



## Badanie wydajności wybranych środowisk budowy platform integracyjnych

TOMASZ GÓRSKI

Wojskowa Akademia Techniczna, Wydział Cybernetyki, Instytut Systemów Informatycznych,  
00-908 Warszawa, ul. S. Kaliskiego 2, gorski@wat.edu.pl, tomasz.gorski@rightsolution.pl

**Streszczenie.** W artykule przedstawiono sposób badania wydajności platformy integracyjnej utworzonej przy wykorzystaniu różnych środowisk. W porównaniu środowisk uwzględniono także aspekt czasu potrzebnego na implementację rozwiązań na platformie. Analizie poddano także zakres narzędzi wspierających projektowanie rozwiązań. Artykuł zawiera opis przypadku biznesowego implementowanego na platformie integracyjnej. Przedstawiono w nim także konfigurację środowiska testowego oraz dokonano analizy wyników badań wydajnościowych.

**Słowa kluczowe:** platforma integracyjna, wydajność, projektowanie systemów informatycznych

### 1. Wprowadzenie

Firmy i organizacje posiadają wiele aplikacji wspomagających ich działalność. Szczególnie w korporacjach istotna jest szybkość wprowadzania nowych usług zapewniających skuteczne działanie na szybko zmieniającym się rynku. Przez wiele lat wprowadzanie nowych usług lub udoskonalanie istniejących wiązało się z wprowadzeniem nowych systemów, często wytworzonych w technologiach nowszych niż już istniejące systemy informatyczne. Takie podejście zaowocowało powstaniem wielu aplikacji działających w obrębie tej samej organizacji, posiadających podobną funkcjonalność oraz przechowujących podobne lub wręcz te same dane. Sytuacja taka podnosi koszty utrzymania systemów informatycznych i komplikuje realizację procesów biznesowych przebiegających przez wiele działów tej samej organizacji. Czynniki te spowodowały zapotrzebowanie na technologie i rozwiązania umożliwiające integrację istniejących systemów i aplikacji, a także na ponowne wykorzystanie już utworzonych funkcjonalności [3, 11, 16, 17, 18].

Odrębnym problemem jest sposób zarządzania dużą liczbą powiązanych funkcjonalnie systemów informatycznych będący częścią architektury korporacyjnej organizacji [8]. Powstało wiele koncepcji integracji systemów informatycznych, a obecnie stosowana jest architektura zorientowana na usługi (ang. *Service-Oriented Architecture* — SOA) [6, 15]. Centralnym elementem w tego typu rozwiązaniach jest szyna usług (ang. *Enterprise Service Bus* — ESB) [4, 14]. Na rynku dostępnych jest wiele środowisk umożliwiających budowę szyny usług, która jest częścią platformy integracyjnej, zgodnej z podejściem SOA [5, 12, 13]. Rozwiązania te różnią się od siebie nie tylko kosztem zakupu, lecz także łatwością rozwijania rozwiązania integracyjnego czy potrzebami sprzętowymi oraz systemowymi. Istotne staje się dobranie środowiska budowy platformy integracyjnej dostosowanego do potrzeb i wymagań organizacji z punktu widzenia budowy platformy integracyjnej i jej systemów informatycznych. Kluczowe jest także zastosowanie podejście do projektowania platformy integracyjnej [1, 7].

## 2. Parametry platform integracyjnych

Integracja systemów spowodowana jest przeważnie przez konieczność realizacji wymagań biznesowych. Podstawowym założeniem jest ograniczenie wydatków na tworzenie nowego systemu, gdy część jego funkcjonalności realizują już systemy istniejące. Decydując się na wybór środowiska budowy platformy, rozpatrywane są dwa główne aspekty:

- parametry techniczne środowiska budowy platformy oraz jego wydajność i łatwość w zarządzaniu,
- cena środowiska budowy platformy (licencja i wymagany sprzęt) oraz koszt jego utrzymania.

Najważniejszym elementem przy wyborze środowiska budowy platformy integracyjnej jest oszacowanie obecnych oraz przewidywanych przyszłych wymagań. Podstawowe z nich to:

- przepustowość platformy  $P_p$  — liczba możliwych do obsłużenia komunikatów na platformie, przesyłanych między integrowanymi systemami w jednostce czasu,
- liczba integrowanych systemów  $N$  — liczba połączonych aplikacji wspólną warstwą, umożliwiającą im komunikację,
- liczba usług wystawionych z każdego integrowanego systemu  $u_i$  — liczba usług z każdego z integrowanych systemów włączana na platformę integracyjną,  $i = \overline{1, N}$ ,
- utrzymanie i eksploatacja — łatwość zarządzania usługami, bezawaryjność pracy systemu oraz dostępność mechanizmów zapobiegania awarii.

- konserwacja oraz utrzymywanie kodu realizującego integrację na platformie — czas implementacji rozwiązania na platformie, koszt rozbudowy systemu, utrzymanie rozwijanego kodu.

Jedną z kluczowych własności platform integracyjnych jest ich wydajność. Przed dokonaniem wyboru środowiska budowy platformy integracyjnej należy zastanowić się nad następującymi kwestiami:

- szacowany ruch na platformie  $R_p$  — liczba komunikatów wymienianych w jednostce czasu między każdym z integrowanych systemów,

$$R_p = \sum_i^N \sum_j^N l_{ij}, \quad (1)$$

gdzie:  $l_{ij}$  — liczba komunikatów wysyłanych z systemu  $i$  do systemu  $j$ ,

- wymagania sprzętowe — zasoby potrzebne każdemu ze środowisk do obsługi funkcjonowania platformy.

Pomocny przy dokonaniu trafnego wyboru może okazać się jasno zdefiniowany zbiór miar wydajności platformy integracyjnej. Do takich miar możemy zaliczyć:

- ilość pamięci RAM alokowaną przez platformę integracyjną,
- ilość pamięci stronicowanej wykorzystywaną przez platformę integracyjną,
- procentowe obciążenie procesora(ów),
- liczbę obsługiwanych transakcji w jednostce czasu,
- liczbę przetwarzanych komunikatów na platformie w jednostce czasu.

Wymagania sprzętowe możemy opisać następującymi wskaźnikami:

- minimalna potrzebna przestrzeń dyskowa,
- minimalna ilość pamięci RAM przydzielonej dla platformy,
- maksymalna ilość pamięci RAM przydzielonej dla platformy,
- minimalna liczba procesorów potrzebna do uruchomienia platformy,
- maksymalna liczba procesorów, na której można uruchomić platformę.

### 3. Wybór platform

W chwili obecnej istnieje na rynku wiele środowisk budowy platform integracyjnych. Do porównania zostały wybrane dwa z nich:

- webMethods — produkt dostarczany przez Software AG,
- Mule — produkt dostarczany przez MuleSoft.

Wybrane środowiska różnią się od siebie znacznie pod względem:

- wymagań sprzętowych — pierwsza z nich wymaga zdecydowanie większych zasobów sprzętowych,
- ceny za licencję — pierwszy z produktów jest droгим rozwiązaniem, a Mule jest rozwiązaniem typu OpenSource,
- łatwości instalacji oraz konfiguracji,

- projektowania i wdrażania procesów biznesowych.
- Wybrane środowiska, choć tak różne, umożliwiają implementację rozpatrywanego przypadku biznesowego na platformie integracyjnej.

#### 4. Przypadek biznesowy do wdrożenia na platformach

Podstawowym zadaniem platformy integracyjnej jest zapewnienie komunikacji między systemami do niej włączonymi. Realizacja tego założenia przenosi ciężar transformacji komunikatów oraz wykonywania podstawowych operacji na warstwę integracyjną. Platforma integracyjna potrafi zapewnić komunikację zarówno synchroniczną jak i asynchroniczną. Dalej pokazano projekt i zagadnienia implementacyjne integracji dla wybranych środowisk budowy platform integracyjnych [19].

W rozpatrywanym przypadku biznesowym przyjęto, że integrować swoje działania mają trzy systemy informatyczne:

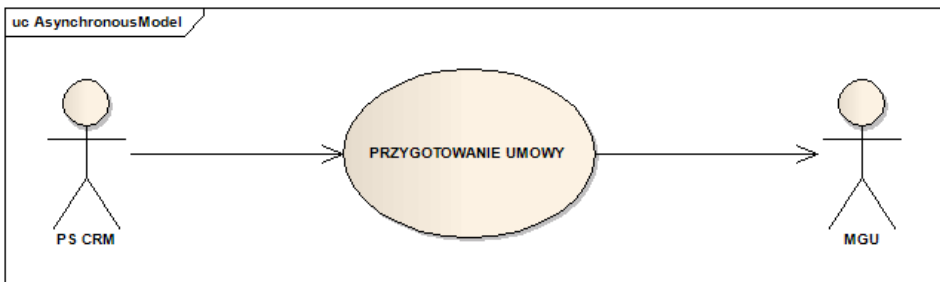
- PS CRM — System PeopleSoft CRM,
- CBD — Centralna baza dłużników,
- MGU — Moduł generatora umów.

Integracja rozpatrywanych systemów jest wymagana ze względu na realizację następujących przypadków użycia systemów informatycznych:

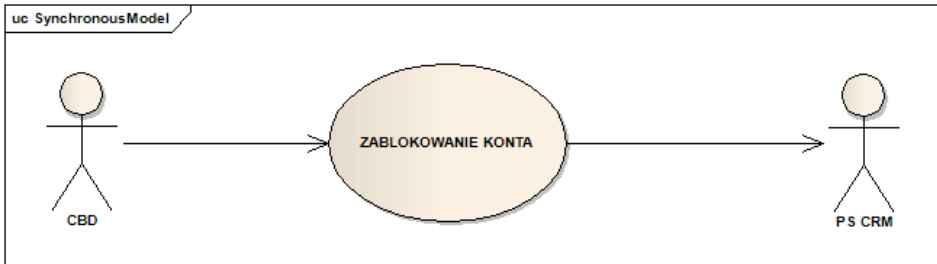
- **przygotowanie umowy** — przypadek ten realizuje funkcjonalność przygotowania umowy dla klienta, która ma być wysłana do podpisania,
- zablokowanie konta.

W realizacji przypadku użycia „Przygotowanie umowy” zachodzi potrzeba komunikacji asynchronicznej pomiędzy systemem PS CRM a platformą integracyjną oraz asynchronicznej pomiędzy platformą integracyjną i systemem MGU (rys. 1).

W realizacji przypadku użycia „Zablokowanie konta” zachodzi potrzeba komunikacji synchronicznej systemu CBD z platformą integracyjną oraz asynchronicznej między platformą integracyjną a systemem PS CRM (rys. 2).



Rys. 1. Komunikacja asynchroniczna między systemami PS CRM oraz MGU



Rys. 2. Komunikacja synchroniczna między systemami CBD oraz PS CRM

## 5. Opis przypadków użycia oraz wysyłanych komunikatów

Przypadek użycia „Przygotowanie umowy” wymaga realizacji komunikacji asynchronicznej między systemami PS CRM a MGU. Komunikat wysyłany z systemu PS CRM zawiera żądanie wygenerowania umowy dla klienta. Komunikat zawiera pola, które jednoznacznie identyfikują klienta oraz typ umowy (rys. 3).

```
<?xml version="1.0" ?>
<WCY_GEN_CONTRACT_REQ_MSG>
  <requestId>${requestId}</requestId>
  <clientId>${clientId}</clientId>
  <productId>${productId}</productId>
  <contractType>R342</contractType>
  <requestSystem>PS_CRM</requestSystem>
  <destinationSystem>MGU</destinationSystem>
  <orderSubType>CREATION</orderSubType>
  <orderStatus>COMPLETED</orderStatus>
  <orderDate>2010-03-25 00:00:00</orderDate>
</WCY_GEN_CONTRACT_REQ_MSG>
```

Rys. 3. Postać komunikatu wysłanego z systemu PS CRM

W tabeli 1 przedstawiono znaczenie przesyłanych pól w komunikacie.

Komunikat wysyłany z platformy integracyjnej do systemu MGU zawiera żądanie wygenerowania umowy dla klienta. Komunikat ten jest zgodny z definicją interfejsu systemu MGU. Komunikat ten został przygotowany na podstawie komunikatu wejściowego z systemu PS CRM oraz logiki zawartej w implementacji usług.

W tabeli 2 przedstawiono znaczenie przesyłanych pól w komunikacie.

TABELA 1

## Opis pól komunikatu z PS CRM

Nr	Nazwa pola	Biznesowy opis pola
1	WCY_GEN_CONTRACT_REQ_MSG	Root element — zlecenie
2	requestId	Id zlecenia w systemie PS CRM
3	clientId	Id klienta zarejestrowanego w systemie PS CRM
4	contractType	Typ umowy w PS CRM
5	requestSystem	Nazwa systemu zlecającego
6	destinationSystem	Nazwa systemu wykonującego zlecenie
7	orderSubType	Typ zlecenia
8	orderStatus	Status
9	orderDate	Data wysłania zlecenia do realizacji

TABELA 2

## Opis biznesowy komunikatu z platformy do MGU

Nr	Nazwa pola	Biznesowy opis pola
1	WCY_WM_GEN_CONTRACT_REQ_MSG	Root element — zlecenie
2	requestId	Id zlecenia w systemie PS CRM
3	clientId	Id klienta zarejestrowanego w systemie PS CRM
4	templateId	Numer typu dokumentu systemu MGU
4	contractType	Typ umowy w PS CRM
5	requestSystem	Nazwa systemu wysyłającego komunikat
6	orderSubType	Typ zlecenia
7	orderStatus	Status
8	orderDate	Data wysłania zlecenia do realizacji
9	Result	Root element — wynik przetwarzania na warstwie EAI
10	errorCode	Kod błędu — „0” w przypadku poprawnego przetwarzania
11	errorDescription	Opis błędu — „” w przypadku poprawnego przetwarzania

Przypadek użycia „Zablokowanie konta” wymaga realizacji komunikacji synchronicznej między systemami CBD a PS CRM. Systemem inicjującym przepływ na platformie jest system CBD, który zleca zablokowanie konta klienta z powodu zaciągniętych długów. Komunikat wysyłany z CBD do systemu PS CRM, zawiera żądanie zablokowania konta klienta w systemie PS CRM. System CBD wywołuje

tą usługę przy użyciu protokołu http i dostaje zwrótną odpowiedź o wstawieniu stosownego komunikatu do kolejki MQSeries systemu PS CRM.

W tabeli 3 przedstawiono znaczenie przesyłanych pól w komunikacie.

TABELA 3

Opis pól komunikatu z CBD

Nr	Nazwa pola	Biznesowy opis pola
1	customerId	Numer klienta w Centralnej Bazie Dłużników
2	serviceName	Nazwa usługi
3	serviceId	Szczegółowy kod usługi klienta
4	orderReason	Przyczyna wysłania zgłoszenia o zablokowaniu konta

Komunikat wysyłany z platformy integracyjnej do systemu PS CRM zawiera żądanie zablokowania konta klienta w systemie PS CRM. Komunikat zawiera dane o typie usługi, kliencie oraz okresie, na jaki ma być zablokowana usługa.

W tabeli 4 przedstawiono znaczenie przesyłanych pól w komunikacie.

TABELA 4

Opis pól komunikatu z platformy do PS CRM

Nr	Nazwa pola	Biznesowy opis pola
1	WCY_WM_CUSTOMER_SERVICE_REQ_MSG	Root element — zlecenie
2	lockTime	Okres, na jaki ma być zablokowane konto
3	orderDate	Data wysłania zlecenia
4	customerId	Id klienta w systemie PS CRM
5	serviceId	Id usługi, która ma być zablokowana
6	orderReason	Przyczyna wysłania zlecenia o zablokowaniu klienta
7	serviceName	Nazwa blokowanej usługi
8	businessErrorCode	Biznesowy kod błędu systemu CBD

Role platformy w realizacji obydwu przypadków użycia jest odpowiednie sterowanie przepływem komunikatów oraz dostosowanie ich struktury tak, aby docelowy system mógł go odebrać i poprawnie przetworzyć.

## 6. Opis realizacji przypadków użycia na platformie webMethods

Realizacja przypadków użycia na platformie webMethods obejmowała następujące czynności:

- utworzenie pakietów webMethods dla kanałów i adapterów systemów zewnętrznych oraz warstwy enterprise,
- utworzenie oraz przeniesienie na serwer webMethods pliku .bindings zawierającego definicję połączenia oraz aliasów kolejek MQ Series,
- uruchomienie instancji IntegrationServera oraz konfigurację brokera webMethods,
- Zdefiniowanie połączeń do kolejek MQ łączonych systemów,
- utworzenie notyfikatorów JMS dla kanałów, które pobierają komunikaty z kolejki do dalszego przetwarzania,
- utworzenie serwisów realizujących odbieranie komunikatów z kanałów oraz przeniesienie realizacji procesu na warstwę enterprise,
- utworzenie serwisów odpowiedzialnych za właściwe procesowanie komunikatów,
- utworzenie serwisów adaptera systemów zewnętrznych.

W tabeli 5 przedstawiono zestawienie powstałych komponentów webMethods.

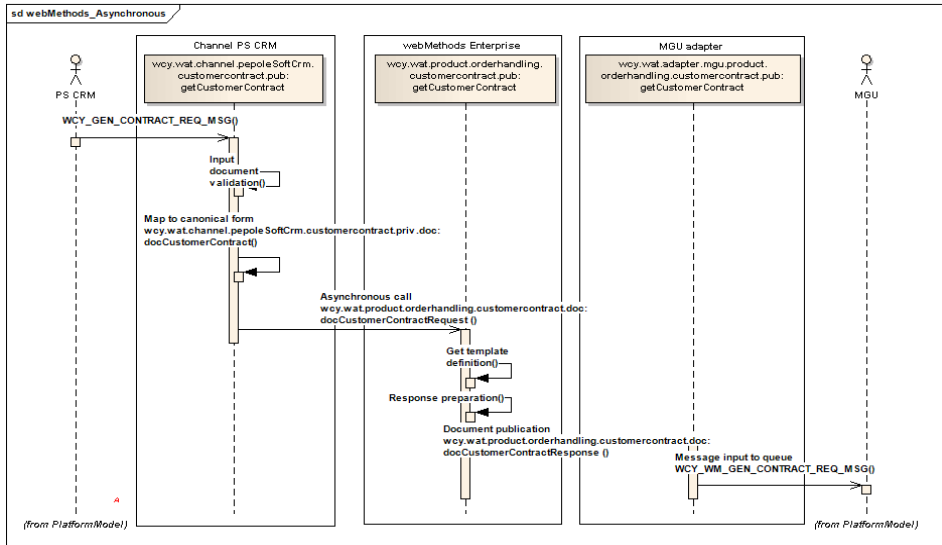
TABELA 5

Lista komponentów realizujących integrację między systemami

Nr	Nazwa komponentu	Opis realizowanej funkcjonalności
1	WcyWatChannelPeopleSoftCrm	Pakiet webMethods zawierający serwisy odpowiedzialne za odbieranie komunikatów z kanału systemu PS CRM
2	WcyWatEnterprise	Pakiet webMethods zawierający całą logikę warstwy enterprise
3	WcyWatAdapterMgu	Pakiet webMethods zawierający serwisy odpowiedzialne za wysyłanie komunikatu do systemu MGU
4	WcyWatCommonDoc	Pakiet webMethods zawierający definicję podstawowych typów dokumentów
5	WcyWatConfig	Pakiet, w którym znajdują się zdefiniowane połączenia jdbc oraz jms
6	WcyWatChannelCbd	Pakiet webMethods zawierający serwisy warstwy kanału

Na rysunku 4 przedstawiony został diagram sekwencji pokazujący realizację przypadku użycia „Przygotowanie umowy”.

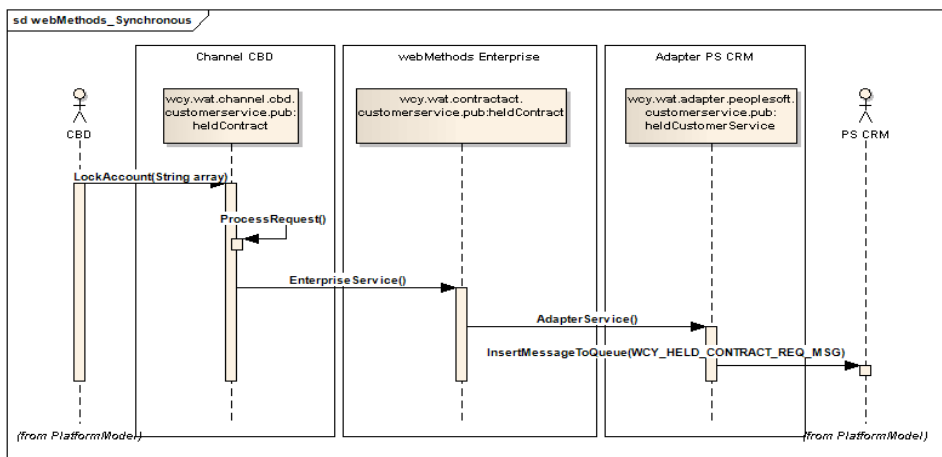




Rys. 4. Realizacja przypadku „Przygotowanie umowy” na platformie webMethods

Poszczególne czynności wykonywane na platformie są realizowane przy użyciu usług stworzonych na potrzeby projektu lub dostarczanych przez platformę.

Na rysunku 5 przedstawiony został diagram sekwencji pokazujący realizację przypadku użycia „Zablokowanie konta”.



Rys. 5. Realizacja przypadku „Zablokowanie konta” na platformie webMethods

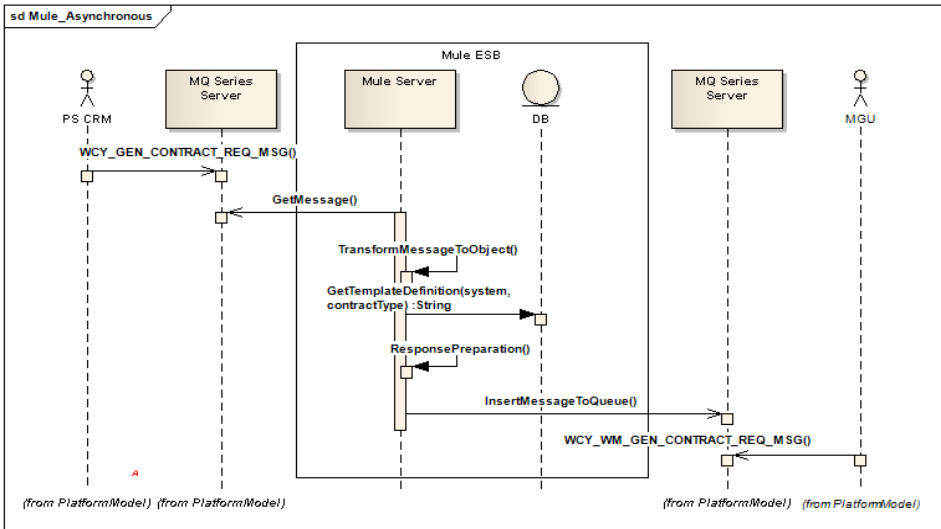
## 7. Opis realizacji przypadków użycia na platformie Mule

W realizacji przypadków użycia na platformie Mule kluczowym elementem było zdefiniowanie konektora do kolejki MQ Series IBM. Definicja konektora określa wszystkie niezbędne ustawienia do połączenia się z zarządcą kolejek:

- `wmq.host` — IP serwera, gdzie jest zarządca kolejek,
- `wmq.port` — numer portu, na którym nasłuchuje zarządca kolejek,
- `wmq.queue.manager` — nazwa zarządcy kolejek,
- `wmq.username` — nazwa użytkownika na serwerze z uprawnieniami `rwx` dla katalogów, gdzie jest zainstalowany serwer kolejek MQ Series,
- `wmq.password` — hasło na serwerze.

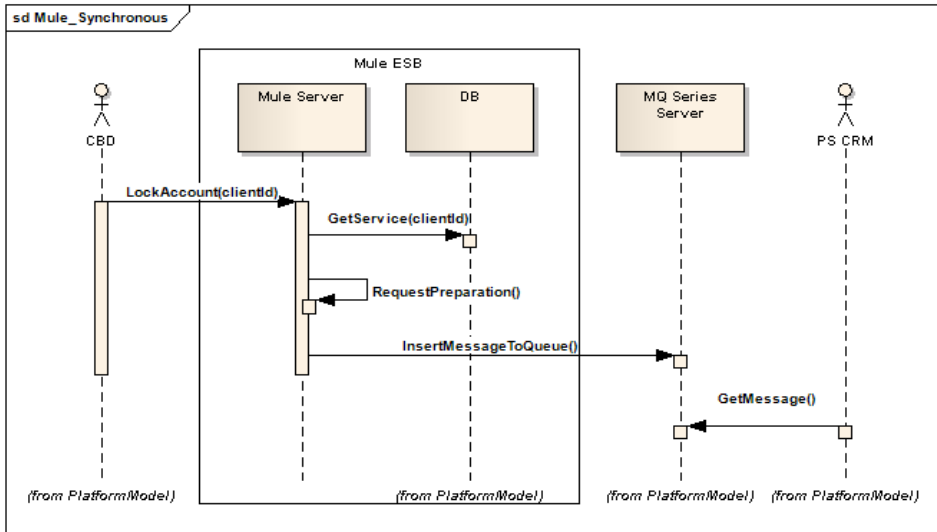
Po zdefiniowaniu w taki sposób konektora, możliwy jest zarówno odczyt, jak i zapis komunikatów do kolejki MQ.

Na rysunku 6 przedstawiono diagram sekwencji pokazujący realizację przypadku użycia „Przygotowanie umowy”. Poszczególne czynności wykonywane na platformie są realizowane przy użyciu wbudowanych mechanizmów platformy.



Rys. 6. Realizacja przypadku użycia „Przygotowanie umowy” na platformie Mule

Na rysunku 7 przedstawiono diagram sekwencji pokazujący realizację przypadku użycia „Zablokowanie konta”. Poszczególne czynności wykonywane na platformie realizowane są przy użyciu usług stworzonych w ramach implementacji lub dostarczanych przez platformę. Wszystkie usługi realizujące przepływ są wywoływane synchronicznie.



Rys. 7. Realizacja przypadku użycia „Zablokowanie konta” na platformie Mule

Cechą implementacji na platformie Mule jest prostota w tworzeniu kompletnych przepływów (ang. *end-to-end*) przy pomocy plików konfiguracyjnych. Cała konfiguracja jest wyniesiona do jednego pliku (*config.xml*), w którym może być zdefiniowanych wiele modeli.

## 8. Analiza implementacji oraz utrzymania platformy

Obydwa środowiska budowy platform integracyjnych zostały zainstalowane na systemie operacyjnym Linux, który dostarcza narzędzi i mechanizmów do zarządzania procesami oraz monitorowania wydajności platformy.

Mule jest lekkim środowiskiem do łączenia systemów, zapewniając im możliwość komunikacji. Proces instalacji środowiska jest prosty, przebiega bardzo szybko (3 godziny) i nie wymaga zaawansowanej wiedzy. Zaletą tego środowiska jest jego przenaszalność, zagwarantowana przez technologię Java. W formie wtyczki do Eclipse dostępne jest narzędzie umożliwiające stworzenie struktury projektu mule. W pliku *config.xml* zapisywana jest zarówno logika przetwarzania jak i konfiguracja. Środowisko to posiada także dedykowane narzędzia do mapowania danych. Brakuje narzędzi do zarządzania liczbą wątków, a także liczbą połączeń do bazy na definiowanym połączeniu. Sporym brakiem w Mule jest brak narzędzi do projektowania procesów biznesowych w języku BPMN (ang. *Business Process Modeling Notation*).

Rozpatrywany w artykule przypadek biznesowy został zrealizowany w ciągu 8 godzin pracy. Projektowanie w Mule pociąga za sobą potrzebę definiowania

modeli, które tworzą logicznie odrębny obszar wykorzystywanych komponentów. W związku z tym nie można istniejącego kodu użyć ponownie do projektowania kolejnych przepływów na platformie. Takie rozwiązanie jest ukierunkowane na budowę rozwiązań z niewielką liczbą prostych przepływów. Natomiast przy dużej liczbie przepływów części kodu będą się powtarzały, zamiast być wyniesione do stosownych bibliotek. Biorąc pod uwagę sposób implementacji, platforma zbudowana z Mule wydaje się trudna w projektowaniu i zarządzaniu przy znacznej liczbie przepływów.

Do najmocniejszych stron webMethods zalicza się serwer usług, zakres wspierających narzędzi oraz jakość gwarantowanych usług. W pakiecie instalacyjnym znajdują się: IntegrationServer, Broker Server oraz My webMethods Portal. Są to podstawowe narzędzia, które połączone tworzą logiczną całość zdolną do tworzenia, uruchamiania oraz zarządzania usługami. Środowisko webMethods ma spore wymagania pod względem zasobów dyskowych (1,5 GB na katalog instalacyjny). Instalacja tego środowiska jest dużo bardziej skomplikowana i czasochłonna (czas trwania instalacji — 2 dni). Dodatkowo potrzebna jest przestrzeń dla logów zapisywanych przez serwer, broker oraz portal.

Środowisko webMethods posiada Broker komunikatów zapewniający mechanizm równoważenia obciążenia (ang. *load-balancing*). Broker jest podstawowym elementem w trójwarstwowej architekturze (kanał–enterprise–adapter), który występuje między każdą z warstw.

Rozpatrywany w artykule przypadek biznesowy został zrealizowany w ciągu 24 godzin pracy. Czas ten jest trzy razy dłuższy niż w przypadku platformy Mule. Konieczne okazało się zaplanowanie rozkładu usług w różnych pakietach, stworzenie wszystkich adapterów, definicji dokumentów oraz serwisów realizujących całą logikę. Natomiast stworzone elementy mogą być z łatwością zarządzane oraz ponownie użyte przy projektach kolejnych przepływów, a implementacja wykorzystująca przyjęty wzorzec pozwoli w łatwy sposób zmieniać logikę serwisów. Dużą zaletą platformy jest dostępność narzędzi do projektowania oraz wgrzywania definicji procesów. Daje to możliwość stworzenia zarządzanej warstwy procesów z przypisanymi usługami z systemów informatycznych.

## 9. Parametry techniczne środowiska

Każde z wybranych środowisk zostało zainstalowane na tym samym środowisku testowym. Serwer sprzętowy, na którym zainstalowano obydwa środowiska, posiada następujące parametry:

- system operacyjny — Ubuntu 8.04.3 LTS,
- pamięć RAM — 4 GB, 4 GB pamięci stronicowania (SWAP),
- dysk twardy — 160 GB,

— procesor — 8 × Intel(R) Pentium(R) D CPU 2.80 GHz.

Ponadto, na serwerze sprzętowym zainstalowano następujące oprogramowanie:

- baza danych: Oracle Database 10g Release 10.2.0.1.0 — 64 bit,
- Java: JRE 1.5, JDK 1.5.0\_16,
- środowiska budowy platform: Mule, webMethods,
- serwer kolejek IBM MQSeries.

W tabeli 6 zawarto zestawienie komponentów oraz ich rozlokowanie na serwerze.

TABELA 6

## Komponenty środowiska testowego

Nr	Nazwa komponentu	Lokalizacja na serwerze	Przeznaczenie
1	JDK	/opt/java-1.5.0_14/	Środowisko uruchomieniowe dla komponentów napisanych w języku JAVA
2	Mule	/opt/webmeth/mule-enterprise-standalone-2.2.5	Platforma integracyjna mule — katalog domowy
3	webMethods	/opt/webmeth/61_1/IntegrationServer	Platforma Integracyjna webMethods — katalog domowy (instancja serwera)
4	webMethods Broker	/opt/webmeth/61_1/Broker	Instancja brokera komunikatów platformy webMethods
5	MQ library	/opt/mqm/Java/lib	Zestaw niezbędnych bibliotek dla połączenia serwerów integracyjnych oraz frameworka testowego z serwerem kolejek MQ
6	JNDI dir	/opt/webmeth/jndi	Katalog zawierający plik .bindings, który zawiera definicję połączenia do qManagera oraz definicję aliasów kolejek MQ.
7	Reports dir	/opt/webmeth/reports	Katalog, w którym odkładane są wyniki narzędzia raportującego obciążenie środowiska przy wykorzystaniu skryptów raportujących.
8	Oracle	/usr/lib/oracle/xe/app/oracle/product/10.2.0/Server	Katalog domowy serwera Oracle, wykorzystywany przy przesyłaniu komunikatów między systemami.
9	MQSeries	/opt/mqm	Katalog domowy serwera kolejek IBM MQ Series

W celu monitorowania obciążenia serwera został wykorzystany skrypt shelowy getStats.sh. Skrypt ten odpowiada za rejestrowanie zmian zasobów serwera w czasie przeprowadzania testów. Jest napisany z wykorzystaniem komendy vmstats, która raportuje informacje związane z procesami, pamięcią, stronicowaniem oraz aktywnością procesora.

## 10. Projekt aplikacji testowej generującej komunikację na platformie

W celu wygenerowania komunikacji między rozpatrywanymi systemami utworzono aplikację wykonaną w technologii Java. Opracowana aplikacja testowa realizowała możliwość symulowania komunikacji synchronicznej oraz asynchronicznej rozpatrywanych systemów z platformą integracyjną. Dla komunikacji synchronicznej aplikacja testowa była odpowiedzialna za wywołanie wystawionej usługi z wykorzystaniem protokołu http oraz sprawdzenie rezultatu wysłanego żądania. Dla realizacji przypadku z obsługą komunikacji asynchronicznej aplikacja odpowiadała za:

- przygotowanie komunikatu wejściowego do przetworzenia,
- umieszczenie komunikatu w zdefiniowanej kolejce MQSeries,
- oczekiwanie na nadejście komunikatu w kolejce wejściowej drugiego systemu,
- weryfikację poprawności odebranego komunikatu,
- wygenerowanie raportu z czasu trwania testu.

Zaprojektowana aplikacja testowa wykorzystywała API kolejek MQSeries realizując zadania:

- przygotowania i wysłania komunikatu do kolejki MQ,
- odebrania komunikatu wyjściowego z kolejki MQ.

Dla zapewnienia komunikacji synchronicznej aplikacja umożliwia wywołanie protokołem http żądania realizacji usługi. Dla platformy Mule wywołana zostaje usługa wystawiona jako web service. Natomiast na platformie webMethods wołana jest usługa z wykorzystaniem API webMethods.

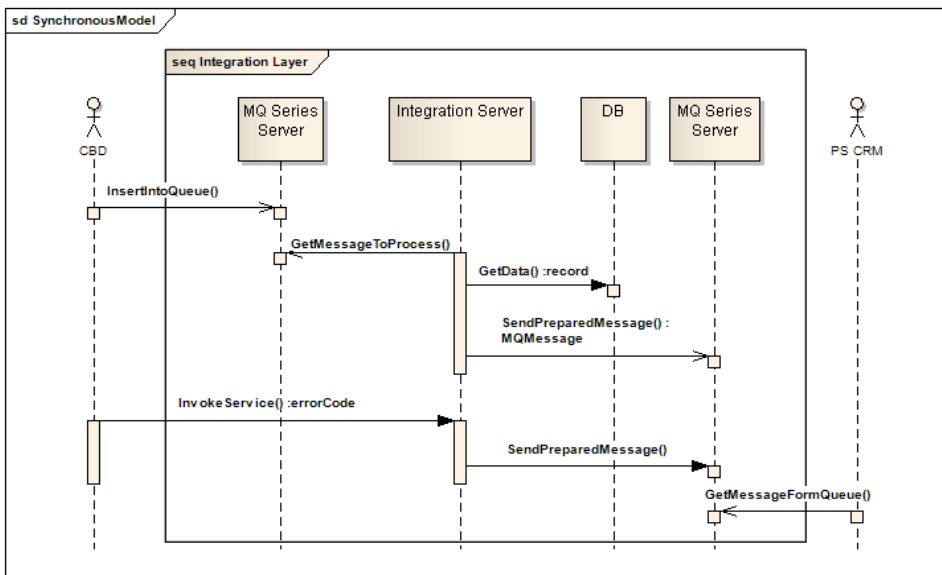
Dla realizacji przypadku użycia „Przygotowanie umowy” wykonywane są poniższe czynności:

- wstawienie przez aplikację testową (framework) komunikatu do kolejki MQ,
- pobranie na platformie komunikatu, przy wykorzystaniu adaptera JMS lub konektora JMS,
- dokonanie transformacji odebranego komunikatu,
- dokonanie translacji danych na podstawie konfiguracji w bazie danych,
- przygotowanie komunikatu do systemu MGU,
- wysłanie odpowiedzi zwrotnej z platformy do systemu zewnętrznego przez umieszczenie komunikatu w kolejce MQ,
- przechwycenie przez aplikację testową (framework) przygotowanej odpowiedzi,
- weryfikacja odebranego komunikatu.

Dla realizacji przypadku użycia „Zablokowanie konta” wykonywane są poniższe czynności:

- wywołanie usługi przy wykorzystaniu protokołu http przez aplikację testową,
- transformacja odebranego komunikatu,
- dokonanie translacji danych na podstawie konfiguracji w bazie danych,
- przygotowanie komunikatu do systemu PS CRM,
- wysłanie odpowiedzi zwrotnej z platformy do systemu zewnętrznego poprzez umieszczenie komunikatu w kolejce MQ,
- przechwycenie przez aplikację testową przygotowanej odpowiedzi,
- weryfikacja odebranego komunikatu.

Na rysunku 8 przedstawiony został diagram sekwencji obrazujący komunikację synchroniczną systemów zewnętrznych z platformą integracyjną wraz z przebiegiem przetwarzania komunikatów na platformie.



Rys. 8. Komunikacja systemów zewnętrznych z platformą

## 11. Analiza wyników badań wydajnościowych

W tak przygotowanym środowisku testowym przeprowadzono serię badań wydajnościowych zarówno dla komunikacji synchronicznej jak i asynchronicznej dla obydwu wybranych środowisk budowy platform integracyjnych.

Dalej przedstawiono wyniki zebrane na podstawie przeprowadzonych badań przy użyciu powyżej opisanej aplikacji testowej oraz narzędzia zbierania wartości miar wydajności platformy integracyjnej. Zbierano wartości miar wydajności platformy integracyjnej przy obciążeniu platformy następującą liczbą jednocześnie realizowanych komunikatów: 10, 100, 500, 1000, 3000. Wyniki zostały przedstawione w postaci tabelarycznej z podziałem na rodzaj testowanej komunikacji.

W tabeli 7 przedstawiono wyniki dla platformy integracyjnej zbudowanej przy wykorzystaniu webMethods.

TABELA 7

## Zestawienie wyników testów wydajnościowych dla webMethods

webMethods								
Komunikacja synchroniczna					Komunikacja asynchroniczna			
Liczba komunikatów	Czas przetwarzania [s]	Średnie zużycie CPU [us%]	Średnie zużycie RAM serwera [KB]	Rozmiar logów [KB]	Czas przetwarzania [s]	Średnie zużycie CPU [us%]	Średnie zużycie RAM serwera [KB]	Rozmiar logów [KB]
10	1,2	1,0	499244	14,2	8	0,7	456917	13030
100	8	1,4	499454	203	35	1,3	454305	189171
500	70	2,9	496996	508	170	2,4	454206	651500
1000	146	3,6	497265	1002	369	3,5	451039	3199171
3000	390	3,8	526863	2978	1378	3,6	461379	9238936
Wykorzystanie przydzielonych zasobów								
Minimalna pamięć ram przydzielona serwerowi					256MB			
Maksymalna pamięć ram przydzielona serwerowi					512MB			
Zaalokowana pamięć ram przez serwer					366MB			

W tabeli 8 przedstawiono wyniki dla platformy integracyjnej zbudowanej przy wykorzystaniu Mule.

Analizie poddano czas przetwarzania określonej liczby komunikatów jednocześnie uruchomionych na platformie integracyjnej. Na rysunku 9 przedstawiono diagram pokazujący zależność czasu przetwarzania określonej liczby komunikatów dla obydwu platform w przypadku komunikacji synchronicznej.

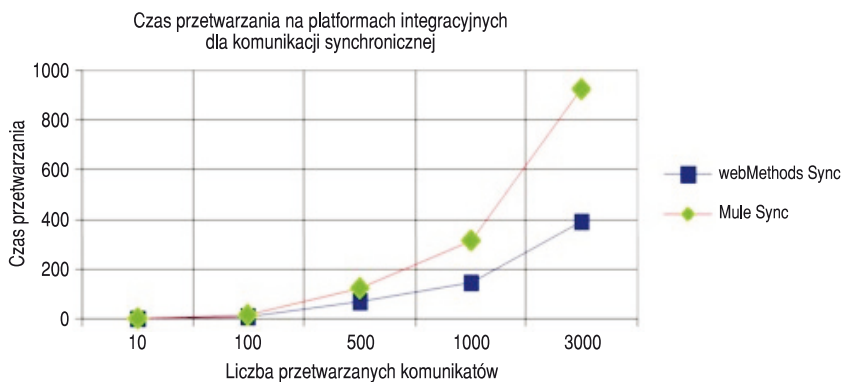
Na rysunku 10 przedstawiono diagram pokazujący zależność czasu przetwarzania określonej liczby komunikatów dla obydwu platform w przypadku komunikacji asynchronicznej.



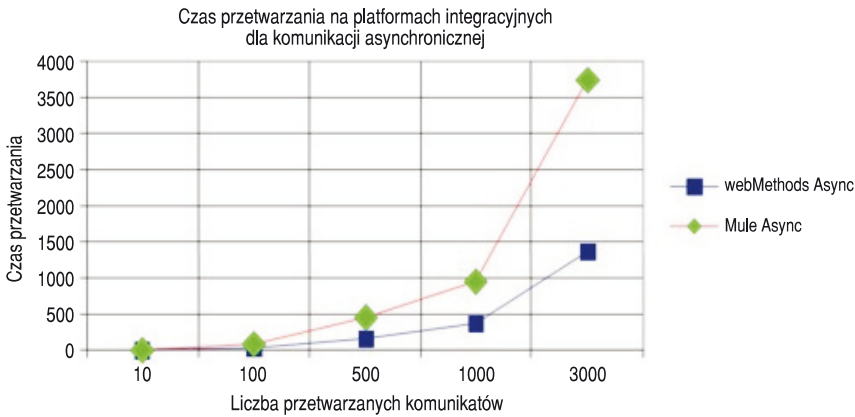
TABELA 8

## Zestawienie wyników testów wydajnościowych dla Mule

Mule								
Komunikacja synchroniczna					Komunikacja asynchroniczna			
Liczba komunikatów	Czas przetwarzania [s]	Średnie zużycie CPU [us%]	Średnie zużycie RAM serwera [KB]	Rozmiar logów [KB]	Czas przetwarzania [s]	Średnie zużycie CPU [us %]	Średnie zużycie RAM serwera [KB]	Rozmiar logów [KB]
10	1,5	1,2	701246	9715	7	1,2	602229	14745
100	15,2	1,3	701301	52881	81	1,2	602947	77881
500	120	1,35	732602	163206	450	2,0	605987	314406
1000	313	1,4	797405	321608	938	2,1	605919	569008
3000	926	1,52	833682	950342	3733	2,3	623982	1700325
Wykorzystanie przydzielonych zasobów								
Minimalna pamięć ram przydzielona serwerowi					brak			
Maksymalna pamięć ram przydzielona serwerowi					brak			
Zaalokowana pamięć ram przez serwer					41,8 MB			



Rys. 9. Czas przetwarzania określonej liczby komunikatów dla komunikacji synchronicznej



Rys. 10. Czas przetwarzania określonej liczby komunikatów dla komunikacji asynchronicznej

Z analizy uzyskanych wyników widać, że platforma zbudowana z wykorzystaniem webMethods nie tylko ok. 2-3-krotnie szybciej przesyła komunikaty, lecz także zdecydowanie lepiej radzi sobie ze wzrastającym obciążeniem komunikacyjnym. Czasy odpowiedzi pozostają praktycznie na tym samym poziomie. Dla przykładu, dla komunikacji asynchronicznej przy 1000 i 3000 komunikatów czasy odpowiedzi pozostają w granicach 0,5 sekundy. W przypadku Mule czasu odpowiedzi komunikatu platformy obsługującej 3000 w stosunku do czasu odpowiedzi komunikatu platformy obsługującej 1000 to ok. 266% (z 1 sekundy na 2,66 sekundy).

Natomiast przewagą Mule jest zużycie zasobów. Platforma zbudowana przy wykorzystaniu Mule potrzebuje prawie 9 razy mniej pamięci RAM do swojego funkcjonowania.

## 12. Podsumowanie

W artykule pokazano sposób badania wydajności platformy integracyjnej utworzonej przy wykorzystaniu różnych środowisk. W porównaniu środowisk uwzględniono także aspekt czasu potrzebnego na implementację rozwiązań na platformie. Analizie poddano także zakres narzędzi wspierających projektowanie rozwiązań.

Istotne jest, że znając szacowany ruch na platformie integracyjnej, można świadomie dobrać rozwiązanie dopasowane do rzeczywistych potrzeb budowanej platformy integracyjnej. Ponadto, wymagania wydajnościowe można wtedy zweryfikować z kwestiami kosztów, jakie należy ponieść na budowę platformy w kwestii potrzeb sprzętowych i licencji.

Na podstawie przeprowadzonych badań i zebranych charakterystyk można stwierdzić, że każda z porównywanych platform ma swoje mocne i słabe strony.

Mule jest środowiskiem przeznaczonym dla małych i średnich przedsięwzięć. Środowisko to ma niewielkie wymagania sprzętowe i świetnie radzi sobie z grupą podstawowych typów komunikacji. Natomiast przy użyciu WebMethods można budować platformy o wysokiej wydajności do integracji wielu systemów. Tego typu rozwiązania najczęściej mają narzucone wymagania odnośnie czasu realizacji zleceń. Rozwiązanie jest przeznaczone raczej dla dużych firm, nie tylko z racji dobrych osiągnięć wydajnościowych oraz wielu narzędzi, ale przede wszystkim pod względem wymagań sprzętowych oraz ceny.

Z analizy wydajnościowej trybów komunikacji wynika, że obydwie platformy zdecydowanie krótsze czasy odpowiedzi uzyskują przy komunikacji synchronicznej. Należy jednak pamiętać, że komunikacja asynchroniczna dzięki mechanizmom rozładowywania kolejek komunikatów pozwala na rozłożenie obciążenia systemu w czasie, zachowując pełną funkcjonalność systemu oraz eliminując możliwość krytycznego przeciążenia systemu.

Artykuł wpłynął do redakcji 25.02.2011 r. Zweryfikowaną wersję po recenzji otrzymano w lutym 2012 r.

#### LITERATURA

- [1] N. BIEBERSTEIN, *Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap*, IBM Press, 2005.
- [2] C.A. BINILDAS, *Service Oriented Java Business Integration*, Packt Publishing, 2008.
- [3] Business Process Model and Notation (BPMN), OMG, <http://www.omg.org/spec/BPMN/>.
- [4] D. CHAPPELL, *Enterprise Service Bus*, O'Reilly, 2004.
- [5] R. CREDLE i in., *Patterns: SOA Design Using WebSphere Message Broker and WebSphere ESB*, IBM, 2007.
- [6] T. ERL, *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall, 2005.
- [7] T. GÓRSKI, *Zwinność i dyscyplina w podnoszeniu efektywności zespołów projektowych*, Biuletyn Instytutu Systemów Informatycznych, 6, 2010.
- [8] T. GÓRSKI, *Metoda zarządzania architekturą korporacyjną*, Biuletyn Instytutu Systemów Informatycznych, 3, 2009.
- [9] M. HAVEY, *Essential Business Process Modeling*, O'Reilly, 2005.
- [10] G. HOHPE, B. WOLF, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging*, Addison Wesley, 2003.
- [11] JSR-000208 Java Business Integration 1.0, Sun Microsystems 2005 <http://jcp.org/aboutjava/communityprocess/final/jsr208/index.html>.
- [12] M. KEEN i in., *Getting Started with WebSphere Enterprise Service Bus V6*, IBM, 2006.
- [13] M. KEEN i in., *Patterns: SOA with an Enterprise Service Bus in WebSphere Application Server V6*, IBM, 2005.
- [14] M. KEEN, *Patterns: Implementing an SOA Using an Enterprise Service Bus*, IBM, 2004.
- [15] C. SADTLER, *Patterns: SOA Foundation Service Creation Scenario*, IBM, 2006.
- [16] UDDI Version 3.0.2, OASIS 2004, <http://uddi.org/pubs/uddi-v3.0.2-20041019.pdf>.

- [17] Web Services Business Process Execution Language Version 2.0, OASIS 2007, <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.doc>.
- [18] Web Services Description Language (WSDL) Version 2.0, W3C 2007, <http://www.w3.org/TR/wsdl20/>.
- [19] S. ZIELIŃSKI, *Analiza porównawcza platform integracji systemów informatycznych*, praca magisterska WAT, 2010.

T. GÓRSKI

**Performance analysis of selected frameworks for integration platforms development**

**Abstract.** The article demonstrates how to analyze the performance of the integration platform created with using different frameworks. In frameworks comparison also included the aspect of time required to implement solutions on the platform. The range of tools to support designing of solution was also analyzed. The article contains a description of the business case which was implemented on integration platform. The article also encompasses description of test environment and an analysis of performance test results.

**Keywords:** integration platform, performance, information systems design