

CBPEL - Linguagem para definição de processos de negócio interorganizacionais

António Teófilo¹ e Alberto Rodrigues da Silva²

¹ ISEL/DEETC, ateofilo@deetc.isel.ipl.pt

² IST/INESC-ID, alberto.silva@inesc-id.pt

Resumo: Têm emergido recentemente várias linguagens XML para descrever processos de negócio, nomeadamente: BPEL, WSCI, XLANG, ou BPML. Contudo, estas linguagens são inadequadas para descrever processos de negócio interorganizacionais, que são processos com vários participantes em que o controle de execução é partilhado por todos. Para processos com estes requisitos poderia utilizar-se as linguagens WS-CDL ou BPSS/ebXML, mas que apresentam algumas limitações na sua utilização. Este trabalho visa propor uma linguagem para modelar processos de negócio interorganizacionais, baseada nos conceitos da linguagem BPEL, à qual designamos por CBPEL (*Common Business Process Execution Language*).

1 Introdução

O desenvolvimento dos Serviços Web (“*Web Services*”) [13] como uma tecnologia modular de interoperabilidade global, coloca-os como uma peça fundamental para a área dos processos de negócio (PN). A sua adopção como base para os processos descritos em notações XML [8], define a tecnologia de PN baseados em XML. Esta tecnologia visa a normalização da descrição, execução, e supervisão dos PN, proporcionando aos utilizadores um elevado grau de liberdade face às soluções proprietárias que caracterizaram a era antes-do-XML/WS, da qual se destaca somente as iniciativas de normalização do grupo *Workflow Management Coalition* [17].

A tecnologia de PN baseadas em XML, utiliza a pilha de protocolos básicos dos Serviços Web para os requisitos mais elementares, como: o SOAP [7], para a troca de mensagens; o WSDL [11], para a descrição dos serviços; e o UDDI [16] para publicação e descoberta de serviços. Pode, também, utilizar outros protocolos para providenciar características adicionais nas interações, como por exemplo: o WS-Security [5], para segurança, o WS-ReliableMessaging [6], para entrega fiável de mensagens, e o WS-Transaction¹ [9] para comportamento transaccional.

Actualmente existem várias linguagens XML para descrever PN que utilizam como tecnologia de suporte as várias especificações relacionadas com os Serviços Web, como por exemplo: BPEL [2], XLANG [15], BPML [3], BPSS/ebXML [10], WS-CDL [12], ou o WSCI [4].

¹ Esta designação abrange as especificações: WS-Coordination, WS-AtomicTransaction e WS-BusinessTransaction

Contudo o foco deste artigo é sobre os PN que envolvem várias organizações na sua execução, e que são designados de PN interorganizacionais. Estes PN são caracterizados por possuírem uma execução repartida pelos vários participantes, existindo portanto uma distribuição da sua execução.

Verifica-se que as linguagens anteriormente mencionadas, com excepção das linguagens *WS-CDL* e *BPSS/ebXML*, estão vocacionadas para a composição de Serviços Web centrada num processo, pelo que são inadequadas para a descrição de PN interorganizacionais. Este trabalho visa, preencher esse espaço, propondo uma linguagem baseada na linguagem *BPEL* capaz de descrever processos interorganizacionais.

Este artigo tem cinco secções, em que a primeira introduz o contexto do artigo, a segunda discute a descrição dos processos interorganizacionais, a terceira apresenta a linguagem interorganizacional baseada na linguagem *BPEL*, a quarta contém um exemplo descrito nessa linguagem, e por último são apresentadas as considerações finais do trabalho.

2 Descrição de processos interorganizacionais

Em relação à descrição de processos interorganizacionais será primeiro apresentada a distinção para com os processos centralizados, e depois será apresentada a forma global unificada, que é uma forma de descrever os processos interorganizacionais de forma mais concisa e menos propensa a erros.

Como já referido algumas linguagens adoptam um ponto de vista centrado no processo em questão, e assumem que a interacção com os outros processo é coordenada por ele, e portanto, que os outros processos se comportam de forma passiva. Este modelo é designado de *composição* ou *orquestração* de Serviços Web. É um modelo em que os parceiros, do processo central, são entidades com uma lógica simplista tipo pedido-resposta. As linguagens *BPEL*, *WSCI*, *XLANG* e *BPML* são exemplos deste tipo de linguagem.

O contexto de um processo que envolva várias organizações, em que todas possuem parte do controle do processo global, implica que o seu controle encontra-se distribuído pelos vários participantes. Nestes contextos o modelo centralizado não pode ser aplicado directamente, pois cada participante não tem o controle absoluto do processo. Este contextos são designados de *coreografia* de processos, pois todos os parceiros são, ou implementam, processos de negócio.

A descrição dos processos interorganizacionais poderá assumir duas formas. Uma delas consiste no somatório das descrições individuais dos processos de todos os participantes envolvidos. Essa forma implica que os processos sejam concebidos em paralelo e que haja perfeita correlação em todos os aspectos, como por exemplo o fluxo de controle, e o fluxo de dados. A outra forma tem em conta que: as interacções têm de ser complementares nos parceiros envolvidos; e que o fluxo de controle pode ser partilhado entre os vários participantes. Pelo que o processo interorganizacional pode ser descrito por um único fluxo global em que as actividades de interacção indicam quem é o participante emissor e o receptor, e que posteriormente será decomposto nos

fluxos correspondentes a cada participante. A esta forma de descrever o processo interorganizacional designamos de forma global unificada.

2.1 A forma global unificada

A forma global unificada é uma forma de descrever os processos interorganizacionais, de um modo mais sintético. Um processo interorganizacional pode ser descrito na sua totalidade pelo somatório dos processos das várias entidades participantes. Contudo, à medida que os processos vão crescendo em dimensão, a complexidade da sua edição vai aumentando, tornando-a propensa a erros, e de difícil observação.

Assim torna-se premente uma representação mais sintética, e se possível que proporcione uma redução da complexidade total. Esse é o objectivo da representação global unificada, que se baseia nos seguintes pontos:

- As interacções envolvem uma emissão e a respectiva recepção, que podem ser representadas numa única actividade de emissão-recepção.
- Apesar dos fluxos de controle existentes nos vários participantes poderem ser diferentes, eles devem partilhar padrões comuns para que o processo global seja isento de erros. Por exemplo, um envio de três mensagens diferentes em sequência, requer a recepção de três mensagens, que pode ser em paralelo, em sequência, ou por uma ordem qualquer. Contudo a utilização da sequência como ordem de recepção será o modo mais natural.

Juntando as interacções emissão-recepção num fluxo de controle único e global, que envolva todos os participantes, teremos a descrição do processo interorganizacional descrito de forma global unificada. Nas Figura 2 servem como exemplos de processos interorganizacionais, envolvendo três e quatro participantes, que do lado esquerdo são descritos de forma discreta, ou seja, cada participante tem o seu processo, e do lado direito são descritos pelo processo total na forma global unificada.

No exemplos das figuras mencionadas, utiliza-se a notação gráfica das Redes de Petri [14] para descrever os processos. Onde cada círculo representa um lugar, que é um contentor de unidades, e cada quadrado/rectângulo representa uma transição que ao disparar: retira uma unidade de cada lugar de entrada; deposita uma unidade em cada lugar de saída; e executa uma acção associada (se houver).

A decomposição do processo descrito de forma global unificada nos processos dos vários participantes tem os seguintes passos: 1) replicar o processo global para cada participante; 2) no processo de cada participante fazer: a) substituir todas as actividades em que o próprio não participa por actividades inertes, ou seja que não têm quaisquer efeitos; b) substituir todas as actividades de interacção por actividades de emissão ou recepção, consoante o seu papel; c) substituir as primitivas de decisão, no fluxo de controle, em que o próprio não seja o participante activo na tomada de decisão, por decisões tardias em função das mensagens que são recebidas; d) eliminar as actividades inertes e simplificar os fluxos de controle.

2.2 Suporte à forma global unificada

A linguagem *BPSS/ebXML* permite a descrição de processos que envolvem vários parceiros, por intermédio de colaborações com múltiplas partes (“*multiparty collaborations*”) mas que resultam da junção de várias colaborações binárias (“*binary collaborations*”) por intermédio de transições. Cada colaboração binária possui a descrição da interacção entre dois participantes. Utiliza, portanto, parcialmente a forma global unificada. Esta linguagem também apresenta as desvantagens de ser mais complexa e requerer uma base de implementação mais complexa que as linguagens anteriormente apresentadas. A linguagem *WS-CDL* utiliza directamente a forma global unificada, contudo encontra-se em fase de desenvolvimento (“*draft*”) e apresenta uma série de conceitos pouco claros, tais como o alinhamento de estado, e a utilização de canais (“*channels*”).

A linguagem BPEL (*Business Process Execution Language*) foi escolhida para servir de base à nova, agora proposta, linguagem de descrição de processos interorganizacionais especificados de forma global unificada, porque se encontra estável, possui uma sustentação teórica clara, que é baseada nas redes de Petri, e porque já existiam, à altura do início deste trabalho, implementações de executores de processos BPEL.

3 A linguagem Common BPEL (CBPEL)

Nesta secção é apresentada a linguagem de descrição de processos interorganizacionais, baseada na BPEL, que designamos de *Common Business Process Execution Language*, abreviadamente CBPEL. A descrição da CBPEL encontra-se organizada em dois grupos: dados e parceiros; e interacções e controle de fluxo.

3.1 Primeiro grupo: dados e parceiros

Neste grupo são abordados os elementos: *partnerLinkType*, *partnerLink*, *partners*, *variables*, *correlation sets*, *process* e *assign*.

3.1.1 PartnerLinkType e PartnerLinks

Na BPEL os *partnerLinkTypes* (PLT) são declarações ao nível do WSDL, que identificam uma ligação entre eventualmente dois intervenientes, em que cada interveniente assume um papel (“*Role*”) que é desempenhado por um *portType*. Os *partnerLinks* definem, no processo, para cada PLT, quem implementa cada uma das suas *roles*, indicando com *myRole* o papel do próprio processo, e com *partnerRole* o papel do outro parceiro.

Na CBPEL: o elemento PLT permanece igual, devendo contudo definir sempre dois papéis; nos *partnerLinks* dado que a vista do processo é global e única para todos os participantes, a noção de *myRole* será substituída por outro elemento *partnerRole*.

Estrutura do elemento *partnerLinks* na CBPEL:

```

<partnerLinks>?
  <partnerLink name="ncname" partnerLinkType="qname" >+
    <role name="ncname" partnerLinkTypeRoleName="ncname"/>
    <role name="ncname" partnerLinkTypeRoleName="ncname"/>
  </partnerLink> </partnerLinks>

```

3.1.2 Partners

A definição de *partners*, num processo BPEL, identifica quem são os outros parceiros do próprio processo, e quais os *partnerlinks* em que eles participam.

A definição de parceiros (“*partners*”) é fundamental na CBPEL, pois vai definir quais os papéis globais existentes e para cada um deles qual a sua participação em cada *partnerLink*. Cada *partner*, como será descrito de forma imparcial, e terá que possuir uma associação a cada *partnerlink-role* em que participa.

Estrutura do elemento “*partners*” na CBPEL fica então:

```

<partners>
  <partner name="ncname">+
    <partnerLink name="qname" role="ncname"/>+
  </partner> </partners>

```

3.1.3 Variable(s), Assign, copy e CorrelationSets

Na BPEL: o elemento *variable*, permite definir variáveis; os elementos *assign* e *copy* permitem realizar afectações e transformações de dados; e os conjuntos de correlação (“*correlation sets*”) permitem definir um conjunto de propriedades (“*properties*”) como um identificador de correlação. As *properties* são referências para campos de dados existentes nas mensagens que têm que ser conhecidos pelo processo. Este tipo de informação é normalmente designado de dados do protocolo (“*protocol data*”). Todos estes elementos permanecem inalterados na CBPEL.

3.1.4 Process

O elemento *process* é o elemento que contém as definições de um processo na linguagem BPEL. Na CBPEL o nome deste elemento será alterado para *commonProcess*, para elucidar a diferença de domínio.

Estrutura deste elemento na CBPEL:

```

<commonProcess name="ncname" . . .
  xmlns="http://www.temp.org/cbpel/2004/09/common-business-
  process/">
  <partnerLinks/>? <partners/>? <variables/>?
  <correlationSets/>? <faultHandlers/>?
  <compensationHandler/>? <eventHandlers/>?
  activity
</commonprocess>

```

3.2 Segundo grupo: interacções e controle de fluxo

Neste grupo são abordados os seguintes elementos: interacções (*invoke*, *receive*, *reply*), *wait*, *empty*, *sequence*, *switch*, *while*, *pick*, e *flow*.

3.2.1 Interacções

Para modelar interacções, na BPEL, existem os elementos: *receive*, *reply* e *invoke*. O elemento *receive* permite receber uma mensagem de um parceiro. O *reply* permite responder a uma chamada síncrona de um parceiro, do qual já se fez um *receive*. O *invoke* permite fazer o envio de uma mensagem e opcionalmente de forma síncrona, esperar por uma resposta.

A CBPEL, à semelhança com a *WS-CDL*, apresenta uma vista global comum sobre o processo, pelo que cada interacção é modelada por um único elemento, designado de “*send*”, e que representa um *invoke* num parceiro e um *receive* noutra parceiro.

Estrutura do elemento *send* na CBPEL:

```
<send partnerLink="ncname"
  fromPartner="ncname"    toPartner="ncname"
  fromPortType="qname"    toPortType="qname"
  fromOperation="ncname"  toOperation="ncname"
  inputVariable="ncname"  outputVariable="ncname"
  createInstance="yes|no"? <!-- receiver -->
  syncSendName="ncname"?  <!-- receiver -->
  standard-attributes> standard-elements
  <correlations>?
</send>
```

3.2.2 Wait e Empty

Na BPEL, o elemento *wait* permite realizar uma pausa na execução, e o elemento *empty* permite realizar a acção vazia. Estes elementos deverão ter na CBPEL a indicação de quem cumpre essa acção.

Estrutura dos elementos *wait* e *empty* na CBPEL:

```
<wait (for="duration-expr"|until="deadline-expr")
  partner="ncname" standard-attributes>
  standard-elements
</wait>
<empty partner="ncname" standard-attributes>
  standard-elements
</empty>
```

3.2.3 Sequence

O elemento *sequence*, na BPEL, define que a execução das actividades englobadas são realizadas em sequência. Este elemento permanece igual na CBPEL.

3.2.4 Flow

O elemento *flow*, na BPEL, permite definir actividades que serão executadas concorrentemente. Podem ser modeladas dependências, entre essas actividades concorrentes, pelo elemento “*link*”. Este elemento permanece igual na CBPEL. Na Figura 1 encontra-se um exemplo de um fluxo paralelo envolvendo vários participantes, e sua modelação na CBPEL. Na Tabela 1 encontra-se as descrições dos processos BPEL e CBPEL do exemplo da Figura 1.

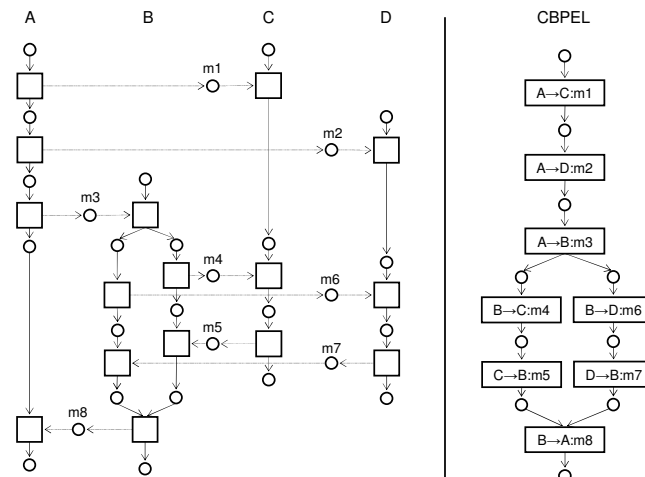


Figura 1 – Exemplo de um fluxo paralelo (*flow*) na CBPEL

Tabela 1 – Listagens BPEL e CBPEL do exemplo da Figura 1

Participante A	Participante B	Participante C	CBPEL
SEQ	SEQ	Seq	Seq
Send m1/c	Rec m3/a	Rec m1/a	a → c: m1
Send m2/d	Flow	Rec m4/b	a → d: m2
Send m3/b	Seq	Send m5/b	a → c: m3
Rec m8/b	Send m4/c		Flow
	Rec m5/c	Participante D	Seq
	Seq	Seq	b → c: m4
	Send m6/d	Rec m2/a	c → b: m5
	Rec m7/d	Rec m6/b	Seq
	Send m8/d	Send m7/b	b → d: m6
			d → b: m7
			b → a: m8

3.2.5 Switch

Este elemento, na BPEL, permite realizar uma decisão de escolha múltipla. Numa definição global unificada de um processo, uma escolha múltipla, terá que ser executada por um participante, o qual deverá iniciar a primeira interação de cada um dos ramos alternativos. Os outros participantes que figuram dentro da escolha múltipla terão que efectuar uma decisão tardia (*pick*) para saberem que ramo alternativo foi escolhido. Na Figura 2 encontra-se um exemplo de um fluxo contendo uma decisão, modelado em processos BPEL, e o respectivo processo CBPEL. As listagens deste exemplo foram omitidas por limitações de espaço.

3.2.6 While

Este elemento, na BPEL, permite a execução cíclica de uma actividade ou de um grupo de actividades. Este elemento tem um comportamento semelhante ao “*switch*”, pois

inclui uma decisão. Na Figura 3 encontra-se um exemplo de um fluxo interorganizacional descrito graficamente na BPEL e na CBPEL, e que contém um ciclo. Na Tabela 2 encontra-se as listagens correspondentes a esse exemplo.

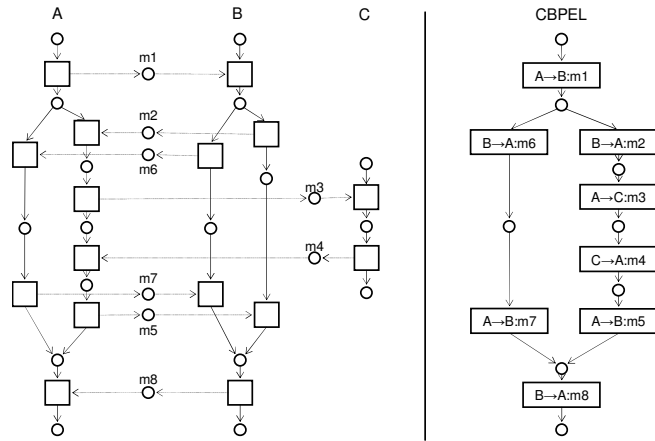


Figura 2 – Exemplo de uma decisão (*switch*) na CBPEL

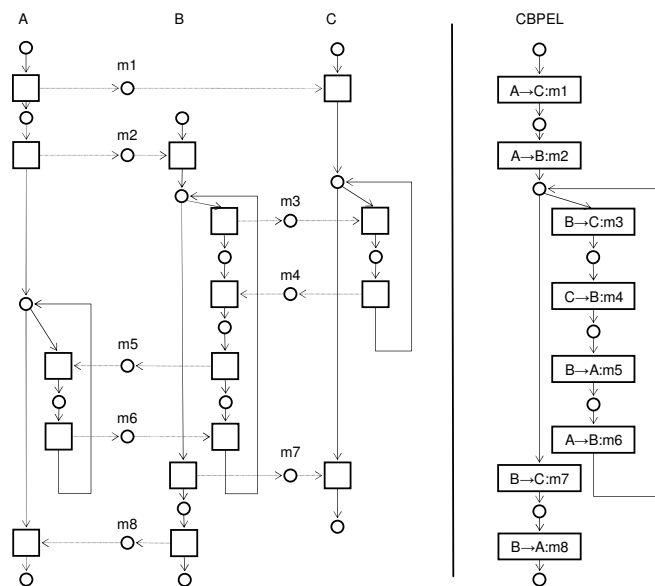


Figura 3 – Exemplo de um ciclo (*while*) na CBPEL

O exemplo da Figura 3 descreve um ciclo envolvendo três participantes. Da sua observação verifica-se que qualquer participante que intervenha dentro de um ciclo terá que também ele implementar esse ciclo, pois eventualmente não saberá quantas iterações irão existir. Contudo se o fluxo do participante é iniciado dentro do ciclo, então cada iteração do ciclo gera uma nova instância do fluxo desse participante.

Tabela 2 – Listagens BPEL e CBPEL do exemplo da Figura 3

Participante A	Participante B	Participante C	CBPEL
SEQ	SEQ	Seq	Seq
Send m1/c	Rec m2/a	Rec m1/a	a → c: m1
Send m2/b	While	While (!sair)	a → b: m2
Sair = 0	Seq	Pick	While
while (!sair)	Send m3/c	Rec m3/b	Seq
Pick	Rec m4/c	Send m4/b	b → c: m3
Rec m5/b	Send m5/a	Rec m7/b	c → b: m4
Send m6/b	Rec m6/a	Sair = 1	b → a: m5
Rec m8/b	Send m7/c		a → b: m6
Sair = 1	Send m8/a		b → c: m7
			b → a: m8

3.2.7 Pick

Este elemento, na BPEL, permite seleccionar a execução de actividades em função do primeiro acontecimento a ocorrer de entre uma série de mensagens (*onMessage*) e de uma série de alarmes temporais (*onAlarm*) que são vulgarmente designados de *timeouts*. O elemento *pick* em si não faz sentido existir na CBPEL pois é um elemento passivo. No entanto a capacidade de modelar *timeouts* deve existir na CBPEL. Assim o elemento *onAlarm* é adicionado ao elemento *send*, permitindo a indicação de *timeouts* na recepção de mensagens.

Estrutura do elemento *onAlarm* na CBPEL fica então:

```
<onAlarm (for="duration-expr" | until="deadline-expr")>*
  activity
</onAlarm>
```

4 Exemplo

Como exemplo, de um processo interorganizacional, é apresentado um cenário referente à realização de uma compra numa livraria electrónica, o qual é um cenário adaptado de [1]. O cenário contém quatro entidades participantes: o cliente (c), a livraria (b), o editor (p), e o serviço de entregas (s). Na Figura 4 apresenta-se o cenário descrito na sua forma global unificada, cujo descrição sumária em CBPEL é a seguinte:

```
<commonprocess name="Livraria Electrónica"
  xmlns="http://tempuri.org/cbpe1/2004/09/common-business-process/">
  <partners>
    <partner name="Customer"/>          <partner name="Bookstore"/>
    <partner name="Publisher"/>        <partner name="Shipper"/>
  </partners>
  <sequence>
    <send fromPartner="Customer" toPartner="Bookstore"
      Operation="c_order" />
    <send fromPartner="Bookstore" toPartner="Publisher"
      Operation="b_order" />
  <switch>
    <case v1>
      <sequence>
```

```

        <send fromPartner="Publisher" toPartner="Bookstore"
            Operation="pb_rejected" />
        <send fromPartner="Bookstore" toPartner="Customer"
            Operation="bb_rejected" />
    </sequence> </case>
<case v2>
    <sequence> // notação ainda mais resumida
        <send "Publisher" to "Bookstore" Op "pb_accepted" />
        <send "Bookstore" to "Customer" Op "bb_accepted" />
        <send "Bookstore" to "Shipper" Op "req_shipment" />
    <switch>
        <case v1>
            <sequence>
                <send "Shipper" to "Bookstore" Op "sb_rejected" />
                <send "Bookstore" to "publisher" Op "bp_rejected" />
                <send "Bookstore" to "Customer" Op "bb2_rejected" />
            </sequence> </case>
        <case v2>
            <sequence>
                <send "Shipper" to "Bookstore" Op "sb_accepted" />
                <send "Bookstore" to "Publisher" Op "inform_pub" />
                <send "Publisher" to "Shipper" Op "send_book" />
                <send "Shipper" to "Customer" Op "ship_book" />
                <send "Shipper" to "Bookstore" Op "notify" />
                <send "Bookstore" to "Customer" Op "send_bill" />
                <send "Customer" to "Bookstore" Op "payment" />
            </sequence> </case> </switch> </sequence> </case>
    </switch> </sequence> </commonprocess>

```

Seguidamente apresenta-se o conteúdo sumário do processo da livraria, descrito em BPEL, e extraído do processo global CBPEL anterior:

```

<process name="Bookstore-process" . . .
    xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/">
    <sequence>
        <receive partner="Customer" operation="c_order" />
        <invoke partner="Publisher" operation="b_order" />
        <pick>
            <onMessage frompartner="Publisher" operation="pb_rejected" />
                <invoke partner="Customer" operation="bb_rejected" />
            </onMessage>
            <onMessage frompartner="Publisher" operation="pb_accepted" />
                <sequence>
                    <invoke partner="Customer" operation="bb_accepted" />
                    <invoke partner="Shipper" operation="req_shipment" />
                <pick>
                    <onMessage frompartner="Shipper" operation="sb_rejected" />
                        <sequence>
                            <invoke partner="publisher" operation="bp_rejected" />
                            <invoke partner="Customer" operation="bb2_rejected" />
                        </sequence>
                    </onMessage>
                    <onMessage frompartner="Shipper" operation="sb_accepted" />
                        <sequence>
                            <invoke partner="Publisher" operation="inform_pub" />
                            <receive partner="Shipper" operation="notify" />
                            <invoke partner="Customer" operation="send_bill" />
                            <receive partner="Customer" operation="payment" />
                        </sequence> </onMessage> </pick> </sequence>
                </onMessage> </pick> </sequence> </process>

```

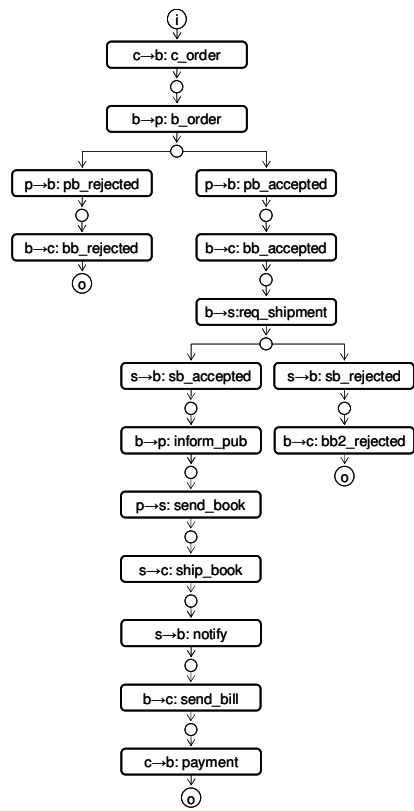


Figura 4 – Cenário interorganizacional da livraria electrónica

5 Considerações finais

Este trabalho propõe uma linguagem designada de *Common Business Process Execution Language*, ou abreviadamente CBPEL, que visa a descrição de processos de negócio interorganizacionais, e que é baseada na linguagem BPEL. A CBPEL descreve os processos interorganizacionais de uma forma global, que é unificadora das perspectivas dos seus vários participantes. O objectivo desta forma de descrição dos processos é diminuir a sua complexidade, que de outro modo seriam descritos pela conjugação dos processos individuais de todos os participantes.

No entanto, esta forma de descrever os processos interorganizacionais apresenta dois problemas: fraca noção da responsabilidade dos processos individuais; e limitação na definição dos fluxos. A fraca noção da responsabilidade dos processos individuais deve-se a que os processos agora estão todos juntos num único fluxo, contudo a ferramenta de edição pode apresentar de forma paralela com o processo global, o processo individual dos participantes de modo a se poder visualizar as responsabilidades individuais. A limitação dos fluxos dos participantes é o resultado da

descrição por fluxo único, contudo os participantes são livres de implementar o fluxo acordado do modo que entendam desde que preservem as suas características [1].

Como trabalho futuro são identificados os seguintes aspectos: (1) demonstrar que características a forma global unificada deve possuir para que os processos gerados, a partir de um processo global bem construído, sejam processos bem construídos; (2) incorporar o tratamento de eventos, falhas e compensações; e (3) incorporar um protocolo de coordenação distribuída como o protocolo *WS-BusinessActivity* [9] para garantir a coerente execução entre os vários participantes.

Referências

1. W. van der Aalst, M. Weske, The P2P Approach to Interorganizational Workflows, In K.R. Dittrich, A. Geppert, M. C. Norrie (eds), Proc. Of the 13th Int. Conf. on Advanced Information Systems(CAiSE'01), Berlin, 2001
2. T. Andrews, et al, Business Process Execution Language for Web Services, Version 1.0, May 2003, <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>
3. A. Arkin, Business Process Modeling Language (BPML), Working Draft 0.4, BPMI, March 2001, <http://www.bpmi.org/>
4. A. Arkin, et al, Web Services Choreography Interface (WSCI), version 1.0, W3C Note, August, 2002, <http://www.w3.org/TR/wsci/>
5. B. Atkinson, et al, Web Services Security (WS-Security), Microsoft IBM VeriSign, April, 2002, <http://www-106.ibm.com/developerworks/webservices/library/ws-secure/>
6. R. Bilorusets, et al, Web Services Reliable Messaging Protocol (WS-ReliableMessaging), March, 2004,
7. D. Box, et al, Simple Object Access Protocol, version 1.1, May, 2000, <http://www.w3.org/TR/SOAP>
8. T. Bray, et al, Extensible Markup Language (XML) 1.0, W3C, February, 1998, <http://www.w3.org/TR/REC-xml>
9. L. Cabrera, et al, Coordinating Web Services Activities with WS-Coordination, WS-AtomicTransaction, and WS-BusinessActivity, Microsoft Corporation, Jan, 2004, <http://msdn.microsoft.com/library/en-us/dnwebsrv/html/wsacoord.asp>
10. D.Chappel, et al, Professional ebXML Foundations, Wrox Press Ltd, UK, 2001 (<http://www.ebxml.org>)
11. E. Christensen, et al, Web Services Description Language (WSDL), version 1.1, W3C Note, March, 2001, <http://www.w3.org/TR/wsd.html>
12. N. Kavantzias, D. Burdett, G. Ritzinger, Web Services Choreography Description Language (WSCDL), version 1.0, April, 2004, <http://www.w3.org/TR/2004/WD-ws-cdl-10-20040427/>
13. H. Kregler, Web Services Conceptual Architecture (WSCA 1.0), IBM Software Group
14. J. Peterson, Petri net theory and the modeling of systems, Prentice Hall, 1981
15. S. Thatte: XLANG - Web Services for Business Process Design, Draft Specification. Microsoft, May 2001,http://www.gotdotnet.com/team/xml_wsspecs/xlang/default.htm
16. UDDI Group, Universal Description, Discovery and Integration, UDDI Technical White Paper, Ariba Microsoft IBM, Sep, 2000
17. WfMC, Workflow Management Coalition: Terminology & Glossary, UK, Feb, 1999