

TOMASZ GÓRSKI¹

Wydział Cybernetyki
Wojskowa Akademia Techniczna w Warszawie

WOJCIECH KUCHTA²

Główny Inspektorat Jakości Handlowej Artykułów Rolno-Spożywczych

Zastosowanie magistrali usług ESB do przesyłania dużych wolumenów danych

1. Wstęp

Firmy i organizacje mają wiele systemów informatycznych wspierających ich działalność. Wynika to z faktu, że 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. Coraz częściej systemy informatyczne eksploatowane w firmach muszą współpracować ze sobą. Spowodowało to zapotrzebowanie na technologie i rozwiązania umożliwiające integrację istniejących systemów i aplikacji, a także na ponowne wykorzystanie już utworzonych funkcjonalności. Obecnie stosowanym podejściem do integracji systemów informatycznych jest architektura usługowa (ang. *service-oriented architecture* – SOA). Centralnym elementem w tego typu rozwiązaniach jest szyna usług (ang. *enterprise service bus* – ESB)³. Rozwiązanie integracyjne składające się z integrowanych systemów oraz warstwy komunikacyjnej umożliwiającej współpracę podłączonych do niej systemów jest nazywane platformą integracyjną⁴. Przy projektowaniu rozwiązań integracyjnych istotna jest możliwość modelowania ich kompletnego opisu architektonicznego. Do tego jest potrzebny model widoków architektonicznych⁵ dostosowany do

¹ gorski@wat.edu.pl.

² wkuchta@ijhars.gov.pl.

³ D. Chappell, *Enterprise Service Bus*, O'Reilly, Sebastopol 2004.

⁴ T. Górski, *Platformy integracyjne. Zagadnienia wybrane*, Wydawnictwo Naukowe PWN, Warszawa 2012.

⁵ T. Górski, *Architectural view model for an integration platform*, „Journal of Theoretical and Applied Computer Science” 2012, vol. 6, no. 1, s. 25–34.

potrzeb modelowania platform integracyjnych oraz zestaw konstrukcji modelowych, dzięki któremu można przedstawić pełną architekturę platform integracyjnych⁶.

2. Model widoków architektonicznych „1+5”

Spójność opisu architektonicznego rozwiązań informatycznych jest istotnym zagadnieniem aktualnie badanym⁷. Istnieje wiele modeli z różnymi zestawami widoków architektonicznych, np.: „4+1”, RM-ODP, model Siemens, widoki SEI⁸. Nie zapewniają jednak one możliwości kompletnego opisu architektury rozwiązań integracyjnych.

Zaproponowano model widoków architektonicznych „1+5” (rysunek 1), dostosowany do potrzeb projektowania platform integracyjnych. W modelu tym wyodrębniono następujące widoki architektoniczne:

- integrowanych procesów (ang. *integrated processes*);
- przypadków użycia (ang. *use cases*);
- logiki (ang. *logical*);
- integrowanych usług (ang. *integrated services*);
- kontraktów (ang. *contracts*);
- rozlokowania (ang. *deployment*).

Podstawowym widokiem architektonicznym jest widok integrowanych procesów, w którym modelowane są procesy biznesowe mające być zautomatyzowane na platformie integracyjnej. Następne cztery widoki (przypadków użycia, logiki, integrowanych usług, kontraktów) służą do przedstawienia projektu platformy integracyjnej. Widok przypadków użycia zawiera wymagania funkcjonalne na system integrowany w ramach platformy integracyjnej. W widoku integrowanych usług są prezentowane usługi wystawiane z systemów informatycznych oraz ich sposób włączenia na magistralę usług. Widok kontraktów przedstawia komponenty reprezentujące systemy informatyczne i kontrakty określone między nimi. W widoku tym są także prezentowane przepływy mediacyjne dla każdego kontraktu. Ostatni widok, rozlokowania, przedstawia sposób osadzenia zaprojektowanej platformy integracyjnej w określonym środowisku uruchomieniowym.

⁶ T. Górski, *UML profiles for architecture description of an integration platform*, „Biuletyn Wojskowej Akademii Technicznej” 2013, nr 2, s. 43–56.

⁷ M. Abi-Antoun, J. Aldrich, N. Nahas, B. Schmerl, D. Garlan, *Differencing and merging of architectural views*, „Automated Software Engineering” 2008, vol. 15, s. 35–74.

⁸ N. Rozanski, E. Woods, *Software Systems Architecture. Working with stakeholders using Viewpoints and Perspectives*, Addison-Wesley, Crawfordsville 2005.



Rysunek 1. Model widoków architektonicznych „1+5”

Źródło: opracowanie własne.

Szczegółowy opis przedstawianego modelu widoków architektonicznych, wraz z przykładami zastosowań, znajduje się w literaturze⁹. W rozpatrywanym podejściu zostały zaproponowane modele i diagramy języków BPMN oraz UML z rozszerzeniem o konstrukcje języka SoaML do modelowania architektury platformy integracyjnej (tabela 1).

Tabela 1. Elementy do modelowania architektury platformy integracyjnej

Model	Widok	Diagram
Procesów	integrowanych procesów	(BPMN) procesu biznesowego
Przypadków użycia	przypadków użycia	(UML) przypadków użycia
		(UML) aktywności
Projektowy	logiki	(UML) sekwencji
		(UML) komunikacji
		(UML) klas
Usług	integrowanych usług	(UML) komponentów
	kontraktów	(UML) komponentów
		(UML) aktywności
		(UML) struktury
Rozłokowania	rozłokowania	(UML) wdrożenia

Źródło: opracowanie własne.

⁹ T. Górski, *Architektura platformy integracyjnej dla elektronicznego obiegu recept, „Roczniki” Kolegium Analiz Ekonomicznych*, z. 25, Oficyna Wydawnicza SGH, Warszawa 2012.

3. Prace powiązane tematycznie

Architektura usługowa¹⁰ i integracja systemów informatycznych z zastosowaniem magistrali¹¹ usług są aktualnymi zagadnieniami rozpatrywanymi w literaturze. Przegląd literatury dotyczącej zagadnienia modeli procesów w przypadku aplikacji usługowych można znaleźć w tekście *Process models for service-based applications: A systematic literature review*¹². Wsparcie procesów biznesowych przez systemy informatyczne w kontekście dynamicznego i optymalnego doboru usług do realizacji zadań w procesie biznesowym przedstawiono w artykułach *Per-flow optimal service selection for Web services based processes*¹³ i *Modeling functional requirements for configurable content- and context-aware dynamic service selection in business process models*¹⁴. Kolejnym istotnym zagadnieniem w obszarze architektury usługowej jest wsparcie dynamicznie rekonfigurowanych procesów biznesowych¹⁵ oraz uwzględnienie wymagań pozafunkcyjnych¹⁶.

Integracja wielu systemów informatycznych powoduje wysoką złożoność projektu integracji¹⁷. Ponadto, w literaturze przedmiotu są obecne prace opisujące elementy konieczne do budowy nowej generacji zintegrowanych środowisk biznesowych i analizujących cechy magistrali usług jako rdzenia tej infrastruktury¹⁸. Porównanie cech magistral usług różnych producentów zostało także przedstawione w książce *Platformy integracyjne*¹⁹. W budowie rozwiązań integracyjnych niezwykle istotnymi zagadnieniami są zarówno poprawne dostarczenie komunikatu, jak i dostarczenie go w wymaganym

¹⁰ T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall, Crawfordsville 2005.

¹¹ M. Keen, A. Acharya, *Implementing an SOA Using an Enterprise Service Bus*, IBM, 2010.

¹² S. Lane, I. Richardson, *Process models for service-based applications: A systematic literature review*, „Information and Software Technology” 2011, vol. 53, s. 424–439.

¹³ D. Ardagna, R. Mirandola, *Per-flow optimal service selection for Web services based processes*, „The Journal of Systems and Software” 2010, vol. 83, s. 1512–1523.

¹⁴ A. Frece, M.B. Juric, *Modeling functional requirements for configurable content- and context-aware dynamic service selection in business process models*, „Journal of Visual Languages and Computing” 2012, vol. 23, s. 223–247.

¹⁵ F. Cicirelli, A. Furfaro, L. Nigro, *A service-based architecture for dynamically reconfigurable workflows*, „The Journal of Systems and Software” 2010, vol. 83, s. 1148–1164.

¹⁶ P. Potena, *Optimization of adaptation plans for a service-oriented architecture with cost, reliability, availability and performance tradeoff*, „The Journal of Systems and Software” 2013, vol. 86, s. 624–648.

¹⁷ R. Chalmeta, V. Pazos, *A step-by-step methodology for enterprise interoperability projects*, „Enterprise Information Systems” 2015, vol. 9, s. 436–464.

¹⁸ M.A. Martínez-Carreras, F.J. García Jimenez, A.F. Gómez Skarmeta, *Building integrated business environments: analysing open-source ESB*, „Enterprise Information Systems” 2015, vol. 9, s. 401–435.

¹⁹ T. Górski, *Platformy integracyjne...*, op.cit.

czasie. Kwestię detekcji błędów w rozwiązaniach integracyjnych przedstawia artykuł *A proposal to detect errors in Enterprise Application Integration solutions*²⁰. Natomiast zagadnienie analizy wydajnościowej różnych środowisk budowy rozwiązań integracyjnych zostało opisane w artykule *Performance Analysis of Selected Frameworks for an Integration Platform Development*²¹.

4. Miary wydajności platformy integracyjnej

Jednym z podstawowych wymagań wydajnościowych jest czas uzyskania wyników realizacji usługi. Można przyjąć następujące miary wydajności platformy integracyjnej związane z wywołaniami usług:

- opóźnienie w jedną stronę (ang. *one-way delay* – OWD) – jest to czas, jaki zajmuje wiadomości osiągnięcie docelowej lokalizacji, czyli jest on liczony od wysłania z punktu źródłowego do dotarcia do celu; na OWD składają się dwa elementy, które są związane z przesyłaniem danych:
 - opóźnienie propagacji – jest to czas przemieszczenia się komunikatu od jednego odcinka do drugiego przy uwzględnieniu jedynie szybkości fizycznej łącza, jakim jest przesyłany,
 - opóźnienie przetwarzania (ang. *serialization delay*) – jest to czas potrzebny do konwersji czy przetwarzania komunikatu przez całą drogę, którą przebywa;
- opóźnienie w obie strony (ang. *round trip time* – RTT) – jest to czas, w jakim komunikat pokona drogę do punktu docelowego i z powrotem; mogłoby się wydawać, że jest to po prostu dwukrotność czasu opóźnienia w jedną stronę, lecz w rzeczywistości mogą to być całkiem różne czasy;
- maksymalne opóźnienie (ang. *maximum delay*) – największa wartość czasu opóźnienia w jedną stronę; jest on ważny dla aplikacji, które mają założone maksymalne odchylenie czasu opóźnienia;
- zmiana opóźnienia (ang. *delay variation*) – jest to zmiana opóźnienia w jedną lub w dwie strony w czasie; może to zależeć od intensywności wysyłanych w jednej chwili komunikatów.

²⁰ R.Z. Frantz, R. Corchuelo, C. Molina-Jiménez, *A proposal to detect errors in Enterprise Application Integration solutions*, „The Journal of Systems and Software” 2012, vol. 85, s. 480–497.

²¹ T. Górski, *Performance Analysis of Selected Frameworks for an Integration Platform Development*, „Bulletin of IT Systems Institute” 2011, vol. 7, s. 9–17.

Dla jednostronnych wywołań mierzone jest opóźnienie w jedną stronę OWD. Dla tego typu wywołań typową miarą jest wartość liczby komunikatów obsługiwanych w zadanym okresie. Przy wywołaniach, gdy oczekiwana jest odpowiedź, czas jest mierzony od wysłania żądania do momentu odbioru odpowiedzi przez żądającego (RTT).

5. Wzorce architektoniczne

W rozwiązaniach integracyjnych są stosowane następujące wzorce architektoniczne²²:

- oddzielne wywołanie – jest to wzorec, który rozwiązuje zagadnienie powiązań tymczasowych oraz kwestie związane z wydajnością, ale daje pewne opóźnienia, wpływa na skalowalność i wydajność;
- potoki równoległe – jest to wzorec, w którym podzadania składają się na cały proces; ze względu na ciągłe utrzymywanie się dużej liczby żądań następuje ich gromadzenie w kolejce, co może doprowadzić do jej przepełnienia;
- przetwarzanie sieciowe – jest to wzorec wraz z technologią, na której został zaprojektowany, mogący sobie poradzić z wydajnością i dostępnością; spełnienie atrybutów jakościowych jest możliwe do zrealizowania za pomocą mechanizmów mogących równoważyć obciążenia obliczeniowe na niewykorzystywane zasoby, a łączenie zasobów w pule oraz stała analiza warunków sieciowych wpływa w znaczący sposób na skalowalność;
- instancja usługi – jest to wzorec ściśle związany z dostępnością; głównym wyzwaniem związanym z tą usługą jest możliwość skalowania w przypadku wysokiego obciążenia; wzorec ten opiera się na wprowadzeniu dużej liczby kopii tej samej usługi;
- wirtualny punkt końcowy – jest to wzorec, który ukrywa rzeczywiste adresy komponentów brzegowych; polega na wdrożeniu nadmiarowych instancji usługi dostępnych pod jednym adresem IP funkcjonujących jako jedna usługa, zapewniając jednocześnie przejrzystość lokalizacji;
- strażnik usługi – jest to wzorec skupiający się na autonomiczności przez aktywne monitorowanie swojego stanu, podejmowanie działań w momencie ewentualnych problemów, poprawianie własnej kondycji oraz permanentne informowanie o swoim statusie odpowiedniego operatora; w celu zwiększenia dostępności usługi należy znaleźć i zdiagnozować ewentualne problemy oraz usunąć błędy dotyczące usługi.

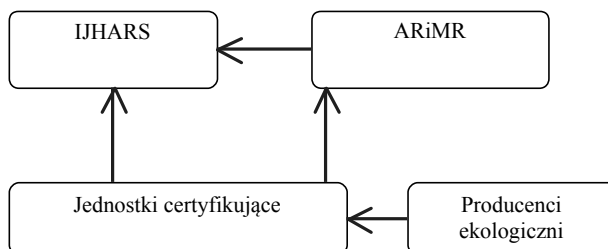
²² A. Rotem-Gal-Oz, *Wzorce SOA*, Helion, Gliwice 2013.

6. Przypadek biznesowy

Rozpatrywanym przypadkiem biznesowym systemu obsługi dużych wolumenów danych wykorzystującego szynę usług jest system Inspekcji Jakości Handlowej Artykułów Rolno-Spożywczych (IJHARS). Inspekcja jest instytucją zajmującą się kontrolą jakości handlowej artykułów rolno-spożywczych na rynku krajowym, a także w obrocie z zagranicą, działającą na mocy ustawy z dnia 21 grudnia 2000 r. o jakości handlowej artykułów rolno-spożywczych (Dz. U. z 2001 r. Nr 5, poz. 44 z późn. zm.). Jednym z zadań Inspekcji jest również nadzór nad jednostkami certyfikującymi oraz nadzór nad produkcją ekologiczną zgodnie z ustawą z dnia 25 czerwca 2009 r. o rolnictwie ekologicznym (Dz. U. z 2009 r. Nr 116, poz. 975). Jednostki certyfikujące to podmioty upoważnione przez Ministra Rolnictwa i Rozwoju Wsi do przeprowadzania kontroli oraz wydawania i cofania certyfikatów w zakresie rolnictwa ekologicznego. Zbierają one dane od producentów ekologicznych. Jednym z obowiązków jednostek certyfikujących jest również przekazywanie Agencji Restrukturyzacji i Modernizacji Rolnictwa (ARiMR) wykazu producentów ekologicznych, którzy spełnili określone wymagania w zakresie rolnictwa ekologicznego. ARiMR jako akredytowana agencja płatnicza zajmuje się wdrażaniem instrumentów współfinansowanych z budżetu Unii Europejskiej oraz udziela pomocy ze środków krajowych. Odpowiada m.in. za udzielanie pomocy finansowej producentom ekologicznym realizującym pakiet rolnictwo ekologiczne w ramach Programu Rozwoju Obszarów Wiejskich, finansowanego z Europejskiego Funduszu na Rzecz Rozwoju Obszarów Wiejskich. Płatności są przyznawane m.in. na podstawie danych zawartych w wykazie przekazanym przez jednostki certyfikujące. Oddzielny wykaz producentów ekologicznych jest przedstawiany przez jednostki certyfikujące do IJHARS. ARiMR okresowo przeprowadza kontrole u producentów ekologicznych mające na celu sprawdzanie realizacji zobowiązań rolno-środowiskowych oraz w konsekwencji – prawidłowości sporządzenia wykazu.

W przypadku wykrycia nieprawidłowości w wykazie prezes ARiMR składa do IJHARS wnioski o ukaranie jednostki certyfikującej. Wniosek taki często zawiera wiele załączników (zdjęć, dokumentów w formacie pdf) dotyczących działek rolnych, przekazywanych na płycie CD/DVD. Pliki takie dotyczące jednej sprawy mogą mieć średnio objętość ok. 700 MB. Rysunek 2 przedstawia schemat przepływu danych. W związku z dużą objętością danych jest niestety wykluczone przekazanie ich w całości za pomocą platformy ePUAP, ponieważ uniemożliwia ona przekazywanie tak dużych załączników. Platforma ePUAP z pewnością byłaby doskonałym rozwiązaniem pozwalającym na uwierzytelnienie użytkownika, dając możliwość wykorzystania koncepcji

pojedynczego logowania SSO (ang. *single-sign on*) czy podpisu dokumentu za pomocą profilu zaufanego z wykorzystaniem usługi sieciowej.



Rysunek 2. Schemat przepływu danych w systemie rolnictwa ekologicznego w Polsce

Źródło: opracowanie własne.

Niezależnie od wykazu przedstawianego prezesowi ARiMR, jednostki certyfikujące przekazują IJHARS także dane i informacje o producentach ekologicznych. Dane te są wykorzystywane do sprawowania nadzoru nad jednostkami certyfikującymi i nad produkcją ekologiczną, a także mogą być wykorzystywane podczas rozpatrywania wniosku złożonego przez prezesa ARiMR. Jednostki przekazują IJHARS dane w formacie plików XML (w formie off-line, np. płyta CD/DVD), które są następnie ręcznie importowane do zintegrowanego systemu informatycznego IJHARS (ZSI IJHARS), co również z pewnością można byłoby usprawnić.

Mając powyższe na uwadze oraz uwzględniając perspektywę integracji systemu IJHARS z systemem Służby Celnej oraz platformą ePUAP, należy stwierdzić, że wdrożenie szyny usług wydaje się najlepszym rozwiązaniem. Integracja systemów informatycznych wspiera realizację konkretnych potrzeb biznesowych, w związku z tym istotne jest rozważenie różnych modeli oddziaływań biznesowych. Można wyróżnić następujące modele biznesowe:

- ukierunkowany na użytkownika (ang. *user-centric*);
- ukierunkowany na zdarzenia (ang. *event-centric*);
- ukierunkowany na procesy biznesowe (ang. *business-centric*).

Przykładowo, zdarzenie z systemu ARiMR mogłoby powodować uruchomienie procesu związanego z jego obsługą w systemie IJHARS. ZSI IJHARS jest kluczowym systemem informatycznym IJHARS umożliwiającym rejestrację w formie elektronicznej zadań realizowanych przez Inspekcję. Struktura systemu składa się z następujących podsystemów:

- rejestracji kontroli;
- rejestracji kontrahentów;
- inspektorów;

- bezpieczeństwa;
- słowników;
- *front-end*;
- analizy danych.

Docelowo system ma obejmować wszystkie zadania realizowane przez Inspekcję, umożliwiać komunikację z klientem zewnętrznym za pomocą skrzynek podawczych założonych na platformie ePUAP oraz pozwalać na łatwą integrację z systemami zewnętrznymi. W związku z tym istnieje konieczność wyodrębnienia dodatkowego podsystemu komunikacji, którego zadaniem powinna być obsługa dużych wolumenów danych.

Wymagania funkcjonalne dla takiego systemu to:

- udostępnienie jednolitego interfejsu do przekazywania i odbierania danych;
- umożliwienie łatwego dodawania filtrowania danych;
- automatyzacja procesu przekazania danych;
- obsługa doręczeń komunikatów;
- autoryzacja nadawcy i odbiorcy;
- sprawdzanie i odrzucanie komunikatów niezgodnych ze wzorem – weryfikacja dokumentów XML pod kątem zgodności ze schematem XSD;
- sprawdzanie i odrzucanie komunikatów mających niewłaściwą zawartość;
- zapewnienie przetwarzania komunikatu tylko raz, tzw. operacje idempotentne (ang. *idempotent*);
- weryfikacja podpisu potwierdzonego profilem zaufanym ePUAP złożonego na dokumencie;
- możliwość kolejkowania i późniejszego dostarczenia komunikatów, które nie mogły zostać dostarczone z powodu niedostępności systemów zewnętrznych.

Wymagania niefunkcjonalne dla takiego systemu to:

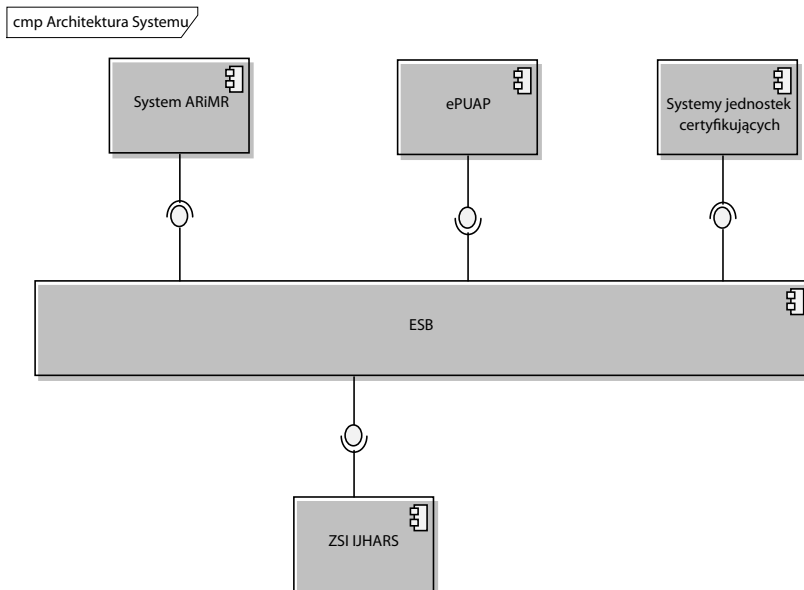
- wydajność – zapewnienie odpowiedniej liczby transakcji na sekundę, odpowiedniego czasu odpowiedzi;
- skalowalność – możliwość rozbudowy systemu w łatwy sposób w przypadku zwiększenia obciążenia;
- elastyczność – usługi wchodzące w skład systemu powinny być od siebie niezależne, aby zapewnić jego elastyczność;
- bezpieczeństwo – zapewnienie atrybutów bezpieczeństwa (poufność, integralność, dostępność, rozliczalność).

Rysunek 3 przedstawia architekturę systemu uwzględniającą szynę usług jako element odpowiedzialny za integrację systemów. W procesie komunikacji wyróżnia się następujące modele realizowane w rozwiązaniach integracyjnych:

- pytanie/odpowiedź (ang. *request/response*);
- pytanie/wiele odpowiedzi (ang. *request/multiresponse*);

- rozsyłanie zdarzeń (ang. *event propagation*);
- publikuj/subskrybuj (ang. *publish/subscribe*).

W przypadku rozpatrywanego systemu odpowiedni wydaje się pierwszy model. Zakłada on przesłanie komunikatu od konsumenta do usługi docelowej, przetworzenie go, a następnie zwrócenie odpowiedzi. Należy zauważyć, że w tym wzorcu można wyróżnić tryb synchroniczny, polegający na zablokowaniu wątku po stronie konsumenta, oraz tryb asynchroniczny, pozwalający na dalsze działanie konsumenta usługi.



Rysunek 3. Diagram komponentów architektury systemu

Źródło: opracowanie własne.

W zakresie obsługi żądań przesyłanych do systemu można wyróżnić następujące wzorce mediacyjne²³:

- przełączanie protokołów (ang. *protocol switch*);
- transformacja (ang. *transform*);
- wzbogacanie (ang. *enrich*);
- trasowanie (ang. *route*);
- dystrybucja (ang. *distribute*);
- monitorowanie (ang. *monitor*);
- korelacja (ang. *correlate*).

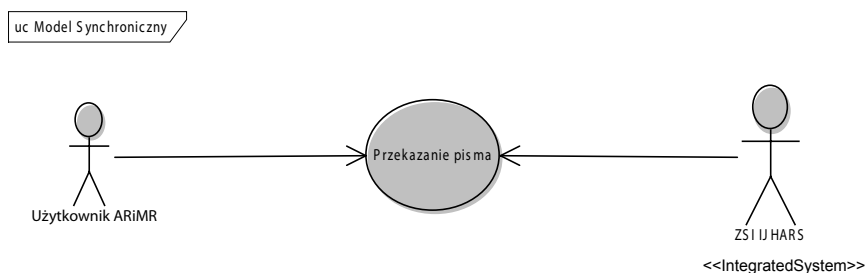
²³ T. Górski, *Platformy integracyjne...*, op.cit.

W rozpatrywanym systemie zostanie wykorzystany wzorzec transformacji, który zapewni transformację komunikatu z obiektu obsługiwanego przez szynę usług na format XML.

7. Zakres funkcjonalności realizowany na magistrali ESB

W rozpatrywanym systemie została zrealizowana funkcja pozwalająca na przekazanie dokumentu z systemu ARiMR do szyny usług, a następnie pobranie go z szyny do systemu IJHARS. Rysunek 4 przedstawia diagram przypadków użycia dla aplikacji ARiMR z wyodrębnionym systemem integrowanym za pomocą szyny usług.

Przekazanie pisma ze wszystkimi załącznikami w formie elektronicznej i zastąpienie papierowego druku z załącznikami płytą CD/DVD z pewnością wpłynęłyby na usprawnienie działań ARiMR oraz IJHARS. Przypadek użycia „przekazanie pisma” został opisany za pomocą języka WSDL oraz udostępniony w formie usługi na szynie ESB.



Rysunek 4. Diagram przypadków użycia dla systemu ARiMR z wyodrębnionym systemem integrowanym ZSI IJHARS

Źródło: opracowanie własne.

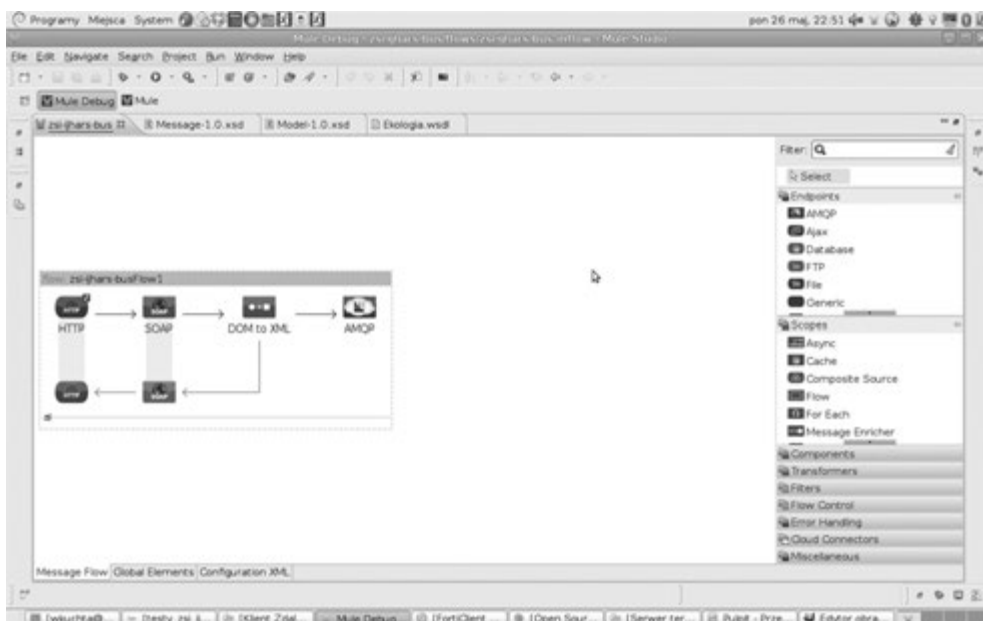
Aplikacja ARiMR za pomocą protokołu HTTP prześle komunikat do udostępnionej usługi. Pismo wraz z załącznikami zostanie przekazane w formie pliku XML. Komunikat po transformacji z wewnętrznego modelu obiektowego szyny do formatu XML zostanie umieszczony w kolejce komunikatów. Następnie komunikat zostanie pobrany z kolejki do systemu IJHARS. Implementacja systemu po stronie ARiMR oraz po stronie IJHARS została przygotowana w formie prostych skryptów napisanych w języku PHP. Została również wykorzystana szyna usług Mule ESB wraz brokerem komunikatów RabbitMQ, będącym implementacją protokołu *Advanced Message Queuing Protocol* (AMQP).

Formularz przedstawiający system po stronie ARiMR pozwala na wprowadzenie znaku sprawy, daty pisma oraz treści. Załączniki w postaci plików PDF zawierających szkice działek oraz plików JPG ze zdjęciami działek producenta ekologicznego są pobierane z dysku lokalnego, odpowiednio z folderów – szkice i zdjęcia. Skrypt imitujący system IJHARS pobiera komunikat za pomocą API, udostępnianego przez brokera RabbitMQ, oraz wyświetla liczbę załączników.

8. Zaprojektowane rozwiązanie

W celu przeprowadzenia badania spełnienia wybranych wymagań funkcjonalnych i pozafunkcyjnych zostało przygotowane środowisko testowe. Zainstalowane środowisko ma następującą konfigurację:

- nadzorca maszyn wirtualnych serwer IBM 3500 (45056 MB RAM, procesor 1x4 rdzenie Intel Xeon 2.33 GHz) CitrixXenServer 5.6 Enterprise Edition;
- pamięć masowa NetApp FAS 2020;
- system operacyjny maszyny wirtualnej – Linux Debian 6.0;



Rysunek 5. Widok przepływu w środowisku Mule Studio

Źródło: opracowanie własne.

- pamięć RAM maszyny wirtualnej 4 GB;
- dysk twardy maszyny wirtualnej 80 GB;
- procesor 1 rdzeń 2.33 GHz.

Oprogramowanie, które zostało zainstalowane, to:

- Java JRE 1.6, JDK 1.6.0_26;
- szyna usług Mule ESB Enterprise Edition 3.5.0;
- Mule Management Console 3.5.0;
- broker komunikatów RabbitMQ 3.3.1.

Zamodelowany w środowisku Mule Studio 3.5.0 przepływ implementuje komunikację synchroniczną. Na rysunku 5 przedstawiono widok środowiska programistycznego.

9. Analiza wyników testów

W realizacji testów wykorzystano przypadek użycia „przekazanie pisma”. W ramach dokumentacji działki występują następujące dokumenty:

- pismo przewodnie;
- szkic działki (pliki PDF);
- zdjęcia działki (plik JPG).

W sumie dokumentacja do przesłania dotycząca działek ma ok. 700 MB.

Na rysunku 6 przedstawiono przykład komunikatu SOAP, jaki będzie przesyłany. Elementy `skicDzialki` i `zdjecieDzialki` zawierają odpowiednio pliki PDF i plik JPG zakodowane algorytmem `base64`. Należy tu dodać, że nie jest to zalecany sposób przesyłania plików tego typu. SOAP Message Transmission Optimization Mechanism (MTOM) to standard rozwijany przez konsorcjum W3C, zapewniający efektywne przekazywanie danych binarnych za pomocą usług sieciowych. Dzięki mechanizmowi XML Optimized Packaging (XOP) dane są przesyłane jako załączniki MIME komunikatu SOAP. Referencja do danych w komunikacie SOAP jest realizowana za pomocą elementu `<xop:Include>`. Dzięki temu oddzielamy dane tekstowe zawarte w komunikacie od danych binarnych. Należy wziąć pod uwagę to, że kodowanie danych binarnych algorytmem `base64` zwiększa ich rozmiar zgodnie ze współczynnikiem 1.33, przez co niepotrzebnie jest zwiększany rozmiar całego komunikatu. Załączniki MIME również są kodowane algorytmem `base64`, lecz są oddzielone od treści wiadomości, co w pewnych okolicznościach może mieć istotne znaczenie.

Test nr 1 polegał na przesłaniu i odebraniu jednego komunikatu zawierającego wszystkie zdjęcia i szkice działek dotyczące wybranej sprawy. Dokumentacja do celów testów miała 675,71 MB. Po ok. 10 minutach od uruchomienia skryptu testu pojawiła

się seria ostrzeżeń i skrypt przerwał działanie. Mimo zwiększenia limitów przydziału pamięci na skrypt oraz innych opcji konfiguracyjnych zgodnych z dokumentacją serwera PHP, nie przyniosło to oczekiwanego rezultatu.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="http://www.ijhars.gov.pl/SOA/message/1.0"
  xmlns:ns1="http://www.ijhars.gov.pl/SOA/model/1.0">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:wyslijPismo>
      <ns1:Pismo>
        <pismoData>2002-01-01</pismoData>
        <pismoZnakSprawy>ZP-370-DD-X</pismoZnakSprawy>
        <pismoTresc>dasdasdasdasdasd adasdasdasd</pismoTresc>
        <pismoSzkiceDzialek>
          <szkicDzialki>/awKKapGGyQ=</szkicDzialki>
          <szkicDzialki>/awPKapGGyQ=</szkicDzialki>
          <szkicDzialki>/awOKapGGyQ=</szkicDzialki>
        </pismoSzkiceDzialek>
        <pismoZdjeciaDzialek>
          <zdjecieDzialki>/awOKapGGyQwwwww</zdjecieDzialki>
          <zdjecieDzialki>/awOKapGGyQSSwww</zdjecieDzialki>
          <zdjecieDzialki>/awOKapGGyQwwGGGW</zdjecieDzialki>
        </pismoZdjeciaDzialek>
      </ns1:Pismo>
    </ns:wyslijPismo>
  </soapenv:Body>
</soapenv:Envelope>
```

Rysunek 6. Postać komunikatu wysłanego z systemu ARiMR

Źródło: opracowanie własne.

Test nr 2 polegał na przesłaniu każdego pliku oddzielnym komunikatem. Dokumentacja do celów tego testu także miała rozmiar 675,71 MB, natomiast liczba wszystkich plików wynosiła 169 i każdy z nich został wysłany oddzielnym komunikatem. Wyniki testów przedstawiono w tabeli 2.

Tabela 2. Wyniki testu nr 2

Czas wysłania wszystkich komunikatów [s]	Minimalna wielkość komunikatu [MB]	Średnia wielkość komunikatu [MB]	Maksymalna wielkość komunikatu [MB]	Minimalny czas przesłania komunikatu [s]	Średni czas przesłania komunikatu [s]	Maksymalny czas przesłania komunikatu [s]	Czas odebrania [s]
127,86	1,32	4	6,11	0,15	0,49	1,31	70,68

Źródło: opracowanie własne.

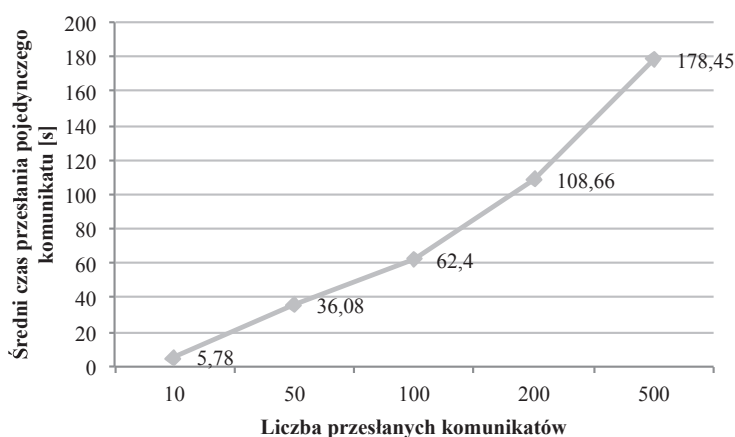
Przeprowadzone testy jasno pokazały, jak duże znaczenie ma odpowiedni dobór sposobu przesyłania dokumentacji przy wykorzystaniu magistrali usług. Ponadto, przeprowadzono test nr 3, którego celem było pokazanie wpływu liczby i rozmiaru przesyłanych komunikatów na czas ich przesyłania. Polegał na przesłaniu komunikatów o wielkości 4,2 MB przy zwiększającej się liczbie jednocześnie przesyłanych komunikatów. Test ten został wykonany za pomocą programu JMeter.

Tabela 3. Wyniki testu nr 3

Liczba komunikatów	Minimalny czas przesłania pojedynczego komunikatu [s]	Średni czas przesłania pojedynczego komunikatu [s]	Maksymalny czas przesłania pojedynczego komunikatu [s]	Wielkość przesłanych wszystkich komunikatów [MB]
10	4,63	5,78	6,10	42
50	25,73	36,08	45,62	210
100	24,83	62,40	92,543	420
200	25,10	108,66	185,02	840
500	26,62	178,45	310,4	2100

Źródło: opracowanie własne.

Ponadto, na rysunku 7 przedstawiono wykres zależności średniego czasu przesłania pojedynczego komunikatu od liczby jednocześnie wysyłanych komunikatów na magistralę usług.



Rysunek 7. Wykres średniego czasu przesłania pojedynczego komunikatu

Źródło: opracowanie własne.

Należy dodać, że powyżej 500 komunikatów pojawiały się ostrzeżenia o przekroczonym czasie połączenia i komunikaty przestały docierać. Może to być związane ze zbyt dużym obciążeniem szyny. Odpowiednio przeprowadzony proces strojenia wydajności mógłby rozwiązać problem. Z przeprowadzonych testów wynika, że wymaganie funkcjonalne polegające na przesłaniu i odebraniu danych za pomocą szyny usług zostało zrealizowane. Częściowa implementacja wzorca architektonicznego oddzielne wywołanie pozwoliła na zwiększenie wydajności, polegające na możliwości obsługi krótkotrwałego zwiększonego obciążenia szyny. W rozważanym przypadku ma to duże znaczenie, ponieważ dane przesyłane są rzadko, ale na stosunkowo krótki czas mogą znacznie ją obciążyć.

10. Podsumowanie i kierunki dalszych badań

W niniejszym artykule przedstawiono na przykładzie Inspekcji Jakości Handlowej Artykułów Rolno-Spożywczych wykorzystanie magistrali usług do obsługi dużych wolumenów danych. Szyny ESB ze względu na duże możliwości integracyjne mogą znacznie wspomóc urzędy administracji państwowej przy budowaniu rozwiązań z zakresu *Big Data*, będąc elementem pośredniczącym w pozyskiwaniu danych z wielu źródeł. Wykorzystanie szyny może wpłynąć na obniżenie kosztów działania podmiotu publicznego, minimalizując liczbę połączeń między systemami, skracając czas integracji z zewnętrznymi systemami, zwiększając możliwość ponownego wykorzystania czy ułatwiając wdrażanie elektronicznej obsługi klienta zewnętrznego.

W przykładowym rozwiązaniu pokazano, jak szybkie może być przekazanie danych drogą elektroniczną. Należy podkreślić fakt, że przesłanie danych standardowo za pomocą poczty zajmuje średnio 3 dni, natomiast przesłanie w wersji elektronicznej zajęło dla przykładowego przypadku niecałe 128 sekund. Jak widać, potencjalne korzyści wdrożenia tego typu rozwiązania są ogromne. Do wad szyn usług można zaliczyć to, że wprowadzają kolejny element opóźnienia wymagający zapewnienia środków bezpieczeństwa. Prostsze przepływy integracyjne łatwiej i szybciej można wdrożyć bez szyny usług.

Dalsze prace ukierunkowane będą na wdrożenie rozwiązania integracyjnego w Inspekcji Jakości Handlowej Artykułów Rolno-Spożywczych w wersji produkcyjnej.

Bibliografia

- Abi-Antoun M., Aldrich J., Nahas N., Schmerl B., Garlan D., *Differencing and merging of architectural views*, „Automated Software Engineering” 2008, vol. 15, s. 35–74.
- Boehm M., Habich D., Lehner W., *On-demand re-optimization of integration flows*, „Information Systems” 2014, vol. 45, s. 1–17.
- Chalmeta R., Pazos V., *A step-by-step methodology for enterprise interoperability projects*, „Enterprise Information Systems” 2015, vol. 9, s. 436–464.
- Chappell D., *Enterprise Service Bus*, O’Reilly, Sebastopol 2004.
- Cicirelli F., Furfaro A., Nigro L., *A service-based architecture for dynamically reconfigurable workflows*, „The Journal of Systems and Software” 2010, vol. 83, s. 1148–1164.
- Erl T., *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall, Crawfordsville 2005.
- Górski T., *Architectural view model for an integration platform*, „Journal of Theoretical and Applied Computer Science” 2012, vol. 6, no. 1, s. 25–34.
- Górski T., *Architektura platformy integracyjnej dla elektronicznego obiegu recept*, „Roczniki” Kolegium Analiz Ekonomicznych, z. 25, Oficyna Wydawnicza SGH, Warszawa 2012.
- Górski T., *Performance Analysis of Selected Frameworks for an Integration Platform Development*, „Bulletin of IT Systems Institute” 2011, vol. 7, s. 9–17.
- Górski T., *Platformy integracyjne. Zagadnienia wybrane*, Wydawnictwo Naukowe PWN, Warszawa 2012.
- Górski T., *UML profiles for architecture description of an integration platform*, „Biuletyn Wojskowej Akademii Technicznej” 2013, nr 2, s. 43–56.
- Martínez-Carreras M. A., García Jimenez F. J., Gómez Skarmeta A. F., *Building integrated business environments: analysing open-source ESB*, „Enterprise Information Systems” 2015, vol. 9, s. 401–435.
- Potena P., *Optimization of adaptation plans for a service-oriented architecture with cost, reliability, availability and performance tradeoff*, „The Journal of Systems and Software” 2013, vol. 86, s. 624–648.
- Rozanski N., Woods E., *Software Systems Architecture. Working with stakeholders using Viewpoints and Perspectives*, Addison-Wesley, Crawfordsville 2005.

* * *

Using Enterprise Service Bus to transfer large volumes of data

Summary

In this article, referring to the example of Agricultural and Food Quality Inspection (www.ijhars.gov.pl), Enterprise Service Bus (ESB) was used to handle transfer of large volumes of data. Due to the high integration capabilities, ESB can greatly assist government agencies

in building solutions in the field of Big Data. The use of EBS may lead to the reduction of operating costs of the public entity minimizing the number of connections between systems, shortening the time of integration with external systems, and increasing the reusability of the written software. The design of integration was done according to the '1+5' architectural views model adjusted to the description of integration solutions. Authorial UML profiles – *UML Profile for Integration Platform* and *UML Profile for Integration Flows* – have been used. The paper presents a solution showing how fast data can be transferred electronically using the ESB.

Keywords: Enterprise Service Bus, IT systems integration, '1+5' architectural views model