



POLITECHNIKA WARSZAWSKA  
Wydział Elektroniki i Technik Informacyjnych  
Instytut Telekomunikacji

## **PRACA DYPLOMOWA INŻYNIERSKA**

Marcin Maciorowski

**Tworzenie narzędzi wspomagających projektowanie BPEL  
w architekturze SOA**

Praca wykonana pod kierunkiem  
dra inż. Andrzeja Ratkowskiego

.....  
Ocena pracy

.....  
Podpis Przewodniczącego Komisji

Warszawa, 2014

## **Życiorys**

Urodziłem się 22 września 1988 roku w Radzynie Podlaskim. W 2004 roku rozpocząłem naukę w I Liceum Ogólnokształcącym w Radzynie Podlaskim, gdzie uczęszczałem do klasy o profilu matematyczno-fizyczno-informatycznym. Po uzyskaniu świadectwa dojrzałości w 2007 roku, rozpocząłem studia na Politechnice Warszawskiej na Wydziale Elektroniki i Technik Informatycznych. W trakcie studiów wybrałem specjalizację Systemy informacyjno-decyzyjne prowadzoną przez Instytut Automatyki i Informatyki Stosowanej.

Marcin Maciorowski

## **Streszczenie**

BPEL.

## **Abstract**

Implementation of .... (Font TNRoman 12 Normal).

This thesis includes the design and testing procedure of .... .

## Spis treści.

1.	Wstęp.....	5
2.	Cel pracy.....	6
3.	Układ pracy.....	6
4.	Opis języka BPEL.....	7
	4.1. Bla bla bla.....	7
	4.2. Bla bla bla.....	7
	4.3. Bla bla bla.....	7
5.	Wtyczka Eclipse BPEL Designer.....	8
	5.1. Opis interfejsu użytkownika.....	8
	5.2. Przykładowy proces BPEL.....	10
6.	Wtyczka generująca instrukcje kopiujące w blokach przepisania danych.....	13
	6.1. Konfiguracja wtyczki (PDE).....	13
	6.2. Transformacja procesu z postaci EMF do postaci grafu.....	13
	6.3. Analizator grafu procesu.....	13
	6.4. Graficzny interfejs użytkownika.....	13
7.	Testy.....	14
8.	Podsumowanie.....	15
	8.1. Napotkane problemy.....	15
	8.2. Możliwości rozwoju.....	15
9.	Bibliografia.....	16
10.	Załączniki.....	17
	10.1. Płyta CD.....	17
	10.2. Instrukcja instalacji wtyczki BPELag (BPEL assign generator).....	17

## 1. Wstęp.

=====

We wstępie znajdzie się ogólne rozwinięcie streszczenia, czego praca dotyczy, z czym czytelnik się zetknie w kolejnych rozdziałach. Opisany układ dokumentu.

=====

## **2. Cel pracy.**

Celem pracy jest napisanie wtyczki do zintegrowanego środowiska programistycznego jakim jest Eclipse, która umożliwi automatyczne uzupełnienie zaprojektowanego procesu BPEL o instrukcje kopiujące dane. Wtyczka ma za zadanie dokonać analizy procesu oraz na podstawie wyników analizy wygenerować instrukcje kopiujące. Analizowany proces BPEL oraz uczestniczące w procesie usługi wykorzystują tę samą konwencję nazewnictwa.

## **3. Układ pracy**

Opis układu pracy. Powstanie na końcu, gdy zostaną napisane już wszystkie rozdziały.

## 4. Opis języka BPEL.

=====

Wstępny opis języka BPEL wprowadzający czytelnika w aspekty języka, których dotyczy niniejsza praca dyplomowa. Na pewno będą to: Struktura procesu, bloki assign, invoke.

=====

### 4.1.Bla bla bla.

=====

Bla bla bla.

=====

### 4.2.Bla bla bla.

=====

Bla bla bla.

=====

### 4.3.Bla bla bla.

=====

Bla bla bla.

=====

## 5. Wtyczka Eclipse BPEL Designer.

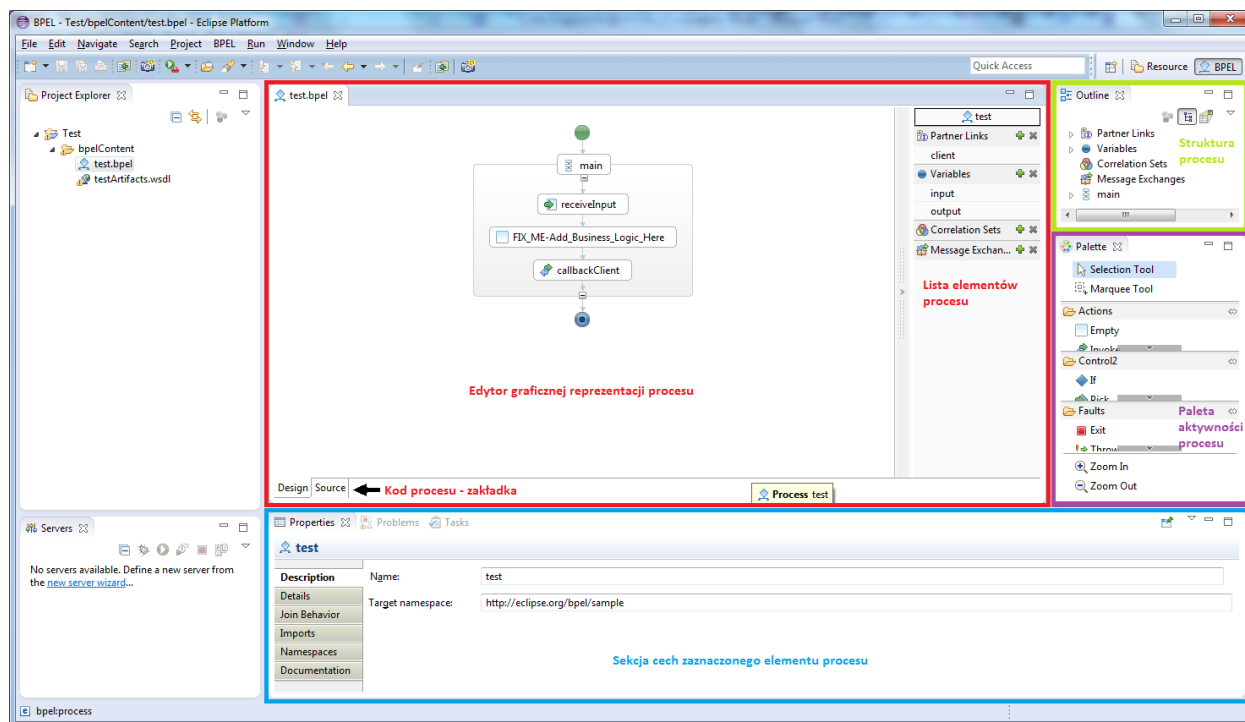
Eclipse BPEL Designer jest wtyczką rozszerzającą zintegrowane środowisko programistyczne (*ang. Integrated Development Environment – IDE*) Eclipse, która wspiera definiowanie, edytowanie, instalację oraz testowanie i debuggowanie procesów WS-BPEL 2.0, czyli języka do definiowania procesów biznesowych opartego o usługi sieciowe, dostarczonego przez konsorcjum OASIS. Główne cechy wtyczki:

- Designer – edytor graficzny (oparty o GEF – Graphical Editing Framework) wprowadzający graficzne oznaczenia elementów procesu BPEL.
- Model – reprezentacja modelu BPEL (specyfikacja WS-BPEL 2.0) reprezentowana przez model oparty o EMF (Eclipse Modelling Framework).
- Validation – operujący na modelu EMF walidator informujący o błędach i ostrzeżeniach dotyczących procesu BPEL, wynikających ze specyfikacji.
- Runtime Framework – zestaw narzędzi umożliwiających instalację oraz wykonanie procesu BPEL.
- Debug – zestaw narzędzi umożliwiający śledzenie kolejnych kroków wykonywanego procesu oraz dostarczających obsługę przerwania wywołania.

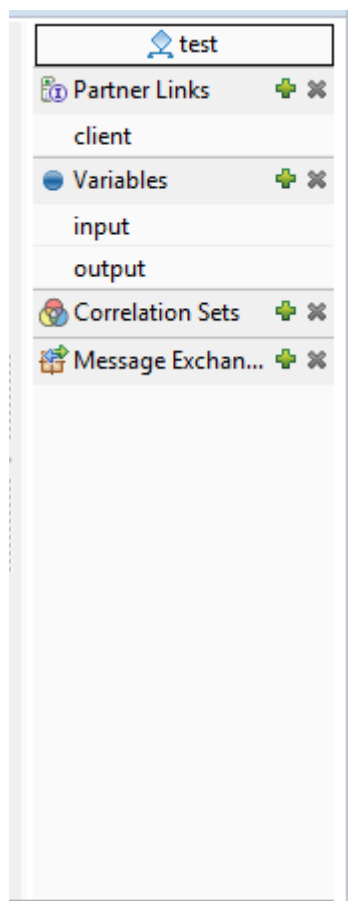
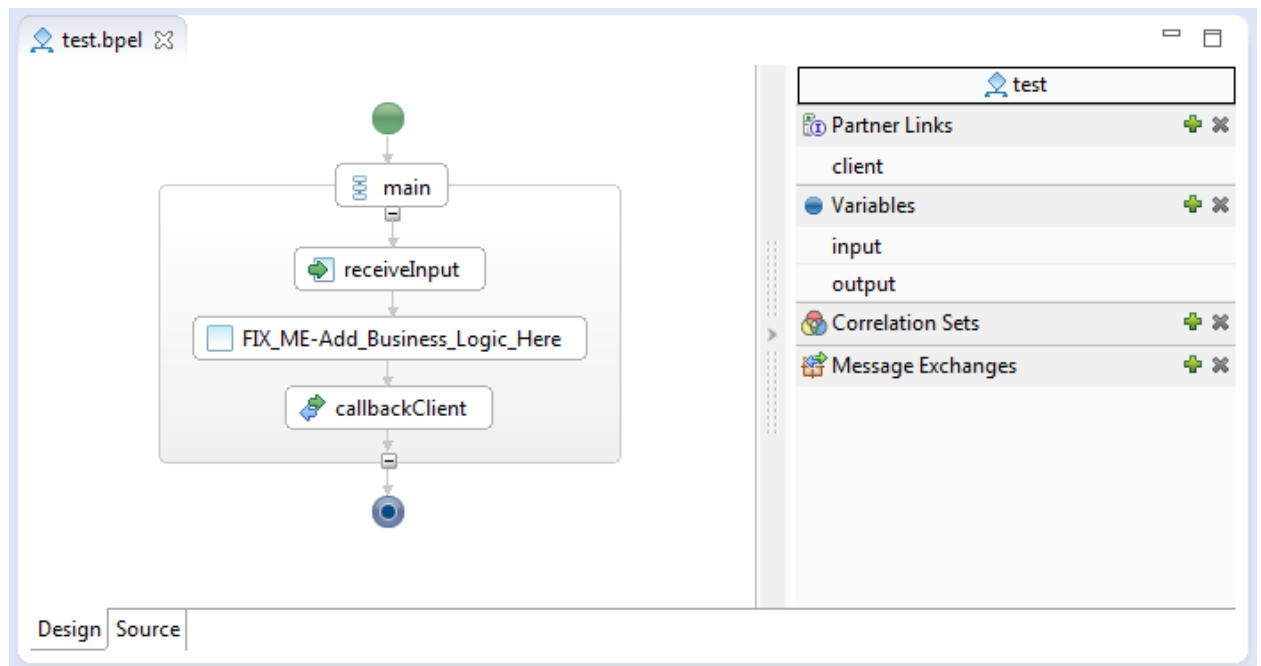
W niniejszej pracy wykorzystywana jest wtyczka Eclipse BPEL Designer w wersji 1.0.3.

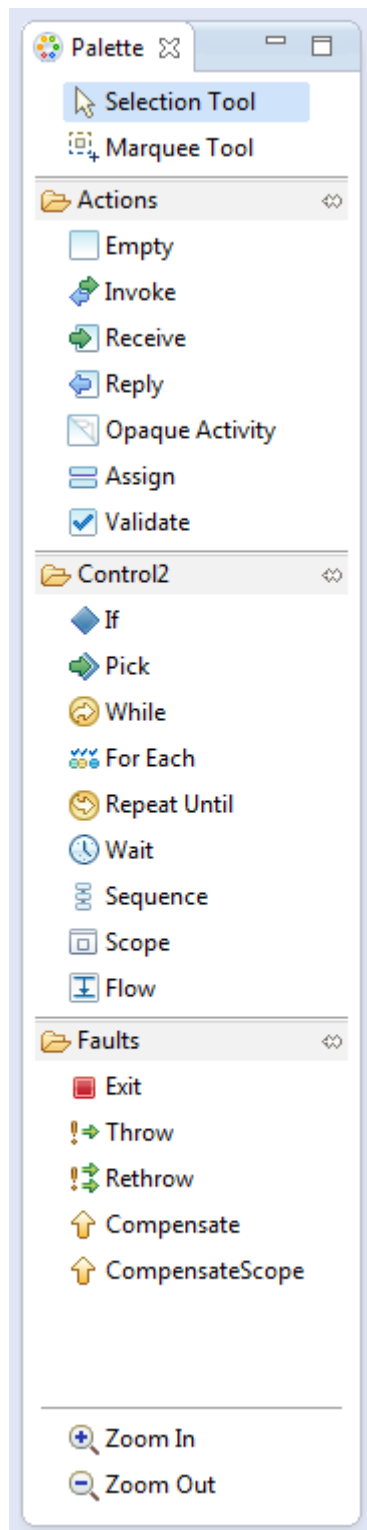
### 5.1.Opis interfejsu użytkownika.

Elementy graficznego interfejsu użytkownika i funkcje poszczególnych elementów.





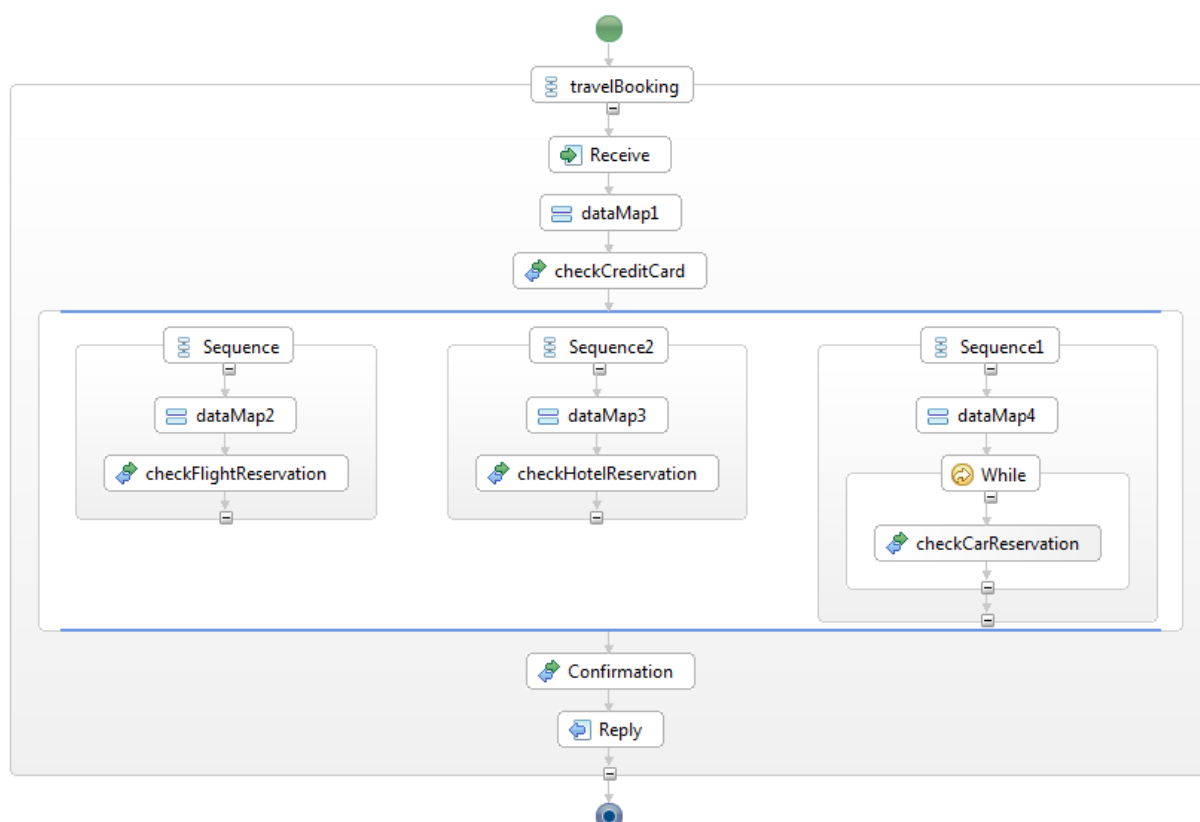




## 5.2. Przykładowy proces BPEL.

Na Rys. 5.1 przedstawiony został przykładowy proces BPEL utworzony przy użyciu Eclipse BPEL Designera, na podstawie procesu rezerwacji wycieczki [2]. W momencie wywołania procesu rezerwacji, zostają mu przekazane informacje dotyczące karty kredytowej, celu oraz okresie podróży. Poprzez wywołanie zewnętrznych usług następuje najpierw sprawdzenie dostępności środków – na karcie kredytowej, następnie równolegle rezerwacja lotu, hotelu oraz samochodu. Po zakończeniu równoległych przebiegów

do konsumenta usługi trafia żądanie potwierdzenia rezerwacji, po którym zostaje wysłana informacja o poprawnym zakończeniu procesu.

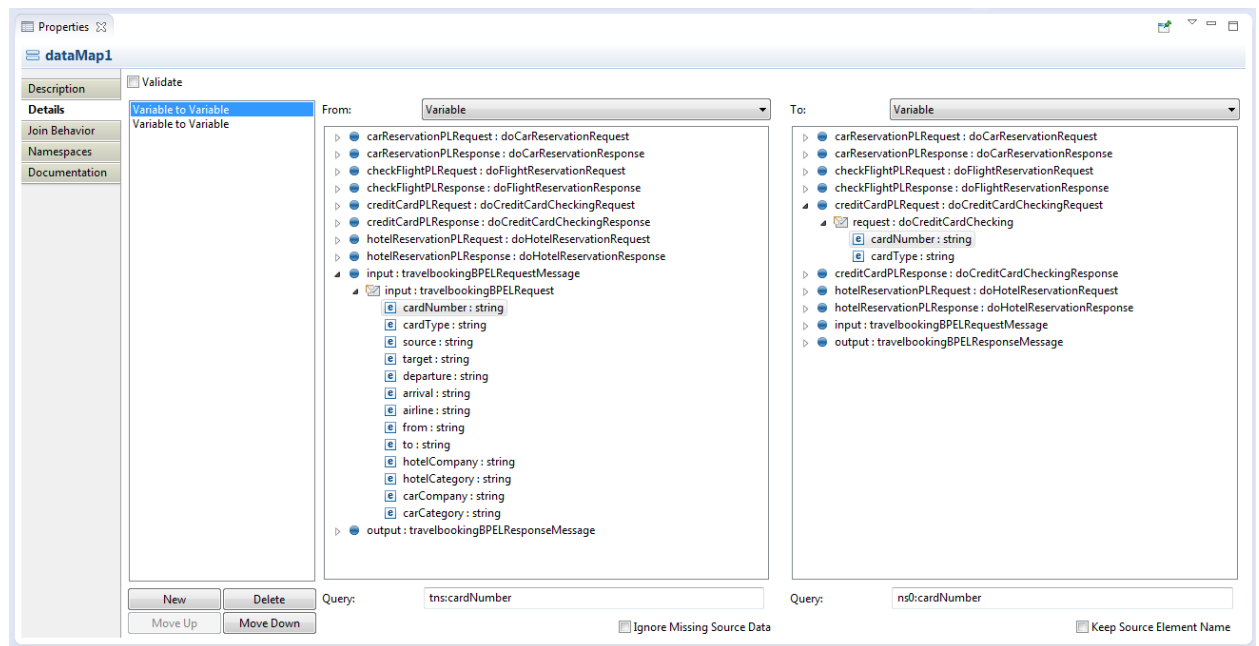


Rys. 5.1 Przykładowy proces BPEL utworzony w Eclipse BPEL Designer – *travelBooking*

W przedstawionym procesie występują trzy bloki przepisania danych (*Assign*):

- *dataMap1* – zawiera instrukcje kopiujące odpowiednie wartości wejściowe procesu do zmiennych będących elementami parametru wywołania usługi *checkCreditCard*.
- *dataMap2* – analogicznie do *dataMap1* dla usługi *checkFlightReservation*.
- *dataMap3* – analogicznie do *dataMap1* dla usługi *checkHotelReservation*.
- *dataMap4* – analogicznie do *dataMap1* dla usługi *checkCarReservation*.

Na Rys. X przedstawiona została konfiguracja instrukcji kopiujących dane na przykładzie bloku przepisywania danych *dataMap1*. Sekcja zawiera listę instrukcji kopiujących opisanych jako para typów (elementu źródłowego oraz elementu docelowego dla instrukcji kopiowania) oraz dwie listy zmiennych o zasięgu nie mniejszym niż aktualnie konfigurowany blok *Assign*. W obu listach *From* oraz *To* zaznaczono zmienne odpowiednio źródłowa i docelowa.



Rys. 3. Sekcja *Properties* bloku przepisywania danych *dataMap1*, zakładka z listą instrukcji kopiujących.

Na Rys. 4. Przedstawiono wygenerowany przez Eclipse BPEL Designer kod w języku BPEL odpowiadający konfiguracji przedstawionej na Rys. 3. dla bloku *dataMap1*.

```

69<bpel:assign validate="no" name="dataMap1">
70  <bpel:copy>
71    <bpel:from part="input" variable="input">
72      <bpel:query queryLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath1.0">
73        <![CDATA[tns:cardNumber]]>
74      </bpel:query>
75    </bpel:from>
76    <bpel:to part="request" variable="creditCardPLRequest">
77      <bpel:query queryLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath1.0">
78        <![CDATA[ns0:cardNumber]]>
79      </bpel:query>
80    </bpel:to>
81  </bpel:copy>
82  <bpel:copy>
83    <bpel:from part="input" variable="input">
84      <bpel:query queryLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath1.0">
85        <![CDATA[tns:cardType]]>
86      </bpel:query>
87    </bpel:from>
88    <bpel:to part="request" variable="creditCardPLRequest">
89      <bpel:query queryLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath1.0">
90        <![CDATA[ns0:cardType]]>
91      </bpel:query>
92    </bpel:to>
93  </bpel:copy>
94</bpel:assign>

```

Rys. 4. Kod BPEL bloku *dataMap1*.

## **6. Wtyczka generująca instrukcje kopiujące w blokach przepisania danych.**

=====

Wprowadzenie do rozwiązania wdrożonego dzięki zrealizowanemu projektowi.

=====

### **6.1.Konfiguracja wtyczki (PDE).**

=====

Opis konfiguracji wtyczki pod plugin Eclipse BPEL Designer – extension points.

=====

### **6.2.Transformacja procesu z postaci EMF do postaci grafu.**

=====

Opis procesu transformacji procesu z postaci EMF do postaci grafu gotowego do analizy.

=====

### **6.3.Analizator grafu procesu.**

=====

Opis procesu analizy grafu procesu będącego wynikiem transformacji z postaci EMF.

=====

### **6.4.Graficzny interfejs użytkownika.**

=====

Opis elementów graficznego interfejsu użytkownika dostarczonego pluginu.

=====

## 7. Testy.

=====				
Opis	przeprowadzonych	testów:	przebieg,	wyniki.
=====				

## 8. Podsumowanie.

=====

Ogólne	podsumowanie	projektu.
--------	--------------	-----------

=====

### 8.1.Napotkane problemy.

=====

Problemy,	z	którymi	stykano	się	podczas	realizacji	projektu.
-----------	---	---------	---------	-----	---------	------------	-----------

=====

### 8.2.Możliwości rozwoju.

=====

Dalsze perspektywy rozwoju projektu, jak można go rozwinąć, co można ulepszyć.
--

=====

## **9. Bibliografia.**

[1] <http://www.eclipse.org/bpel/>

[2] <http://pic.dhe.ibm.com/infocenter/adiehelp/v5r1m1/index.jsp?topic=%2Fcom.ibm.etools.ctc.bpel.doc%2Fsamples%2Ftravelbooking%2FtravelBooking.html>



## **10. Załączniki.**

Bla bla bla.

### **10.1. Płyta CD.**

Bla bla bla.

Dołączona płyta CD zawiera:

- Zestaw testowy przeznaczony dla kodera/dekoder
  - \*.v – pliki poszczególnych modułów kodera/dekoder (Verilog)
  - \*.mif – pliki inicjacyjne pamięci ROM modułów kodera/dekoder
  - Idcelp\_encoder\_tester.qar – archiwum projektu Quartus II zawierające zestaw testowy kodera
  - Idcelp\_decoder\_tester.qar – archiwum projektu Quartus II zawierające zestaw testowy dekodera
  - EncoderUSBReader.java – moduł programowy testera kodera odczytujący dane z portu USB i zapisujący je do pliku (Java)
  - DecoderUSBReader.java – moduł programowy testera dekodera odczytujący dane z portu USB i zapisujący je do pliku (Java)
  - jd2xx.jar – biblioteka procedur komunikacji z układem FT245BM (Java)
- Zestaw wektorów testowych wraz ze stanami wewnętrznymi
- Elektroniczną wersję pracy dyplomowej magisterskiej

### **10.2. Instrukcja instalacji wtyczki BPELag (BPEL assign generator).**

Instrukcja