

Henryk MAŁYSIAK, Marcin BONK  
Politechnika Śląska, Instytut Informatyki

## ZDALNE WSPOMAGANIE NAUCZANIA NA PRZYKŁADZIE BUDOWY I DZIAŁANIA MIKROPROCESORÓW IA-64

**Streszczenie.** Dostęp do informacji przedstawianych w Internecie jest dziś powszechny. Stwarza to interesujące możliwości wspomaganie nauczania. W pracy przedstawiono przykład rozwiązania systemu wspomagającego nauczanie. Umożliwia on rejestrację użytkowników i grup, wygodny dostęp do informacji, łatwe modyfikowanie bazy wiedzy, niezbędną ochronę zasobów oraz dyskusje między użytkownikami.

**Słowa kluczowe:** zdalne nauczanie, Internet, mikroprocesory IA-64, systemy wspomagające nauczanie.

## EXAMPLE OF DISTANCE AIDED TEACHING OF STRUCTURE AND WORKING OF IA-64 MICROPROCESSORS

**Summary.** Access to information presented in Internet is general today. It giving interesting possibilities of support of teaching. In the article the example of solution of system aided teaching was presented. It makes possible registration of users and groups, simple access to information, simple modifications of knowledge base, necessary security of resources and discussions between users.

**Keywords:** distance aided teaching, Internet, IA-64 microprocessors, systems aided teaching.

### 1. Wprowadzenie

Edukacja za pośrednictwem Internetu, tak popularna w wielu krajach świata, zaczyna zdobywać należne sobie miejsce również w naszym kraju. Zalety takiego zdalnego kształcenia są powszechnie znane. Nauczanie tego typu może być:

- aktywne - zapewniające kontakt z nauczycielem i innymi uczestnikami kursu w czasie rzeczywistym (komunikacja synchroniczna). Jest ono realizowane na zasadzie wideokonferencji,
- bierne - zapewniające kształcenie na podstawie zdalnie udostępnianych materiałów. Komunikacja z nauczycielem lub innymi uczestnikami kursu jest w tym przypadku asynchroniczna, korzysta się np. z poczty elektronicznej czy grup dyskusyjnych.

Zastosowanie Internetu jako medium informacyjnego wynika z jego popularności, coraz większej dostępności oraz szybko rozwijających się narzędzi programowych i infrastruktury tej sieci. Hipertekst w połączeniu z obrazami, animacjami, dźwiękiem i elementami interaktywnymi umożliwia tworzenie efektownych prezentacji materiałów edukacyjnych. Zastosowanie technologii strumieniowych (np. Real Player) pozwala użytkownikom na płynne odtwarzanie materiału audiowizualnego nawet na łączu o niższej przepustowości. Rozwój technik kryptograficznych oraz wysoki poziom bezpieczeństwa dzisiejszych systemów umożliwiają organizowanie zdalnych kursów i rejestrowanie uczestników bez ryzyka poznania poufnych danych przez osoby trzecie.

Do bardziej popularnych systemów zdalnego nauczania można zaliczyć Smart Force firmy SkillSoft [4], iLearning firmy Oracle [4] oraz Learning Space firmy Lotus [4]. Analiza istniejących systemów udostępniania wiedzy pozwoliła sprecyzować założenia niezbędne do realizacji prezentowanego rozwiązania.

## 2. Koncepcja realizacji systemu

Przy projektowaniu systemu przyjęto następujące założenia [1]:

- otwarta, modułowa architektura systemu ułatwiająca jego rozbudowę,
- przyjazny interfejs użytkownika,
- możliwość korzystania z serwisu niezależnie od sprzętu i systemu operacyjnego,
- zapewnienie bezpieczeństwa dostępu do danych na poziomie wymaganym w tego typu systemach,
- realizacja systemu w oparciu o narzędzia z licencją open source lub udostępniane darmowo dla celów edukacyjnych.

System zdalnego wspomagania nauczania powinien umożliwiać samodzielną naukę z możliwością sprawdzenia stopnia opanowania wiedzy za pośrednictwem nowoczesnych technologii przekazu określonych zasobów informacji. Część tych zasobów może być dostępna publicznie, a część może wymagać rejestracji i uzyskania określonych uprawnień. W projektowanym systemie przyjęto, że tworzenie nowych zasobów wiedzy powinno być

proste i dokonywane z poziomu interfejsu aplikacji. Tworzone dokumenty powinny mieć standardowy format. Przy ich opracowaniu można korzystać z języków HTML, DHTML, Java Script oraz arkuszy styli CSS. W prezentowanym rozwiązaniu wprowadzono dwa rodzaje dokumentów:

- oparte na jednej głównej stronie www, na której mogą znajdować się odnośniki do innych stron,
- składające się z uporządkowanego ciągu stron tworzących prezentację z elementami ułatwiającymi nawigację po nich.

Każdy dokument posiada swojego autora, nazwę i opis określający kategorię, do której należy. W systemie występują dokumenty dostępne dla wszystkich użytkowników (publiczne), dla użytkowników zarejestrowanych i użytkowników posiadających specjalne uprawnienia nadane przez administratora serwisu.

Użytkownikom mogą być przydzielane role. Użytkownik może pełnić kilka ról jednocześnie. Aktualnie dostępne są następujące role:

- gość, otrzymuje ją każdy odwiedzający witrynę przed zalogowaniem się, może on przeglądać zasoby publiczne,
- użytkownik, otrzymuje ją każda osoba zarejestrowana, może ona tworzyć własne zasoby, dyskutować na forum, należeć do grupy,
- menadżer grupy, ta rola pozwala utworzyć i zarządzać jedną grupą użytkowników,
- administrator, posiada najwyższe z możliwych uprawnienia.

### 3. Projekt systemu

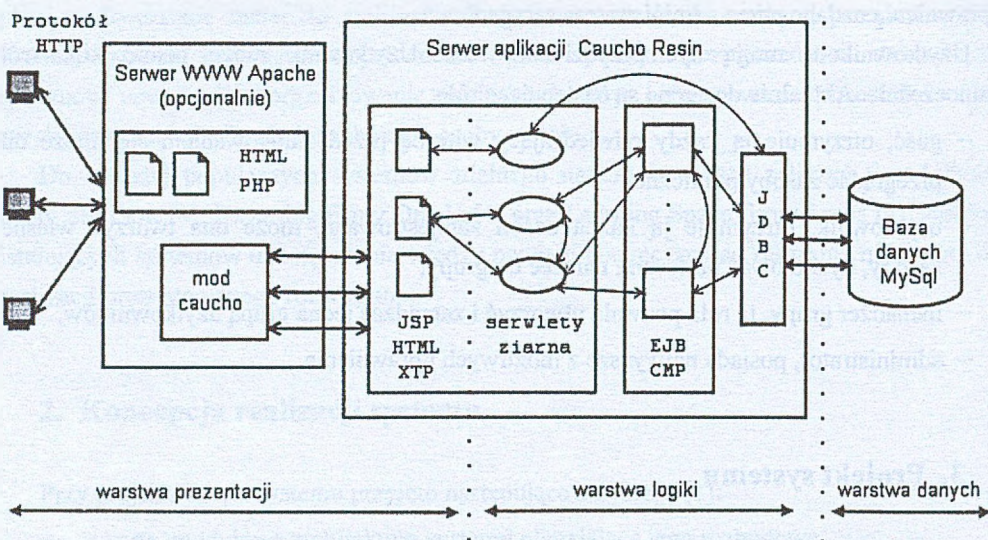
System posiada architekturę trójwarstwową. Poszczególne warstwy systemu przedstawiono na rys. 1.

Warstwa prezentacji zawiera elementy związane z wizualizacją danych u użytkowników. W jej skład wchodzi:

- przeglądarka internetowa u klienta stanowiąca platformę dla graficznego interfejsu użytkownika. Wymagane jest użycie przeglądarek obsługujących poprawnie DHTML, JavaScript oraz arkusze styli CSS. Założenia te spełniają między innymi Internet Explorer 5+, Netscape Navigator 6+ oraz Mozilla,
- opcjonalnie serwer WWW. Przewidziano tutaj użycie najbardziej popularnego serwera WWW Apache, rozpowszechnianego za darmo na licencji open source. Serwer ten cechuje się bardzo wysoką wydajnością i stabilnością, ponadto ma możliwość instalowania dodatkowych modułów. Zainstalowanie pakietu mod\_caucho pozwala na

komunikację z kontenerem sieciowym i przekazywanie mu określonych żądań od klientów. Element ten nie jest konieczny, ponieważ zastosowany serwer aplikacji posiada wbudowaną funkcjonalność serwera HTTP. Jednak takie rozwiązanie pozwala na wykorzystanie ciekawych mechanizmów, takich jak dynamiczne równoważenie obciążenia. Ponadto umieszczenie statycznych plików HTML i zaimplementowanie szyfrowania SSL (ang. *Secure Socket Layer*) na serwerze WWW może zaowocować lepszą wydajnością,

- strony JSP, HTML, XTP udostępniane w serwerze aplikacji (Caucho Resin) zawierające dynamiczną zawartość generowaną przez skrypty w języku Java. Stanowią one szablony prezentujące dane uzyskane w wyniku odwołań do warstwy logiki.



Rys. 1. Architektura systemu  
Fig. 1. System architecture

Warstwę logiki tworzy serwer aplikacji Caucho Resin. Zawiera ona obiekty reprezentujące operacje możliwe do przeprowadzenia w systemie oraz umożliwiające dostęp do danych. Komponenty te są udostępniane w serwerze aplikacji. Są to:

- serwlety – do zarządzania obiektami systemu, generowania elementów dynamicznych,
- komponenty CMP – stanowiące obiektowy interfejs do bazy danych,
- ziarna JavaBeans – pełniące funkcje filtrów w forum dyskusyjnym.

Do warstwy logiki zalicza się również sterownik JDBC dla bazy danych. W systemie użyto sterownika dostarczanego z serwerem aplikacji, który pisany był pod kątem jak największej wydajności. W przypadku jednoczesnego dostępu do bazy danych wielu użytkowników występuje problem zachowania spójności danych. Jeżeli np. jakiś rekord

zostałby zmodyfikowany przez kilku użytkowników jednocześnie, otrzymany rezultat mógłby być błędny. Sterownik firmy Caucho rozwiązuje ten problem, implementując mechanizmy transakcji, mimo tego że zastosowana baza danych transakcji nie wspiera. Dla obiektów CMP serwer aplikacji, (kontener sieciowy) automatycznie dba o spójność, zwalniając programistę z konieczności ręcznego synchronizowania operacji.

Warstwę danych tworzy serwer relacyjnej bazy danych MySQL. Wszystkie operacje są wykonywane za pośrednictwem sterownika JDBC. Dane mogą być pobierane przez komponenty CMP, bezpośrednio z serwleta lub ziarna JavaBean.

Jak wynika z rys. 1, zastosowana architektura zapewnia dobrą skalowalność i łatwość rozbudowy systemu. Istnieje możliwość rozproszenia systemu dla serwisów wymagających dużych wydajności.

Serwer Caucho Resin zapewnia sprzętowe równoważenie obciążenia (ang. *hardware load balancing*). Jest to mechanizm, który przewiduje możliwość zastosowania kilku serwerów aplikacji działających na różnych komputerach. Żądania od klientów są rozdzielane pomiędzy te serwery (z uwzględnieniem faktu, iż połączenia dotyczące jednej sesji muszą być przekazywane do tego samego serwera). Dynamiczne równoważenie obciążenia między poszczególne serwery może być realizowane na dwa sposoby:

- za pośrednictwem serwletu *LoadBalanceServlet*, dotyczy to przypadku systemu bez zewnętrznego serwera WWW,
- z użyciem pluginu *mod\_caucho* w systemie z serwerem Apache.

Istnieje również możliwość uruchomienia dwóch instancji (podstawowej i zapasowej) serwera Resin na jednej maszynie. Dzięki temu, jeśli zawiedzie pierwsza z nich, cały ruch jest kierowany na drugą. Można także skonfigurować współpracę kilku zewnętrznych serwerów WWW z jednym serwerem aplikacji.

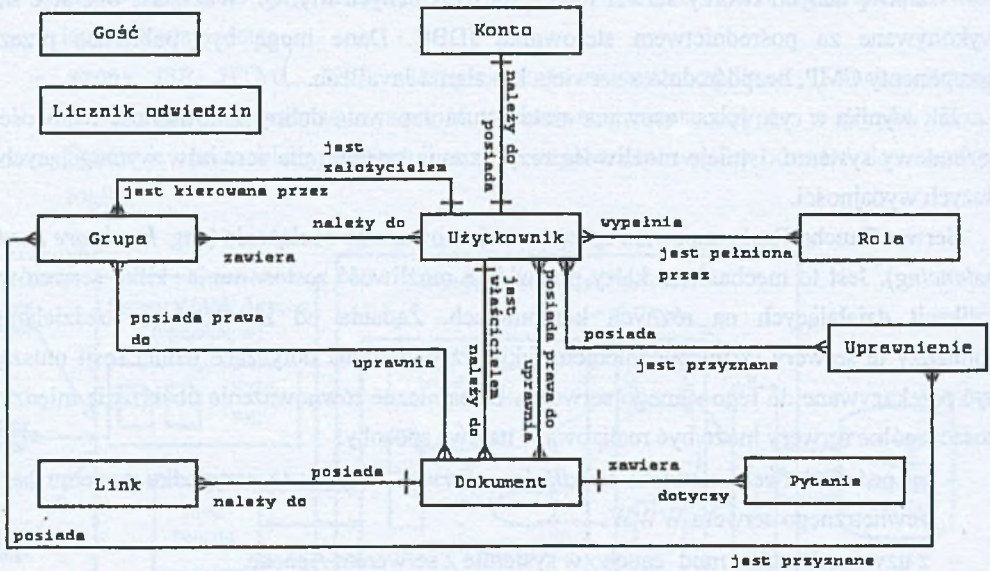
Baza danych i komponenty EJB mogą być umieszczone na oddzielnych maszynach, przez co odciążony zostaje serwer podstawowy. Modułowość aplikacji i rozdzielenie warstw upraszcza dodawanie nowych komponentów i funkcji systemu. Struktura i metody dostępu do danych zgromadzonych na serwerze bazy MySQL, jak i plików znajdujących się w przestrzeni dyskowej, ma istotny wpływ na wydajność systemu. Przyjęty w rozwiązaniu model logiczny danych przedstawiono na rys. 2.

Użytkownik stanowi w nim podstawową encją systemu powiązaną następującymi relacjami:

- posiada jedno konto,
- może należeć do wielu grup,
- może zarządzać wieloma grupami,
- może pełnić wiele ról,
- może posiadać wiele uprawnień.

Dokument zaś jest encją powiązaną następującymi relacjami:

- ma jednego właściciela,
- może posiadać wiele linków,
- można do niego przydzielać prawa dla użytkowników i grup,
- może zawierać pytania testowe.

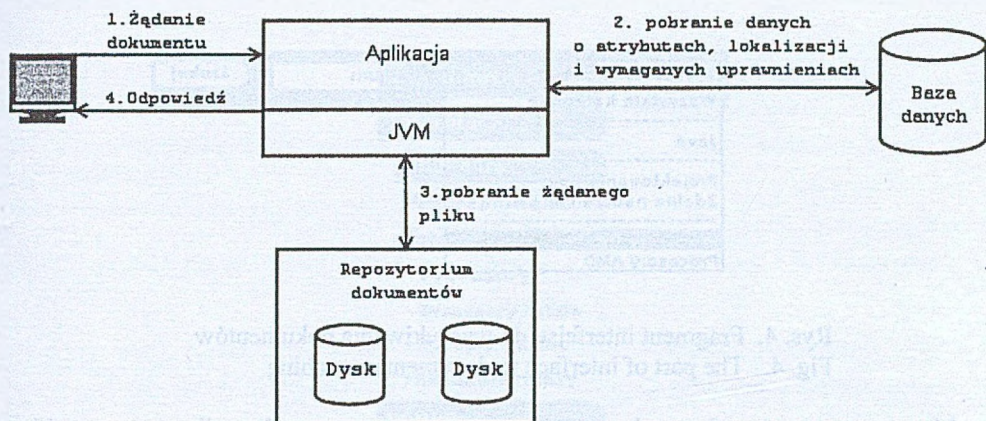


Rys. 2. Model logiczny danych

Fig. 2. Data logical model

Na podstawie tego modelu opracowano strukturę fizycznej bazy danych (tablice: użytkowników, kont użytkowników, grup, kojarząca użytkowników z grupami, ról pełnionych przez użytkowników, kojarząca użytkowników z rolami, uprawnień, dokumentów, kojarząca użytkowników o określonych uprawnieniach z dokumentami, kojarząca grupy o określonych uprawnieniach z dokumentami, pytań testowych dla dokumentów translacji odnośników do dokumentów, gości serwisu i liczników gości.

Dokumenty bądź prezentacje umieszczone w systemie są przechowywane na dysku w określonym katalogu do aplikacji. Może on mieć uporządkowaną strukturę logiczną (np. podział na rozdziały) wtedy jest z nim skojarzony plik XML. Dostęp do dokumentów ilustruje rys. 3.



Rys. 3. Metoda przechowywania dokumentów użytkowników  
 Fig. 3. The method of storing user's documents

Takie rozwiązanie nie obciąża bazy i nie generuje zbędnego ruchu w sieci.

Dostęp do danych (tabeli) zawartych w bazie jest zrealizowany za pośrednictwem komponentów obiektowych CMP. Każda tabela bazy danych jest skojarzona z obiektem typu encja. Tworzone są 3 klasy dla każdego komponentu CMP:

- klasa interfejsu obiektu lokalnego (odpowiedzialna za tworzenie nowych i wyszukiwanie istniejących obiektów),
- klasa interfejsu lokalnego (zawierająca metody biznesowe reprezentujące logikę aplikacji,
- klasa implementacji (zawierająca wszystkie implementacje metod zawartych w poprzednich dwóch interfejsach).

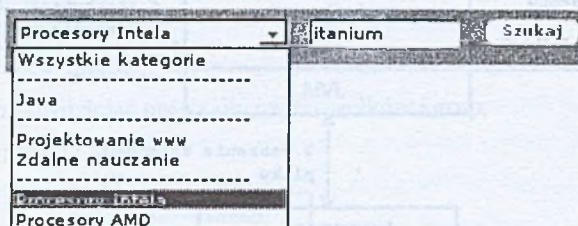
Zwiększa to przejrzystość programu i jego elastyczność.

#### 4. Interakcja z użytkownikiem

Serwis jest zbudowany z wykorzystaniem ramek. W górnej części znajduje się nagłówek tytułowy, przycisk logowania / wylogowania, pola do wyszukiwania dokumentów oraz menu systemowe. Wyszukiwanie opiera się na dwóch elementach:

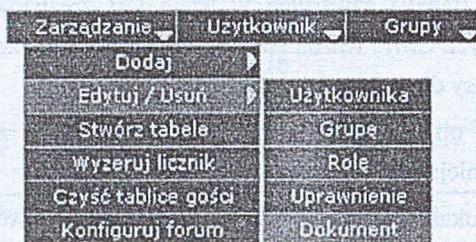
- kategorii dokumentów, których szukamy (można wybrać wszystkie dostępne),
- liście interesujących nas słów kluczowych.

Okienka do wyszukiwania umieszczone zostały na rysunku 4.



Rys. 4. Fragment interfejsu do wyszukiwania dokumentów  
Fig. 4. The part of interface to documents searching

Menu systemowe zawiera polecenia, które może wykonać użytkownik systemu, zaś jego zawartość zależy od roli, jaką pełni. Przykładowy fragment tego menu przedstawiono na rysunku 5.



Rys. 5. Fragment rozwiniętego menu systemowego dla administratora  
Fig. 5. The part of popup system menu for administrator

Menu główne serwisu pokazano na rysunku 6.

Menu *Administratora* zawiera kompletny zestaw opcji i komend.

Menu *Zarządzanie* zawiera polecenia: dodaj (użytkownika, grupy, rolę, uprawnienia do dokumentu), edytuj/usuń (jeden z obiektów dodanych), stwórz tabelę i wyzeruj licznik, czyść tablicę gości.

Menu *Użytkownik* posiada polecenia: twój profil, twoje grupy, twoje dokumenty (umożliwia między innymi umieszczanie treści prezentacji na serwerze tworzenie pytań testowych, przydzielanie praw do dokumentu grupom i użytkownikom).

Okno edycji pytania w teście pozwala określić treść pytania, odpowiedzi, zaznaczyć poprawne i typ pytania (wielokrotny wybór lub jedna odpowiedź prawidłowa). Dane te zostaną umieszczone w bazie i posłużą do automatycznego wygenerowania strony z testem. Pytania można dodawać, edytować i usuwać.





Rys. 6. Fragment Menu głównego serwisu

Fig. 6. The part of main menu of service

Menu *Grupy* zawiera polecenia dodaj i edytuj (grupę). Menu *Zasoby* obejmuje polecenia dodaj i edytuj (dokument). Menu *Statystyka* zawiera polecenia lista zalogowanych, lista top10 dokumentów (wyświetla 10 najczęściej przeglądanych prezentacji).

Menu *użytkownika* zawiera elementy *użytkownik*, *zasoby* i *statystyka*. Menu *menadżera grupy* zawiera elementy *użytkownik*, *grupy*, *zasoby* i *statystyka*.

Strona logowania, dostępna po wciśnięciu przycisku *Zaloguj*, zawiera interfejs do podania nazwy użytkownika i hasła, a także link do formularzy umożliwiających rejestrację.

Jak już wspomniano, każdy dokument może zawierać test, który obejmuje 20 pytań z zakresu poznanego materiału. Interfejs użytkownika przewiduje wygodne dodawanie i edycję tych pytań dostępne dla każdej osoby z prawami zapisu do dokumentu.

## 5. Podsumowanie

System został uruchomiony i przetestowany na jednym serwerze dołączonym do Internetu. Należy go traktować jako podstawę do opracowania systemu zdalnego nauczania o funkcjach zbliżonych do oferowanych przez profesjonalne systemy tego typu. Dalsze prace nad rozwojem systemu zmierzające do tego celu są kontynuowane.

**LITERATURA**

1. Bonk M.: System komputerowy wspomagający nauczanie w zakresie budowy i działania mikroprocesorów IA-64, praca dyplomowa wykonana pod kierunkiem dr inż. H. Małysiaka.
2. Kocher C., Falkner J.: Professional IS8. Wrox Press Limited. 2001.
3. Romowicz W.: Java serwer Pages oraz inne komponenty Java Platform. Helion, Gliwice 2001.
4. [WWW.smartforce.com](http://WWW.smartforce.com), [ilearning.oracle.com](http://ilearning.oracle.com), [www.okno.pw.edu.pl](http://www.okno.pw.edu.pl).
5. [msdn.microsoft.com](http://msdn.microsoft.com), [java.sun.com/www.w3.org](http://java.sun.com/www.w3.org) (style/XSL,[www.ph.net](http://www.ph.net)).

Recenzent: Dr inż. Ryszard Winiarczyk

Wpłynęło do Redakcji 12 maja 2003 r.

**Abstract**

Access to information presented in Internet is general today. It giving interesting possibilities of support of teaching. In the article the example of solution of system aided teaching was presented.

The system aided teaching was created as on internet portal that supports learning of structure and working of IA-64 microprocessors. Conditions of effective designing and proper arrangement of this system was described. Tools used to create web pages and user's interface were also shown.

The system makes possible registration of users and groups, simple access to information, simple modifications of knowledge base, necessary security of resources and discussions between users.

**Adresy**

Henryk MAŁYSIAK: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44 - 100 Gliwice, Polska, [henryk@zmitac.iinf.polsl.gliwice.pl](mailto:henryk@zmitac.iinf.polsl.gliwice.pl).

Marcin BONK: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44 - 100 Gliwice, Polska, [marbonk@poczta.onet.pl](mailto:marbonk@poczta.onet.pl).