International Journal of Applied Mathematics and Computing Science. Volume 3, Issue 4, (2014) 70-84
http://www.cayley-nielson.com/ijamcs

Choreography modeling for BPMN based on Pi calculus

Lisa Marcoux
Mathematics Computer Science Department, Glendale Community College, Maricopa Community Colleges, 6000 West Olive Avenue, Glendale, Arizona 85302, USA
E-mail: *LIS2171314@maricopa.edu*

Abstract

Business process modeling is a key step in business process management activity; it plays a crucial role in process analysis and optimization. When modeling the complex business process interaction, the original BPMN modeling approach is not applicable, need special modeling language and methodology for process choreography? In order to solve the semantic ambiguity and integrity of business process modeling, proposed a formalize business process choreography modeling method based on π calculus. First using BPMN2.0 Choreography for business process choreography graphical instance design; Then formal modeling conducted for BPMN2.0 choreography's basic activities and structured activities based on π calculus; Instances of business process choreography auction scene model is given, simulate and deduce the model, which verifies the correctness of the model and the feasibility of modeling method. Results show that the formal modeling method based on π calculus makes choreography modeling more precise and specification, which ensures the consistency between the business process interoperability.

Keywords: choreography; BPMN2.0; π calculus; formal modeling.

1. Introduction

With the development of information technology and operational complexity rises, enterprise can't complete all the necessary business through their own operating, in the business process execution they need to connect with other enterprises, to complete the exchange of information through collaboration. This cross-organizational way brings the issue that prompt enterprise collaborative relationship dynamic, open and uncertainty. This means that it is difficult to construct and test them correctly, so before the deployment of such systems to ensure its correctness and reliability of the interaction behavior becomes very important. BPMN can provide support of business process graphical design for business personnel [1], it is a set of standards developed by BPMI (The Business Process Management Initiative), which is known as BPMN (Business Process Modeling Notation), and released BPMN1.0 specification in 2004. BPMN 1.0 enjoy great popularity with its simple and easy operation, but BPMN 1.0 only provides a business process modeling notation, without the definition of exchange format in order to make sure the BPMN tools of different manufacturers' user exchange model in a standard way, therefore, the model has incompatible problems; Computer implementation of business process described by BPMN 1.0, must solve problems such as unknown elements of the process execution semantics [2].

In order to solve these defects of BPMN1.0, BPMI incorporated into the Object Management Group (OMG), BPMN 2.0 standard is introduced by OMG in 2011[3], which is redefined as Business Process Model and Notation 2.0(BPMN 2.0). Compare with BPMN1. 0, the elements of the process execute semantics greatly enhanced in BPMN 2.0, it emphasizes the role of enterprise modeling and choreograph of cross-process owners [4].

Choreography is listed to participating organization, the whole business goals, the sequence and

ISSN 2333-2425
December, 2014

conditions of message exchange, the rules and limitations of the model participating organization comply with. Choreography as the convention of organizations can be used to verify if all the organizational behavior is correct or not. [5] use Petri net for formal modeling, but it often makes even a simple question can also produce the state explosion. [6] focus on orchestration of BPMN model for π calculus formalization description, modeling the single entity in control flow of business process, ignoring the modeling for information exchange between participants, and the research of relationships between different choreography process. Faced with these problems, [7] put forward that process choreography can be used to represent and management cross-organization process, which even can be used as a means of multilateral environment interaction. Therefore, process choreography most belongs to the category of interconnection model, namely, describing the observable behavior of the different partners, then associated through the message flow. This modeling method is usually lead to modeling error, cannot support the choreography of the basic design principles. [8] Propose interactions modeling scheme, define the model decomposition and constraints specification from global level. This paper based on interaction modeling, introducing BPMN 2.0 as new interaction modeling language, use π calculus for formal definition of choreography basic activities structured activities. Combined with concrete example, show a specific choreography modeling prototype based onπcalculus, and to verify the consistency of the proposed model, finally summarizes the formal execution semantics based on π calculus choreography modeling and the tool set of implementation method.

2. Related knowledge and work

2.1 π calculus

π calculus is proposed by Turing Award winner Robin Milner and his collaborators, which is a concurrent computing model used to describe the dynamic structure of intra-process and inter-process [9]. π calculus promotes the communication process calculus (CCS), allowing delivery channel name in communication. Therefore, there are two basic entity in π calculus: name (in lowercase letters a, b, c,...) and process (in capital letters P, Q, R,...).

We assume that a set of infinite name N, the elements of N represent in lowercase, x, y, z, etc. Action prefix for π calculus is π, which can trigger an action to perform, the grammar of π calculus is: $\pi ::= x(y)|\overline{x}\langle y\rangle|\tau$. $\overline{x}\langle y\rangle$ represents send name y from channel x, which can be abbreviated to $\overline{x}$; $x(y)$ indicates input name y through channel x, that can be abbreviated to x; τ only do process internal migration without message delivery.

2.2 BPMN 2.0

Using BPMN notation to describe business processes, can provide a simple mechanism in creating a business process model, at the same time to deal with the complexity of business processes, which is easier to understand for business personnel, the way to processing the two contradictory requirements is to mark the graphical classifications for a particular category, so that business people can easily identify the elements of the basic types, thereby to understand graphics. BPMN 2.0 has five types of element object: Flow Objects is a major graphic element defines the behavior of the business process. Data provide information about process or store information. Connection objects, connected stream Objects or other Objects. Swimlanes, to group the basic modeling elements. Artifacts, provide additional information to process, which will not affect the process of semantic.

(1) Flow objects can be divided into three categories: Events, Activities and Gateways.

Events, according to the influence on process at the different time, divided into start event,

intermediate catch event and end event. Event element symbols are shown in figure 1.

Figure 1. event elements

Activities, usually refers to the work enterprise executed, which can be atomic or compound, this paper only describes the tasks in activity. In figure 2, from left to right are send task, receive task, standard cycle task, multi-instance concurrent cycle task and multi-instance sequence cycle task.

Figure 2. activity elements

Gateway, which is used to control the branching and aggregation of sequence flow. Gateway is represented by diamond; the gateway type determines its behavior.

(2) Data has data object, data input, and data output and data store. Figure 3 shown data element respectively.

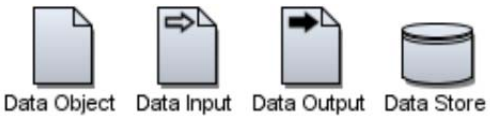

Figure 3. data elements

(3) Connecting Object has sequence flow and message flow, shown in figure 4.

Figure 4. connecting object elements

Sequence flow indicated by solid arrows, it derives from the start event, through the selection path, parallel path and activity, and stops at the end event. Message flow represent by hollow arrows, which refers to the sending and receiving of messages between the two participants. In BPMN specification, two pools represent independent participants respectively.

(4) Swimlanes used to organize the basic elements of the model, it has two objects: pool and lane, as shown in figure 5. Pool stands for participants in the process, and at the same time as the container of activities, which used to isolated activities between different pools. Lane is an area inside of pool, and it extended to the end of the pool. Lane used to organize and classify activities in the pool.

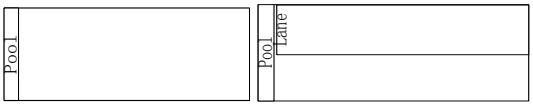

Figure 5. swimlanes elements

Attention: the difference between pool and lane is that connection of activities between different pools can only use message flow, while connection between different lanes can only use sequence flow.

(5) Artifacts have group and text annotations. Group is used to organize some activities together, which generate documentation and facilitate the analysis of process model, and will not affect the execution of the process sequence. Text annotations provide additional information of BPMN process model for reader, which increase the readability of the model diagram. As shown in figure 6.

Figure 6. artifacts elements

2.3 BPMN 2.0 Choreography

The object of choreography describes is not limited to a single pool, every step of it contains two or more participants. In the choreography, what we focus on is not participants' internal information interaction, but the communication between the participants. Choreography defines the order of participants' interaction in the business scene, figure 7 shows several core elements of choreography.

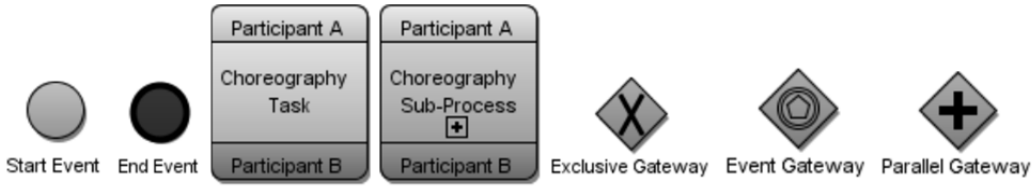

Figure 7. core elements of choreography

Start event, there is usually a cause (trigger), triggering event startup process, and affect the path of business process execution.

End event, also has a trigger condition that performs it makes the process ends.

Choreography task, the task of choreography has two participants, Participant A is the sender of choreography task, and Participant B is the receiver of choreography task.

Choreography sub - process, contains sub choreography of task, which can define multi-choreography task and can also contain more than one roles.

Exclusive gateway, represented as a diamond within X. It's created in the process or choreography, which judges each output stream on its conditional expression, mutually exclusive execute the activities of path, as shown in figure 8.

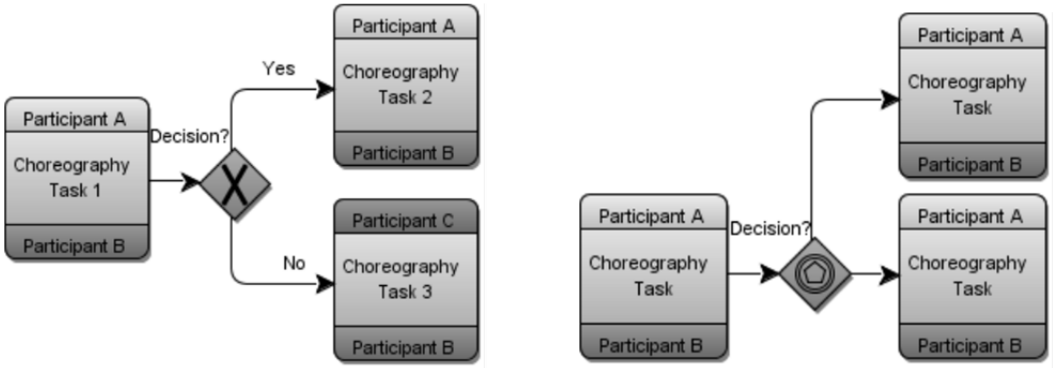

Figure 8. Exclusive gateway and Event Gateway

Event Gateway, in the process or choreography, the gateway based on event represent a branch point, output branch selection is based on the occurrence of some event when the process run to the gateway, rather than condition expression. As shown in figure 8.

Parallel Gateway, can create and implement multiple paths at the same time in process or choreography. Parallel gateway provides a mechanism of synchronous and production parallel flow, which mainly used for branching and merging. As shown in figure 9.

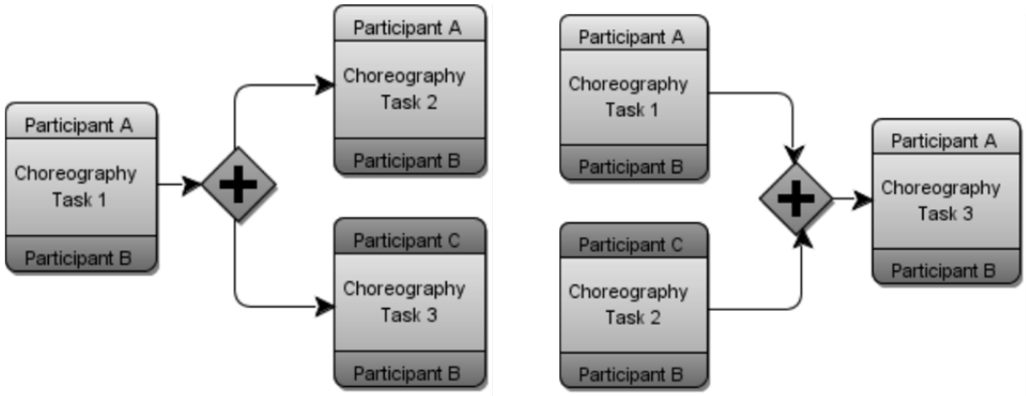

Figure 9. Parallel Gateway

3. Choreography modeling based on π calculus

Enterprise business process is composed of a series of activities. Activities of choreography can be divided into two kinds: basic activity and structured activity. Basic activity shows the interactions between the processes, whereas structured activity defines the execution method and sequence of basic activity. And we give some definitions of elements.

3.1 π calculus description of some element for BPMN 2.0

Not all elements of BPMN2.0 affect the execution of the process behavior, such as group and text annotations in artifacts, their function is to facilitate the people's understanding of the process model. Therefore, we ignore elements which have not directly affected on the external performance of the business process execution of BPMN2.0 specification, such as swimlanes, artifacts, etc., then extract a minimal elements set describing the behavior of the business process model. Elements of BPMN2.0 related to business process execution behavior have event, activity, gateway, sequence flow and message flow. Use these elements for π calculus describing of choreography model, can characterize the execution behavior of business processes completely. Gateway description of π calculus is detailed described in 3.2.

1) Event, indicate the message sending and receiving process, start and end events are represented below respectively. Channel y_{Start} sends a process start complete signal; τ_{End} is internal action of process, performs the internal action and the process will be end.

Start event: $[\![C_{Start}]\!]_{\bar{y}_{Start}} = \bar{y}_{Start}$

End event: $[\![C_{End}]\!] = \tau_{End}.0$

2) Activity, including atomic activity, compound activity and choreography activity, that we use name represent activity. For choreography activities, the first letter of Choreography and subscript is activity name. For example, choreography request is expressed as C_{Req}.

3) Sequence flow, it can limit the sequence of activities in model, usually we use the symbol ". " to limit, only when the activity front of the symbol ". " is completed, can continue to perform follow-up activities.

4) Message flow, usually interact with external processes, can use one end of the communication channel to represent:

m: indicates receiving message flow; $\overline{m}$: indicates sending message flow ($m \in N_m$).

3.2 BPMN 2.0 choreography basic activity modeling

Before the modeling for the semantic of choreography activities, we define a function: $[\![\ ]\!]_{\overline{y}}: A \to P$, which from a set of choreography activity [10] to the semantic expression of π calculus. $\overline{y}$ is a signal completed channel, when a name send from channel $\overline{y}$, it means that the corresponding activity has done executing. In general, suppose all choreography activities have their unique channel $\overline{y}$. When it is not sensitive to input and output data, we often omit parameters. For example，we write $\overline{y}\langle x \rangle$ as $\overline{y}$, write $y(x)$ as *y*. When the channel has no follow-up action, we add 0 behind the action, such as $y.0$. We promise that the capital letter C is the prefix of choreography action, the followings are modeling for basic activity of choreography.

Basic activity1. Start choreography task: $[\![C_{Start}]\!]_{\overline{y}} = \overline{y}$

Here，Start indicate the beginning of choreography, and it send out the completion signal through channel y.

Basic activity2. End choreography task: $[\![C_{End}]\!] = \tau.0$

Here, End indicate the end of choreography, when C_{End} receive the completion signal through y, it will end the process.

Basic activity 3. Unilateral interactive of basic interaction pattern:

$$[\![C_{Unilateral}(r_1, r_2, m)]\!]_{\overline{y}} = \tau_{r_1} \cdot \overline{m} | m \cdot \tau_{r_2} \cdot \overline{y_0} | y_0 \cdot \overline{y}$$

In the unilateral interaction pattern, there are two roles r_1 and r_2 involved in the choreography, m is the message channel that r_1 sends to r_2, y_0 is the information channel that the action of r_2 is completed, y is the signal channel that choreography activity is finished.

Therefore, the specific implementation of this choreography is: r_1 via internal action send message through channel m, when r_2 received this message, it will send out received completed signal through channel y_0 , at last, channel y send out the interaction finished signal.

Basic activity 4. Bilateral interactive of basic interaction pattern

$$[\![C_{Bilateral}(r_1, r_2, m_1, m_2)]\!]_{\overline{y}} = \tau_{r_1}.\overline{m_1}.m_2.\tau_{r_1}.\overline{y_0} | m_1.\tau_{r_2}.\overline{m_2} | y_0.\overline{y}$$

The same to unilateral interaction pattern, there are also two roles r_1 and r_2 involved in bilateral interactive, y is the activity completed signal channel. The different is that there are two message channel in bilateral interactive pattern, among which, m_1 is the message channel that r_1 sends to r_2, m_2 is the message channel that responses to r_1 when r_2 received news from channel m_1, y_0 is the channel that m_2 sends out news successfully.

The specific implementation of this choreography is: r_1 triggered by internal action and sends news

to r_2 through message channel m_1, at the same time, r_1 received the response from r_2 through channel m_2, r_1 through an internal action, sends out the execution completed signal by channel y_0; r_2 received the message from channel m_1, after the internal action, channel m_2 sends out r_2 finished signal , at last, channel y sends out the signal that bilateral interactive is completed.

3.3 BPMN 2.0 choreography structured activity modeling

Structured activity model defines the execution method and sequence of basic activities. BPMN structured activities model described by π calculus mainly include Sequence and Gateway. We promise that $\wedge$ and $\vee$ represent the condition true and false respectively. Followings are their definitions.

Structured activity 1. Sequence: Sequential execute its internal choreography activity.

$$[\![C_{Seq}(C_1,C_2,...,C_n)]\!]_{\bar{y}} = (vy_1y_2...y_n)\left([\![C_1]\!]_{\bar{y}_1}\,|\,y_1.[\![C_2]\!]_{\bar{y}_2}\,|\ldots|\,y_{n-1}.[\![C_n]\!]_{\bar{y}_n}\,|\,y_n.\bar{y}\right)$$

$C_1,C_2,...,C_n$ is choreography activities of Sequence, they can be the basic activities or structured activities. $y_1,y_2,...,y_n$ are completion signal channel of $C_1,C_2,...,C_n$ respectively, y is the completion signal transmission channel for sequential executed choreography activities.

The specific implementation process is: after y_1 received the completion signal of C_1, C_2 starts and when it sends out the completion signal, the subsequent activity will be started. By this analogy, when the last activity C_n sends out the completion signal, start channel y to send out the completion signal of Sequence.

Structured activity 2. Exclusive Gateway: Choose the single task to execute choreography activities which meet the conditions.

$$[\![C_{EG}(b,C_1,C_2)]\!]_{\bar{y}} = (vy_1y_2)\left((b=\wedge)[\![C_1]\!]_{\bar{y}_1}\,|(b=\vee)[\![C_2]\!]_{\bar{y}_2}\,|(y_1+y_2).\bar{y}\ \right)$$

C_1,C_2 are choreography activities included in the Exclusive Gateway. They can be basic choreography activity or structured activity. y_1,y_2 are completed signal channel of C_1,C_2 respectively, b is condition, y is the completed signal transmission channel of exclusive gateway.

The specific implementation process is: C_1 is started when condition b is $\wedge$, C_2 is started when condition b is $\vee$. Finally, when the completion signal of C_1 or C_2 is sent out, start channel y to send out the signal that exclusive gateway is completed.

Structured activity 3. Parallel Gateway: Execute choreography activities concurrently.

$$[\![C_{PG}(C_1,C_2)]\!]_{\bar{y}} = (vy_1y_2)(\ [\![C_1]\!]_{\bar{y}_1}\,|\,[\![C_2]\!]_{\bar{y}_2}\,|\,(y_1.y_2+y_2.y_1).\bar{y}\ )$$

The definition of C_1,C_2,y_1,y_2 has the same meaning with Structured activity 2. C_1 and C_2 can execute concurrently, when both of them are completed, they will send out the completion signal. Finally, send out the completion signal of parallel gateway through y.

Structured activity 4. Event Gateway: choose the task to perform based on event.

We introduce the event channel e as sources channel of expected event b in gateway. Event b can be Boolean, Integer, Character, String and other types of messages, which according to the concrete instance. According event that gateway decide whether to execute or implement which subsequent process. Therefore, we have the following five types of event gateway. Among them, C_1,C_2 and C are choreography activities of event gateway, they can be basic activities or structured activities. e is event channel, b is specific event, y is the signal channel that activities of event gateway are finished.

Type 1 . $[\![C_{EG_1}(e,C)]\!]_{\bar{y}} = e.[\![C]\!].\bar{y}$

When any event coming from channel e, it will trigger the execution of event gateway, at last send out the completion signal of event gateway through y.

Type 2 . $[\![C_{EG_2}(e,C)]\!]_{\bar{y}} = (vb)\big(e(b).[b=\wedge][\![C]\!].\bar{y}\big)$

When event b is $\wedge$ coming from channel e, start C, then send out the completion signal of event gateway by y.

Type 3 . $[\![C_{EG_3}(e,C)]\!]_{\bar{y}} = (vb)\big(e(b).[b=\vee][\![C]\!].\bar{y}\big)$

When event b is $\vee$ coming from channel e, start C, then send out the completion signal of event gateway by y.

Type 4 . $[\![C_{EG_4}(e,C_1,C_2)]\!]_{\bar{y}} = (vb,y_1,y_2)(\ e(b).\big([b=\wedge][\![C_1]\!]_{\bar{y}_1} \big| [b=\vee][\![C_2]\!]_{\bar{y}_2}\big)\big|(y_1+y_2).\bar{y}\ )$

When channel e received event b, judged it. Start C_1 if b is $\wedge$, start C_2 if b is $\vee$. Finally, when the completion signal of C_1 or C_2 is sent out, then send out the completion signal of event gateway by y.

Type 5 .

$$[\![C_{EG_5}(e_1,e_2,C_1,C_2)]\!]_{\bar{y}} = (vb_1,b_2,y_1,y_2)(\ e_1(b_1).\big([b_1=\wedge][\![C_1]\!]_{\bar{y}_1} \big| e_2(b_2).[b_2=\vee][\![C_2]\!]_{\bar{y}_2}\big)\big|(y_1+y_2).\bar{y})$$

In this type, e_1 , e_2 are event channel, b_1 , b_2 are specific events, y_1 , y_2 are private channel of C_1, C_2 respectively. When b_1 is $\wedge$ and received by e_1, then perform C_1; When b_2 is $\vee$ and received by e_2, then perform C_2. Finally, when the completion signal of C_1 or C_2 is sent out, then send out the completion signal of event gateway by y.

Structured activity 5. Choreography Loop Type:

According to the execution characteristics of enterprise's business process, some business need loop execution, according to the specific instances, we introduce the event trigger condition b to start the cycle process of choreography. We promise that b is the trigger condition, we start the process when b is true, y is the completion signal transmission channel. We have the following three types of cycle choreography model.

Mode 1. Standard Loop：

$$[\![C_{SL}(b,C)]\!]_{\bar{y}} = !\big([b=\wedge][\![C]\!]_{\bar{y}}\big) \quad \text{or} \quad [\![C_{SL}(C)]\!]_{\bar{y}} = ![\![C]\!]_{\bar{y}}$$

In standard loop, start the process when b is $\wedge$, then execute C by sequence. When y sends out the completion signal, end the loop.

Mode 2. MultiInstance Sequential Loop：

$$[\![C_{MSL}(b,n,C)]\!]_{\bar{y}} = (vy_1y_2...y_n)\big([b=\wedge][\![C_1]\!]_{\bar{y}_1}\big|y_1.[\![C_2]\!]_{\bar{y}_2}\big|...\big|y_{n-1}.[\![C_n]\!]_{\bar{y}_n}\big|y_n.\bar{y}\big)$$

In multiInstance sequential loop, n is the number of choreography activities. Start the process when b is $\wedge$, process instance has been stimulated to run successively. Only the former process completed, can start the next process. We send out the completion signal by y and end the cycle until all the process finished successively.

Mode 3. MultiInstance Parallel Loop：

$$[\![C_{MPL}(b,n,C)]\!]_{\bar{y}} = (vy_1y_2...y_n)\big([b=\wedge][\![C_1]\!]_{\bar{y}_1}\big|[\![C_2]\!]_{\bar{y}_2}\big|...\big|[\![C_n]\!]_{\bar{y}_n}\big|y_1.y_2.....y_n.\bar{y}\big)$$

In MultiInstance Parallel Loop, n is the number of choreography activities. Start the process when b

is $\wedge$, then all process instances will be concurrent execution. We send out the completion signal by y and end the cycle until all the process finished.

So far, by the method of π calculus，We give the fully definitions of elements, basic activities and structured activities of BPMN2.0. Provide standardized criteria for formal modeling method of choreography.

4. Instance of choreography modeling

4.1 Collaboration diagram of BPMN2.0

The following gives a business process modeling instance of auction scene. In this scenario often involves three roles: seller, auction service and bidders. The basic commercial transaction is that seller sells products to bidders, in order to determine the buyer; the highest bidder in the auction is winner. The purpose of the introduction of auction service is to attract more potential buyers, which is more efficient processing the auction, or deal with the auction service of seller outsourcing.

Specific interaction process is: the seller send auction creation request to the auction service, when receiving the request, it sends a creation confirmation to seller. Once the auction begins, bidders can bid that auction service will be confirmed in turn. At a particular point in time the auction ended, the auction service will send the notification to seller and bidders that which one gives the highest price and the corresponding amount. Finally, products payment and shipment executed at the same time. Seller sends payment details, such as to provide the bank accounts of bid winner, and confirm the payment. In product shipment, the seller will send the notification to the bid winner as soon as the product sent out successfully, at last the bid winner received the goods. The BPMN collaboration diagram of this process is shown in figure 10.

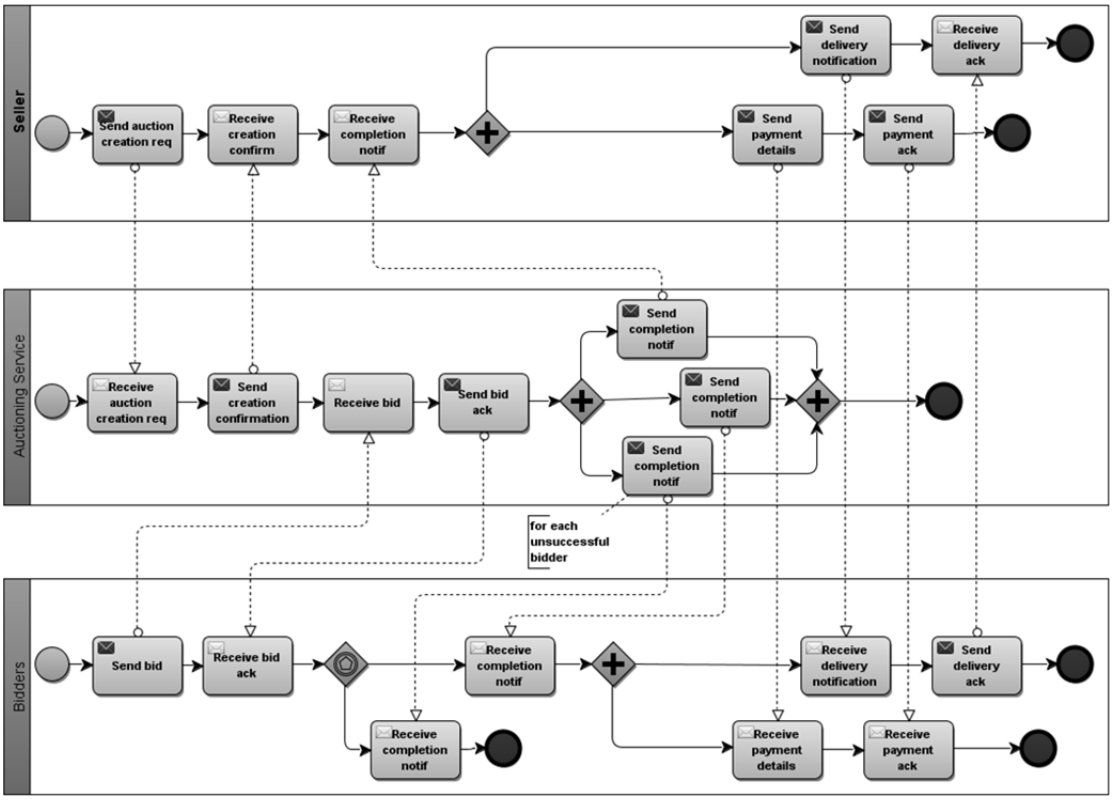

Figure 10. BPMN collaboration diagram of auction scene

4.2 Choreography modeling of auction scene

1) Choreography is the complex interaction of interdependent behavior between entities, which demand multi-cooperation, and at least two roles participate in choreography. For the business process in figure 10, logic abstract about it and get the choreography diagram of auction scene in figure 11. We define some tasks before modeling the choreography.

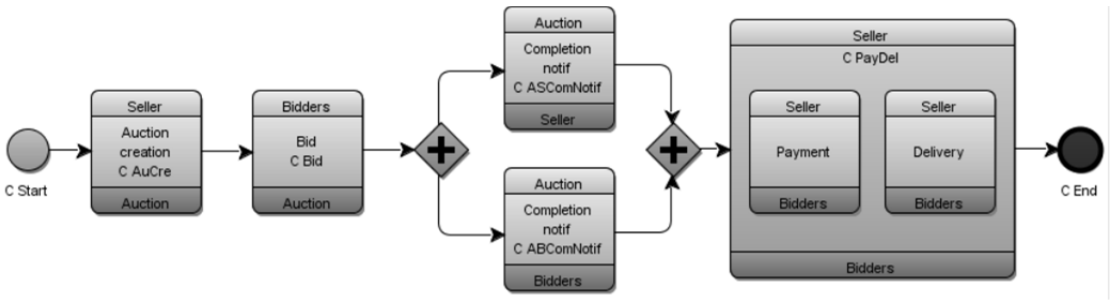

Figure 11. BPMN choreography flow chart of auction scene

In this choreography modeling of auction scene, involves the following eight choreography tasks: choreography begins C_{Start}, auction create C_{AuCre},bid C_{Bid}, the auction completion notification between Auction and Seller $C_{ASComNotif}$, the auction completion notification between Auction and Bidders $C_{ABComNotif}$, parallel gateway C_{PG}, payment and product delivery C_{PayDel}, choreography ends C_{End}.

Among which, the completion signal channel of internal process for choreography task C_{AuCre}, C_{Bid}, $C_{ASComNotif}$, $m_{ABComNotif}$, C_{PayDel} is y_{AuCre}, y_{Bid}, $y_{ASComNotif}$, $y_{ABComNotif}$, y_{PayDel} respectively; The message channel of internal process is m_{AuCre}, m_{Bid}, $m_{ASComNotif}$, $m_{ABComNotif}$, m_{PayDel} respectively; The completion signal channel of choreography task is y_{AuCreS}, y_{BidS}, $y_{ASComNotifS}$, $y_{ABComNotifS}$, $y_{PayDelS}$ respectively; The completion signal channel of C_{Start}, C_{PG}, C_{Seq}, is y_{Start}, y_{PGS}, y_{SeqS} respectively.

2) Followings are models of choreography tasks：

Model 1. Choreography start task C_{Start} gives a trigger condition to start the process, then the process sends out start finished signal through channel y_{Start}.

$$[\![C_{Start}]\!]_{\overline{y}_{Start}} = \overline{y}_{Start}$$

Model 2.The auction creation choreography task between Seller and Auction:

$$[\![C_{AuCre}(Seller, Auction, m_{AuCre})]\!]_{\overline{y}_{AuCreS}} = \tau_{Seller}.\overline{m}_{AuCre} \,|\, m_{AuCre}.\tau_{Auction}.\overline{y}_{AuCre} \,|\, y_{AuCre}.\overline{y}_{AuCreS}$$

Seller prepared data by internal action τ_{Seller}, and send out auction creation information by channel m_{AuCre}. When Auction received it, execute internal action $\tau_{Auction}$, and send out auction creation finished

signal through channel y_{AuCre}. Finally, send out auction creation completed signal of choreography task by channel y_{AuCreS}.

Model 3. The bid choreography task between Bidders and Auction:

$$[\![C_{Bid}(Bidders, Auction, m_{Bid})]\!]_{\overline{y}_{BidS}} = \tau_{Bidders}.\overline{m}_{Bid}\,|\,m_{Bid}.\tau_{Auction}.\overline{y}_{Bid}\,|\,y_{Bid}.\overline{y}_{BidS}$$

Bidders prepared data by internal action $\tau_{Bidders}$, and send out bid information by channel m_{Bid}. When Auction received it, executes internal action $\tau_{Auction}$, and sends out bid finished signal through channel y_{Bid}. Finally, sends out bid completed signal of choreography task by channel y_{BidS}.

Model 4. The auction completed notification choreography task between Auction and Seller：

$$[\![C_{ASComNotif}(Auction, Seller, m_{ASComNotif})]\!]_{\overline{y}_{ASComNotifS}} = \tau_{Auction}.\overline{m}_{ASComNotif}\,|\,m_{ASComNotif}.\tau_{Seller}.\overline{y}_{ASComNotif}\,|\,y_{ASComNotif}.y_{ASComNotifS}$$

Auction prepared data by internal action $\tau_{Auction}$, and sends out auction completed notification by channel $m_{ASComNotif}$. When Seller received it, executes internal action τ_{Seller}, and sends out the success signal that auction completed notification through channel $y_{ASComNotif}$. Finally, sends out auction completed notification of choreography task by channel $y_{ASComNotifS}$.

Model 5. The auction completed notification choreography task between Auction and Bidders：

$$[\![C_{ABComNotif}(Auction, Bidders, m_{ABComNotif})]\!]_{\overline{y}_{ABComNotifS}} = \tau_{Auction}.\overline{m}_{ABComNotif}\,|\,m_{ABComNotif}.\tau_{Bidders}.\overline{y}_{ABComNotif}\,|\,y_{ABComNotif}.y_{ABComNotifS}$$

Auction prepared data by internal action $\tau_{Auction}$, and sends out auction completed notification by channel $m_{ABComNotif}$. When Bidders received it, executes internal action $\tau_{Bidders}$, and sends out the success signal that auction completed notification through channel $y_{ABComNotif}$. Finally, sends out auction completed notification of choreography task by channel $y_{ABComNotifS}$.

Model 6. The parallel gateway of $C_{ASComNotif}$ and $C_{ABComNotif}$ is:

$$[\![C_{PG}(C_{ASComNotif}, C_{ABComNotif})]\!]_{\overline{y}_{PGS}} = (vy_{ASComNotifS}.y_{ABComNotifS})(\ [\![C_{ASComNotif}]\!]_{\overline{y}_{ASComNotifS}}\ |\ [\![C_{ABComNotif}]\!]_{\overline{y}_{ABComNotifS}}$$

$$|\ (y_{ASComNotifS}.y_{ABComNotifS} + y_{ABComNotifS}.y_{ASComNotifS}).\overline{y}_{PGS}\ )$$

$C_{ASComNotif}$ and $C_{ABComNotif}$ are executed concurrently, the following three conditions may occur: the former finished first, the latter finished first or both of them finished at the same time. Only when both execution is completed, can send out parallel gateway of choreography completed signal by channel y_{PGS}.

Model 7. The payment and delivery choreography task between Seller and Bidders:

$$[\![C_{PayDel}(Seller, Bidders, m_{PayDel})]\!]_{\overline{y}_{PayDelS}} = \tau_{Seller}.\overline{m}_{PayDel}\,|\,m_{PayDel}.\tau_{Bidders}.\overline{y}_{PayDel}\,|\,y_{PayDel}.y_{PayDelS}$$

Seller prepared data by internal action τ_{Seller}, and sends out payment and delivery information by channel m_{PayDel}. When Bidders received it, execute internal action $\tau_{Bidders}$, and send out payment and

delivery success signal through channel y_{PayDel}. Finally, send out payment and delivery completed signal of choreography task by channel $y_{PayDelS}$.

Model 8. The auction scene was executed sequentially:

$$\left[\!\left[C_{Seq}\left(C_{AuCre},C_{Bid},C_{PG},C_{PayDel}\right)\right]\!\right]_{\overline{y}_{SeqS}}=\left(vy_{AuCreS}y_{BidS}y_{PGS}y_{PayDelS}\right)\left(\ \left[\!\left[C_{AuCre}\right]\!\right]_{\overline{y}_{AuCreS}}\mid\ y_{AuCreS}.\left[\!\left[C_{Bid}\right]\!\right]_{\overline{y}_{BidS}}\mid\ y_{BidS}.\left[\!\left[C_{PG}\right]\!\right]_{\overline{y}_{PGS}}\right.$$

$$\left.\mid\ y_{PGS}.\left[\!\left[C_{PayDel}\right]\!\right]_{\overline{y}_{PayDelS}}\left|y_{PayDelS}.\overline{y}_{SeqS}\right.\ \right)$$

Perform tasks of choreography according to the sequence of auction tasks created. Auction creation task C_{AuCre} executes first, when successor task receives the completion signal of it, then start current task. In this way, execute sequentially, end the sequence tasks until send out the completion signal of choreography task C_{Seq}.

Model 9. Choreography task C_{End} through an internal action τ_{End} ended the process.

$$\left[\!\left[C_{End}\right]\!\right]=\tau_{End}.0$$

Model 10. The integrity formalized definition of auction scene is:

$$Auction\overset{def}{=}\left(v\tilde{c}\right)\left[\!\left[C_{Start}\right]\!\right]_{\overline{y}_{Start}}\mid\ y_{Start}.\left[\!\left[C_{Seq}\left(C_{AuCre},C_{Bid},C_{PG},C_{PayDel}\right)\right]\!\right]_{\overline{y}_{SeqS}}\mid\ y_{SeqS}.\left[\!\left[C_{End}\right]\!\right]$$

This auction scene is composed of C_{Start}, C_{Seq} and C_{End}. $\tilde{c}$ is the set of choreography tasks completed channel in the model. The implementation of auction scene is that C_{Start} starts the process. When the sequence activities received the completed signal of C_{Start}, execute C_{AuCre} and C_{Bid} sequentially. Finnish the bidding completion notification to Sellers and Bidders through a parallel gateway task C_{PG}, then execute the product payment and transport task C_{PayDel}. Finally, ended the auction process with the choreography task C_{End}.

3.3 Validation of choreography model

Choreography modeling of business process based on the premise of its correctness. In order to ensure the accuracy, integrity and consistency of the choreography modeling semantics, we deduce and verify the choreography model we have built. Following are analysis for every step of process which to validate whether each step is executed according to the design.

$$Auction\overset{def}{=}\left(v\tilde{c}\right)\left[\!\left[C_{Start}\right]\!\right]_{\overline{y}_{Start}}\mid\ y_{Start}.\left[\!\left[C_{Seq}\left(C_{AuCre},C_{Bid},C_{PG},C_{PayDel}\right)\right]\!\right]_{\overline{y}_{SeqS}}\mid\ y_{SeqS}.\left[\!\left[C_{End}\right]\!\right]$$

Choreography process starts, expand task C_{Start}:

$$\rightarrow\left(v\tilde{c}\right)\overline{y}_{Start}\mid\ y_{Start}.\left[\!\left[C_{Seq}\left(C_{AuCre},C_{Bid},C_{PG},C_{PayDel}\right)\right]\!\right]_{\overline{y}_{SeqS}}\mid\ y_{SeqS}.\left[\!\left[C_{End}\right]\!\right]$$

Choreography start task C_{Start} send out start completed signal through channel y_{Start}. When C_{Seq} received this signal, expand and execute C_{Seq} sequentially. Now we expand and execute the interaction tasks of C_{AuCre}:

$$\xrightarrow{\overline{y}_{Start}}(v\tilde{c})\tau_{Seller}.\overline{m}_{AuCre}\left|m_{AuCre}.\tau_{Auction}.\overline{y}_{AuCre}\right|y_{AuCre}.\overline{y}_{AuCreS}\mid\ y_{AuCreS}.[\![C_{Bid}]\!]_{\overline{y}_{BidS}}\mid\ ...[\![C_{End}]\!]$$

Seller of C_{AuCre} prepares data through an internal action τ_{Seller}, and sends out auction creation news through channel m_{AuCre}. Auction receives this news:

$$\xrightarrow{\tau_{Seller}.\overline{m}_{AuCre}}(v\tilde{c})\tau_{Auction}.\overline{y}_{AuCre}\left|y_{AuCre}.\overline{y}_{AuCreS}\right|\ y_{AuCreS}.[\![C_{Bid}]\!]_{\overline{y}_{BidS}}\mid\ ...[\![C_{End}]\!]$$

Auction of C_{AuCre} prepares data through an internal action $\tau_{Auction}$, and sends out auction creation news delivery succeeded signal through private channel y_{AuCre}:

$$\xrightarrow{\tau_{Auction}.\overline{y}_{AuCre}}(v\tilde{c})\overline{y}_{AuCreS}\mid\ y_{AuCreS}.[\![C_{Bid}]\!]_{\overline{y}_{BidS}}\mid\ ...[\![C_{End}]\!]$$

C_{AuCre} sends out completion signal of auction creation choreography task through channel y_{AuCreS}, and expands task C_{Bid}:

$$\xrightarrow{\overline{y}_{AuCreS}}(v\tilde{c})\tau_{Bidders}.\overline{m}_{Bid}\left|m_{Bid}.\tau_{Auction}.\overline{y}_{Bid}\right|y_{Bid}.\overline{y}_{BidS}\mid\ y_{BidS}.[\![C_{PG}]\!]_{\overline{y}_{PGS}}\mid\ ...[\![C_{End}]\!]$$

Bidders of C_{Bid} prepares data through internal action $\tau_{Bidders}$, and send out bid news through channel m_{Bid}, and Auction receives this news:

$$\xrightarrow{\tau_{Bidders}.\overline{m}_{Bid}}(v\tilde{c})\tau_{Auction}.\overline{y}_{Bid}\left|y_{Bid}.\overline{y}_{BidS}\right|\ y_{BidS}.[\![C_{PG}]\!]_{\overline{y}_{PGS}}\mid\ ...[\![C_{End}]\!]$$

Auction of C_{Bid} prepares data through internal action $\tau_{Auction}$, and sends out bid news delivery successful signal through private channel y_{Bid}:

$$\xrightarrow{\tau_{Auction}.\overline{y}_{Bid}}(v\tilde{c})\overline{y}_{BidS}\mid\ y_{BidS}.[\![C_{PG}]\!]_{\overline{y}_{PGS}}\mid\ ...[\![C_{End}]\!]$$

C_{Bid} sends out completion signal of bid choreography task through channel y_{BidS}, and expands task C_{PG}. C_{PG} is parallel gateway, so $C_{ASComNotif}$ and $C_{ABComNotif}$ are executed in parallel. When both of them are finished, can execute successor task C_{PayDel}. Now, we verify the process of $C_{ASComNotif}$:

$$\xrightarrow{\overline{y}_{BidS}}(v\tilde{c})\tau_{Auction}.\overline{m}_{ASComNotif}\left|m_{ASComNotif}.\tau_{Seller}.\overline{y}_{ASComNotif}\right|y_{ASComNotif}.\overline{y}_{ASComNotifS}$$
$$\mid\ [\![C_{ABComNotif}]\!]_{\overline{y}_{ABComNotifS}}\mid\ (\ y_{ASComNotifS}.y_{ABComNotifS}+y_{ABComNotifS}.y_{ASComNotifS}\ ).\overline{y}_{PGS}\mid\ ...[\![C_{End}]\!]$$

Auction of $C_{ASComNotif}$ prepares data through internal action $\tau_{Auction}$, and send out completion notification of auction through channel $m_{ASComNotif}$, and Seller receives the completion notification:

$$\xrightarrow{\tau_{Auction}.\overline{m}_{ASComNotif}}(v\tilde{c})\tau_{Seller}.\overline{y}_{ASComNotif}\left|y_{ASComNotif}.\overline{y}_{ASComNotifS}\right|\ [\![C_{ABComNotif}]\!]_{\overline{y}_{ABComNotifS}}\mid\ (\ y_{ASComNotifS}.$$
$$y_{ABComNotifS}+y_{ABComNotifS}.y_{ASComNotifS}\ ).\overline{y}_{PGS}\left|...[\![C_{End}]\!]\right.$$

Seller of $C_{ASComNotif}$ prepares data through internal action τ_{Seller}, and send out completion notification of auction delivery successful signal through private channel $y_{ASComNotif}$:

$$\xrightarrow{\tau_{Seller}.\overline{y}_{ASComNotif}}(v\tilde{c})\overline{y}_{ASComNotifS}\left|[\![C_{ABComNotif}]\!]_{\overline{y}_{ABComNotifS}}\right|\ (y_{ASComNotifS}.y_{ABComNotifS}+y_{ABComNotifS}.y_{ASComNotifS}\ ).\overline{y}_{PGS}\mid\ ...[\![C_{End}]\!]$$

Like the $C_{ASComNotif}$, when $C_{ABComNotif}$ finished execution:

$$\xrightarrow{\tau_{Bidders},\overline{y}_{ABComNotif}}(v\tilde{c})\,\overline{y}_{ASComNotifS}\mid\ \overline{y}_{ABComNotifS}\mid\ \left(y_{ASComNotifS}.y_{ABComNotifS}+y_{ABComNotifS}.y_{ASComNotifS}\right).\overline{y}_{PGS}\mid\ ...[\![C_{End}]\!]$$

When both of $C_{ASComNotif}$ and $C_{ASComNotif}$ are finished execution, they send out completion signal of choreography through channel $\overline{y}_{ASComNotifS}$ and $\overline{y}_{ABComNotifS}$ respectively.

$$\xrightarrow{\overline{y}_{ASComNotifS},\overline{y}_{ABComNotifS}}(v\tilde{c})\,\overline{y}_{PGS}\mid y_{PGS}.[\![C_{PayDel}]\!]_{\overline{y}_{PayDelS}}\mid\ ...[\![C_{End}]\!]$$

Parallel gateway C_{PG} sends out completion signal of choreography through channel $\overline{y}_{PGS}$, and expands C_{PayDel}:

$$\xrightarrow{\overline{y}_{PGS}}(v\tilde{c})\,\tau_{Seller}.\overline{m}_{PayDel}\mid m_{PayDel}.\tau_{Bidders}.\overline{y}_{PayDel}\mid y_{PayDel}.\overline{y}_{PayDelS}\mid y_{PayDelS}.\overline{y}_{SeqS}\mid y_{SeqS}.[\![C_{End}]\!]$$

Seller of C_{PayDel} prepares data through internal action τ_{Seller}, then sends out product payment and delivery information through channel m_{PayDel}, and Bidders receive this news:

$$\xrightarrow{\tau_{Seller},\overline{m}_{PayDel}}(v\tilde{c})\,\tau_{Bidders}.\overline{y}_{PayDel}\mid y_{PayDel}.\overline{y}_{PayDelS}\mid y_{PayDelS}.\overline{y}_{SeqS}\mid y_{SeqS}.[\![C_{End}]\!]$$

Bidders of C_{PayDel} prepares data through internal action $\tau_{Bidders}$, then send out completion signal of product payment and delivery through private channel y_{PayDel}:

$$\xrightarrow{\tau_{Bidders},\overline{y}_{PayDel}}(v\tilde{c})\,\overline{y}_{PayDelS}\mid y_{PayDelS}.\overline{y}_{SeqS}\mid y_{SeqS}.[\![C_{End}]\!]$$

C_{PayDel} sends out completion signal of choreography through channel $\overline{y}_{PayDelS}$, and expands C_{End}:

$$\xrightarrow{\overline{y}_{PayDelS}}(v\tilde{c})\,\overline{y}_{SeqS}\mid y_{SeqS}.\tau_{End}.0$$

C_{Seq} sends out completion signal of execute choreography task sequentially through channel $\overline{y}_{SeqS}$, and C_{End} receives this news:

$$\xrightarrow{\overline{y}_{SeqS}}(v\tilde{c})\,\tau_{End}.0$$

C_{End} ends this process through an internal action:

$$\xrightarrow{\tau_{End}}(v\tilde{c})\,0$$

At present, we have finished the correctness verification of the choreography model for auction scene, among which doesn't have deadlocks. Through the validation process, we find that after the execution of choreography modeling, we get 0 at the last step that means each step of choreography model is completed. It is different with general collaboration diagram model. In collaboration diagram model, after the

execution interaction of each node is completed, it will return to the initial state of the node. Therefore, when verify the collaboration diagram model, the last verification step will return to the initial state of the model.

4. Conclusion and outlooks

This paper uses BPMN 2.0 combines with π calculus, proposed a formalism method which focused on choreography basic activity and structured activity of enterprise business process. Choreography makes interaction between participants of business process more standardized, which reflects the powerful expressiveness of π calculation through choreography modeling. We proposed explicit formalized definition of interaction modeling process of choreography, and eliminate conflict of business process interoperability. Use specific examples, in view of the choreography modeling for enterprise business, we give formal verification to the model and eliminate the possibility of deadlocks in interaction model. The concern of formal model shift from business process participants internal orchestration execution to choreography execution between participants. In the further study, on the one hand, we will further study the application of π calculus in BPMN2.0 choreography modeling; On the other hand, we will explore new verification method to simulate and detect the model.

References

[1] Ouyang, Chun, et al. "Formal semantics and analysis of control flow in WS-BPEL." Science of Computer Programming 67.2 (2007): 162-198.

[2] Davenport, Thomas H., and J. Short. "Information technology and business process redesign." Operations management: critical perspectives on business and management 1 (2003): 97.

[3] Wohed, Petia, et al. "Pattern-based Analysis of BPMN-An extensive evaluation of the Control-flow, the Data and the Resource Perspectives." (2005).

[4] El-Bakry, Hazem M., and Nikos Mastorakis. "Business process modeling languages for information system development." Proceedings of the 9th WSEAS international conference on Applied informatics and communications. World Scientific and Engineering Academy and Society (WSEAS), 2009.

[5] Koskela, Mika, and Jyrki Haajanen. "Business Process Modeling and Execution." Tools and technologies report for SOAMeS project (2007).

[6] Model, Business Process. "Notation (BPMN) Version 2.0." Object Management Group specification (2011).

[7] Barros, Alistair, Gero Decker, and Marlon Dumas. "Multi-staged and multi-viewpoint service choreography modelling." Proceedings of the Workshop on Software Engineering Methods for Service Oriented Architecture (SEMSOA), Hannover, Germany. CEUR Workshop Proceedings. Vol. 244. 2007.

[8] Decker, Gero, and Mathias Weske. "Interaction-centric modeling of process choreographies." Information Systems 36.2 (2011): 292-312.

[9] Milner, Robin. Communicating and mobile systems: the pi calculus. Cambridge university press, 1999.

[10] Leymann, Frank, Dimka Karastoyanova, and Michael P. Papazoglou. "Business process management standards." Handbook on Business Process Management 1. Springer Berlin Heidelberg, 2015. 595-624.