

Modelo de Referência para Linguagens de Especificação de PNWS

Paula Ventura Martins
Universidade do Algarve / INESC-ID
Campus de Gambelas
8005-139 Faro, Portugal
pventura@ualg.pt

Alberto Rodrigues Silva
Instituto Superior Técnico / INESC-ID
Rua Alves Redol, 9
1000-029 Lisboa, Portugal
alberto.silva@acm.org

RESUMO

Este documento apresenta uma proposta de modelo de referência para análise e comparação de linguagens de especificação de “processos de negócio baseados em *Web Services*” (PNWS). Descrevem-se os aspectos e tópicos que caracterizam o modelo referência. Segue-se a análise das várias linguagens de especificação PNWS, propostas a standards, referenciando os padrões comuns à modelação de processos de negócio. Para concluir mencionam-se aspectos fundamentais ainda não incorporados nas linguagens candidatas a standards para especificação de PNWS.

PALAVRAS-CHAVE

Processos de Negócio, *Web Services*, Linguagens.

1. INTRODUÇÃO

O uso crescente de sistemas distribuídos em larga escala originou um novo tipo de tecnologia associada ao conceito de *Web Services* [Rior03]. A integração de aplicações dentro da organização e entre organizações não requer apenas a implementação sobre várias plataformas e sistemas operativos, mas também implementações sujeitas a alterações que não afectem as interfaces, isto é, sistemas fracamente ligados.

Começamos por clarificar o significado do termo “*Web Services*” (WS). Não existindo uma definição formal e consensual deste conceito, adoptaremos a seguinte definição: WS são componentes de software que descrevem uma colecção de operações disponíveis em rede através de mensagens XML [Kreg01]. O serviço é descrito através de uma notação formal standard XML, designado por descrição de serviço. A descrição do serviço apresenta todos os detalhes de interacção com o serviço, ou seja, formato das mensagens, protocolos de transporte e localização. Porém, a interface de um WS esconde detalhes de implementação, permitindo o uso independentemente do software e plataforma de hardware em que é instalado, bem como da linguagem de programação em que é escrito. Os WS podem ser aplicados individualmente ou em conjunto com outros WS de forma a criar uma agregação complexa de serviços ou de uma transacção de negócio.

No suporte à definição e execução de processos de negócio é relevante a arquitectura de sistemas de informação. As características inerentes a um WS exigem a definição de uma arquitectura baseada em serviços e com três operações fundamentais: publicação, procura e ligação. Os fornecedores publicam os seus serviços num mediador de serviços. Por outro lado, os clientes procuram os serviços requeridos através do mediador, ligando-se posteriormente aos serviços suportados pelos fornecedores [Kreg01].

Um processo de negócio corresponde a uma sequência de actividades, que processam vários *inputs* e produzem vários *outputs* e que possuem objectivos, também designado por orquestração ou fluxo de trabalho [Silva01]. Um processo de negócio pode ser realizado por pessoas e/ou de forma automática. A maior parte

dos processos de negócio são definidos e suportados no contexto de sistemas de informação de forma a permitir um suporte computacional e eventual automação de procedimentos.

Considera-se que a necessidade de definir as características, terminologia e componentes inerentes às linguagens de especificação de “processos de negócio baseados em *Web Services*” (PNWS) exige a elaboração de um modelo de referência para uma sua mais fácil análise e classificação. Este modelo de referência deve ser descrito de forma geral e independente de qualquer das propostas e linguagens concretas.

Este artigo está estruturado em cinco Secções. Na Secção 2 é apresentado o estado da arte das linguagens de especificação de PNWS e sua relação com a pilha de protocolos adoptados. A Secção 3 aborda os conceitos de suporte aos PNWS e descreve o modelo de referência proposto. A aplicação do modelo de referência na análise e comparação de linguagens propostas para *standards* é apresentada na Secção 4. A Secção 5 conclui o artigo descrevendo as características ainda não incorporadas nas linguagens existentes e que deverão merecer investigação num futuro próximo.

2. LINGUAGENS DE ESPECIFICAÇÃO DE PNWS

Começamos por abordar a pilha genérica da arquitectura dos WS apresentada na Figura 1. A mesma Figura apresenta uma análise comparativa de tecnologias de algumas linguagens de especificação de PNWS. A maior parte das linguagens de especificação de PNWS aparecem no topo da pilha genérica de descrição de serviços, designando-se por camada de orquestração. Esta camada, em alguns casos, posiciona-se sobre o WSDL (*Web Service Description Language*) [Chris01], o qual descreve em detalhe a informação necessária para invocar um WS. No entanto, O WSDL não define mecanismos que permitam a invocação sequencial de vários WS ou de como invocar uma operação num WS.

As linguagens de orquestração, por seu lado, descrevem como organizar e coordenar um conjunto de invocações de WS na implementação de um conjunto de processos de negócio.

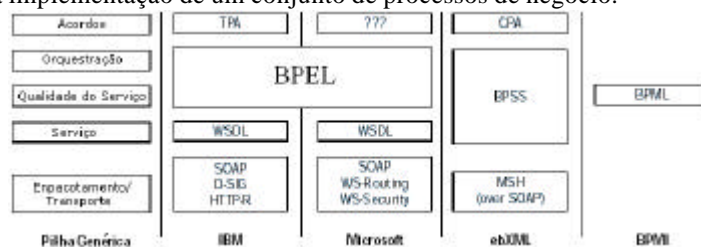


Figura 1: Pilha genérica da arquitectura dos WS (adaptado de [Rior03]).

Em seguida apresentam-se as linguagens propostas para a especificação de PNWS.

O *Business Process Execution Language for Web Services* (BPEL) [Curb03] é uma linguagem de modelação de processos de negócio baseados em WS, resultante da fusão das linguagens *Web Services Flow Language* (WSFL) [Leym01] e *XLANG* [That01], propostas anteriores apresentadas respectivamente pela IBM e pela Microsoft.

O *Business Process Modeling Language* (BPML) é uma proposta para especificação de PNWS apresentada pela *Business Process Management Initiative* (BPMI.org) [Arki03].

Vários consórcios apresentam propostas para especificação de fluxos de trabalho, assim como para especificação de interações entre dois ou mais WS. Entre outras referem-se:

?? **WSCI** (*Web Service Choreography Interface*) [Arki02] é a linguagem de especificação de PNWS proposta pelo consórcio Sun/BEA/Intalio/SAP “ é uma linguagem de especificação de interfaces baseada em XML que descreve o fluxo de mensagens trocadas entre participantes num WS durante uma interacção coordenada com outros serviços” [Vird03].

?? **XPDL** (*XML Process Definition Language*) [wfmc02] é uma proposta para a especificação de PNWS apresentada pela *Workflow Management Coalition* (WfMC). XPDL aparece como uma linguagem estruturada por grafos acrescida de outros conceitos para manipulação de blocos.

?? **BPSS** (*Business Process Specification Schema*) [Rior03] pertence ao conjunto de especificações ebXML B2B. BPSS é uma linguagem simples que apenas descreve processos públicos. A orquestração das transacções é definida através de fluxos de trabalho baseados na semântica dos diagramas de actividades do UML [Silva01],[Levi01].

3. MODELO DE REFERÊNCIA PROPOSTO

3.1 Conceitos de Suporte

As primitivas definidas nas linguagens de especificação de PNWS derivam de várias experiências anteriores sobre processos de negócio, na Figura 2 definem-se as relações existentes entre essas primitivas. Porém a abordagem do PNWS apresenta a noção de interface para composição de serviços.

Uma **interface** define um conjunto de operações, as quais são implementadas por **componentes**. Cada entidade participante no negócio tem um **papel** que é descrito pelas operações definidas na interface. A relação entre os papéis das entidades participantes (responsabilidade) no negócio é modelada por um serviço. Um **serviço** serve para representar relações directas entre duas entidades de negócio. Se um serviço é definido por agregação de vários serviços designa-se por **composto**, caso contrário é chamado **elementar**.

Os processos descrevem a sequência de interações entre entidades participantes no negócio. Um processo de negócio (**processo genérico**) pode ser representado (Figura 2), e decomposto com maior detalhe, como um conjunto de outros processos mais concretos os quais se designam por **subprocessos**, relativamente ao processo pai. Um processo elementar, ou seja que não pode ser decomposto, designa-se por **actividade**.

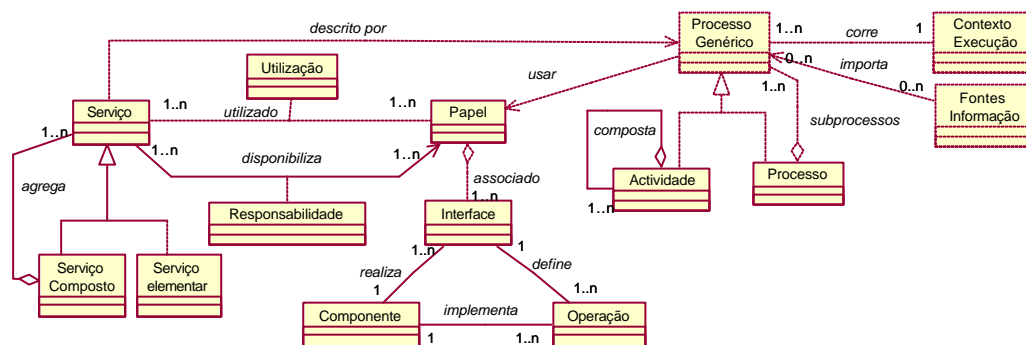


Figura 2: Conceitos de suporte aos PNWS.

Os **contextos de execução** definem o ambiente restrito de execução de processos directamente relacionados. A execução num mesmo contexto permite troca de informação e de coordenação das suas execuções. Um contexto contém definições locais apenas acessíveis a entidades que lhe pertencem.

Outro conceito consiste na possibilidade de importar elementos (**Fontes de Informação**) definidos noutras linguagens e que podem ser combinados com outros elementos, na mesma máquina ou em outras máquinas, para definir um novo serviço.

Nas linguagens de especificação de PNWS, processos que interactuam através de troca de mensagens recorrem a protocolos transaccionais para suporte a transacções de longa duração e encadeadas. A noção da compensação é muitas vezes aplicada na recuperação a falhas em transacções de longa duração bem como em processos do negócio suportados por sistemas de gestão de fluxos de trabalho. Em geral, no contexto dos processos de negócio, o desenho de tarefas de compensação não é trivial, e.g., deve ser especificada explicitamente (**Gestão de Transacções**).

O funcionamento correcto de um processo depende do método de coordenação das várias actividades que o constituem, surgindo por conseguinte o conceito de sinal como mecanismo que permite sincronizar a execução de várias actividades. Se ocorre uma excepção durante a execução de um processo de negócio, então é importante que o modelo apresente acções de recuperação de falhas.

3.2 Aspectos Técnicos

Os WS são sistemas fracamente ligados, e.g., a lógica da aplicação esconde-se atrás de uma interface que implementa um ou vários serviços. É mais bem sucedida a ideia da arquitectura cujos componentes dependem menos da implementação dos outros componentes. A arquitectura WS declara na interface a

descrição das mensagens transmitidas durante a invocação de serviços. Todos os detalhes da implementação do processo são escondidos do processo invocador.

Naturalmente, as interfaces têm necessidade de evoluir, surgindo o conceito de versões em WS. A **gestão de versões** é uma noção essencial, uma vez que a implementação de processos de negócios é constantemente reformulada. Como em qualquer modelo de programação de componentes, as aplicações clientes de WS dependem de versões específicas dos serviços publicados. Sendo o XML repositório das linguagens de especificação de PNWS, é essencial incluir nas alterações aos *schemas* XML e interfaces WSDL, um esquema de versões para evitar que os clientes existentes deixem de funcionar. A forma correcta de manter a compatibilidade entre componentes é criar os *schemas* para as interfaces e processos em *namespaces* únicos.

As mensagens enviadas a processos de negócio não necessitam apenas de ser entregues na interface correspondente, mas devem ser recebidas nos tipos de processo correctos. No contexto dos WS, essa informação deve ser incluída nos cabeçalhos dos protocolos de comunicação e nos dados dos serviços que definem as interfaces entre parceiros de negócio. A **correlação** é a forma de identificar um tipo de processo de negócio dentro de um conjunto do mesmo processo.

As linguagens devem incluir modelos que possibilitem a **composição de serviços**, ou seja, criar novos serviços por agregação de serviço existentes. Outro aspecto a considerar é a **localização de serviços**. Antes de executar uma operação, a actividade deve localizar o serviço responsável pela realização da operação. Deve também incluir uma **linguagem de interrogação** que permita interrogação sobre dados, extracção de valores de propriedade, e saber informação de estado das comunicações (geralmente XPath).

Na secção 4.1, a primeira coluna do Quadro 1 apresenta os conceitos de suporte e os aspectos técnicos, definidos anteriormente, que caracterizam as linguagens de especificação de PNWS.

3.3 Processos

Sendo os processos, elementos básicos na especificação de Sistemas de Informação, surge no contexto dos WS a necessidade de separar a sua classificação em processos privados e públicos face ao objectivo de preservar a confidencialidade nos processos de cada organização. Os **processos públicos** descrevem como a informação deve fluir entre as empresas. Os **processos privados** permitem às empresas preservar a confidencialidade dos seus processos e métodos de gestão de informação.

As **interfaces** são descritas por processos que especificam o comportamento de troca de mensagens de cada uma das partes envolvidas no negócio. Em algumas linguagens de especificação de PNWS, o conceito de **processo abstracto** aparece com o mesmo significado de interface.

A classificação de processos apresenta ainda a noção de **processo executável**, o qual pode recorrer a um vasto leque de características de manipulação e de atribuição de dados (mas sem permissão para tratar valores não deterministas). Os processos executáveis servem para definir as regras do negócio. A **execução de processos** pode ser feita através de mensagens, sinais ou por invocação numa actividade. O processo invocador poderá **esperar ou não** pela conclusão das actividades do processo invocado.

Os **processos aninhados** correspondem a processos definidos no contexto de um outro processo, designado por super-processo. Os processos criados nesse contexto, apenas podem aceder às propriedades e instanciar outros processos que sejam definidos nesse contexto.

Na secção 4.2, a primeira coluna do Quadro 2 resume os aspectos característicos dos processos.

3.4 Mensagens

Segundo o padrão de comunicação do modelo cliente/servidor, a invocação de operações de serviços tem dois tipos: comunicação síncrona (e.g., *request/response*, *solicit/response*, espera síncrona) ou comunicação assíncrona (e.g., *one-way*, *notification*).

Os tipos de PNWS exigem informação adicional nas mensagens recebidas no processo invocado, de modo a tomar conhecimento do estado em que se encontra o processo que invocou o serviço (Quadro 3). Essa **informação de estado** requer o recurso a variáveis de estado, como é o caso dos **conjuntos de correlação**. Os conjuntos de correlação servem para identificar os estados dos tipos de processos das partes envolvidas no negócio.

As linguagens apresentam um conjunto de elementos que permitem: (1) **referência a serviços** para especificação dos valores iniciais de serviços invocados; (2) **propriedades de mensagens** para definição do contexto em que uma mensagem se enquadra de acordo com os participantes envolvidos.

A prestação de serviços em sistemas distribuídos em larga escala requer tratamento especial sobre aspectos de segurança e autorização. As mensagens podem ser modificadas indevidamente (ou esquecidas); pelo que devem ser implementados **protocolos de segurança** para garantir que tal não possa acontecer enquanto são transmitidas ou quando chegam aos processos destino. Neste cenário é preciso manter inalterada a informação encapsulada numa mensagem (**fiabilidade da mensagem**). Adicionalmente, a aplicação cliente deve ter confiança no sistema e nas mensagens trocadas com o(s) seu(s) parceiro(s).

3.5 Sinais

Um sinal serve para coordenar, num determinado contexto, a execução de várias actividades. Também serve para apresentar condições que resultam da execução de uma actividade, permitindo a outras, dentro do mesmo contexto, detectar e reagir a essas condições. Existem várias funcionalidades que permitem gerir o recurso aos sinais para fins de coordenação (Quadro 4):

?? **Definição:** Instrução para geração de sinais num determinado processo de negócio.

?? **Instanciação:** Instrução para criar tipos de sinais.

?? **Sincronização:** Instrução que permite coordenar a execução de actividades e de determinados tipo de sinais, dentro de um contexto restrito.

3.6 Actividades

As actividades correspondem a processos elementares, i.e., a acções não interrompíveis que ocorrem nos processos de negócios. Os tópicos que se descrevem em seguida (Quadro 5) correspondem a tipos de operações atómicas que aparecem geralmente na sintaxe das linguagens de especificação de PNWS:

?? **Atribuição:** Permite alteração dos dados das mensagens.

?? **Faltas:** Indicação de uma condição de erro que impede a execução normal de uma actividade.

?? **Finalização:** Operação que permite o cancelamento de actividades.

?? **Interrupção:** Possibilidade de interromper as actividades por determinados períodos de tempo.

?? **Actividade encapsulada:** Actividade reconhecida num contexto perdendo significado no seu exterior.

?? **Compensação num contexto:** Actividade associada a acções de compensação num processo de negócio.

?? **Nula:** Actividade sem trabalho associado.

3.7 Tipos de Fluxos

A modelação de processos de negócio consiste em criar descrições formais dos fluxos de trabalho, incluindo as actividades a realizar e as relações de dependência entre si. Esta modelação pode ser realizada através de 3 tipos de fluxos: (1) fluxos de dados; (2) fluxos de controlo e (3) fluxos de excepção. Os **fluxos de dados** permitem a troca de informação entre actividades/processos. Os **fluxos de controlo** impõem regras e relações de dependência na transição entre processos. Os **fluxos de excepção** estão associados à ocorrência de erros ou excepções e consequente realização de processos para tratamento desses casos.

Os fluxos de controlo para modelação de processos baseiam-se nos seguintes construtores (Quadro 6):

?? **Sequência:** Uma sequência contém zero ou mais actividades para execução em série, pela ordem em que aparecem na lista de declaração do elemento.

?? **Paralelismo:** Permite especificar concorrência e sincronismo entre actividades.

?? **Seleccção condicional:** Permite especificar condições com correspondentes fluxos alternativos.

?? **Seleção por evento:** Permite o bloqueio até à ocorrência de um ou vários eventos e posterior execução da actividade associada.

?? **Iteração:** Execução das mesmas actividades sobre todos os elementos de uma lista.

?? **Ciclos:** Permite especificar a repetição cíclica (zero ou mais vezes) de várias actividades.

4. APLICAÇÃO DO MODELO DE REFERÊNCIA

4.1 Aspectos Técnicos

Iniciamos com a aplicação do meta-modelo do modelo de referência (Figura 2) a duas das linguagens de especificação de PNWS: BPEL e BPML. O BPEL define um modelo e uma gramática para descrever o comportamento de processos de negócio baseados em interacções entre processos e entidades de negócio (*partners*). A interacção entre entidades de negócio ocorre através de interfaces WS designadas por *porttype*. Outras características são mecanismos para tratamento de excepções no negócio e processamento de faltas.

Por outro lado, o BPML apresenta um modelo abstracto para descrição de processos e entidades de negócio. O modelo formal do BPML permite definir processos executáveis que contemplam actividades complexas, transacções com compensação de actividades, gestão de dados, concorrência, tratamento de excepções e contexto de execução. O BPML inclui uma gramática no formato XML *Schema* para troca de definições entre sistemas heterogéneos e ferramentas de modelação.

O BPML apresenta dois tipos de transacções: transacções atómicas e transacções embutidas (*embedded transaction*). A gestão de transacções é realizada por protocolos de transacção, tais como o BTP [Pott02] e o *WS-Transaction* [Cabr02]. Em ambas as linguagens, o modelo de gestão de transacções implica a invocação de um tipo próprio de actividades para situações de compensação (Quadro 1).

Quadro 1. Aspectos Técnicos das linguagens BPEL e BPML.

Aspectos Técnicos	BPEL	BPML
Gestão de Transacções	Não	Não
Repositório	XML	XML
Gestão de versões	Não	Não
Contexto de Execução	Sim	Sim
Fontes de Informação	Sim	Sim
Mensagem	Sim	Sim
Sinais	Sim	Sim
Processamento de Excepções	Sim	Sim
Interface de Comunicação	Sim	Não
Linguagem de Interrogação	XPath	XPath, XSTL e XQuery
Correlação	Sim	Sim
Localização de Serviços	Não	Sim
Composição de Serviços	Não	Não

Um contexto pode incluir tratamento de faltas, de interrupções e de compensações, variáveis e conjuntos de correlação. No BPEL, o contexto de execução tem uma aplicação restrita a actividades. Cada contexto (*scope*) tem uma actividade principal que define o seu comportamento normal. Uma actividade principal pode ser um conjunto encapsulado de actividades. No BPML o contexto de execução tem uma aplicação mais vasta ao nível de outros processos: Um processo inclui a definição de um contexto que é partilhado por todas as actividades executadas no seu contexto (contexto do processo).

A facilidade de importação de elementos de outras linguagens permite ao BPEL e ao BPML a reutilização de especificações de *namespaces* WSDL, incluindo no BPEL a possibilidade de importar *porttypes*.

Uma das aplicações de exemplo do BPEL é a descrição de interacções entre empresas. Um requisito importante é a facilidade com que se modela o processo do parceiro. Esta linguagem apresenta interfaces,

onde a noção de *service link* serve para modelar directamente as relações entre parceiros. Porém, o serviço do parceiro pode ser determinado dinamicamente no processo. O conceito de *service reference* destina-se a representar os dados dinâmicos necessários para descrever tal serviço requerido.

Actualmente, no BPEL e BPML, a linguagem de interrogação XPath permite realizar perguntas sobre as variáveis, extracção de valores das propriedades e saber o estado das ligações.

Como o BPML não apresenta interfaces para comunicação entre processos de parceiros de negócio, o atributo *location* no elemento *service* e o atributo *locate* na actividade *action* identifica o endereço do documento que contém a definição do serviço. O documento pode incluir a definição de vários serviços.

Actualmente o BPEL não suporta a composição de serviços, o seu antecessor WSFL incluía este aspecto.

4.2 Processos

O BPEL descreve os processos das entidades de negócio sem diferenciar aspectos públicos e privados, designando-os por processos executáveis (Quadro 2). Os processos abstractos especificam mensagens mutuamente visíveis que descrevem o comportamento dos participantes no negócio. A sintaxe do BPEL diferencia processos executáveis de processos abstractos essencialmente através das extensões que cada um inclui. Para a definição de processos executáveis, inclui linguagem de interrogação, variáveis, atribuições, correlações, operações WS, compensação, tratamento de interrupções. Os processos abstractos apenas permitem variáveis e atribuições.

No BPEL, o facto de não existirem processos aninhados, restringe a execução de outros processos através das actividades *receive*, *reply* e *invoke*. Uma actividade *invoke* permite ao processo activar uma operação no *porttype* de um parceiro. As actividades *receive* e *reply* permitem a um processo esperar ou enviar uma mensagem de ou para um parceiro de negócio.

No BPML a actividade *call* e *spawn* permitem criar tipos de processos, a primeira espera que o processo invocado termine a sua execução, o que não acontece com a actividade *spawn*.

Apenas o BPML inclui a possibilidade de definir processos aninhados (*nested process*).

Quadro 2. Processos nas linguagens BPEL e BPML.

Processos	BPEL	BPML
Processos públicos ou privados	-	Sim
Processos executáveis ou abstractos	Sim	Não
Execução de processos	<i>Receive ou reply ou invoke</i>	<i>Call, spawn</i>
Criação de instância com e sem espera	-	<i>Call, spawn</i>
Processos aninhados	-	Sim

⁽¹⁾ A operação é realizada através de uma das actividades

4.3 Mensagens

No BPEL, a correlação entre os tipos dos processos dos parceiros é realizada através de conjuntos de correlação (*correlationSets*), os quais são integrados nas mensagens como forma de identificação do estado do serviço parceiro (Quadro 3). O BPML apresenta na definição do processo o atributo *correlation* para identificar uma ou mais definições de correlações, as correlações permitem associar mensagens de entrada a tipos de processos. Adicionalmente, importa correlações definidas em documentos WSDL.

Quadro 3. Definição de Mensagens no BPEL e BPML.

Mensagem	BPEL	BPML
Informação de Estado	<i>correlationSets</i>	Sim
Referência a Serviços	<i>Propertyalias ou assign ou query ou variable</i>	-
Tipos de Comunicação	Síncrona e assíncrona	Síncrona e assíncrona
Propriedades de Mensagens	<i>Property</i>	<i>Property</i>
Protocolo de segurança	-	-
Fiabilidade da mensagem	-	-

Ambas as linguagens incluem os dois tipos de comunicação: síncrona (*request-reponse*) e assíncrona (*one-way*). As propriedades aparecem nas mensagens de entrada e de saída no atributo *property*.

No BPEL, a combinação das actividades *reply* e *receive* corresponde a uma operação *request-response*. O construtor *invoke* permite operações *one-way* e *request-response*. No BPML a actividade *action* invoca uma operação que inclui troca de mensagens entre processos de parceiros de negócio. A comunicação é síncrona quando a operação envolve mensagens de entrada e de saída. A comunicação é assíncrona quando a operação inclui apenas uma mensagem de entrada ou de saída.

4.4 Sinais

De acordo com a informação apresentada no Quadro 4, o BPML é uma linguagem mais flexível no que diz respeito ao tratamento de excepções. São provas desse facto, a possibilidade de criar tipos de sinais e a facilidade de realizar sincronismo de actividades com uma única instrução, ao contrário do BPEL que necessita de 3 instruções para sincronismo de actividades.

Quadro 4. Sinais nas linguagens BPEL e BPML.

Sinais	BPEL	BPML
Definição	<i>Throw</i>	<i>Signal</i>
Instanciação	-	<i>Raise</i>
Sincronização	<i>links, source e target</i> ⁽¹⁾	<i>Synch</i>

⁽¹⁾ A operação é composta pelo conjunto das 3 actividades

4.5 Actividades

O conjunto de tipos de actividades apresentadas por ambas as linguagens é idêntico (Quadro 5). O BPEL apresenta mais uma actividade que permite abortar a execução de um processo. No BPML, as noções de actividades encapsuladas e de compensação são mais abrangentes pelo facto de serem aplicadas a processos.

Quadro 5. Actividades simples nas linguagens BPEL e BPML.

Actividades Simples	BPEL	BPML
Atribuição	<i>Assign</i>	<i>Assign</i>
Faltas	<i>Throw</i>	<i>Fault</i>
Finalização	<i>Terminate</i>	-
Interrupção	<i>Wait</i>	<i>Delay</i>
Actividade encapsulada	<i>Scope</i>	<i>Context</i>
Compensação num contexto	<i>Compensate</i>	<i>Compensation e Compensate</i>
Nula	<i>Empty</i>	<i>Empty</i>

4.6 Tipos de Fluxos

As actividades que permitem fluxos de controlo apresentam quase sempre a mesma terminologia (Quadro 6).

Quadro 6. Fluxos de controlo nas linguagens BPEL e BPML.

Fluxos de controlo	BPEL	BPML
Sequência	<i>Sequence</i>	<i>Sequence</i>
Paralelismo	<i>Flow</i>	<i>All</i>
Seleccção condicional	<i>Switch</i>	<i>Switch</i>
Seleccção por evento	<i>Pick</i>	<i>Pick</i>
Iteração	-	<i>Foreach</i>
Ciclos	<i>While</i>	<i>While ou Until</i>

As excepções são: (1) o caso de paralelismo de actividades, no BPEL designa-se por *flow* e no BPML por *all*; (2) o BPML apresenta mais o fluxo de controlo *until* para a realização de ciclos; (3) o tipo de fluxo iteração apenas aparece no BPML.

5. CONCLUSÃO

O sucesso dos WS na integração de infra-estruturas de negócio é um facto que foca a necessidade de *standards* ao nível dos PNWS. Das duas linguagens mencionadas e candidatas a *standard* de especificação de processos de negócio, o BPEL é uma boa solução para integração de serviços

No entanto, o BPEL e o BPML não contemplam ainda aspectos essenciais como sejam: segurança e autorização, gestão de transacções e gestão de versões. Em ambas as linguagens, a gestão de transacções é feita por protocolos transaccionais externos à linguagem, tal como o *WS-Transaction*. A especificação de protocolos de segurança e de autorização é um tópico crítico, porém estas linguagens ainda não lhe fazem referência.

Por outro lado, a constante reformulação e evolução dos serviços fornecidos pelas organizações exige mecanismos de controlo de versões de forma a evitar que os serviços baseados em implementações anteriores deixem de funcionar. Este aspecto também ainda não é referido em qualquer das linguagens emergentes.

Por conseguinte e como conclusão geral: o estado actual das linguagens de especificação de PNWS começa a ser adequado à concretização de projectos concretos; faltando ainda, contudo, um conjunto de aspectos que serão alvo de trabalho futuro.

Este artigo apresenta um modelo de referência como uma ferramenta para analisar as características e adequação das linguagens de especificação de PNWS, bem como a comparação das funcionalidades apresentadas pelas linguagens.

REFERÊNCIAS

- [Arki02] A. Arkin et al, 2002. *Web Service Choreography Interface 1.0*.
<http://www.sun.com/software/xml/developers/wsci/wsci-spec-10.pdf>
- [Arki03] A. Arkin, 2003. *Business Process Modeling Language*. BPMI Proposed Recommendation Janeiro 24, 2003.
<http://www.bpmi.org>
- [Cabr02] F. Cabrera et al, 2002, *Web Services Transaction WS-Transaction*.
<http://www-106.ibm.com/developerworks/webservices/library/ws-transpec/>
- [Chris01] E. Cristensen et al, Março 2001. *Web Service Description Language (WSDL) 1.1*
<http://www.w3.org/TR/wsd1>
- [Curb03] Curbera F et al., 2003. *Business Process Execution Language for Web Services, Version 1.1*.
<http://www106.ibm.com/developerworks/webservices/library/ws-bpel/>
- [Holl95] D. Hollingsworth, 1995. *Workflow Management Coalition -The Workflow Reference Model*.
- [Kreg01] H. Kreger, 2001. *Web Services Conceptual Architecture (WSCA 1.0)*.
<http://www4.ibm.com/software/solutions/webservices/>
- [Levi01] P. Levine et al, Maio 2001. *ebXML Business Process Specification Schema Version 1.01*.
<http://www.ebxml.org/specs/ebBPSS.pdf>
- [Leym01] F. Leymann, 2001. *Web Services Flow Language (WSFL 1.0)*.
<http://www3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf>
- [Pott02] M. Potts et al, 2002, *Business Transaction Protocol*.
<http://www.oasis-open.org/committees/documents.php>
- [Rior03] D. O'Riordan. *Business Process Standards for Web Services*.
<http://www.webservicesarchitect.com/content/articles/BPSFWSBDO.pdf>
- [Silva01] A. Silva, C. Videira, 2001. *UML, Metodologias e Ferramentas CASE*. Edições Centro Atlântico, Portugal.
- [That01] S. Thatte, 2001. *XLANG- Web Services for Business Process Design*.
<http://www.servicearchitecture.com/webservices/articles/xlang.html>
- [Vird03] M. Virdell, 2003. *Business processes and workflow in the Web services world*.
<http://www.106.ibm.com/developerworks/webservices/library/ws-work.html>
- [wfmc02] 2002. *Workflow Process Definition Interface -- XML Process Definition Language*.
<http://www.wfmc.org>