

A Methodology to Define QoS and SLA Requirements in Service Choreographies

Authors

Victoriano Alfonso Phocco Diaz
Daniel Macedo Batista

Department of Computer Science
University of Sao Paulo
`alfonso7@ime.usp.br`, `batista@ime.usp.br`

September 18, 2012

Agenda

- 1 Introduction
- 2 Problem
- 3 Methodology
- 4 Performance Evaluation
- 5 Conclusions and Future Works

- 1 Introduction
- 2 Problem
- 3 Methodology
- 4 Performance Evaluation
- 5 Conclusions and Future Works

SOC (Service Oriented Computing)

It is a new computing paradigm that utilizes services as the basic constructs to support the development of rapid, low-cost and easy composition of distributed applications even in heterogeneous environments. [Papazoglou et al., 2006].

Key elements:

SOC (Service Oriented Computing)

It is a new computing paradigm that utilizes services as the basic constructs to support the development of rapid, low-cost and easy composition of distributed applications even in heterogeneous environments. [Papazoglou et al., 2006].

Key elements:

- **Services** (mainly Web services).

SOC (Service Oriented Computing)

It is a new computing paradigm that utilizes services as the basic constructs to support the development of rapid, low-cost and easy composition of distributed applications even in heterogeneous environments. [Papazoglou et al., 2006].

Key elements:

- **Services** (mainly Web services).
- **SOA** (Service Oriented Architecture)

SOC (Service Oriented Computing)

It is a new computing paradigm that utilizes services as the basic constructs to support the development of rapid, low-cost and easy composition of distributed applications even in heterogeneous environments. [Papazoglou et al., 2006].

Key elements:

- **Services** (mainly Web services).
- **SOA** (Service Oriented Architecture)
- **Service Composition:**

SOC (Service Oriented Computing)

It is a new computing paradigm that utilizes services as the basic constructs to support the development of rapid, low-cost and easy composition of distributed applications even in heterogeneous environments. [Papazoglou et al., 2006].

Key elements:

- **Services** (mainly Web services).
- **SOA** (Service Oriented Architecture)
- **Service Composition:**
 - ▶ **Service Orchestration**.

SOC (Service Oriented Computing)

It is a new computing paradigm that utilizes services as the basic constructs to support the development of rapid, low-cost and easy composition of distributed applications even in heterogeneous environments. [Papazoglou et al., 2006].

Key elements:

- **Services** (mainly Web services).
- **SOA** (Service Oriented Architecture)
- **Service Composition:**
 - ▶ **Service Orchestration**
 - ▶ **Service Choreography**

SOC (Service Oriented Computing)

It is a new computing paradigm that utilizes services as the basic constructs to support the development of rapid, low-cost and easy composition of distributed applications even in heterogeneous environments. [Papazoglou et al., 2006].

Key elements:

- **Services** (mainly Web services).
- **SOA** (Service Oriented Architecture)
- **Service Composition:**
 - ▶ **Service Orchestration**
 - ▶ **Service Choreography**
- **QoS** (Quality of Service).

Service Orchestration

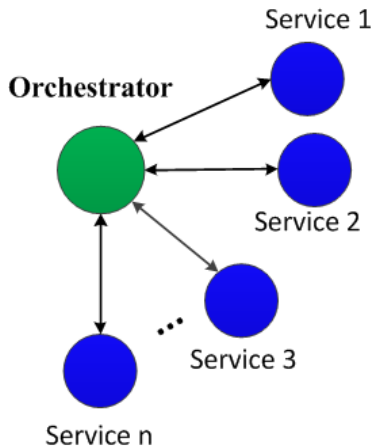


Figure: Service Orchestration

Service Choreography

- Allows service composition in a **collaborative** manner.
- Don't have a single point of control or coordination.
- Describes the **P2P interactions** of the externally **observable behavior of its participants**.

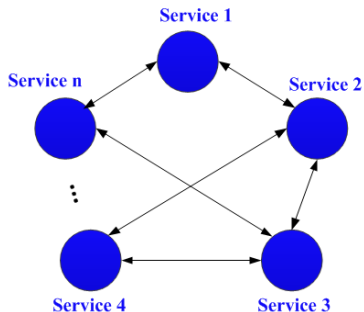


Figure: Service Choreography

Service Choreography

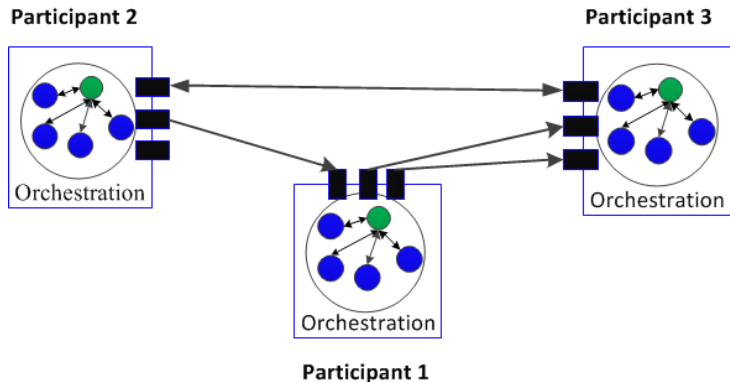


Figure: Service Choreography

Process Choreography

- A Choreography is also a process.

Process Choreography

- A Choreography is also a process.
- **BPMN** (Business Process Model and Notation) is a standard for modeling business process.
- **BPMN** supports choreographies modeling.

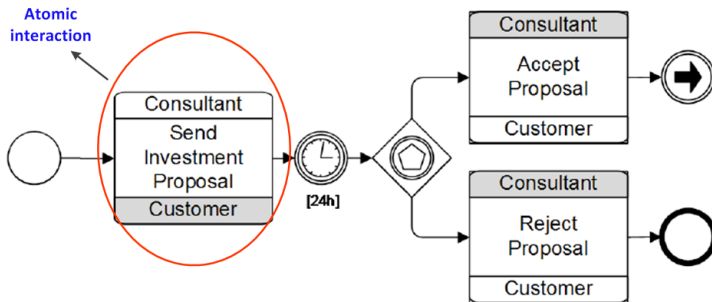
Process Choreography

- A Choreography is also a process.
- **BPMN** (Business Process Model and Notation) is a standard for modeling business process.
- **BPMN** supports choreographies modeling.
- Two modeling approaches:
 - ▶ Interconnection Model

- A Choreography is also a process.
- **BPMN** (Business Process Model and Notation) is a standard for modeling business process.
- **BPMN** supports choreographies modeling.
- Two modeling approaches:
 - ▶ Interconnection Model
 - ▶ Interaction Model

Interaction Model

- Interactions **globally captured**.
- Basic building block: **atomic interaction** between two parties.
- Supported from BPMN 2.



BPMN Choreography elements

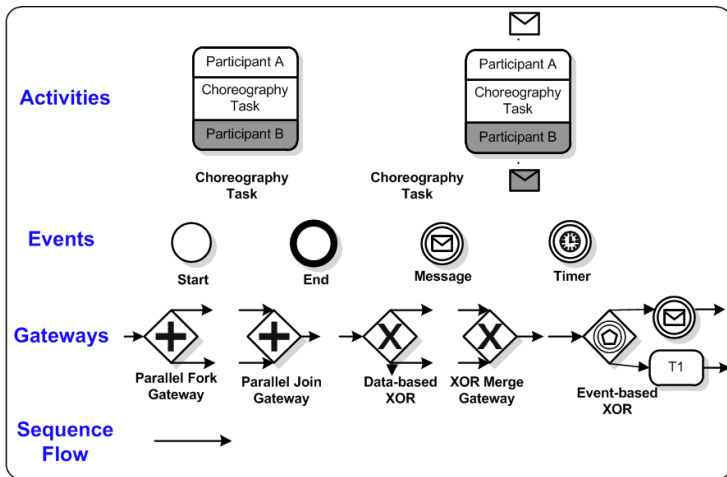


Figure: BPMN elements for modeling choreographies (BPMN 2.0).

- 1 Introduction
- 2 Problem
- 3 Methodology
- 4 Performance Evaluation
- 5 Conclusions and Future Works

- The most approaches don't evaluate choreographies:
 - ▶ focusing on **QoS** or
 - ▶ in earlier stages of development.

- The most approaches don't evaluate choreographies:
 - ▶ focusing on **QoS** or
 - ▶ in earlier stages of development.
- Planning of resources before/during development of choreography.

- The most approaches don't evaluate choreographies:
 - ▶ focusing on **QoS** or
 - ▶ in earlier stages of development.
- Planning of resources before/during development of choreography.
- To guarantee QoS about communications (network) is important.

Objectives

- To assess the **impact of QoS** attributes in a **choreography interaction model**.
- To propose a novel methodology to establish **requirements for QoS and SLA** in **early stages of development**.
- To plan the capacity of the network elements in choreographies.

- 1 Introduction
- 2 Problem
- 3 Methodology**
- 4 Performance Evaluation
- 5 Conclusions and Future Works

- 1 **Mapping** of a choreography to a Generalized Stochastic Petri Net (GSPN).

- 1 **Mapping** of a choreography to a Generalized Stochastic Petri Net (GSPN).
- 2 **Configurations** of the resulting GSPN.

- 1 **Mapping** of a choreography to a Generalized Stochastic Petri Net (GSPN).
- 2 **Configurations** of the resulting GSPN.
- 3 **Simulations** of scenarios.

- Defining the QoS attributes involved in **service**, **network** and **message** aspects.
- QoS attributes:

- Defining the QoS attributes involved in **service**, **network** and **message** aspects.
- QoS attributes:
 - ▶ In service operation: **time to complete the service**.

- Defining the QoS attributes involved in **service**, **network** and **message** aspects.
- QoS attributes:
 - ▶ **In service operation**: **time to complete the service**.
 - ▶ **In network**: delay and **communication errors**.

- Defining the QoS attributes involved in **service**, **network** and **message** aspects.
- QoS attributes:
 - ▶ In service operation : **time to complete the service**.
 - ▶ In network : delay and **communication errors**.
 - ▶ In message : **message format**.

Mapping BPMN to GSPN (I)

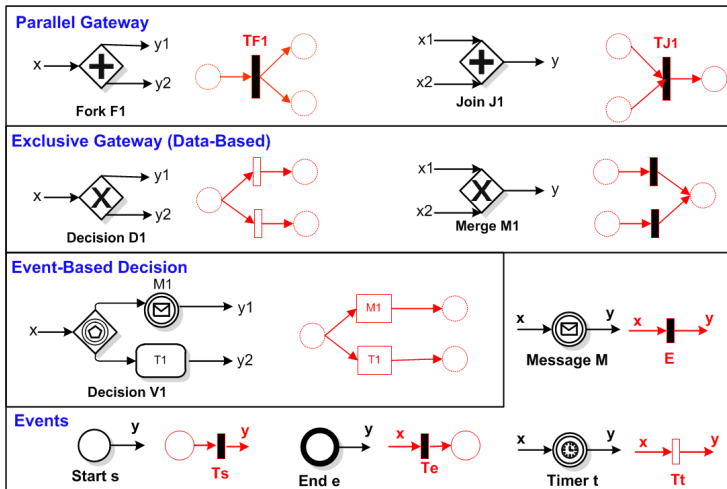
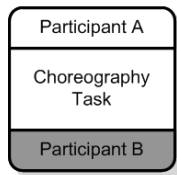


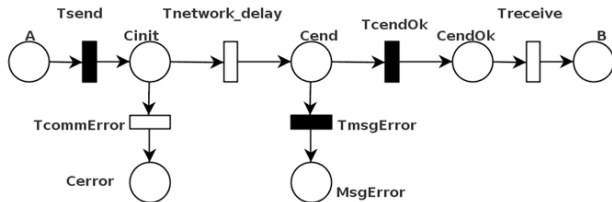
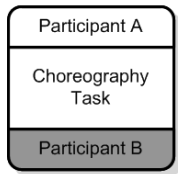
Figure: Mapping of events and gateways elements to modules of Petri nets

Mapping BPMN to GSPN (II)



A) Interaction in BPMN 2

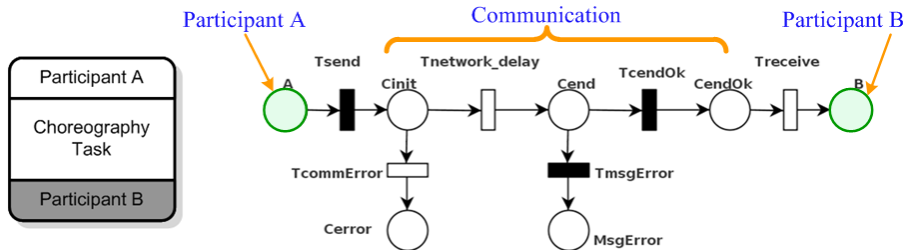
Mapping BPMN to GSPN (II)



A) Interaction in BPMN 2

B) GSPN Mapping with QoS

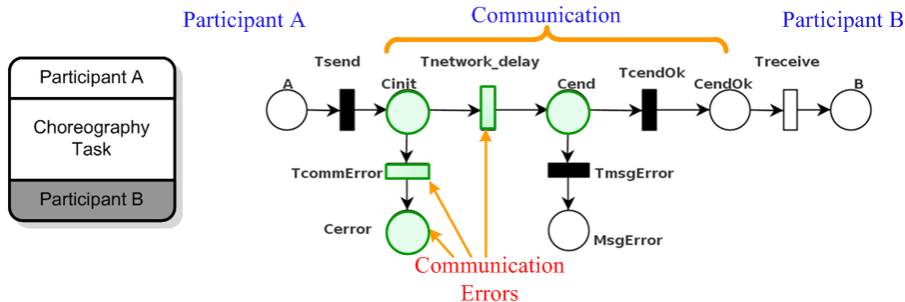
Mapping BPMN to GSPN (II)



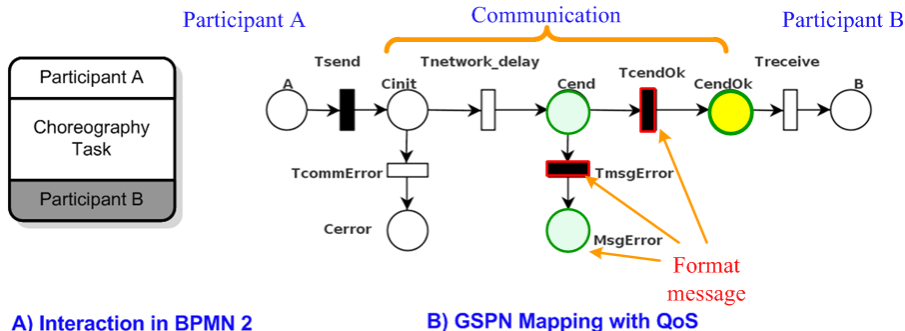
A) Interaction in BPMN 2

B) GSPN Mapping with QoS

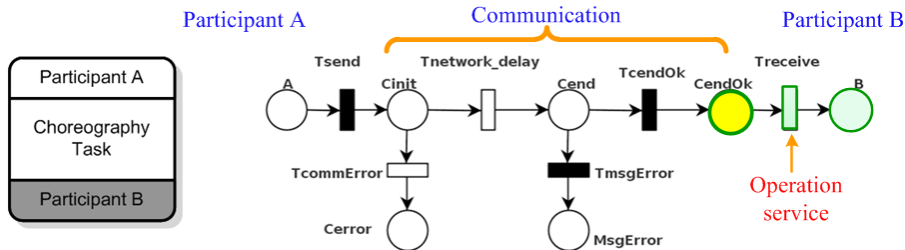
Mapping BPMN to GSPN (II)



Mapping BPMN to GSPN (II)



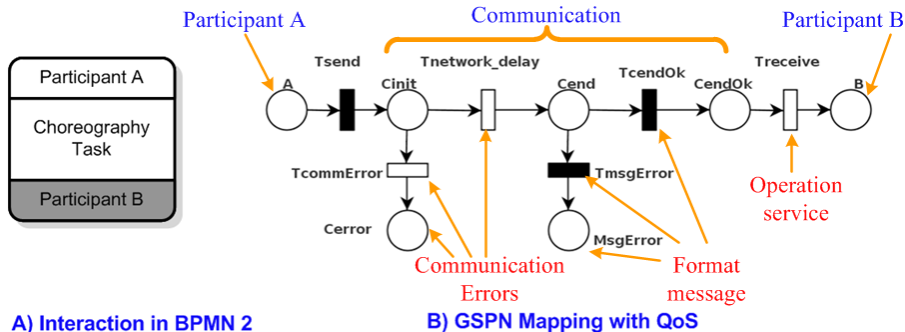
Mapping BPMN to GSPN (II)



A) Interaction in BPMN 2

B) GSPN Mapping with QoS

Mapping BPMN to GSPN (II)



Mapping of choreography in BPMN 2.0 to GSPN with QoS model

Input: **Process Choreography** $PC = (\mathcal{O}, \mathcal{A}, \mathcal{E}, \mathcal{G}, \mathcal{T}, \{e^S\}, \mathcal{E}^I, \{e^E\}, \mathcal{E}^{I_M}, \mathcal{E}^{I_T}, \mathcal{G}^F, \mathcal{G}^J, \mathcal{G}^X, \mathcal{G}^M, \mathcal{G}^V, \mathcal{F})$ in BPMN 2.0.

Mapping Algorithm

Mapping of choreography in BPMN 2.0 to GSPN with QoS model

Input: Process Choreography $PC = (\mathcal{O}, \mathcal{A}, \mathcal{E}, \mathcal{G}, \mathcal{T}, \{e^S\}, \mathcal{E}^I, \{e^E\}, \mathcal{E}^{I_M}, \mathcal{E}^{I_T}, \mathcal{G}^F, \mathcal{G}^J, \mathcal{G}^X, \mathcal{G}^M, \mathcal{G}^V, \mathcal{F})$ in BPMN 2.0.

Output: Generalized Stochastic Petri Net $GSPN_{QoS}$.

Mapping Algorithm

Mapping of choreography in BPMN 2.0 to GSPN with QoS model

Input: **Process Choreography** $PC = (\mathcal{O}, \mathcal{A}, \mathcal{E}, \mathcal{G}, \mathcal{T}, \{e^S\}, \mathcal{E}^I, \{e^E\}, \mathcal{E}^M, \mathcal{E}^T, \mathcal{G}^F, \mathcal{G}^J, \mathcal{G}^X, \mathcal{G}^M, \mathcal{G}^V, \mathcal{F})$ in BPMN 2.0.

Output: Generalized Stochastic Petri Net $GSPN_{QoS}$.

$CT_i \in \mathcal{T}$, $G_j \in \mathcal{G}$ and $E_k \in \mathcal{E}$. where $i, j, k \in \mathbb{N}$.

$PNQoS(CT_i)$, $PNQoS(G_j)$, $PNQoS(E_k)$ are functions return a GSPN according to mapping rules.

\oplus as the operator composition 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 beginning of the $GSPN_{QoS}$.

End

For $G_j \in \mathcal{G}$ **Do**

$GSPN_{QoS} \leftarrow GSPN_{QoS} \oplus PN(G_j)$

End

For $E_k \in \mathcal{E}$ **Do**

$GSPN_{QoS} \leftarrow GSPN_{QoS} \oplus PN(E_k)$

End







Add a starting Place and **immediate Transition** at the beginning of the $GSPN_{QoS}$.

Add a ending Place and **immediate Transition** at the end of the $GSPN_{QoS}$.

Return $GSPN_{QoS}$







Mapping Algorithm (II)

1) Mapping rules

BPMN Element		GSPN module
Choreography Activity		
Events		
Gateways		

Mapping Algorithm (II)

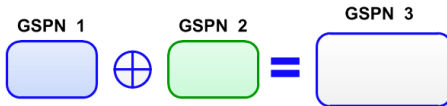
1) Mapping rules

BPMN Element		GSPN module
Choreography Activity		
Events		
Gateways		

2) Composition

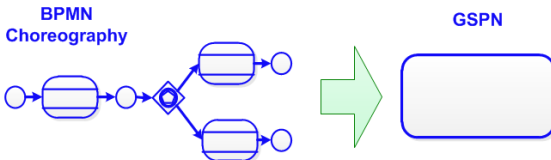


Composition Operator



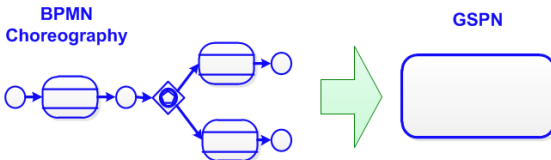
Mapping Algorithm (II)

3) Replacing and composing

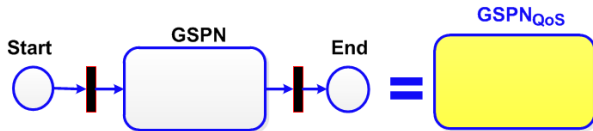


Mapping Algorithm (II)

3) Replacing and composing



4) Reducing and adding final elements



- 1 Introduction
- 2 Problem
- 3 Methodology
- 4 Performance Evaluation**
- 5 Conclusions and Future Works

Scenario

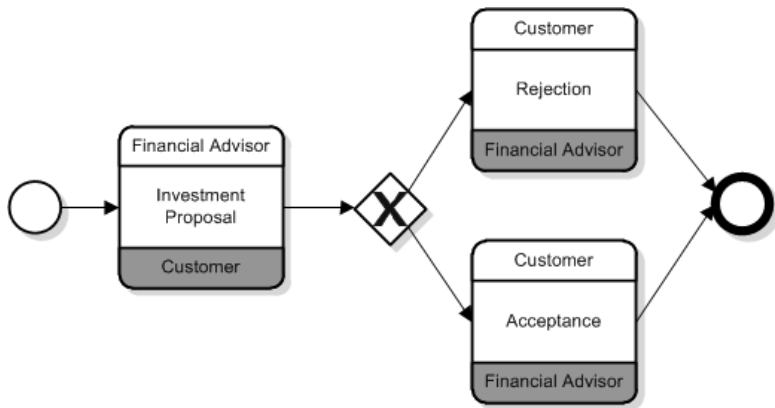


Figure: Choreography example using BPMN2 elements.

Scenario

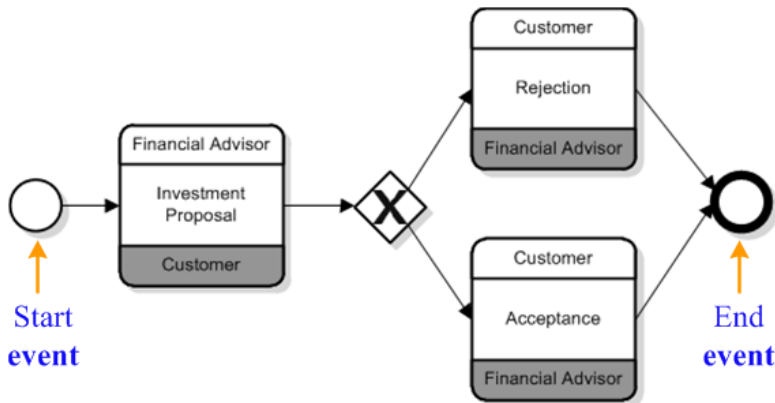


Figure: Choreography example using BPMN2 elements.

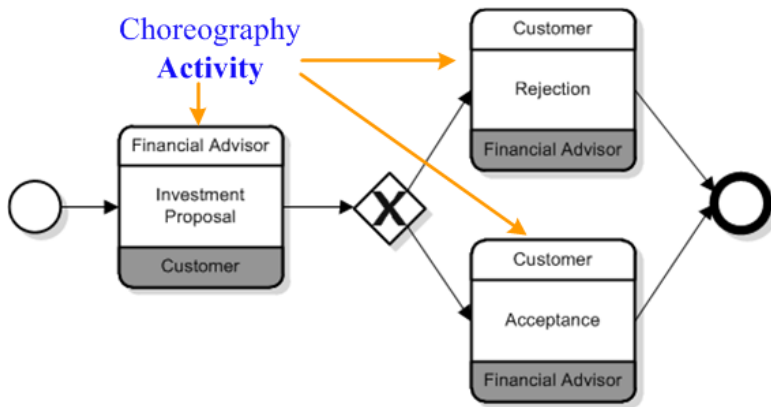


Figure: Choreography example using BPMN2 elements.

Scenario

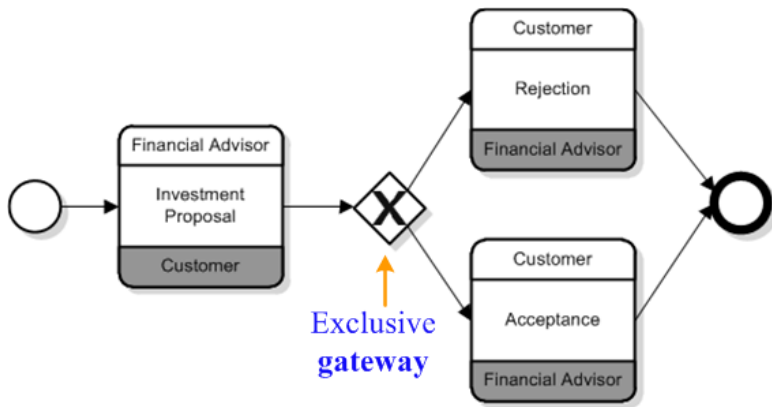


Figure: Choreography example using BPMN2 elements.

Scenario

Participant A: Financial Advisor

Participant B: Customer

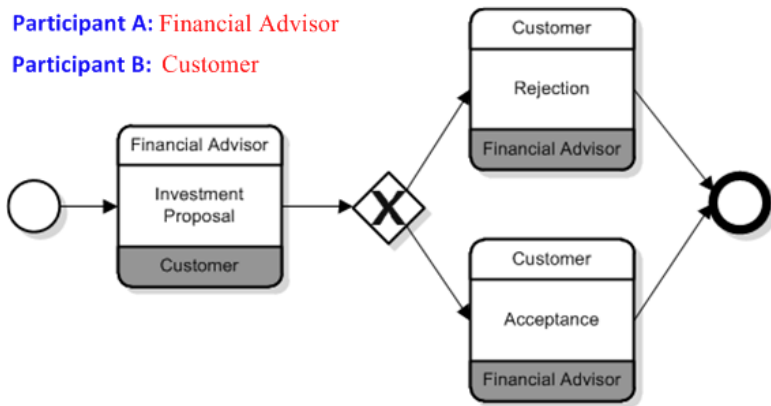


Figure: Choreography example using BPMN2 elements.

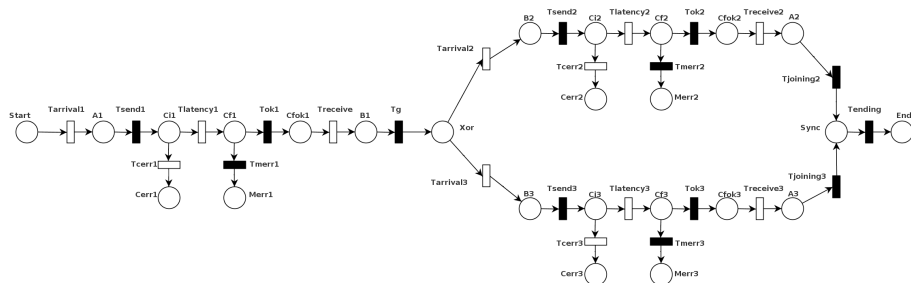
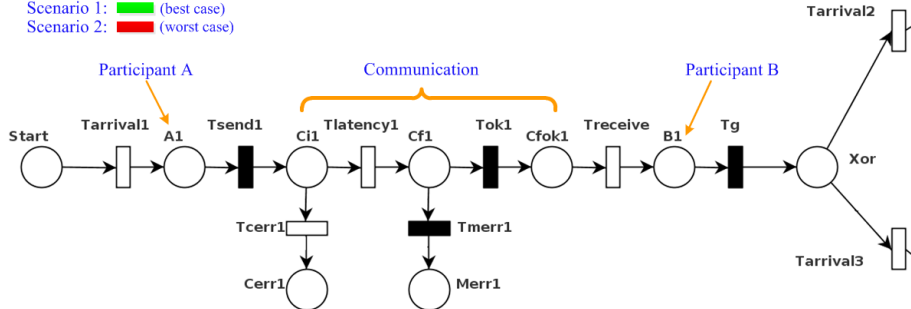


Figure: GSPN obtained from the choreography.

Configuration (I)

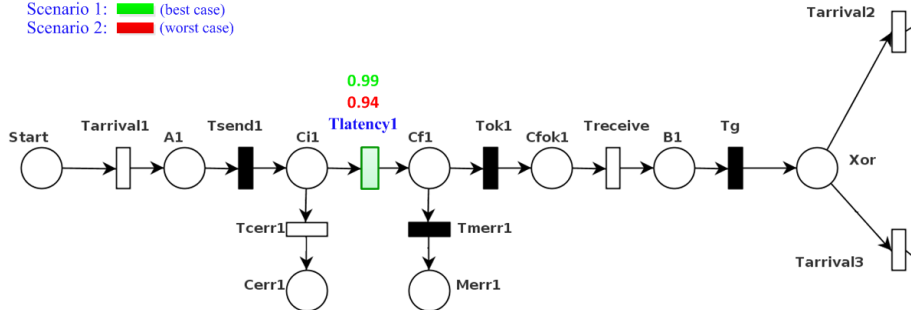
Scenario 1:  (best case)
Scenario 2:  (worst case)



Configuration (I)

Scenario 1:  (best case)

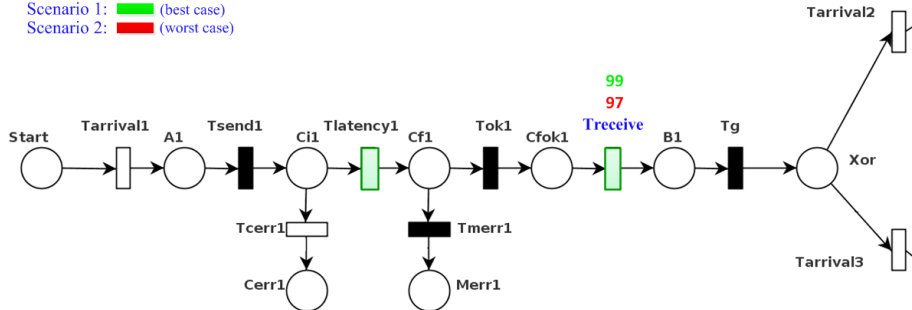
Scenario 2:  (worst case)



Configuration (I)

Scenario 1:  (best case)

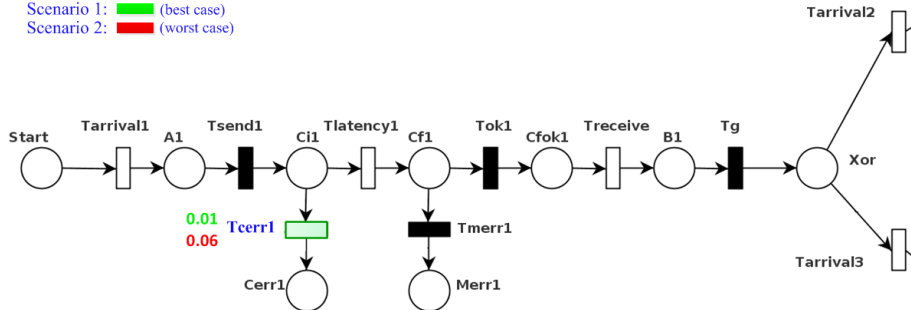
Scenario 2:  (worst case)



Configuration (I)

Scenario 1: █ (best case)

Scenario 2: █ (worst case)



Configuration (II)

Table: Weights of Scenario 1 and Scenario 2

Transition	Weights	
	Scenario 1	Scenario 2
$T_{latency1}, T_{latency2}, T_{latency3}$	0.99	0.94
$T_{cerr1}, T_{cerr2}, T_{cerr3}$	0.01	0.06
$T_{receive}, T_{receive2}, T_{receive3}$	99	97
$T_{merr1}, T_{merr2}, T_{merr3}$	1	3
$T_{arrival2}, T_{arrival3}$	0.5	0.5

- The **Pipe2** tool was used to **model** and **simulate** the **GSPN**.

- The **Pipe2** tool was used to **model** and **simulate** the **GSPN**.
- 1 **token** = 1 **choreography instance**.
- 100 **tokens** are considered to each scenario at the **place Start**.
- 100 **concurrent instances** were executed (**multiple-server semantic**).

- The **Pipe2** tool was used to **model** and **simulate** the **GSPN**.
- 1 **token** = 1 **choreography instance**.
- 100 **tokens** are considered to each scenario at the **place Start**.
- 100 **concurrent instances** were executed (**multiple-server semantic**).
- 1500 fires and 10 replications.
- Confidence level of 95%.

Table: Simulation results

Place	Average number of tokens (%)	
	Scenario 1	Scenario 2
<i>Start</i>	35.28	40.15
<i>End</i>	41.95	38.78
M_{err1}	0.39	0.91
M_{err2}	0.00	0.93
M_{err3}	0.00	0.66
C_{err1}	0.74	2.94
C_{err2}	0.00	0.00
C_{err3}	0.78	0.16
C_{i1}	8.32	8.90
C_{i2}	0.63	0.69
C_{i3}	0.75	8.90

Table: Simulation results

Place	Average number of tokens (%)	
	Scenario 1	Scenario 2
<i>Start</i>	35.28	40.15
<i>End</i>	41.95	38.78
M_{err1}	0.39	0.91
M_{err2}	0.00	0.93
M_{err3}	0.00	0.66
C_{err1}	0.74	2.94
C_{err2}	0.00	0.00
C_{err3}	0.78	0.16
C_{i1}	8.32	8.90
C_{i2}	0.63	0.69
C_{i3}	0.75	8.90

Message
Format Errors

Lost instances:

1.52%
3.10%

Table: Simulation results

		Average number of tokens (%)	
Place		Scenario 1	Scenario 2
<i>Start</i>		35.28	40.15
<i>End</i>		41.95	38.78
Message Format Errors	M_{err1}	0.39	0.91
	M_{err2}	0.00	0.93
	M_{err3}	0.00	0.66
Communication Errors	C_{err1}	0.74	2.94
	C_{err2}	0.00	0.00
	C_{err3}	0.78	0.16
	C_{i1}	8.32	8.90
	C_{i2}	0.63	0.69
	C_{i3}	0.75	8.90

Lost instances:
■ 0.39%
■ 2.50%

Table: Simulation results

		Average number of tokens (%)	
Place		Scenario 1	Scenario 2
<i>Start</i>		35.28	40.15
<i>End</i>		41.95	38.78
Message Format Errors	M_{err1}	0.39	0.91
	M_{err2}	0.00	0.93
	M_{err3}	0.00	0.66
Communication Errors	C_{err1}	0.74	2.94
	C_{err2}	0.00	0.00
	C_{err3}	0.78	0.16
Bottleneck	C_{i1}	8.32	8.90
	C_{i2}	0.63	0.69
	C_{i3}	0.75	8.90

- 1 Introduction
- 2 Problem
- 3 Methodology
- 4 Performance Evaluation
- 5 Conclusions and Future Works

Conclusions

- We have proposed a novel methodology to define QoS and SLA requirements in service Choreography.

Conclusions

- We have proposed a novel methodology to define QoS and SLA requirements in service Choreography.
- It's a initial approach using the “interaction model” (supported by BPMN 2.0).

Conclusions

- We have proposed a novel methodology to define QoS and SLA requirements in service Choreography.
- It's a initial approach using the “interaction model” (supported by BPMN 2.0).
- The GSPN is good to model and analyze several aspects involved into service choreography.

Conclusions

- We have proposed a novel methodology to define QoS and SLA requirements in service Choreography.
- It's a initial approach using the “interaction model” (supported by BPMN 2.0).
- The GSPN is good to model and analyze several aspects involved into service choreography.
- The simulation is useful for supporting analysis of complex processes (e.g. process choreography).

- We have proposed a novel methodology to define QoS and SLA requirements in service Choreography.
- It's a initial approach using the “interaction model” (supported by BPMN 2.0).
- The GSPN is good to model and analyze several aspects involved into service choreography.
- The simulation is useful for supporting analysis of complex processes (e.g. process choreography).
- The simulation results can be used to establish early QoS requirements and initial SLAs.

- To extend the mapping to support more choreography BPMN elements.
- To make more analysis and to use complex scenarios, where correlation problems could happen.
- To include more QoS attributes.



Baile Project
Enabling Scalable Cloud Service Choreographies

Thanks!