Choreography modelling based on π calculus

Jiulei Jiang, Jiao Zhang*, Wenxing Bao

College of Computer Science and Engineering, Beifang University of Nationalities, 750021, Yinchuan, China

Received 18 August 2014, www.cmnt.lv

Abstract

Business process modelling is a key step in business process management, and it plays a crucial role in process analysis and optimization. When modelling complex business process interaction, the original BPMN modelling approach is not applicable, need special modelling language and methodology for process choreography. In order to solve semantic ambiguity and integrity of business process modelling, proposed a formalize business process choreography modelling method based on π calculus. First business process choreography graphical instance design is presented by BPMN2.0 choreography. Then formal definition based on π calculus of BPMN2.0 choreography's basic activities and structured activities is presented. Then an instance of business process choreography auction scene model is given, and the model is analysed and validated manually and automatically respectively. Finally, we propose a π calculus model validation algorithm with the model XML document of Visio BPMN2.0 Modeller tool as input and then implement the algorithm. Theoretical analysis and experimental results show that this method can describe BPMN choreography by π calculus and be able to validate the semantic error of BPMN in terms of translate semantics and automatic deductive of π calculus, which makes choreography modelling more precise and specification.

Keywords: choreography, BPMN2.0, π calculus, formal modelling

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 business process execution they need to connect with other enterprises to complete the exchange of information through collaboration [1]. This cross-organizational way brings the issue that prompt enterprise collaborative relationship dynamic, open and uncertainty. This means it is difficult to construct and test them correctly, so before deployment to ensure its correctness and reliability of the interaction behaviour is vital [2]. BPMN (Business Process Modelling Notation) can provide support of business process graphical design for business personnel [3].

OMG (Object Management Group) introduced BPMN2.0 standard in 2011 [4]. Compare with BPMN1.0, it emphasizes the role of enterprise modelling and choreograph of cross-process owners [5]. BPMN is not a formalized language and can't conduct formal verification by itself. Therefore, how to describe BPMN2.0 as a formal language, and identify BPMN semantic error by its inference ability is of great scientific research value.

There are lots of formal modelling methods and mainly about UML activity diagram [6], BPEL [7]. The mainly user of UML is system architect and worker, their business analysis ability is not strong as BPMN. BPMN is standard of graphical modelling [8]. There are BPMN modelling methods based on Petri net [9]. Petri net tends to produce state explosion when BPMN model is very large. At present, research on BPMN2.0 modelling and validation based on π calculus is less. Existing research mainly focus on orchestration modelling design of BPMN model by π calculus. [7] puts forward a variation π calculus of BPEL's basic and structured activities formal description, but there is no analysis for model validation, it cannot guarantee the correctness of the model.

To solve above problems, we propose a BPMN2.0 choreography modelling method based on π calculus, the following is main contributions:

1) Introduce BPMN2.0 as new interaction modelling language, use π calculus to conduct formal definition of basic and structured activities of choreography.
2) Combined with concrete example, we give out a specific model prototype based on π calculus, and simulate the model by manual deduction. Then use MWB to analyze and verify the model. And the result is same as manual simulation exactly.
3) MWB do not provide any programming interfaces. We have to type π calculus process command and check location of deadlock and circulation structure in π calculus process command line manually. According to relationship between message flow of BPMN2.0 and channel name of π calculus we can find error location.

2 Related knowledge and work

2.1 π CALCULUS

π calculus is a concurrent computing model used to describe the dynamic structure of intra-process and inter-process. π calculus allows delivery channel name in communication, which makes it has the ability to create new channel. It is especially suitable to describe distributed loosely coupled concurrent systems.

* ***Corresponding author's*** e-mail:zhangjiao923@163.com

We assume that a set of infinite name N, the elements of N represent in lowercase, x, y, z, etc. [10]. Action prefix for π calculus is π, which can trigger an action, the grammar of π calculus is: $\pi ::= x(y)|\bar{x}\langle y\rangle|\tau$. $\bar{x}\langle y\rangle$ means send name y from channel x, which can be abbreviated to $\bar{x}$; $x(y)$ indicates input name y through channel x, that can be abbreviated to x; τ only do process internal migration without message delivery.

2.2 BPMN2.0

Using BPMN notation to describe business process, can provide a simple mechanism in creating a business process model, meanwhile to deal with complex business process. BPMN2.0 has following five element objects:

1) Flow objects, is a major graphic elements define the behaviour of the business process. It can be divided into three categories: Events, Activities and Gateways.

Events, according to the influence at different time can divided into start event, intermediate catch event and end event, as shown in Figure 1.

FIGURE 1 Event elements

Activities, can be atomic or compound. It has send, receive, standard cycle, multi-instance concurrent cycle and multi-instance sequence cycle task shown in Figure 2.

FIGURE 2 Activity elements

Gateway, which is used to control the branching and aggregation of sequence flow.

2) Data, provide information about process or store information. has data object, data input, data output and data store. Figure 3 shown data element respectively.

FIGURE 3 Data elements

3) Connecting Object, has sequence flow and message flow, as shown in Figure 4.

FIGURE 4 Connecting object elements

4) Swimlanes used to organize the basic elements of the model, it has two objects: pool and lane, as shown in Figure 5.

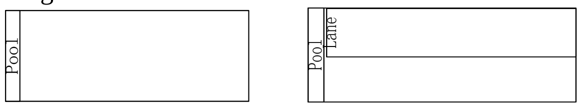

FIGURE 5 Swimlanes elements

5) Artefacts, provide additional information to process, which will not affect the semantic of process. It has group and text annotations, as shown in Figure 6.

FIGURE 6 Artefacts elements

2.3 BPMN2.0 CHOREOGRAPHY

The object of choreography is not limited to a single pool, each step contains two or more participants. We focus on communication among participants. Choreography [11] defines order of interaction in business scene, Figure 7 shows several core elements of choreography.

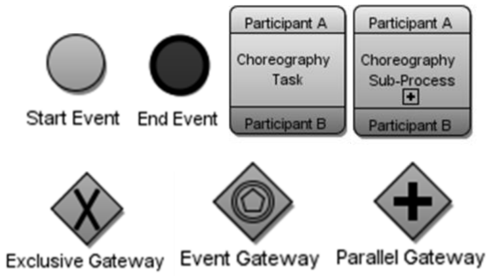

FIGURE 7 Core elements of choreography

- Start event, trigger event startup process.
- End event, is a process end trigger condition.
- Choreography task, has two participants, Participant A is sender and B is receiver.
- Choreography sub – process, contains sub choreography, which can define multi-choreography task and can also contain more than one roles.

Exclusive gateway, judges each output stream on its conditional expression, mutually exclusive execute the activities of path, as shown in Figure 8.

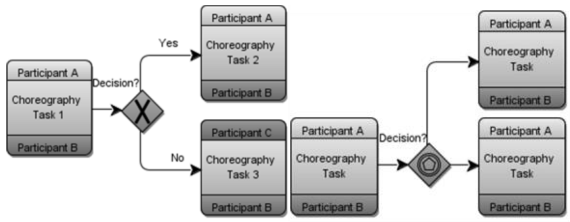

FIGURE 8 Exclusive gateway and Event Gateway

Event Gateway, based on event represent a branch point, as shown in Figure 8.

Parallel Gateway, provides a mechanism of synchronous and production parallel flow, which mainly used for branching and merging, as shown in Figure 9.

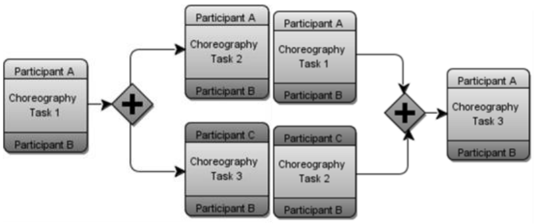

FIGURE 9 Parallel Gateway

Choreography is an appointment between organizations, which can be used to verify if all the organizational behaviour is correct [12]. Enterprise business process is composed of a series of activities. Activities of choreography can be divided into two kinds: basic and structure activities [13]. Basic activities show the interactions between the processes, whereas structured activity defines the execution method and sequence of basic activity. We give some definitions of elements.

3.1 π CALCULUS DESCRIPTION OF SOME ELEMENT BPMN2.0

Elements of BPMN2.0 related to business process execution behaviour have event, activity, gateway, sequence flow and message flow. Use these elements for π calculus to choreography model, can characterize the execution behaviour of business processes completely.

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.

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

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

2) Activity, for choreography activity, we use letter C with a subscript to represent activity name. For example, choreography request is presented as C_{Req}.
3) Sequence flow, it can limit the sequence of activities in model, usually we use the symbol "." to limit the sequence execution from front to back.
4) Message flow, we use m to indicate receiving message; $\bar{m}$ to indicate sending message ($m \in N_m$).

3.2 BPMN2.0 CHOREOGRAPHY BASIC ACTIVITY MODELLING

Before modelling for semantic of choreography activities, we define a function: $[\![\ ]\!]_{\bar{y}}$:$A \rightarrow P$, which is a mapping from a set of choreography activity to the semantic expression of π calculus. $\bar{y}$ is a signal completed channel. In general, suppose all choreography activities have their unique channel $\bar{y}$. When it is not sensitive to input and output data, we often omit parameters. For example, we write $\bar{y}\langle x\rangle$ as $\bar{y}$, write $y(x)$ as y. When the channel has no follow-up action, we add 0 behind the action, such as y.0.

Basic activity 1. Start choreography task:

$$C_{Start}\ _{\bar{y}} = \bar{y}\cdot$$

It is the beginning of choreography, and it send out the completion signal through channel y.

Basic activity 2. End choreography task:

$$C_{End}\ = \tau.0\,.$$

When C_{End} receives the completion signal through y, it will end the process.

Basic activity 3. Unilateral interactive of basic interaction pattern:

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

There are two roles r_1 and r_2, m is message channel that r_1 sends to r_2, y_0 is channel that action of r_2 is completed, y is end signal.

Basic activity 4. Bilateral interactive of basic interaction pattern:

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

There are also two roles r_1 and r_2, y is activity completed signal channel. There are two message channels: m_1 is message channel that r_1 sends to r_2, m_2 represents responses to r_1 when r_2 received news from m_1, y_0 is channel m_2 sends out news successfully.

3.3 BPMN2.0 CHOREOGRAPHY STRUCTURED ACTIVITY MODELLING

Structured activity model defines the execution method and sequence of basic activities. We promise $\wedge$ and $\vee$ represent the condition true and false respectively.

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

$$[\![C_{Seq}(C_1,C_2,...,C_n)]\!]_{\bar{y}} = (\nu y_1 y_2 ... y_n)\big(\ C_1\ _{\bar{y}_1}|y_1.\ C_2\ _{\bar{y}_2}$$
$$|\ldots|y_{n-1}.\ C_n\ _{\bar{y}_n}\big|y_n.\bar{y}\big),$$

$C_1, C_2, ..., C_n$ are 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 channel for sequential executed choreography activities.

Structured activity 2. Exclusive Gateway: choose the single task to execute choreography activities.

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

C_1, C_2 are choreography, they can be basic 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.

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

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

The definition of C_1, C_2, y_1, y_2 has the same meaning with Parallel Gateway C_1 and C_2 execute concurrently, when both of them are completed, send out 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 [14]. Event b can be Boolean, Integer, Character, String and other types of messages. According event that gateway decide whether to execute or implement which subsequent process. Therefore, we have following five types of event gateway.

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: $[\![C_{EG_2}(e,C)]\!]_{\bar{y}} = (\nu b)\left(e(b).[b=\wedge]\ C\ .\bar{y}\right)$

When event b is $\wedge$, start C and send out the completion signal of event gateway by y.

Type 3: $[\![C_{EG_3}(e,C)]\!]_{\bar{y}} = (\nu b)\left(e(b).[b=\vee]\ C\ .\bar{y}\right)$

When event b is $\vee$, start C and send out the completion signal of event gateway by y.

Type 4:

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

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:

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

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: we promise that b is the trigger condition, start process when b is true, y is the completion signal channel. We have following three types.

Mode 1. Standard Loop:

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

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

Mode 2. MultiInstance Sequential Loop:

$$[\![C_{MSL}(b,n,C)]\!]_{\bar{y}} = (\nu y_1 y_2 \ldots y_n)\left([b=\wedge]\ C_1 {}_{\bar{y}_1} \middle| y_1.\ C_2 {}_{\bar{y}_2} \middle| \ldots \middle| y_{n-1}.\ C_n {}_{\bar{y}_n} \middle| y_n.\bar{y}\right),$$

n is the number of choreography activities. Start process when b is $\wedge$. Only when former process completed, can start the next process. We send out completion signal by y and end the cycle until all the process finished successsively.

Mode 3. MultiInstance Parallel Loop:

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

MultiInstance Parallel Loop different from Mode 2 is that all process instances will be concurrent execution.

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

4 Instance of choreography modelling

4.1 COLLABORATION DIAGRAM OF BPMN2.0

Following gives a business process modelling instance of auction scene. Collaboration diagram always represent internal interaction [15]. This scenario often involves three roles: seller, auction service and bidders. The basic commercial transaction is seller sells products to bidders, the highest price bidder in the auction is winner. The purpose 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 seller sends out auction creation request to auction service, when receives request, it sends a creation confirmation to seller. Once auction begins, bidders can bid that auction service will be confirmed in turn. At a particular time the auction ended, the auction service sends notification to seller and bidders which one gives the highest price. Products payment and shipment execute at the same time. Seller sends payment details, such as provides bank accounts and confirm payment. In product shipment, seller will send notification to bid winner as soon as product sent out successfully, bid winner receives goods. This process is shown in Figure 10.

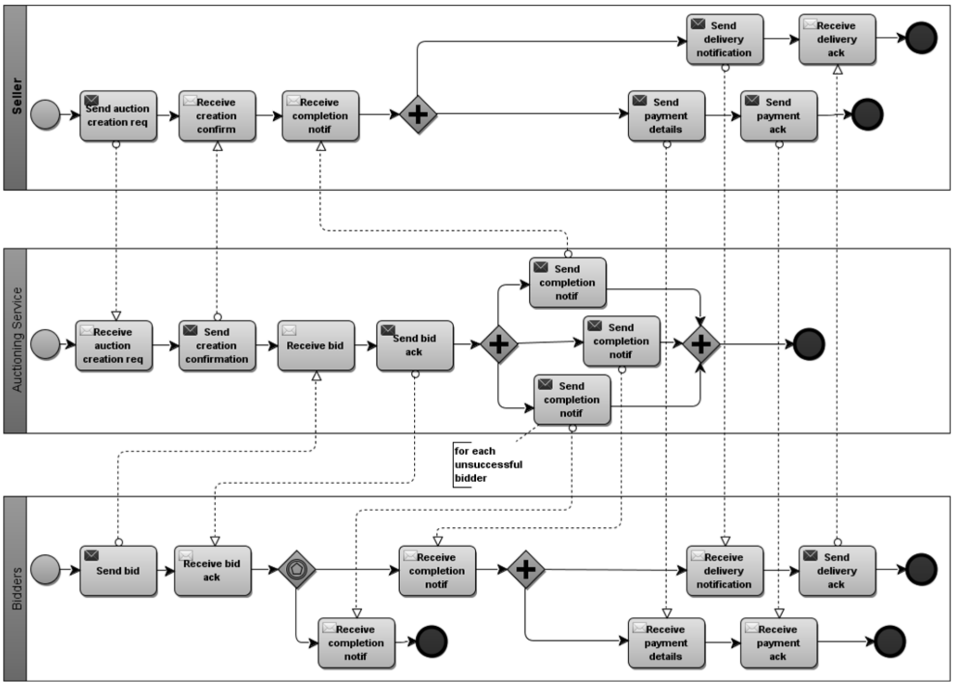

FIGURE 10 BPMN collaboration diagram of auction scene

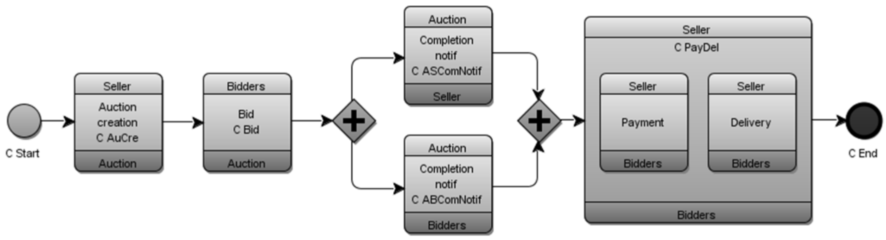

FIGURE 11 BPMN choreography flow chart of auction scene

4.2 CHOREOGRAPHY MODELLING OF AUCTION

Choreography is complex interaction of interdependent behaviour between entities, it demands multi-cooperation and at least two roles in choreography. For business process in Figure 10, logic abstract and the corresponding choreography diagram about it shows in Figure 11.

Before choreography modelling we define some tasks in this auction scene, which involves following eight choreography tasks: choreography begins C_{Start}, auction create C_{AuCre}, bid C_{Bid}, auction completion notification between Auction and Seller $C_{ASComNotif}$, auction completion notification between Auction and Bidders $C_{ABComNotif}$, parallel gateway C_{PG}, payment and product delivery C_{PayDel} and choreography ends C_{End}.

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

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}.

$$C_{Start}{}_{\bar{y}_{Start}} = \bar{y}_{Start}$$

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

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

Seller prepared data by τ_{Seller}, and send out auction creation information by m_{AuCre}. When Auction received it, execute $\tau_{Auction}$, and send out auction creation finished signal through y_{AuCre}. Finally, send out auction creation completed signal of choreography task by y_{AuCreS}.

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

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

Bidders prepare data by $\tau_{Bidders}$ and send out bid information by m_{Bid}. When Auction receives 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 completes notification choreography task between Auction and Seller:

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

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

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

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

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

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

$$\left[\!\left[C_{PG}\left(C_{ASComNotif}, C_{ABComNotif}\right)\right]\!\right]_{\bar{y}_{PGS}} = \left(v y_{ASComNotifS} y_{ABComNotifS}\right)\left(\left[\!\left[C_{ASComNotif}\right]\!\right]_{\bar{y}_{ASComNotifS}}\left|\left[\!\left[C_{ABComNotif}\right]\!\right]_{\bar{y}_{ABComNotifS}}\right|\left(y_{ASComNotifS}.y_{ABComNotifS} + y_{ABComNotifS}.y_{ASComNotifS}\right).\bar{y}_{PGS}\right).$$

$C_{ASComNotif}$ and $C_{ABComNotif}$ execute 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:

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

Seller prepares data by τ_{Seller}, and sends out payment and delivery information by m_{PayDel}. When Bidders receive it, execute $\tau_{Bidders}$ and send out payment and delivery success signal through y_{PayDel}. Finally, send out payment and delivery completed signal by $y_{PayDelS}$.

Model 8. The auction scene is executed sequentially:

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

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. Then it executes sequentially, the sequence tasks end 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.

$$C_{End} = \tau_{End}.0$$

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

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

This auction scene is composed of C_{Start}, C_{Seq} and C_{End}. $\tilde{c}$ is set of choreography tasks completed channel. C_{Start} starts the process, when sequence activities receive complete signal of C_{Start}, execute C_{AuCre} and C_{Bid} sequentially. Finnish the bidding completion notification to Sellers and Bidders through parallel gateway C_{PG}, and execute C_{PayDel}. Finally, ended the auction process with the choreography task C_{End}.

4.3 VALIDATION OF CHOREGRAPHY MODEL

Choreography modelling of business process based on the premise of its correctness [16]. In order to ensure the accuracy, integrity and consistency of the choreography semantics, we deduce and verify the choreography model. Followings are analysis for every step of process which to validate if each step is executed according to the design.

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

Choreography process starts, expand task C_{Start}:

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

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

$$\xrightarrow{\bar{y}_{Start}}(v\tilde{c})\,\tau_{Seller}.\bar{m}_{AuCre}\,|m_{AuCre}.\tau_{Auction}.\bar{y}_{AuCre}\,|y_{AuCre}.\bar{y}_{AuCreS}\,|y_{AuCreS}.\ C_{Bid}\ _{\bar{y}_{BidS}}\,|\ldots\ C_{End}$$

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

$$\xrightarrow{\tau_{Seller},\bar{m}_{AuCre}}(v\tilde{c})\,\tau_{Auction}.\bar{y}_{AuCre}\,|y_{AuCre}.\bar{y}_{AuCreS}\,|y_{AuCreS}.\ C_{Bid}\ _{\bar{y}_{BidS}}\,|\ldots\ C_{End}$$

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

$$\xrightarrow{\tau_{Auction},\bar{y}_{AuCre}}(v\tilde{c})\,\bar{y}_{AuCreS}\,|y_{AuCreS}.\ C_{Bid}\ _{\bar{y}_{BidS}}\,|\ldots\ C_{End}$$

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

$$\xrightarrow{\bar{y}_{AuCreS}}(v\tilde{c})\,\tau_{Bidders}.\bar{m}_{Bid}\,|m_{Bid}.\tau_{Auction}.\bar{y}_{Bid}\,|y_{Bid}.\bar{y}_{BidS}\,|y_{BidS}.\ C_{PG}\ _{\bar{y}_{PGS}}\,|\ldots\ C_{End}$$

Bidders of C_{Bid} prepares data through $\tau_{Bidders}$ and send out bid news through channel m_{Bid}:

$$\xrightarrow{\tau_{Bidders},\bar{m}_{Bid}}(v\tilde{c})\,\tau_{Auction}.\bar{y}_{Bid}\,|y_{Bid}.\bar{y}_{BidS}\,|y_{BidS}.\ C_{PG}\ _{\bar{y}_{PGS}}\,|\ldots\ C_{End}$$

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

$$\xrightarrow{\tau_{Auction},\bar{y}_{Bid}}(v\tilde{c})\,\bar{y}_{BidS}\,|y_{BidS}.\ C_{PG}\ _{\bar{y}_{PGS}}\,|\ldots\ C_{End}.$$

C_{Bid} sends out completion signal of bid through 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 $C_{ASComNotif}$:

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

$C_{ASComNotif}$ prepares data through $\tau_{Auction}$, and sends out completion notification of auction by $m_{ASComNotif}$, and Seller receives the completion notification:

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

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

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

It is same as $C_{ASComNotif}$, when $C_{ABComNotif}$ finished execution:

$$\xrightarrow{\tau_{Bidders},\bar{y}_{ABComNotif}}(v\tilde{c})\,\bar{y}_{ASComNotifS}\,|\bar{y}_{ABComNotifS}\,|(y_{ASComNotifS}.y_{ABComNotifS}+y_{ABComNotifS}.y_{ASComNotifS}).\bar{y}_{PGS}\,|\ldots\ C_{End}.$$

When both of $C_{ASComNotif}$ and $C_{ASComNotif}$ finished execution, they send out completion signal of choreography by $\bar{y}_{ASComNotifS}$ and $\bar{y}_{ABComNotifS}$.

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

Parallel gateway C_{PG} sends out completion signal of choreography by $\bar{y}_{PGS}$ and expands C_{PayDel}:

$$\xrightarrow{\bar{y}_{PGS}}(v\tilde{c})\,\tau_{Seller}.\bar{m}_{PayDel}\,|m_{PayDel}.\tau_{Bidders}.\bar{y}_{PayDel}\,|y_{PayDel}.\bar{y}_{PayDelS}\,|y_{PayDelS}.\bar{y}_{SeqS}\,|y_{SeqS}.\ C_{End}$$

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

$$\xrightarrow{\tau_{Seller},\bar{m}_{PayDel}}(v\tilde{c})\,\tau_{Bidders}.\bar{y}_{PayDel}\,|y_{PayDel}.\bar{y}_{PayDelS}\,|y_{PayDelS}.\bar{y}_{SeqS}\,|y_{SeqS}.\ C_{End}$$

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

$$\xrightarrow{\tau_{Bidders},\bar{y}_{PayDel}}(v\tilde{c})\,\bar{y}_{PayDelS}\,|y_{PayDelS}.\bar{y}_{SeqS}\,|y_{SeqS}.\ C_{End}$$

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

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

C_{Seq} sends out completion signal sequentially by $\bar{y}_{SeqS}$ and C_{End} receives this news:

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

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

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

At present, we have finished the correctness verify-cation of the choreography model for auction scene, among which does not have deadlocks. Through the validation process, we find that after the execution of choreography modelling, we get 0 at the last step that means each step of choreography model is completed. It is different with general collaboration diagram model. After the execution interaction of each node is completed, it will return to initial state of the node. Therefore, when verify collaboration diagram model, the last verification step will return to the initial model.

4.4 ADVANTAGES OF π CALCULUS MODELLING

There are advantages comparing with existing business process formal modelling methods.

1) Use Petri net to BPMN formal modelling can lead to state explosion issue, we choose π calculus for formal modelling. π calculus reduction is a dynamic evolution in mutual communication process[17]. We integrated the advantages of π calculus with BPMN2.0 multi-role modelling and inter-workflow choreography of enterprise characteristics. Therefore, π calculus is the most suitable mathematical theory for BPMN choreography.
2) To verify if there is deadlock in the model. Deadlock refers to two interaction of choreography waiting for each other to send messages, which lead to subsequent choreography can't continue. It is a standstill state among participates interaction [18]. The inference ability of the π calculus can be used to identify BPMN semantic error, verify if there are deadlocks in the model.

5 Conclusions

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

Acknowledgments

This work was supported by project of National Natural Science Foundation, China (No.61162013), the Natural Science Foundation of Ningxia (No.NZ12212), the Scientific Research Projects of Higher Education of Ningxia (No. NGY2012098) and the Postgraduate Innovation Project of Beifang University of Nationalities (No.2012XYC 051).

References

[1] Deng P, Zhang J W, Rong X H and Chen F 2013 *Telecommunication Systems* **52**(2) 1313-26
[2] Crafa S, Varacca D, Yoshida N 2012 Foundations of Software Science and Computational Structures *Springer Berlin Heidelberg* 225-39
[3] Dou H, Wu Y W 2012 *Xi'an University of Architecture and Technology Journal (Natural Science Edition)* **44**(2) 295-9 *(in Chinese)*
[4] OMG 2011 BPMN2.0 (Business process model and notation 2.0) Technical Report http://www.omg.org/spec/BPMN/2.0
[5] Cornax M C, Chessa S D, Rieu D and Dumas M 2011 Business Process Model and Notation *Springer Berlin Heidelberg* 103-17
[6] Einarsson H, Neukirchen H 2012 *Proceedings of the Fifth Workshop on Refactoring Tools* New York 16-23
[7] Jin W, Wang H P, Zhu M X 2012 *Acta Scientiarum Naturalium Universitatis Pekinensis* Beijing **48**(2) 209-16 *(in Chinese)*
[8] Dong Y X, Xia Y N, Sun T H, Zhu Q S 2010 *Journal of Computers* **5**(4) 516-23 *(in Chinese)*
[9] Calderaro V, Hadjicostis C N, Piccolo A, Siano P 2011 *IEEE Transactions on Industrial Electronics* **58**(10) 4613-23
[10] Milner R, Parrow J, Walker D 1992 A calculus of mobile processes (Pro I, II) *Informationand Computation* **100**(1) 1-77
[11] Leite L A F, Oliva G A, Nogueira G M, Gerosa M A, Kon F, Milojicic D S 2013 *Service Oriented Computing and Applications* **7**(3) 199-216
[12] Decker G, Weske M 2011 *Information Systems* **36**(2) 292-312
[13] Zhao Y W, Li J, Li Z Q, Ma D F 2011 Exploring Services Science *Springer Berlin Heidelberg* 1-15
[14] Saeedloei N, Gupta G 2014 Trustworthy Global Computing *Springer International Publishing* 119-35
[15] Anaya A R, Luque M 2013 Natural and Artificial Models in Computation and Biology *Springer Berlin Heidelberg* 235-44
[16] Le H A, Truong N T 2013 Context-Aware Systems and Applications *Springer Berlin Heidelberg* 290-9
[17] Huang Y W, He K Q, Li Z, Yuan K 2013 *Journal of Chinese Computer Systems* **34**(3) 461-5 *(in Chinese)*
[18] Ouederni M, Salaün G, Bultan T 2014 *The 10th International Symposium FACS China* Nanchang 310-4

Authors

Jiulei Jiang, September 1972, Jining, P.R. China.

Current position, grades: professor at the College of Computer Science and Engineering in Beifang University of Nationalities.
University studies: PhD degree in Information Management and Information System at Donghua University.
Scientific interests: information systems analysis, business process modelling, software engineering and formal methods.

Jiao Zhang, September 1987, Jining, P.R. China.

Current position, grades: master's degree student at the College of Computer Science and Engineering in Beifang University of Nationalities.
University studies: graduate student of Beifang University of Nationalities.
Scientific interests: business process modelling, software engineering and formal methods.

Wenxing Bao, February 1971, Ningxia, P.R. China.

Current position, grades: professor at the College of Computer Science and Engineering in Beifang University of Nationalities.
University studies: PhD degree in Electronic Science and Technology at Xi'an Jiaotong University.
Scientific interests: information systems, intelligent computing, computer graphics and image processing.