL using Petri nets. However, WS-CDL is a limited language since it is more oriented to implementation, instead interaction between the participants. Our proposal is oriented to BPMN. For the best of our knowledge, there is no proposal that evaluate the performance of choreographies specified in BPMN.

IV. Methodology to Evaluate the Performance of Service Choreographies

The contribution of this paper is to present a new methodology for evaluate the performance of choreographies with the objective of define QoS requirements. The evaluation is performed in the design stage of the choreography by using GSPNs. The proposal is based on the execution of the following steps:

1) Mapping of the choreography specified in BPMN 2.0 to a GSPN. The GSPN includes the intermediate representation of the choreography and the QoS model that takes into account time and communication constraints of the interactions among participants (each participant represents a peer that is active during the enactment of the choreography)
2) Definition of weights in the transitions with time constraints in the GSPN. In this step, the probability distributions and weights in the interactions related to communications and message exchange are defined.
3) Simulations of scenarios, one with a minimum configuration of failures and another with a higher possibility of failure.

The next subsections present information needed to the execution of all the steps. Subsection IV-A defines choreogra-

phies of processes in BPMN. Subsection IV-B describes the QoS model considered in our proposal, and Subsection IV-C presents the algorithm executed to map BPMN elements to a GSPN.

A. Definition of choreographies specified in BPMN

Based on Figure 1 and on the work presented in [14], choreographies of BPMN processes can be defined as explained below:

Definition 1. *A choreography of BPMN processes is a tuple:* $PC = (\mathcal{O}, \mathcal{A}, \mathcal{E}, \mathcal{G}, \mathcal{T}, \mathcal{S}, \{e^S\}, \mathcal{E}^I, \{e^E\}, \mathcal{E}^{I_M}, \mathcal{E}^{I_T}, \mathcal{G}^F, \mathcal{G}^J, \mathcal{G}^X, \mathcal{G}^M, \mathcal{G}^V, \mathcal{F})$ *where:*

- $\mathcal{O}$ *is the set of objects, which is partitioned in three disjoint sets: activities* ($\mathcal{A}$)*, events* $\mathcal{E}$ *and* gateways $\mathcal{G}$.
- $\mathcal{A}$*, is the set of choreography tasks and it is the same as* $\mathcal{T}$.
- $\mathcal{E}$ *is the set of events. It is partitioned in three disjoint sets: Start event* e^S*, Intermediate events* $\mathcal{E}^{\mathcal{I}}$ *and End event* $e^{\mathcal{E}}$.
- $\mathcal{E}^{\mathcal{I}}$ *is partitioned in the disjoint sets of events of the type messages* $\mathcal{E}^{I_M}$ *and timer* $\mathcal{E}^{I_T}$.
- $\mathcal{G}$ *é the set of gateways and it is partitioned in the five disjoints sets of parallel fork gateways* $\mathcal{G}^F$*, parallel join gateways* $\mathcal{G}^J$*, data-based XOR gateways* $\mathcal{G}^X$*, XOR merge gateways* $\mathcal{G}^V$ *and event-based XOR gateways* $\mathcal{G}^M$.

B. QoS model

In order to analyze the non-functional behavior of choreographies, it is necessary to define the QoS attributes considered. In this paper we consider three entities with QoS attributes: service, network and message. In the operation of the service, the QoS attribute considered is the time needed to complete the service. In the network, the QoS attributes considered are delay and errors in communication, since both significantly affect the response time of the choreography. In the message, the QoS attribute considered is the format of the message, since the message need to be validated.

C. Mapping of BPMN choreography to GSPN

Figure 3 presents the mapping of gateways and events (choreography elements) to the respective modules of Petri nets. Figure 4 exemplifies the mapping of two different choreography tasks (choreography tasks are the basic building blocks of choreographies since they generate messages and service invocation). In Figure 4, positions A, A_2 and B represent the participants. Positions C_{init}, C_{end}, C_{init2}, C_{end2}, C_{endOk} and C_{endOk2} represent the start, end and the correct validation of the communication between two participants. Transitions of type T_{send} and $T_{receive}$ represent the actions of sending and receiving messages. Transitions T_{msg} represent the transmission of the message by the communication channel.

The Step 1 of our methodology is executed by the Algorithm 1. The algorithm maps a choreography of processes specified in BPMN 2.0 to a GSPN including the QoS model. Each BPMN element is mapped to its respective module of

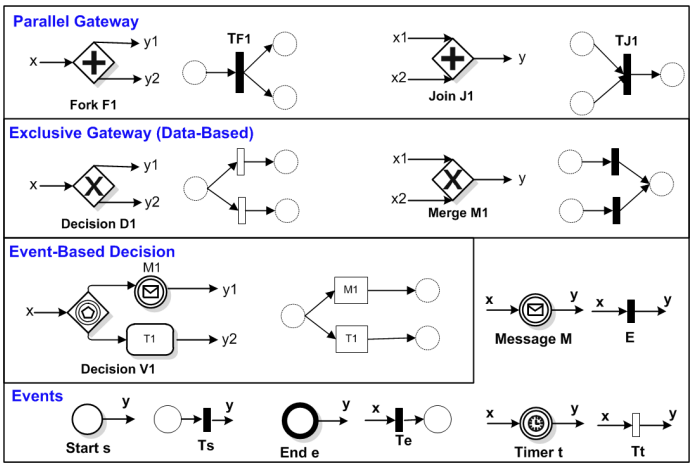

Figure 3: Mapping of events and gateways elements to modules of Petri nets

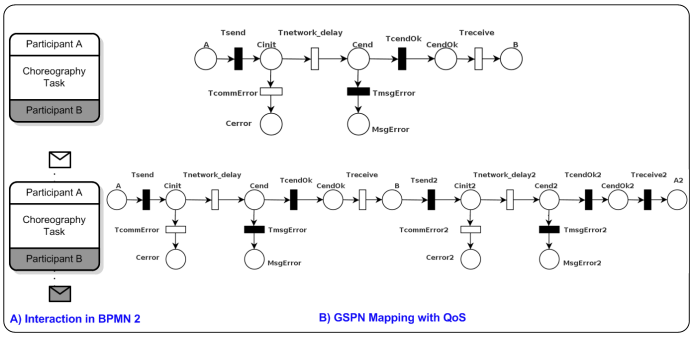

Figure 4: Mapping of two different choreography tasks with the QoS model

Petri nets and the modules are composed as a function of the sequence flows and the gateways. Finally, the Start and the End events are included.

V. Performance Evaluation

To attest the efficacy of our proposal, in this section all the steps of our methodology are performed. The choreography presented in Figure 2 is used in the process. Each one of the next subsections describes one of the three steps of the proposal defined in the Section IV.

A. Mapping

Figure 5 presents the GSPN obtained as the result of the mapping of the choreography of Figure 2. Tables I and II list the interpretation of all places and transitions of the GSPN. The places represent the participants, the start and the end of a communication channel in the interactions, the start and the end of the choreography and the points of decision and errors according to the QoS model.

B. Definitions of weights

The next step is to define the weights representing the timed transitions and the priorities in the immediate transitions. These values depend on the knowledge about the environment

Algorithm 1 Mapping of choreography specified in BPMN 2.0 to a GSPN with QoS model

Input: Process Choreography $PC = (\mathcal{O}, \mathcal{A}, \mathcal{E}, \mathcal{G}, \mathcal{T}, \mathcal{S}, \{e^S\}, \mathcal{E}^I, \{e^E\}, \mathcal{E}^{I_M}, \mathcal{E}^{I_T}, \mathcal{G}^F, \mathcal{G}^J, \mathcal{G}^X, \mathcal{G}^M, \mathcal{G}^V, \mathcal{F})$ in BPMN 2.0.
Output: Generalized Stochastic Petri Net $GSPN_{QoS}$.
Consider $CT_i \in \mathcal{T}$, $G_j \in \mathcal{G}$ and $E_k \in \mathcal{E}$. $i, j, k \in \mathbb{N}$
Consider $PNQoS(CT_i)$ the respective GSPN including QoS as a function of the type of CT_i and the mapping rules specified on Figure 4.
Consider $PNQoS(G_j)$ the respective GSPN including QoS as a function of the type of G_j and the mapping rules specified in Figure 3.
Consider $PNQoS(E_k)$ the respective GSPN including QoS as a function of the type of E_k and the mapping rules specified in Figure 3.
Consider $\oplus$ The binary operator of composition of two GSPNs that returns other GSPN.
$GSPN_{QoS} \leftarrow Empty\ Petri\ Net$
for $CT_i \in \mathcal{T}$ **do**
  $GSPN_{QoS} \leftarrow GSPN_{QoS} \oplus PNQoS(CT_i)$
  Add a arrival timed Transition at first of the $GSPN_{QoS}$.
end for
for $G_j \in \mathcal{G}$ **do**
  $GSPN_{QoS} \leftarrow GSPN_{QoS} \oplus PN(G_j)$
end for
for $E_k \in \mathcal{E}$ **do**
  $GSPN_{QoS} \leftarrow GSPN_{QoS} \oplus PN(E_k)$
end for
Add a starting Place and immediate Transition at the beginning of the $GSPN_{QoS}$.
Add a ending Place and immediate Transition the end of the $GSPN_{QoS}$.
return $GSPN_{QoS}$

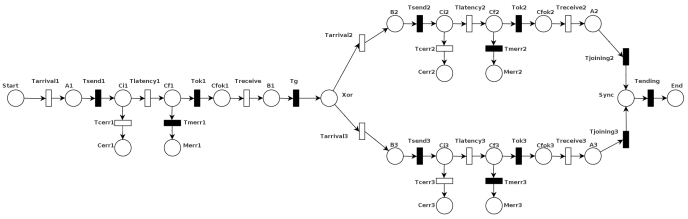

Figure 5: GSPN obtained from the choreography of Figure 2

Table I: Description of the places in the GSPN of Figure 5

Place	Description
$Start$, End	Start and end of choreography process.
A_1, A_2, A_3	Represents the participant "Customer".
B_1, B_2, B_3	Represents the participant "Financial Advisor".
C_{i1}, C_{i2}, C_{i3}	Start of the communication.
C_{f1}, C_{f2}, C_{f3}	End of the communication.
C_{fok1}, C_{fok2}, C_{fok3}	Valid message format .
C_{error1}	Communication error between A and B.
C_{error2}	Communication error between B and A.
C_{error3}	Communication error between B and A.
M_{err1}, 2, 3	Invalid message format.
Xor e $Sync$	*Fork* and *Join* of exclusive gateway XOR.

where the choreography will be executed and the reliability of the independent services. In this paper we don't consider uncertainties in these values. Table III presents the definition of the weights in two scenarios, that will be needed to the simulations on the next step. Scenario 1 is configured to have less probability of failures than Scenario 2.

Table II: Description of the transitions in the GSPN of Figure 5

Transition	Description
$T_{Starting}$	Starting choreography instance.
T_{Ending}	Ending choreography instance.
T_{send1}, T_{send2} e T_{send3}	Activity of message sending.
$T_{arrival1}$, 2, 3	Choreography instances arrival.
$T_{arrival2}$, 3	Flow decision.
$T_{latency1}$, 2, 3	Network latency
$T_{receive1}$, 2, 3	Receive and service execution.
$T_{cerror1}$, 2, 3	Communication error.
$T_{merror1}$, 2, 3	Format fault message.
T_g	Flow sequence continuation.
$T_{joining2}$, 3	Flow joining.

Table III: Weights of Scenario 1 and Scenario 2

	Weights	
Transition	Scenario 1	Scenario 2
$T_{latency1}$, $T_{latency2}$, $T_{latency3}$	0.99	0.94
$T_{cerror1}$, $T_{cerror2}$, $T_{cerror3}$	0.01	0.06
$T_{receive1}$, $T_{receive2}$, $T_{receive3}$	99	97
T_{merr1}, T_{merr2}, T_{merr3}	1	3
$T_{arrival2}$, $T_{arrival3}$	0.5	0.5

C. Simulations

To simulate the scenarios, 100 tokens are considered to each scenario at the place Start. Each token represent one instance of execution of the simulation. So, in each simulation, 100 concurrent instances of the scenarios were executed. Since there are probabilities involved in the experiments, as seen in Table III, the experiments need to be executed several times. There were defined 1500 fires and 10 replications. To simulate the scenarios, the simulator Pipe2 [15] was used.

Table IV shows the results of the simulations. The numbers presented in the table show the average of the tokens that arrived at each one of the places of the GSPN.

Table IV: Results (in %)

	Average number of tokens		95% Confidence interval (+/-)	
Place	Scenario 1	Scenario 2	Scenario 1	Scenario 2
$Start$	35,277	40,14833	5,83475	6,23113
End	41,94523	38,78167	2,53329	3,82088
A_1	0,07825	0,08333	0,00105	0,00369
B_2	0,03599	0,00469	0,00604	0,00651
B_3	0,03991	0,0099	0,00643	0,00742
B_1	0,07668	3,22932	0,00056	0,00168
A_2	0,03599	0,00843	0,00643	0,00724
A_3	0,04069	0,00531	0,00612	0,00638
M_{err1}	0,39202	0,90583	0,94914	1,92248
M_{err2}	0	0,9325	0,62632	0,63929
M_{err3}	0	0,65583	0,86536	0,74018
C_{err1}	0,73787	2,93833	0,81703	2,02391
C_{err2}	0	0	0,66667	1,75059
C_{err3}	0,77856	0.16233	0,91667	1,51997
C_{i1}	8,32316	8,90167	5,32508	7,47872
C_{i2}	0,62676	0,6875	0,22787	0,52219
C_{i3}	0,75039	8,90167	0,39	0,21236

By the results presented in Table IV it is possible to note that in the Scenario 1, an average of 1.51643% ($Cerr1 + Cerr2 +$

$Cerr3$) of the instances didn't finish the process because of error in the network in the moment of the message exchanges. Also, an average of $0,39202\%$ ($Merr1+Merr2+Merr3$) of the instances didn't finish the process because of error in the messages format, detected when the messages were received.

Similarly to the Scenario 1, in Scenario 2 $4,52167\%$ and $2,49416\%$ of the instances didn't finish the process because of error in the network and in the message format respectively. As expected, the results are inline with the parameters defined to each scenario (Table III).

One important finding with the simulation of these scenarios, and that can be useful to developers and network administrators is related with the existing bottleneck in the first iteration (C_{i1}), which avoided $8,32316\%$ and $8,90167\%$ of the instances for being send via network. This information can affect the implementation of the definite choreography or even define new QoS policies to be configured in the network to reduce the delay of the executions.

VI. Conclusions and Future Works

This paper presented a new methodology to evaluate the impact of QoS attributes in choreographies of process specified in BPMN 2.0 and following the interaction model. This methodology is executed in the earlier stages of the development with the objective of defining QoS constraints and SLAs. This definition allows a reduction in the failures when the choreography will be deployed and enacted. The methodology uses a new algorithm to map the choreography into a GSPN. The methodology was employed in two scenarios and the results confirmed its efficacy, since it was observed the presence of a bottleneck in the choreography that can be solved by changing the interaction in the development or by employing QoS mechanisms in the network to prioritize the traffic affected by the bottleneck.

As future work we will expand our methodology to support generic probability distributions in the decision points.

Acknowledgment

This work is supported by the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement number 257178 (project CHOReOS - Large Scale Choreographies for the Future Internet - www.choreos.eu).

References

[1] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-Oriented Computing: State of the Art and Research Challenges," *Computer*, vol. 40, pp. 38–45, 2007.

[2] P. R. Telang and M. P. Singh, "Specifying and Verifying Cross-Organizational Business Models : An Agent-Oriented Approach," *IEEE Transactions on Services Computing*, vol. 99, pp. 1–14, 2011.

[3] A. Barker, C. D. Walton, and D. Robertson, "Choreographing Web Services," *IEEE Transactions on Services Computing*, vol. 2, no. 2, pp. 152–166, 2009.

[4] F. Rosenberg, C. Enzi, A. Michlmayr, C. Platzer, and S. Dustdar, "Integrating Quality of Service Aspects in Top-Down Business Process Development Using WS-CDL and WS-BPEL," *Enterprise Distributed Object Computing Conference, IEEE International*, vol. 0, p. 15, 2007.

[5] A. Barker, P. Besana, D. Robertson, and J. B. Weissman, "The Benefits of Service Choreography for Data-Intensive Computing," in *In Proceedings of the 7th international workshop on Challenges of large applications in distributed environments*, ser. CLADE '09, 2009, pp. 1–10.

[6] O. Kopp, F. Leymann, and S. Wagner, "Modeling Choreographies: BPMN 2.0 versus BPEL-based Approaches," *Enterprise Modelling and Information Systems Architectures: Proceedings of the 4th International Workshop on Enterprise Modelling and Information Systems Architectures, EMISA 2011*, 2011.

[7] G. Decker and A. Barros, "Interaction Modeling using BPMN," in *In Proceedings of the 2007 international conference on Business process management (BPM'07)*. Springer-Verlag, Berlin, Heidelberg, 2008, pp. 208–219.

[8] OMG, "Documents Associated With Business Process Model And Notation (BPMN) Version 2.0," 2011. [Online]. Available: http://www.omg.org/spec/BPMN/2.0/

[9] O. Kopp, F. Leymann, and F. Wu, "Mapping Interconnection Choreography Models to Interaction Choreography Models," *Management*, pp. 1–8, 2010.

[10] Y. Xia, J. Chen, M. Zhou, and Y. Huang, "A Petri-Net-Based Approach to QoS Estimation of Web Service Choreographies," in *Advances in Web and Network Technologies, and Information Management*. Berlin, Heidelberg: Springer-Verlag, 2009, no. 2006, pp. 113–124.

[11] Y. Xia, G. Dai, J. Li, T. Sun, and Q. Zhu, "A model-driven approach to predicting dependability of WS-CDL based service composition," *Language*, no. January, pp. 1127–1145, 2011.

[12] D. Bruneo, S. Distefano, F. Longo, and M. Scarpa, "QoS Assessment of WS-BPEL Processes through non-Markovian Stochastic Petri Nets," 2010.

[13] V. Valero, H. Macià, J. J. Pardo, M. E. Cambronero, and G. Díaz, "Transforming Web Services Choreographies with priorities and time constraints into prioritized-time colored Petri nets," *Science of Computer Programming*, vol. 77, no. 3, pp. 290–313, Mar. 2012.

[14] R. M. Dijkman, M. Dumas, and C. Ouyang, "Formal Semantics and Analysis of BPMN Process Models using Petri Nets," Australian Research Council under Discovery Grant DP0451092, Tech. Rep., 2007. [Online]. Available: http://eprints.qut.edu.au/7115/

[15] P. Bonet, R. Puigjaner, P. D. Mallorca, W. J. Knottenbelt, S. K. Campus, and U. Kingdom, "PIPE v2.5: a Petri Net Tool for Performance Modeling," Tech. Rep., 2007.