

Aleksandra WERNER  
Politechnika Śląska, Instytut Informatyki

## BPEL4WS – NOWY STANDARD ŁĄCZENIA PROCESÓW BIZNESOWYCH W ŚRODOWISKU INTERNETOWYM

**Streszczenie.** W artykule zaprezentowano ideę łączenia procesów biznesowych za pomocą nowego standardu językowego BPEL4WS, który został zaprojektowany z myślą o pełnym wzajemnym zrozumieniu się różnych procesów biznesowych w środowisku internetowym. Artykuł prezentuje również możliwe scenariusze wykorzystania BPEL4WS oraz zawiera przykład ilustrujący sposób kompozycji czynności zapisanych w standardzie BPEL4WS, a składających się na proces biznesowy.

**Słowa kluczowe:** język BPEL4WS, WSDL, WS-Coordination, WS-Transaction, SOAs

## BPEL4WS – NEW STANDARD FOR BUSINESS PROCESSES WEB SERVICES INTEGRATION

**Summary.** This article presents a new language notation for specifying business process behavior based on Web Services, called BPEL4WS, which enables easier execution of those processes, allowing different business processes to understand each other in a web services environment. Possible usage scenarios and an example, how activities may be created and aggregated with BPEL4WS, are also described in the article.

**Keywords:** BPEL4WS language, WSDL, WS-Coordination, WS-Transaction, SOAs

### 1. Wprowadzenie

Biznesowe serwisy internetowe to niezależne, modułarne aplikacje procesów biznesowych, które są oparte na standardowych technologiach przemysłowych: WSDL (gramatyka do opisu serwisów), UDDI (infrastruktura do publikowania serwisów) i SOAP

(protokół do komunikacji między serwisami). Pozwalają one użytkownikom na łączenie różnych komponentów w sposób niezależny od języka i platformy. W chwili obecnej żaden z istniejących standardów nie pozwala na definiowanie semantyki biznesowej w serwisach internetowych, co powoduje, że serwisy internetowe są izolowane i nieprzezroczyste. Przerwanie izolacji oznacza udostępnienie możliwości łączenia serwisów internetowych i specyfikacji połączeń istniejących serwisów internetowych w celu realizacji bardziej kompleksowych zadań i funkcji.

W lutym 2002 r. Microsoft, IBM oraz BEA założyły organizację Web Services Interoperability (WS-I), promującą pełną współpracę serwisów internetowych. Rezultatem porozumienia tych trzech firm było opublikowanie 9 sierpnia 2002 r. informacji o nowym standardzie, specyfikującym sposoby niezawodnego definiowania, tworzenia i łączenia wielu procesów biznesowych w środowisku serwisów internetowych. Podane specyfikacje mają pomóc organizacjom w koordynacji procesów biznesowych i transakcji, zachodzących w systemach heterogenicznych zarówno w obrębie przedsiębiorstwa, jak i obejmujących swoim działaniem również klientów i partnerów projektu. Zaproponowane komponenty standardu, to:

- specyfikacje WS-Coordination i WS-Transaction do kierowania zatwierdzonych przesyłów serwisów internetowych,

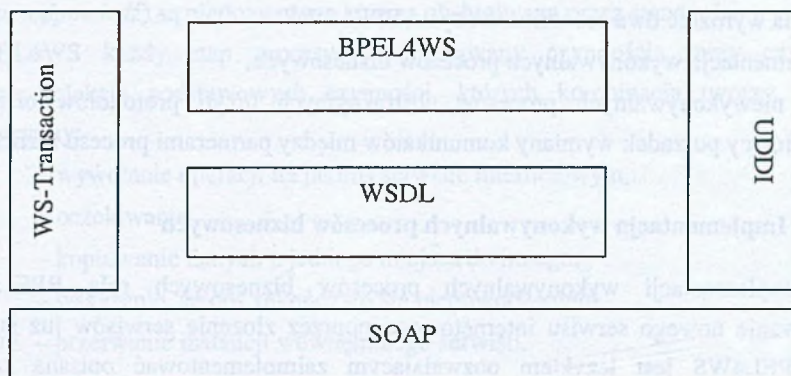
oraz

- nowy język BPEL4WS, opisujący procesy biznesowe i standaryzujący wymianę komunikatów (zarówno w samym przedsiębiorstwie, jak i między partnerami).

Kiedy przy użyciu BPEL4WS proces biznesowy i połączenia z klientami i partnerami oraz wewnętrzne obiekty są już zdefiniowane, następnym etapem jest koordynacja różnych działań występujących w danym procesie, wykonywana za pomocą WS-Coordination oraz WS-Transaction. Komponenty te są dopełnieniem języka BPEL4WS, gdyż dostarczają przedsiębiorstwom sposobów koordynacji i integracji wielu różnych serwisów internetowych i procesów biznesowych w sposób spójny i niezawodny w różnych środowiskach implementacji, w celu zapewnienia prawidłowego wyniku działań. WS-Coordination dostarcza wykonawców (tzw. wywoływaczy) z mechanizmami, zapewniającymi w razie potrzeby wykonanie (również zleconych innym procesom) wielu serwisów lub transakcji, podczas gdy WS-Transaction rezyduje na wyższym poziomie i określa protokół dla długich transakcji zdefiniowanych w BPEL4WS, jak również transakcji atomowych pomiędzy klasycznymi (standardowymi) serwisami internetowymi.

Ogólne związki między standardami przedstawia rys. 1, przy czym BPEL4WS został umiejscowiony na szczycie WSDL, gdyż zarówno procesy, jak i partnerzy są w nim modelowani jako serwisy WSDL. Uogólniając: BPEL4WS używa WSDL do specyfikacji

akcji, które mają mieć miejsce w procesie biznesowym i do opisu serwisów internetowych dostarczanych przez proces biznesowy.



Rys. 1. Związki między standardami

Fig. 1. Relationships among standards

## 2. Język BPEL4WS

Bazujący na gramatyce XML język BPEL4WS (ang. Business Process Execution Language for Web Services) został zaprojektowany z myślą o pełnym wzajemnym zrozumieniu się różnych procesów biznesowych w środowisku internetowym i jest kompozycją dwóch poprzednio konkurujących standardów:

- Web Services Flow Language (WSFL) firmy IBM (procesy zorientowane graficznie) oraz
- XLang firmy Microsoft (strukturalne konstrukcje dla procesów).

BPEL4WS wydaje się być podobny do specyfikacji WSCI (ang. Web Services Choreography Interface) promowanej przez firmę SUN, jednak, mimo zbieżności<sup>1</sup> obu specyfikacji, SUN nie został zaproszony do uczestnictwa w przedsięwzięciu WS-I. W chwili obecnej wszystkie zainteresowane firmy (łącznie z SUN-em) zdecydowanie deklarują chęć współpracy i nie wykluczają dalszego wzajemnego połączenia wysiłków nad standardem automatyzującym transakcje i procesy biznesowe w serwisie internetowym. Biorąc dodatkowo pod uwagę fakt, że specjaliści<sup>2</sup> zgodnie twierdzą, iż opracowywany standard zrewolucjonizuje rynek integracyjny w takim samym stopniu, w jakim SQL

<sup>1</sup>WSCI zawiera wytyczne o wzajemnym dopasowaniu zdarzeń i transakcji zachodzących między komputerami, aplikacjami i serwisami dostępnymi przez Internet.

<sup>2</sup>Przykładowo specjaliści z VITRIA – lidera spośród dostawców innowacyjnych rozwiązań integracji procesów biznesowych – również deklarują wsparcie dla specyfikacji BPEL4WS.

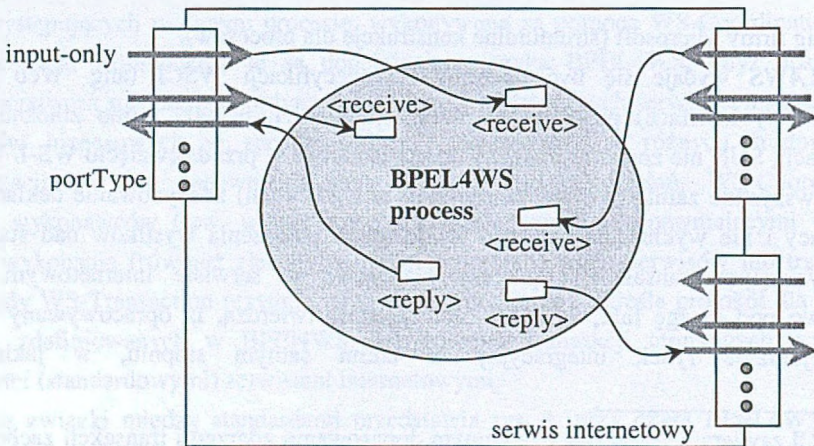
zrewolucjonizował kiedyś rynek baz danych, wzajemne urazy lub uprzedzenia istotnie powinny zejść na dalszy plan.

Można wyróżnić dwa scenariusze użycia BPEL4WS:

- implementacja wykonywalnych procesów biznesowych,
- opis niewykonywalnych procesów abstrakcyjnych (czyli protokołów biznesowych), definiujący porządek wymiany komunikatów między partnerami procesu biznesowego.

### 2.1. Implementacja wykonywalnych procesów biznesowych

W implementacji wykonywalnych procesów biznesowych rolą BPEL4WS jest zdefiniowanie nowego serwisu internetowego poprzez złożenie serwisów już istniejących. Czyli BPEL4WS jest językiem pozwalającym zaimplementować opisaną kompozycję serwisów (interfejs zaistniałego serwisu złożonego jest wówczas opisany jako kolekcja portów *portType* typu WSDL, identyfikujących określone zbiory operacji i powiązane z nimi komunikaty). Kompozycja taka (nazywana po prostu procesem) wskazuje miejsce interfejsu serwisu w swoim łańcuchu wykonawczym. Warto odnotować, że nie każda operacja każdego portu znajduje odzwierciedlenie w oddzielnej części logiki BPEL4WS. Zamiast tego wewnętrzny typ serwisu (zbiór portów serwisu) jest implementowany przez pojedynczy proces BPEL4WS. Tak więc w opisie BPEL4WS jedynie wskazywane są tzw. punkty wejścia do portu, odpowiadające zewnętrznym użytkownikom, wywołującym operacje interfejsu. Te punkty wejścia pobierają (konsumują) przychodzące z operacji typu *input-only* lub *input-output* komunikaty operacji WSDL. W przypadku operacji *input-output* proces musi dodatkowo wskazywać miejsce, gdzie generowany jest komunikat wyjściowy.



Rys. 2. Serwis internetowy jako proces BPEL4WS

Fig. 2. Web service implemented as a BPEL4WS process

Warto tu odnotować, że standard BPEL4WS wspiera tylko operacje typu *input-only* lub *input-output* (żądanie-odpowiedź) WSDL. Operacje *output-only* (ogłaszanie) oraz *output-input* (zajście-odpowiedź) są niedozwolone i nie są obsługiwane przez standard.

W BPEL4WS każdy etap procesu jest nazywany czynnością, przy czym jest zdefiniowana kolekcja podstawowych czynności, których kombinacja tworzy bardziej złożone algorytmy:

- *invoke* – wywołanie operacji na jakimś serwisie internetowym,
- *wait* – oczekiwanie,
- *assign* – kopiowanie danych z jednego miejsca do innego,
- *throw* – wskazanie, że coś zakończyło się niepowodzeniem,
- *terminate* – przerwanie instancji wewnętrznego serwisu,
- *empty* – proces pusty.

Dodatkowo istnieje też możliwość:

- definiowania uporządkowanego ciągu działań (*sequence*),
- używania pól wyboru (*case-statement switch*),
- definiowania pętli (*while*),
- wykonywania wybranej drogi spośród kilku alternatywnych (*pick*),
- wskazania, które ciągi kroków mają być wykonane równolegle (*flow*).

BPEL4WS ma również wbudowane w język wyjątki dostępne przez konstrukcje *throw* i *catch*.

Procesy BPEL4WS składają się głównie z wywołań skierowanych do innych serwisów i/lub odbioru wywołań od klientów (użytkowników serwisu). Te czynności są definiowane przez użycie akcji: *invoke*, *receive*, oraz *reply*.

## 2.2. Definiowanie procesu biznesowego

Procesy biznesowe mogą być opisane w dwojaki sposób:

- jako wykonywalne procesy biznesowe, modelujące aktualne zachowanie uczestników wymiany biznesowej (brak jakichkolwiek założeń odnośnie do separacji publicznych, czyli powszechnych aspektów procesu biznesowego od wewnętrznego działania),
- jako procesy abstrakcyjne, będące opisami procesu na potrzeby protokołów biznesowych. Opisy procesu specyfikują widoczne zachowanie wymiany komunikatów każdej strony, uczestniczącej w protokole, bez ujawniania jej wewnętrznego zachowania.

BPEL4WS, modelując zachowania zarówno wykonywalnych, jak i abstrakcyjnych procesów oraz dostarczając języka dla formalnej specyfikacji procesów biznesowych i protokołów wymiany biznesowej, rozszerza model wymiany serwisów internetowych i umożliwia wsparcie transakcji biznesowych.

W definicji procesu można wyróżnić cztery główne sekcje poprzedzone atrybutami początkowymi, które zostały wyspecyfikowane w tabeli 1:

**sekcja 1:** `<containers>` definiuje zmienne (tzw. kontenery danych), używane przez proces i pozwalające na przechowywanie stanu danych (komunikatów WSDL) (tabela 2),

**sekcja 2:** `<partners>` definiuje strony będące w interakcji z procesem biznesowym w trakcie jego przetwarzania, przy czym każdy partner, będący:

- serwisem, który procesy tylko wywołuje (tzw. partner wywoływany) lub
- serwisem wywołującym procesy (tzw. partner kliencki), lub
- serwisem wywołanym i wywołującym,

jest charakteryzowany przez nazwę roli, którą pełni i typ łącza serwisowego (tabela 3).

**sekcja 3:** opisuje wykonanie żadanego procesu, gdzie reprezentowane są interakcje pomiędzy serwisami (tabela 4). Opis ten może być użyty do reprezentacji zależności pomiędzy serwisami, niezależnie od tego, czy proces biznesowy BPEL4WS jest zdefiniowany dla jednego, czy wielu z tych serwisów.

**sekcja 4:** `<faultHandlers>` definiuje działania wykonywane podczas obsługi błędów (zarówno wewnętrznych, jak i będących wynikiem wywołania serwisu) (tabela 5).

Podstawowa struktura BPEL4WS jest przedstawiona poniżej:

```

<process name = „...”                               </partners>
targetNamespace=“”?
//znak zapytania oznacza opcjonalność
queryLanguage=“”?
abstractProcess=yes/no“?
xmlns=“...”>
<containers>?
<container name = “...”
messageType = „...”?>
<wsdl:message name=“...”?>
...
</wsdl:message>
</container>
</containers>
<partners>?
//przynajmniej jedna rola musi być
//wyspecyfikowana
<partner name = „...”
serviceLinkType=“...”
myRole = „...”?
partnerRole=“...”?>
</partner>
<correlationSets>?
<correlationSet name=“...”
properties=“...”/>
</correlationSets >
<faultHandlers>?
//musi być minimum jeden wyjątek
<catch faultName=“...”?
faultContainer=“...”?>
activity
</catch>
<catchAll>?
activity
</catchAll>
</faultHandlers>
activity
</process>

```

Właściwości komunikatu identyfikujące dane istotne dla protokołu, a zagnieżdżone w komunikacie, mogą być widziane jako dane przezroczyste (istotne dla aspektu publicznego), w przeciwieństwie do danych nieprzezroczystych, które są używane przez wewnętrzne (prywatne) funkcje. Dane przezroczyste mają bezpośredni wpływ na publiczny, czyli ogólnodostępny protokół biznesowy, podczas gdy dane ukryte są znaczące jedynie dla systemów tzw. back-end (w tle) i mają wpływ na protokół biznesowy jedynie przez stworzenie niedeterminizmu w zachowaniu serwisów występujących w protokole biznesowym.

Tabela 1

## Opis atrybutów początkowych

Atrybut	Opis
process name	Nazwa definiowanego procesu (serwisu) internetowego
targetNamespace	Wskazanie docelowej przestrzeni nazw tworzonego pliku WSDL
queryLanguage	Specyfikacja języka zapytań XML, używanego m.in. do definicji (domyślnie XPath1.0)
abstractProcess	Specyfikacja definiowanego procesu jako abstrakcyjny ( <i>yes</i> ) lub wykonywalny ( <i>no</i> – domyślnie)
xmlns	Skojarzenie przestrzeni nazw opisanej atrybutem targetNamespace z URI; w ten sposób parser jest informowany o tym, że element IMG istnieje w innej – niż domyślna – przestrzeni nazw

Tabela 2

## Opis atrybutów sekcji 1 BPEL4WS

Atrybut	Opis
partner name	Nazwa partnera
serviceLinkType	Typ łącza serwisowego, charakteryzującego powiązania pomiędzy dwoma serwisami poprzez definicje tzw. ról, pełnionych przez każdy ze współpracujących serwisów
myRole	Wyszczególnienie roli, dostarczającej specyfikacji portów portType opisywanego procesu
partnerRole	Wyszczególnienie roli partnera

Tabela 3

## Opis atrybutów sekcji 2 BPEL4WS

Atrybut	Opis
container name	Nazwa współdzielonego, globalnie widocznego kontenera do przechowywania komunikatów
messageType	Typ wymienianego komunikatu
wsdl:message name	Nazwa komunikatu (abstrakcyjnej definicji transmitowanych danych – tak aplikacji, jak i protokołu)

Tabela 4

## Opis atrybutów sekcji 3 BPEL4WS

Atrybut	Opis
catch faultName	Unikalna, globalnie dostępna nazwa błędu (wyjątku)
faultContainer	Nazwa współdzielonego, globalnie widocznego kontenera wyjątków, zawierającego dane związane z określoną nazwą faultName błędem
activity	Sekcja przetwarzania, w której może wystąpić dowolny z wymienionych elementów: <receive>, <reply>, <invoke>, <assign>, <throw>, <terminate>, <wait>, <empty>, <sequence>, <switch>, <while>, <pick>, <flow>, <scope>, <compensate>

Tabela 5

## Opis atrybutów sekcji 4 BPEL4WS

Atrybut	Opis
correlationSet name	Nazwa zbioru własności, współdzielonych przez wymieniane komunikaty, a umożliwiających konwersację między partnerami na poziomie aplikacji w danej instancji protokołu biznesowego
properties	Własności („poła” komunikatu) o unikalnej nazwie (takie jak np. cena, ID podatnika), mające określone znaczenie w opisywanym procesie biznesowym

*Przykład:*

Sprzedawca posiada serwis, który dostarcza formularz zakupu i na podstawie szeregu kryteriów reaguje albo akceptacją, albo odrzuceniem zamówienia. Proces podejmowania

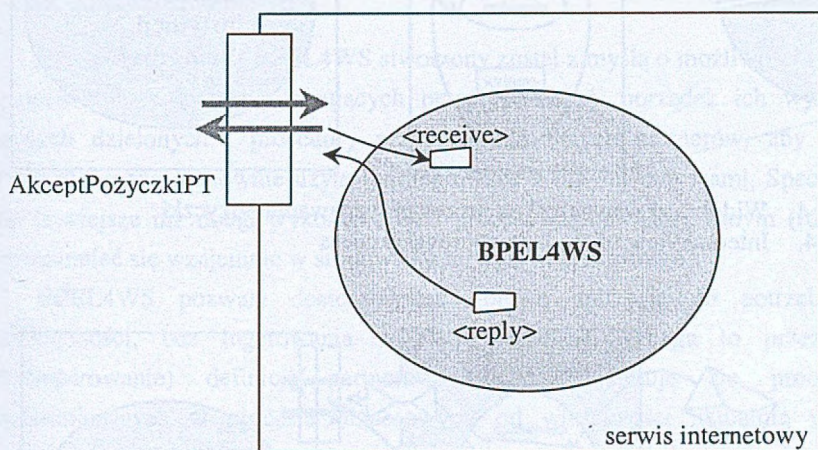


decyzji jest ukryty, ale fakt podjęcia decyzji musi być odzwierciedlony w zewnętrznym procesie jako alternatywa zachowania. Podsumowując: protokół oczekuje na rodzaj przełącznika działania w zachowaniu serwisu sprzedawcy, ale wybór gałęzi jest niedeterministyczny.

### 3. Przykład

Klient wysyła prośbę o pożyczkę i oczekuje na potwierdzenie przydzielenia kredytu.

Z punktu widzenia klienta proces, na który złożą się istniejące serwisy internetowe, będzie konsumował (pobierał) jego aplikację, a następnie wysyłał odpowiedź (rys. 3.).



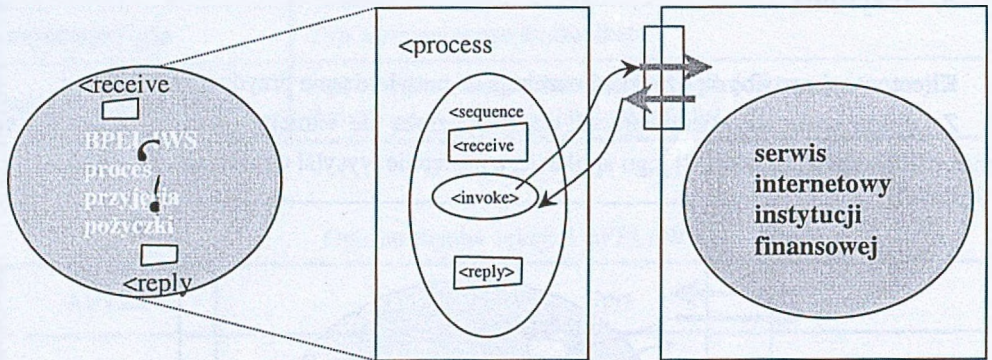
Rys. 3. Zewnętrzny widok na proces przyznawania pożyczki  
Fig. 3. External view of the loan approval process

Można w związku z tym wyróżnić następujące działania:

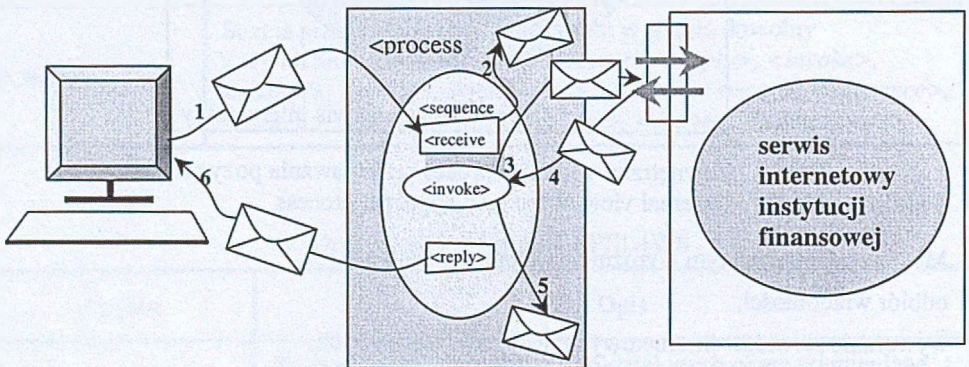
- odbiór wiadomości,
- wywołanie serwisów internetowych instytucji finansowych,
- odpowiedź dla klienta.

Te trzy działania są definiowane w BPEL4WS przez użycie czynności: *receive*, *invoke* i *reply*. W celu dostarczenia procesowi informacji o kolejności i czasie uruchomienia tych podstawowych czynności w BPEL4WS muszą być zdefiniowane wzajemne związki pomiędzy nimi. Dokonywane jest to przez użycie wbudowanych czynności, definiujących ograniczenia w uruchamianiu akcji, które obejmują. W prezentowanym przykładzie czynności mają następować po sobie sekwencyjnie, co można osiągnąć stosując w BPEL4WS operację *sequence* (rys. 4).

Kolejnym etapem jest zdefiniowanie (tzn. opisanie) wprowadzonych serwisów (tu: serwisów instytucji finansowych) i wymienianych komunikatów w standardzie WSDL, a następnie stworzenie procesów i deklaracja partnerów (tu: klient i instytucja finansowa). W następnej fazie następuje zestawienie czynności składających się na proces (wysłanie komunikatu przez klienta, proces odpytuje instytucję finansową czy akceptuje prośbę klienta, odpowiedź zwrotna jest wysyłana do klienta).



Rys. 4. Widok „od wewnątrz” na proces przyznawania pożyczki  
Fig. 4. Internal view of the loan approval process



Rys. 5. Przebieg procesu akceptacji pożyczki  
Fig. 5. Running the loan approval process

Kolejne cyfry umieszczone na rys. 5 obrazują porządek wykonywania poszczególnych kroków, na które składa się:

- wysłanie przez klienta komunikatu do zarządcy procesów (1), w wyniku czego tworzona jest instancja procesu i proces jest uruchamiany (począwszy od operacji <sequence>),

- umieszczenie wiadomości klienta w kontenerze żądań (2), z jednoczesnym wywołaniem (<invoke>) operacji serwisu internetowego instytucji finansowej (3),
- umieszczenie nadchodzącej odpowiedzi w kontenerze zwrotnym (5) i stamtąd przesłanie jej do klienta (6).

Warto przy tym podkreślić, że wiele instancji tego samego procesu może być obsługiwanych równolegle. W BPEL4WS instancje są tworzone niejawnie w momencie przyjęcia komunikatu do serwisu, w związku z czym nie są identyfikowane przez jawny identyfikator, ale przez pola kluczowe zawarte w danych komunikatu.

#### 4. Wnioski

Język wykonawczy BPEL4WS stworzony został z myślą o możliwości takiego wyrażenia procesów biznesowych, opisujących przepływ zadań, porządek ich wykonywania, typy danych dzielonych i procedury przyjmowania innych partnerów, aby mogły być one bezproblemowo ponownie użyte i zintegrowane z innymi procesami. Specyfikacja pozwala na łatwiejsze niż dotąd wykonanie tych procesów, pozwalając innym (różnym) procesom porozumieć się wzajemnie w środowisku serwisu internetowego.

BPEL4WS pozwala dostosowywać gotowe aplikacje do potrzeb i zaistniałych okoliczności, bez ingerowania w samą aplikację. Osiąga to przez rozgraniczenie (odseparowanie) definicji partnerów (którymi zajmuje się proces biznesowy), uczestniczących w procesie biznesowym, od właściwości aktualnie współpracujących partnerów. BPEL4WS dostarcza również mechanizmy do obsługi wyjątków i ewentualnych błędów przetwarzania.

Dzięki rozszerzeniu modelu wymiany serwisów internetowych i umożliwieniu wsparcia transakcji biznesowych przez BPEL4WS wprowadzenie tego standardu spowoduje przyjęcie serwisów internetowych do architektur zorientowanych serwisowo (ang. Serviced Orientated Architecture – SOAs).

W związku z tym, że standard BPEL4WS jest w dalszym ciągu rozwijany, obecna wersja (v1.1) systemu różni się – w pewnym stopniu – od wersji badanej (v1.0). Używana terminologia została częściowo zmieniona (np. kontenery danych są obecnie określane mianem zmiennych, natomiast typ łącza serwisowego jako typ łącza partnerów). W nowym standardzie odseparowano również część wspólną definicji procesów wykonywalnych i procesów biznesowych od właściwości charakterystycznych dla poszczególnych procesów, tworząc tzw. koncepcję rdzenia. Dodatkowo, w nowej wersji systemu przeprowadzono szereg drobnych zmian, wśród których można wyróżnić:

- ograniczenie typów danych, zawartych we właściwościach komunikatu do typów prostych,
- zmianę syntaktyki przy definiowaniu zmiennych w celu wyodrębnienia trzech wzajemnie wykluczających się atrybutów: typu komunikatu (definicja WSDL), typu (typ XML) i elementu (definicja XML),
- zawężenie zakresu czynności przerywania instancji wewnętrznego serwisu (*terminate*) do procesu wykonywalnego,
- usunięcie możliwości niejawnego dostarczenia typu komunikatu WSDL i zastąpienie jej wykorzystaniem prostych typów i elementów globalnych schematu XML.

## 5. Podsumowanie

Podkreślając konieczność opracowania jednolitego standardu łączenia procesów biznesowych w środowisku internetowym można stwierdzić, że w celu zwiększenia swojej efektywności i konkurencyjności w obecnej erze *e-biznesu* firmy zmuszone są do ciągłego monitorowania swoich procesów biznesowych. Dotyczy to zestawienia ze sobą wewnętrznych procesów i procedur, jak również współpracy z dostawcami i klientami. Dzięki obecnym w BPEL4WS zwiększonym możliwościom związanym z usługą dla otwartych środowisk, obejmujących technologie pochodzące od wielu dostawców, użytkownicy otrzymują potrzebną wydajność. Warto wspomnieć, że już w chwili obecnej oferowane są przez firmy informatyczne narzędzia i aplikacje do budowy własnych serwisów internetowych opartych na standardzie organizacji WS-I. Mowa tu przede wszystkim o Oracle9i Application Server, którego przykładowe aplikacje są udostępnione na stronach Oracle Technology Network (OTN) pod adresem: <http://otn.oracle.com/tech/webservices/>.

## LITERATURA

1. „Specifications for Business Transactions and Process Automation”,  
adres strony internetowej: <http://www-106.ibm.com/developerworks/library/ws-bpel/>
2. „Specifications for Business Transactions and Process Automation”,  
adres strony internetowej: <http://dev2dev.bea.com/techtrack/ws-transaction.jsp>
3. „BPEL4WS Concepts”, adres strony internetowej:  
<http://xml.coverpages.org/WS-BusinessProcesses20020809.html>
4. „Business Process Execution Language for Web Services – Main reference page”,  
adres strony internetowej: <http://dev2dev.bea.com/techtrack/BPEL4WS.jsp>

5. BEA News, adres strony internetowej:  
[http://www.bea.com/press/releases/2002/0809\\_transaction\\_specifications.shtml](http://www.bea.com/press/releases/2002/0809_transaction_specifications.shtml)
6. „The XML Specification”,  
adres strony internetowej: <http://www.w3.org/TR/REC-xml>

Recenzent: Prof. dr hab. inż. Leszek Borzemski

Wpłynęło do Redakcji 5 lutego 2003 r.

## Abstract

The new Business Process Execution Language for Web Services, WS-Transaction and WS-Coordination specifications, provide a comprehensive business process automation framework that allows companies to leverage the benefits of the web services architecture, to create and automate business transactions. BPEL4WS supports the specification of a broad spectrum of business processes, from fully executable complex business processes over more simple business protocols to usage constraints of web services. In this article, the main underlying concepts of BPEL4WS are briefly explained. Also, the basic language elements of BPEL4WS are contained in section 2. Simple example of loan requests processing, enclosed in the article, is additionally illustrated by pictures, explaining particular steps of approval process.

## Adres

Aleksandra WERNER: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,  
44-101 Gliwice, Polska, [ola@ares.iinf.polsl.gliwice.pl](mailto:ola@ares.iinf.polsl.gliwice.pl) .