ZESZYTY NAUKOWE WYDZIAŁU ETI POLITECHNIKI GDAŃSKIEJ
Nr 7 Seria: Technologie Informacyjne 2009

Henryk Krawczyk, Paweł Lubomski

Politechnika Gdańska

ARCHITEKTURY SYSTEMÓW INFORMATYCZNYCH WSPOMAGAJĄCYCH ROZWÓJ E-UCZELNI

Streszczenie

Przedstawiono charakterystykę podstawowych kategorii architektur systemów informatycznych. Zaprezentowano koncepcję środowiska zintegrowanego e-uczelni opartą na architekturze zorientowanej na usługi dostępne dla poszczególnych kategorii użytkowników. Uzasadniono wybór takiego podejścia oraz przeanalizowano aspekty bezpieczeństwa. Omówiono również zasady integracji istniejących już systemów.

1. WSTĘP

Rynek edukacyjny wymaga, by dobra uczelnia była atrakcyjna, wyróżniała się wśród innych możliwością reformowania struktury i doskonalenia oferty edukacyjnej. Musi być również przyjazna dla studenta i pracownika, dostosowana do jego indywidualnych potrzeb i oczekiwań [1]. To ukierunkowanie na konkretnego użytkownika wymusza wnikliwe analizy posiadanych danych z szerokiego spektrum: od potencjalnych zainteresowań kandydatów, przez jakość, aktualność i przydatność przedmiotów, po liczbę spełniających się w swoim zawodzie absolwentów.

Obserwujemy powszechną informatyzację kolejnych dziedzin życia. Coraz więcej formalności możemy załatwić elektronicznie w tzw. e-urzędach [2]. Ułatwia to znacznie obsługę zleceniodawcy, nie mówiąc już o zaoszczędzonym czasie pracownika i zainteresowanego. Wprowadzenie komputerów i digitalizacja danych zdecydowanie usprawnia zarządzanie wielką ilością informacji. Analogiczna potrzeba dotyczy każdej większej organizacji, gdzie współpracuje ze sobą dużo osób. Proces informatyzacji dotyczy również uczelni, w których procesy kształcenia i badań są wspomagane systemami komputerowymi.

Kręgosłupem każdego takiego systemu jest odpowiednia architektura. W "Ilustrowanej encyklopedii dla wszystkich" termin architektura jest objaśniany jako sztuka projektowania i kształtowania budowli [3]. Natomiast według Witruwiusza architektura polega na zachowaniu trzech zasad: trwałości (*Firmitas*), użyteczności (*Utilitas*) i piękna (*Venustas*) [4]. Te dwie definicje obrazują konieczność kompleksowego spojrzenia na cały system

informatyczny, jego ogólną budowę i zasadę funkcjonowania, tak aby mógł on funkcjonować wiele lat oraz był użyteczny i przyjazny użytkownikowi. Podobnie jak z budynkami, system informatyczny nie uniknie wprowadzania zmian. Prawidłowo przygotowana architektura pozwoli na przeprowadzenie ich bez uszczerbku dla wspomnianych wyżej zasad. Również sama architektura musi często ewaluować i być doprecyzowywana. Fakt ten implikuje taką jej otwartość, aby potrafiła sprostać nowym wyzwaniom funkcjonalnym i technologicznym.

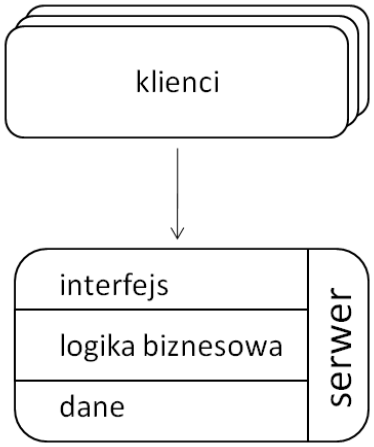
Zorganizowanie takiego ogromnego środowiska wspierającego to bardzo poważne wyzwanie. W niniejszym opracowaniu przedstawiono próbę podjęcia tego typu zadania przy wykorzystaniu najnowocześniejszych technologii informacyjnych.

2. PODSTAWOWE KATEGORIE ARCHITEKTUR SYSTEMÓW INFORMATYCZNYCH

Projektując architekturę dużego systemu informatycznego, należy rozpatrzyć dwa aspekty tej architektury: sprzętową i logiczną. W dobie obecnego rozwoju technologii nie są one tożsame. Istnieje wiele rozwiązań, które pozwalają umieścić wiele jednostek logicznych na jednej sprzętowej. Również odwrotna sytuacja jest możliwa – w sposób transparentny dla jednostki logicznej jest ona umieszczana na kilku jednostkach sprzętowych, co poprawia jej wydajność. Nie ma większego problemu ze zmianą w czasie konfiguracji umieszczenia jednostek logicznych na sprzętowych. Większym wyzwaniem jest architektura logiczna, której wybór będzie bardzo trudno modyfikowalny w czasie. W związku z tym należy poświęcić jej więcej uwagi.

2.1 Architektury scentralizowane

Historycznie najstarszym podejściem jest architektura scentralizowana. Cała logika biznesowa systemu informatycznego oraz dane, na których on operuje, znajdują się w jednej jednostce logicznej – serwerze. Często jest ona podzielona na wiele warstw – są to tzw. aplikacje wielowarstwowe (*ang. n-tier*).



Rys. 1. Architektura scentralizowana

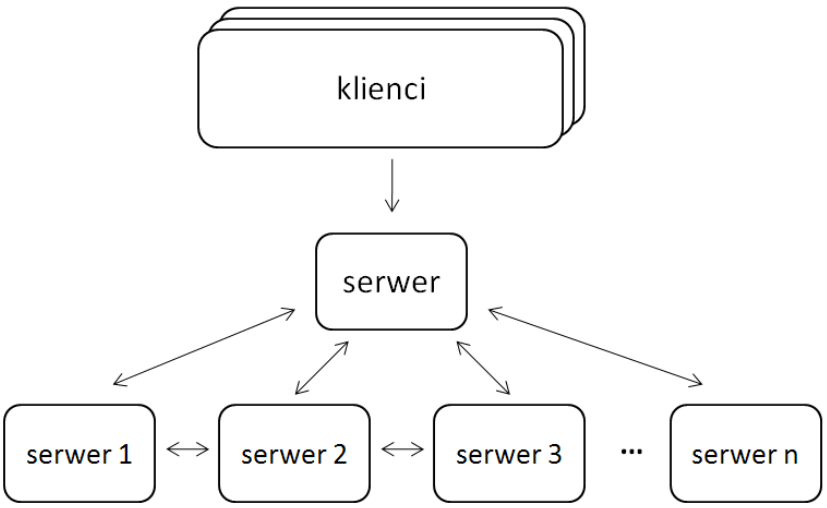
Są to w większości modele typu klient-serwer. Typowym przykładem jest serwer stron WWW, z którym łączą się poszczególni klienci – przeglądarki WWW użytkowników. Można rozróżnić systemy z chudym lub grubym klientem. Rys. 1 prezentuje typową jednostkę centralną trójwarstwową i łączących się z nią klientów.

W podejściu tym nie występują problemy z rozproszeniem danych, ani z komunikacją. Wymagana jest natomiast bardzo wydajna i niezawodna jednostka centralna, której awaria powoduje niedostępność całego systemu informatycznego.

Systemy tego typu to najczęściej spotykane aplikacje. Architektura ta posiada wsparcie w wielu aktualnych, żywych technologiach takich jak JSP, ASP.NET, PHP. Ze względu na swoją specyfikę stosowana jest w małych i średnich systemach informatycznych, co wyklucza zastosowanie jej w systemie wspierającym kompleksowo pracę e-uczelni. Cechuje się również jednolitością technologii – całość systemu musi być wykonana w jednej, obranej na etapie planowania.

2.2 Architektury rozproszone

Wraz ze wzrostem wielkości i stopnia skomplikowania systemów informatycznych, architektury scentralizowane nie były w stanie zapewnić odpowiedniego stopnia niezawodności i wydajności systemom, którym zaczęto powierzać stopniowo coraz bardziej odpowiedzialne zadania i dane. Opracowano więc architektury rozproszone. Są one analogią ludzkiej pracy w zespołach – większych zadań nie da się wykonać w pojedynkę, bo wykonanie takiej ilości pracy przez jedną osobę w tak krótkim czasie jest fizycznie niemożliwe. Tylko dobrze dobrany i zgrany zespół potrafi wykonać duże zadania.



Rys. 2 Architektura rozproszona

Architektury te charakteryzują się rozproszeniem części serwerowej – delegacją logiki biznesowej i zarządzanych danych na osobne jednostki logiczne. Można zaobserwować hierarchię (rys. 2) – komunikacja pomiędzy jednostkami logicznymi odbywa się zarówno w płaszczyźnie pionowej (*ang. vertical*), jak i płaszczyźnie poziomej (*ang. horizontal*). Na

rys. 2 jednostki te przedstawiono jako osobne serwery, ponieważ bardzo często są one lokalizowane na różnych jednostkach sprzętowych. W większości przypadków jest to realizacja modelu klient-pośrednik-serwer.

Rozwiązanie takie pozwala na usprawnienie niezawodności systemu, ponieważ luźne powiązanie poszczególnych jednostek logicznych uniezależnia je od siebie i pozwala pracować pozostałym, nawet gdy jedna z nich ulegnie awarii. Niestety korzyści te są obarczone kosztem komunikacji oraz zarządzania rozproszonymi danymi.

2.2.1 Architektura danych w systemach rozproszonych

Planując architekturę rozproszoną, należy rozważyć wiele różnych aspektów budowanego systemu. Podstawowym z nich jest sposób rozproszenia danych [5]. Rozróżniamy dane specyficzne dla danego systemu składowego, jak np. dane o zarobkach pracowników czy oceny studentów. Istnieją jednak również dane wspólne dla kilku lub wszystkich systemów składowych, jak np. dane personalne osób lub dane słownikowe. Zależnie od podjętej decyzji dane wspólne możemy pobierać „w locie" (ang. *on-line*) lub replikować. Każde z tych dwóch rozwiązań posiada swoje wady i zalety. W pierwszym przypadku uzyskujemy pełną spójność i stuprocentową aktualność danych. Wiąże się to jednak z obniżeniem wydajności systemu o koszt komunikacji. Dodatkowo wzrasta zawodność systemu, ponieważ awaria systemu macierzystego, udostępniającego dane, powoduje niedostępność funkcjonalności pracujących na tych danych w systemach zależnych. Drugie rozwiązanie nie posiada wspomnianych wcześniej wad, jednak wymaga opracowania bardzo szczegółowej polityki synchronizacji tychże danych z uwzględnieniem mechanizmów rozwiązywania konfliktów. Pojawia się również problem aktualności danych.

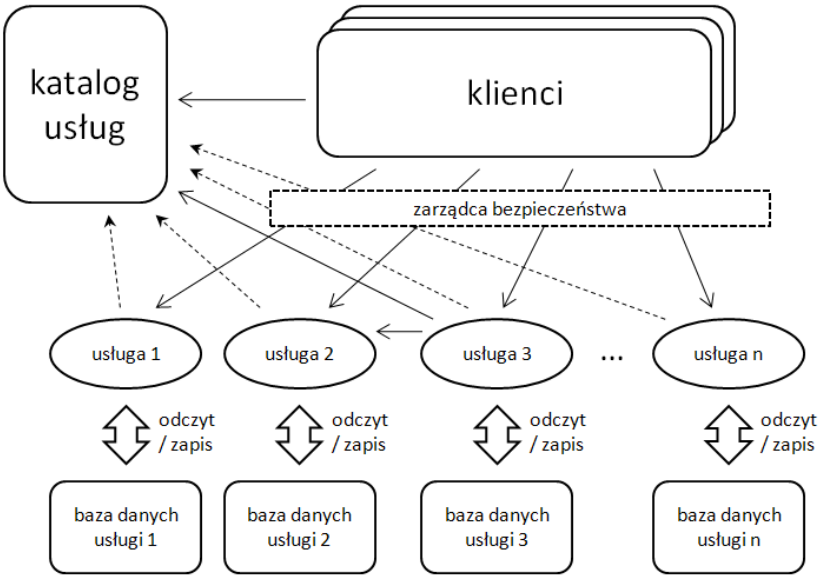
Nie ma jednego, prawidłowego i najlepszego rozwiązania. Każdy przypadek powinien zostać rozpatrzony indywidualnie i dla niego powinna zostać podjęta niezależna decyzja. W większości przypadków decyzja będzie podejmowana osobno nie dla całego systemu, a dla podzbioru danych związanych z danym komponentem systemu.

2.3 Architektury zorientowane na usługi

Zdobywającym coraz większą popularność rozwiązaniem jest architektura zorientowana na usługi (ang. *SOA – Service Oriented Architecture*) [6]. Jest to odwzorowanie normalnych ludzkich zachowań. Nieopłacalnym przedsięwzięciem jest budowanie całej manufaktury, żeby wytworzyć sobie kilka egzemplarzy jakiegoś produktu tylko na swoje potrzeby. Podobnie niemożliwością jest wykształcenie się we wszystkich zagadnieniach związanych z funkcjonowaniem systemu w stopniu eksperckim. W związku z tym zamawia się konkretne produkty u dostawcy-specjalisty zorientowanym w danym zagadnieniu. Rozważając to zagadnienie szerzej, można powiedzieć, iż zamawiane są usługi na wykonanie konkretnego zadania lub produktu materialnego.

Analogiczne podejście można zastosować, projektując architekturę dużego systemu lub nawet całego środowiska zintegrowanego. Im jest on lub ono większe, tym podejście to jest zasadniejsze. Tworzymy wiele systemów-serwisów, dostawców usług, które na konkretne żądanie produkują odpowiedni zestaw danych. Drugim elementem tej struktury są systemy klienckie, które korzystają z wybranych usług (na zasadzie P2P – *ang. peer-to-peer*), często wielu różnych, dla uzyskania konkretnych danych. Systemami klienckimi

mogą być serwisy WWW (do których użytkownik łączy się przez internet) [7], pozostałe systemy uczelniane lub inne systemy typu „gruby klient". Podejście to obrazuje schemat na rys. 3. Zakłada się przy tym, że kolejne usługi są wywoływane przez użytkownika (użytkowników), który w sposób interaktywny (a nie automatyczny) kreuje odpowiednie scenariusze działań (procesy biznesowe). To znacznie upraszcza systemy i daje szansę modyfikacji w przypadku dynamicznie pojawiających się zmian wymagań.



Rys.3. Architektura zorientowana na usługi.

Warto zauważyć, że nie wyklucza się również możliwości interakcji pomiędzy samymi usługami. Często zdarza się tak, że wołana usługa nie posiada wystarczających danych, aby móc wygenerować żądany wynik. W takim przypadku odpytuje ona inną, która jest w stanie dostarczyć jej brakujące informacje. Takie podejście daje możliwość budowy systemów uczelnianych „z dołu do góry".

W dużych środowiskach zintegrowanych opartych na usługach, gdzie dostępnych jest wiele usług jednocześnie, pojawia się problem zarządzania nimi. Niezbędne jest wprowadzenie katalogu usług, który będzie ułatwiał wyszukiwanie potrzebnej usługi. Koncepcję tę zaprezentowano na rys. 3. Warto zauważyć, że z katalogu mogą korzystać zarówno klienci, jak i inne usługi, które potrzebują skomunikować się ze swoim dostawcą dodatkowych danych. Sprawą podstawową jest określenie zbioru usług niezbędnych dla funkcjonowania konkretnej uczelni. Sprawą następną jest ich utworzenie i zgłoszenie do katalogu usług (na rys. 3 zaprezentowane przerywaną linią).

Podjęto prace nad standaryzacją takiego katalogu. Ich efektem jest chociażby specyfikacja UDDI [8]. Ciekawym zagadnieniem wydaje się być zastosowanie ontologii do opisu usług. Pozwoliłoby to na łatwiejsze zarządzanie tymi usługami, jak i korzystanie z katalogu.

2.4 Proces budowania architektury dużego systemu informatycznego

Docelowa architektura często jest wynikiem ewolucji oprogramowania. Niezależne minisystemy wprowadzają komunikację pomiędzy sobą, zastępując wolniejszy „papierowy" przepływ danych. Z czasem ta komunikacja jest tak intensywna i różnorodna, że wymaga uporządkowania i zabezpieczenia. Wprowadzane są pewne standardy komunikacji oraz formaty samych danych. W tym momencie możemy mówić już o środowisku zintegrowanym. Taki ewolucyjny proces posiada jednak jeden poważny mankament: architektura często nie jest optymalna, ponieważ jest dostosowywana do istniejących już rozwiązań częściowych.

Zdecydowanie łatwiejszym i bardziej efektywnym podejściem jest budowanie nowych systemów, zgodnych z przyjętymi w pierwszej kolejności założeniami dotyczącymi kompleksowej architektury [9]. Wiąże się to z zawężeniem dostępnych technologii do pewnego podzbioru, co jest wadą tej metodologii. Jednakże korzyści przeważają, ponieważ system jest wydajniejszy, bardziej niezawodny i bezpieczniejszy – ograniczone jest ryzyko konfliktów na styku technologii oraz potrzeby kosztownego (czasowo) mapowania pomiędzy nimi. Kompleksowa architektura pozwala również na globalne zarządzanie bezpieczeństwem.

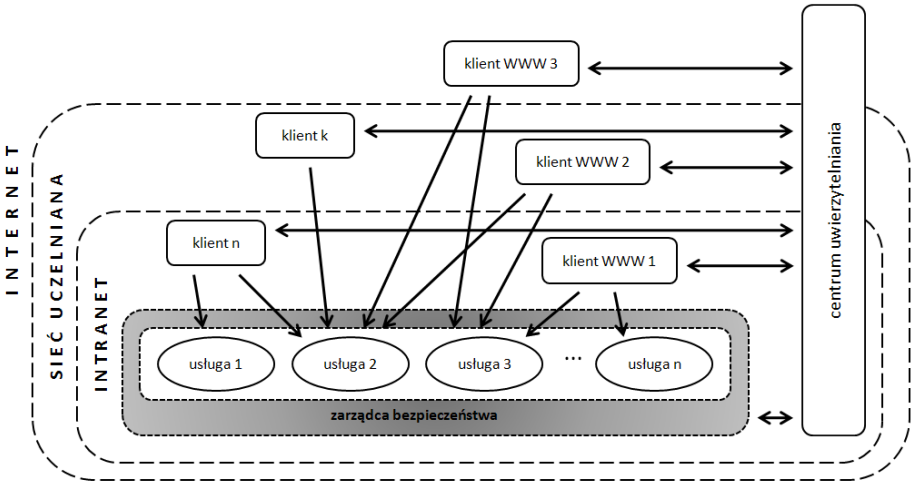
Oczywiście zawsze sukces realizacji systemu rozproszonego w dużej mierze zależy od posiadania wykwalifikowanego architekta środowiska zintegrowanego lub, zależnie od wielkości środowiska, zespołu takich architektów. Tacy specjaliści powinni posiadać spore doświadczenie z eksploatacji środowisk rozproszonych, jak również cechować się dużym stopniem wizjonerstwa i zdolnością przewidywania kierunków rozwoju tego typu systemów.

3. ZAPEWNIANIE BEZPIECZEŃSTWA W ŚRODOWISKU USŁUG

Projektując rozproszone środowisko usług uczelnianych, bardzo dużo uwagi należy poświęcić bezpieczeństwu. Ponieważ usługi są niezależnymi podsystemami i mogą być zlokalizowane nawet na różnych fizycznych serwerach, kontrolowane powinny być wszystkie połączenia do każdej z usług, a więc nie można założyć jednego wejścia do środowiska, ale tyle, ile wynosi całkowita liczba usług [10].

3.1 Autoryzacja i uwierzytelnianie

Zarządzanie tak rozbudowanym i rozproszonym systemem uprawnień w sposób tradycyjny [11] byłoby bardzo skomplikowane i trudne w utrzymaniu. Z tego powodu zaproponowano użycie zarządcy bezpieczeństwa (rys. 4) zbudowanego w oparciu o wzorzec projektowy Front Controller [12]. Zgodnie z rys. 4 każde żądanie do usługi będzie kontrolowane przez zarządcę bezpieczeństwa na zasadzie filtru.



Rys.4. Zarządca bezpieczeństwa i centrum uwierzytelniania.

Kontrola odbywać się będzie dwustopniowo:

1. W pierwszej kolejności zostaną zweryfikowane uprawnienia danego klienta, czy posiada on dostęp do danej usługi. Zakłada się, że tylko skończona liczba klientów będzie autoryzowana za pomocą certyfikatów wydawanych przez zarządcę bezpieczeństwa.
2. W drugiej kolejności sprawdzane będą uprawnienia użytkownika systemu klienckiego, czy jest on uprzywilejowany do wykonania danej operacji. Dane identyfikacyjne będą przechowywane tylko w jednym miejscu – centrum uwierzytelniania. To tam klient będzie musiał uwierzytelnić użytkownika przed wywołaniem usługi, co później zostanie zweryfikowane przez zarządcę bezpieczeństwa.

Dopiero po poprawnym uwierzytelnieniu systemu klienckiego i użytkownika, zarządca bezpieczeństwa dokona autoryzacji żądanej operacji i, w przypadku powodzenia, zezwoli na jej wykonanie.

3.2 Separacja podsieci

Nie istnieją aplikacje i systemy komputerowe w 100% pozbawione luk i błędów. Udostępnienie wszystkich usług w publicznej sieci, jaką jest internet, stwarzałoby duże zagrożenie, szczególnie na uczelni technicznej. W celu zminimalizowania ryzyka wprowadzono logiczny podział sieci komputerowej na 3 podsieci zaprezentowane na rys. 4: intranet (sieć specjalizowaną, do której podłączone są wydzielone komputery), sieć uczelnianą (obejmującą wszystkie komputery w obrębie uczelni) oraz internet. Wszystkie usługi i zarządca bezpieczeństwa zlokalizowane są w najbardziej restrykcyjnym intranecie. Klienci usług zostaną podzieleni i przypisywani do jednej z podsieci zależnie od zadań, jakie mają realizować. Przewiduje się stworzenie 3 podobnych klientów WWW po jednym na każdą podsieć. Jednak zależnie od podsieci będą oni posiadali różne uprawnienia – im bardziej „na zewnątrz", tym bardziej restrykcyjne. W internecie dostępne będą głównie systemy informacyjne uczelni.

Całość koncepcji spina odpowiednia konfiguracja przełączników sieciowych tak, aby dla zewnętrznych podsieci „widoczny" był tylko zarządca bezpieczeństwa. To on będzie pełnił rolę „bramki do usług".

4. ZAKOŃCZENIE

Zagadnienie informatyzacji uczelni jest niezwykle złożone. Wymaga dogłębnej analizy stanu obecnego oraz potrzeb. Kluczem do sukcesu jest fachowy zespół architektów o właściwych predyspozycjach oraz dobrze dobrana architektura systemu.

W obecnej dobie rozwoju technologii informacyjnych można zaobserwować tendencję do decentralizacji większości zagadnień i poczynań ludzkich. Trend ten zauważalny jest również w inżynierii oprogramowania. Odchodzi się od monolitycznych, dużych i jednoelementowych systemów na rzecz integracji rozproszonych, luźniej powiązanych komponentów. Takie komponenty współpracujące ze sobą w celu realizacji wspólnego celu określa się mianem systemów rozproszonych [5].

Podejście takie owocuje wieloma korzyściami. Przede wszystkim pozwala na dystrybucję pracy i odpowiedzialności pomiędzy różne zespoły. Każdy z komponentów jest niezależny i może być wykonany w innej technologii. Powoduje to potrzebę wypracowania technik komunikacji pomiędzy tak różnymi technologiami. Niezależność komponentów niesie ze sobą wzrost niezawodności, ponieważ awaria jednego nie implikuje niedostępności całości systemu. Jest to bardzo istotny czynnik w systemach pracy ciągłej, dostępnych przez znaczną część dobry przez prawie wszystkie dni w roku. Co więcej złożoność poszczególnych komponentów jest mniejsza i łatwiej jest je w znacznie większym stopniu przetestować.

Przedstawione podejście, oparte na bazie usług, wydaje się dobrze pokrywać ze sferami informatyzacji uczelni wyższej. Pozwala również na właściwą kontrolę dostępu do poszczególnych danych. Bardzo dużo uwagi należy poświęcić zagadnieniom związanym z bezpieczeństwem. Jest ono kluczowe ze względu na wrażliwość przechowywanych i przetwarzanych danych. Rozmiar organizacji oraz skala przedsięwzięcia nie pozwalają zapomnieć o aspekcie wydajnościowym, który w przedstawionym rozwiązaniu został osiągnięty za pomocą skalowalności, modularności i łatwości rozproszenia systemu.

Tego typu rozwiązanie jest realizowane na Politechnice Gdańskiej. Jednolity, ale kontrolowany, dostęp do usług dla różnych kategorii użytkowników (studenci, pracownicy, emeryci) umożliwia dostarczenie znacznie większej funkcjonalności w porównaniu z systemami tradycyjnymi zajmującymi się obsługą tylko poszczególnych kategorii użytkowników.

BIBLIOGRAFIA

[1] Dziubich, T., Lubomski, P., Mizgier, A.: *Architektura portalu zarządzania informacjami dydaktycznymi*, Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej, 2008.
[2] Kowalczyk M.: *E-Urząd w Komunikacji z Obywatelem*, Wydawnictwa Akademickie i Profesjonalne, 2009.
[3] Szolginia W.: *Ilustrowana encyklopedia dla wszystkich*, Wydawnictwa Naukowo-Techniczne, 1991.
[4] Witruwiusz: *O architekturze ksiąg dziesięć*, Prószyński i S-ka, 2004.

[5] Coulouris G.: *Systemy rozproszone: podstawy i projektowanie*, Wydawnictwa Naukowo-Techniczne, 1999.
[6] Fryźlewicz Z., Salamon A.: *Podstawy architektury i technologii usług XML sieci WEB*, Wydawnictwo Naukowe PWN, 2008.
[7] Rosen R., Shklar L.: *Web Application Architecture: Principles, Protocols and Practices*, Wiley, 2003.
[8] Zakrzewicz M.: *Wprowadzenie do technologii Web Services: SOAP, WSDL i UDDI*, Materiały XIII Seminarium PLOUG: Web Services: Projektowanie i implementacja architektur zorientowanych na usługi, 2006.
[9] Clements P., Kazman R., Klein M.: *Evaluating Software Architectures: Methods and Case Studies*, MA: Addison-Wesley, 2001.
[10] Andrzejewski W., Zakrzewicz M.: *Problematyka bezpieczeństwa usług Web Services*, Materiały XIII Seminarium PLOUG: Web Services: Projektowanie i implementacja architektur zorientowanych na usługi, 2006.
[11] Szpryngier P.: *Bezpieczeństwo aplikacji internetowych*, materiały wykładowe, 2006.
[12] Crawford W., Kaplan J.: *J2EE Design Patterns*, O'Reilly, 2003.

THE ARCHITECTURES OF INFORMATION SYSTEMS SUPPORTING THE DEVELOPMENT OF E-UNIVERSITY

Summary

The paper presents the characteristics of the main categories of information systems architectures. It describes the service-oriented architecture of distributed system which integrates many service-based smaller systems available for the different categories of users, based on situation of Gdansk University of Technology. Security, safety and efficiency aspects of this system are also mentioned.