

Architektura zorientowana na usługi

5	Temat:	BPEL	Zadania:								Data:
	Autor:	Sylvia Kaleta	1	2	3	4	5	6	7	8	28 XI 2018
	Autor:	Kamil Wanat	E	E	H	E	M	-	-	-	18:00-19:30

Zadanie 1. Instalacja silnika BPEL i integracja z IDE

To zadanie polegało tylko na przygotowaniu środowiska, poprawne wykonanie zadania poświadczają screeny z następnych zadań oraz fakt ich wykonania, gdyż bez poprawnej konfiguracji BPEL nie byłoby możliwe ich wykonanie.

Zadanie 2. Uruchomienie Hello World pod BPEL

W tym zadaniu należało stworzyć i uruchomić program „Hello World” w BPEL. Zadanie zostało wykonane zgodnie z tutorialiem udostępnionym na stronie openESB, a poprawność wykonania poświadczają screeny:

Z testu wykonanego w openESB:

```

1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/ http://schemas.xmlsoap.org/soap/envelope/">
5   <SOAP-ENV:Body>
6     <m:HelloWorldOperationResponse xmlns:m="HelloWorldComposite" xmlns:msgns="http://j2ee.netbeans.org/wsdl/HelloWorld/src/HelloWorld">
7       <part1>Hello Sylwia </part1>
8     </m:HelloWorldOperationResponse>
9   </SOAP-ENV:Body>
10 </SOAP-ENV:Envelope>

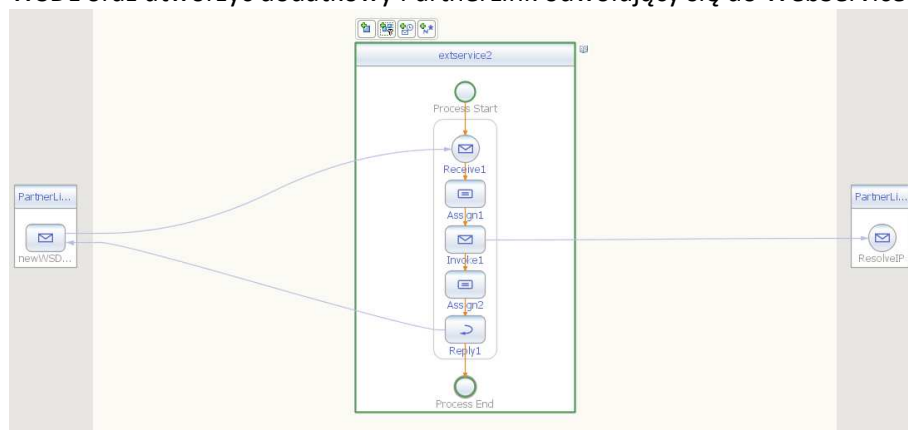
```

Oraz w soapUI:

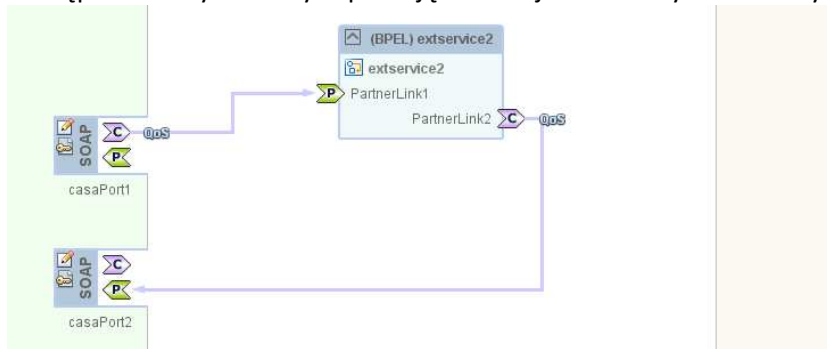
The screenshot shows the soapUI interface with a request and response for a service at `http://localhost:9080/HelloWorldCompositeService1/casaPort1`. The request is a SOAP envelope with a body containing a `HelloWorldOperation` message. The response is a SOAP envelope with a body containing a `HelloWorldOperationResponse` message with the text "Hello Sylwia".

Zadanie 3. Wywołanie innego WebService

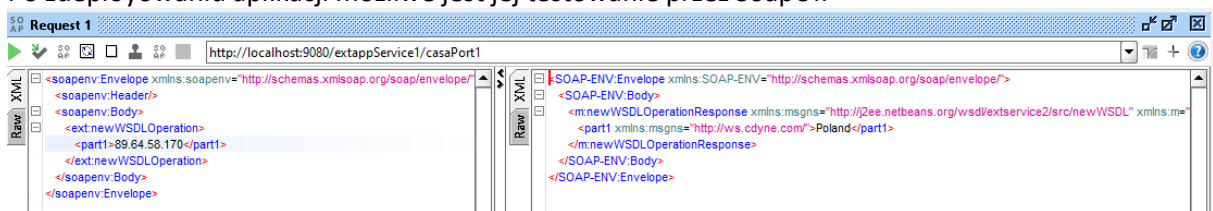
Zadanie polegało na wywołaniu zewnętrznego WebService z poziomu naszego skryptu BPEL. Aby to osiągnąć należało najpierw znaleźć udostępniony publicznie Service oraz plik WSDL. Niestety strona podana przez prowadzącego nie udostępnia już serwisów, co utrudniło nieco pracę. Znalezione zostały natomiast WebService pozwalający na określenie informacji o adresie IP podanym na jego wejście. Aby wywołać zewnętrzny WebService w skrypcie BPEL należy dodać zewnętrzny plik WSDL oraz utworzyć dodatkowy PartnerLink odwołujący się do WebService.



Następnie należy utworzyć aplikację w której Dokładamy dodatkowy port WSDL.



Po zdeployowaniu aplikacji możliwe jest jej testowanie przez SoapUI.



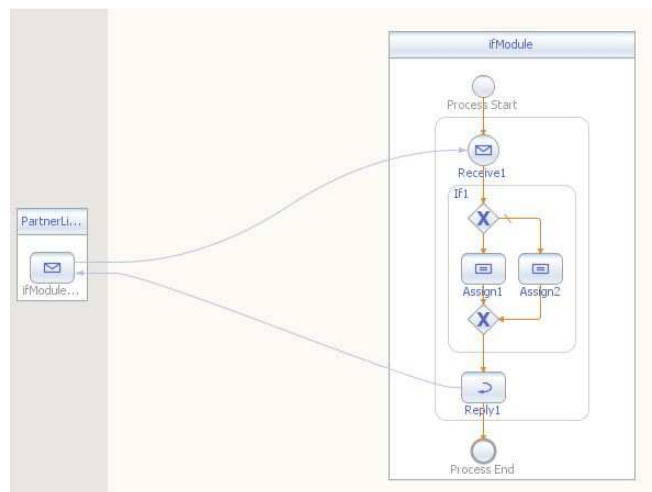
Zadanie 4. Sterowanie przepływem

W zadaniu należało zaprezentować działanie instrukcji if oraz while w BPEL.

Instrukcja if:

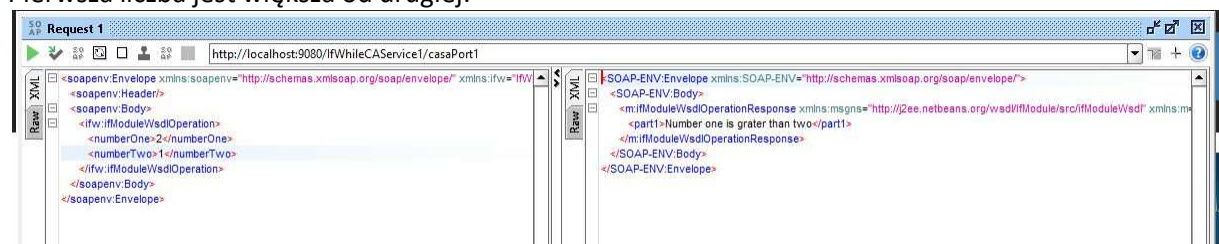
Jest to standardowa instrukcja warunkowa pozwalająca na wykonanie różnych działań w zależności od wartości pewnego wyrażenia logicznego. Stworzony przez nas program przyjmuje 2 liczby i sprawdza czy pierwsza z nich jest większa od drugiej. Wynik porównania jest zwracany jako odpowiedni komunikat tekstowy – komunikaty są ustawiane w instrukcji warunkowej if.

Design programu:

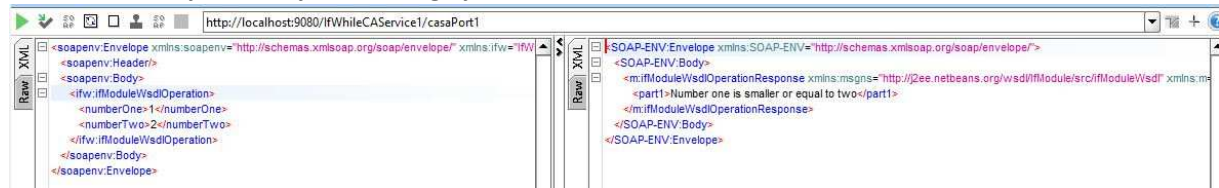


Testy wykonania (SoapUI):

Pierwsza liczba jest większa od drugiej:



Pierwsza liczba jest mniejsza od drugiej:



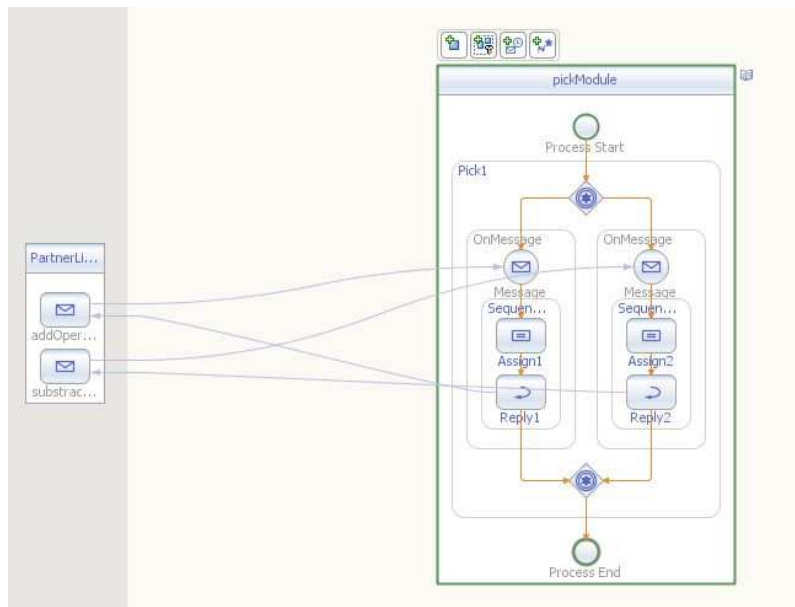
Zadanie 5. Zaawansowane sterowanie przepływem

W tym zadaniu sprawdziliśmy działanie dwóch instrukcji: pick oraz flow.

Instrukcja pick:

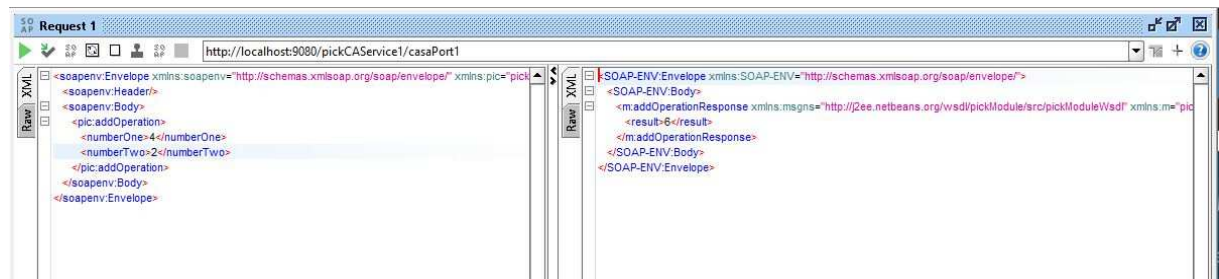
Jest to instrukcja umożliwiająca wykonanie różnych operacji w zależności od otrzymanego żądania. Pozwala na „rozgałęzienie” programu i dzięki niej jeden moduł BPEL może wykonywać różne funkcjonalności wciąż korzystając z jednego partner linku. W naszym programie umożliwiamy dwie operacje określone w pliku WSDL – dodawanie oraz odejmowanie od siebie dwóch liczb. Dzięki instrukcji pick oraz zastosowaniu opcji onMessage reagujemy na odpowiednie wiadomości (addOperation oraz subtractOperation) i zwracamy wyniki odpowiednich operacji.

Design BPEL:

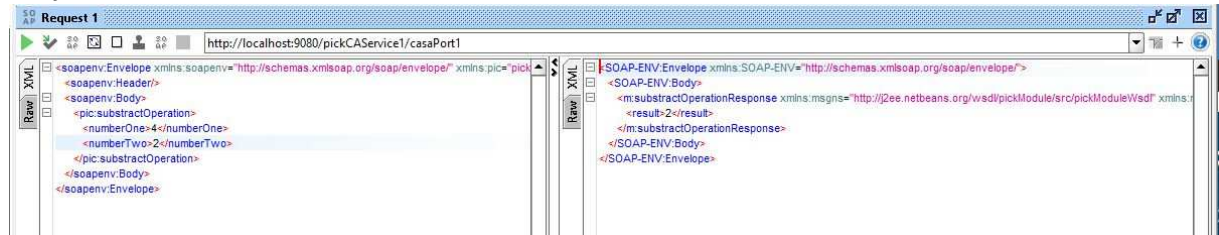


Testy (SoapUI):

Dodawanie:



Odejmovanie:

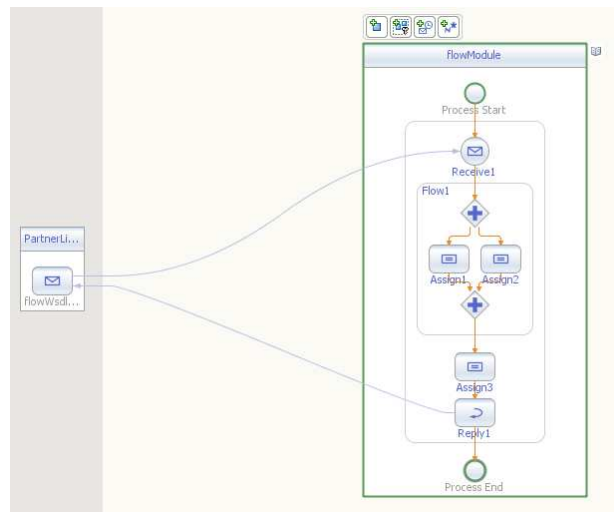


Instrukcja flow:

Instrukcja ta pozwala na wykonanie kilku operacji równolegle, wykorzystywana głównie jest gdy moduł BPEL korzysta z wielu webService'ów, dzięki zastosowaniu tej instrukcji znacząco skracany jest czas oczekiwania na odpowiedzi, gdyż zamiast czekać sekwencyjnie na odpowiedzi kilku serwisów, wysyłamy do nich żądania równolegle, a co za tym idzie również równolegle oczekujemy na odpowiedzi od pytaných serwisów.

Aplikacja stworzona na potrzeby zadania przyjmuje 4 liczby: a, b, c oraz d, a następnie oblicza wartość, którą należy zwrócić według wzoru: $(a+b) \cdot (c+d)$. Obie sumy obliczane są równolegle.

Design w BPEL:



Testy (soapUI):

