

Agata Skowrońska-Kapusta\*, Paweł Kapusta\*, Piotr Goetzen\*

## **Studium wybranych aspektów procesów pozyskiwania i gromadzenia wiedzy w systemach zdalnego nauczania**

### **1. Wprowadzenie**

Wraz z rozwojem technologii opartych na usłudze WWW, takich jak języki tworzenia dynamicznych stron WWW (*PHP, Perl, Java, Python*), technologii komunikacji (*SOAP*) czy formatu dokumentów (*XML, XHTML*), coraz prostsze stało się opracowywanie efektywnych aplikacji internetowych opartych na nowych technologiach. Dzięki nim zaczęto budować zaawansowane systemy internetowe, umożliwiające interakcję użytkownika z systemem poprzez przeglądarkę internetową.

Przykładem takich systemów są platformy aplikacyjne służące do zdalnego nauczania. Jedną z zalet tego rodzaju komunikacji (e-learning) jest możliwość zebrania w jednym miejscu oraz przekazania do korzystania użytkownikowi dużej ilości informacji w atrakcyjnej formie.

E-learning jest nauczaniem na odległość z wykorzystaniem technik komputerowych i Internetu. Dzisiejsze systemy nauczania posiadają setki lub tysiące dokumentów udostępnianych kursantowi za pomocą typowego środowiska e-learningowego – *Virtual Learning Environment (VLE)* z menu nawigacyjnym i ikonami. Dzięki wprowadzeniu dynamicznych elementów na stronach internetowych (technologia *Macromedia Flash* lub technologia *Ajax*), umożliwiono szybszą i prostszą komunikację pomiędzy zdalnym systemem a użytkownikiem – zbliżając się w ten sposób do aplikacji uruchamianych lokalnie [11].

Największą wadą systemów zdalnego nauczania za pomocą sieci Internet jest w dalszym ciągu brak możliwości weryfikacji, czy osoba po drugiej stronie „łącza” jest właściwym studentem oraz jak efektywny, dla danego kursanta, jest proces kształcenia. W przypadku zajęć prowadzonych tradycyjną metodą, prowadzący zajęcia ma pewność, co do tożsamości studenta (lub może to łatwo sprawdzić). Natomiast w przypadku systemów e-learningowych weryfikowanie użytkownika odbywa się z wykorzystaniem nazwy użyt-

---

\* Społeczna Wyższa Szkoła Przedsiębiorczości i Zarządzania w Łodzi

kownika oraz hasła. Uwierzytelnianie odbywa się wyłącznie poprzez określenie tych parametrów, co może wpływać na powstawanie nieprawidłowości, np. podszywanie się pod inną osobę.

## 2. Proces pozyskiwania informacji

Innym aspektem systemów e-learningowych jest opracowanie przyjaznego środowiska dla użytkowników, wspierającego ich efektywność oraz jakość nauczania. Takie podejście do procesu dydaktycznego umożliwi spersonalizowanie ustawień platformy nauczania, dla danego studenta. W celu dokonania implementacji takich ustawień system musi zgromadzić szereg informacji o sposobie pracy danego użytkownika oraz połączyć je z informacjami związanymi z treścią przekazywanej wiedzy, w celu odpowiedniego zaadaptowania systemu [3, 5].

### 2.1. Weryfikacja użytkownika

Obecnie w systemach e-learningowych wyłączną weryfikacją użytkownika jest jego autoryzacja z wykorzystaniem nazwy oraz hasła. Inne, bardziej zaawansowane metody kontroli, takie jak odcisk palca czy weryfikacja twarzy, w zastosowaniu do aplikacji internetowych są technicznie trudne do zrealizowania oraz bardzo kosztowne.

W procesie pozyskiwania wiedzy ważny jest cały przebieg mechanizmów autoryzacji z systemem kształcenia, a nie wyłącznie wyrywkowe akceptowanie lub odrzucanie połączeń). Informacje pochodzące z procesów autoryzacji (wraz z dodatkowymi elementami) tworzą ogólną wiedzę na temat osoby wykorzystującej system.

Przy takich założeniach gromadzenie *informacji o sposobie dostępu oraz korzystania z systemu*, pozwoli na adaptację systemu e-learningowego oraz wykrycie anomalii w korzystaniu z niego [13].

### 2.2. Gromadzenie informacji na etapie autoryzacji użytkownika

Proces autoryzacji jest pierwszym elementem, w którym następuje sprawdzanie tożsamości danego użytkownika. W trakcie tego procesu może być przekazywany szereg dodatkowych informacji, które zostaną wykorzystane do odpowiedniego ustawienia systemu.

Przykładami gromadzenia informacji na potrzeby autoryzacji są:

- 1) **Informacje komunikacyjne** – informacje, z jakiego adresu IP komunikuje się dany użytkownik.

Aktualnie większość systemów, realizujących zarówno zadania e-learningowe, jak i inne aplikacje internetowe, zapisują informacje na temat adresu IP, z jakiego zostało dokonane ostatnie logowanie. Zmodyfikowanie systemu, w celu zapisywania kolejnych adresów IP, z których dokonywane jest logowanie pozwoli na śledzenie pozycji danego kursanta. Mechanizm taki może być w łatwy sposób wykorzystany, w celu

wprowadzenia algorytmu sprawdzającego anomalie. W przypadku wystąpienia nieprawidłowości, system może poprosić o dodatkową autoryzację, np.: wysłanie kodu autoryzacyjnego poprzez e-mail lub SMS lub wprowadzenie hasła autoryzacyjnego dostępnego podczas rejestracji. Mechanizmy takie często są wykorzystywane w systemach bankowych, gdzie wystąpienie anomalii powoduje zablokowanie dostępu, do czasu ponownej weryfikacji użytkownika. Oprócz tworzenia mapy anomalii, system e-learningowy może w ten sposób sprawdzać, czy inna osoba o podanym adresie IP nie logowała się do systemu z wykorzystaniem innej nazwy użytkownika. Badanie adresu IP może być więc zarówno pasywne – informacja o wystąpieniu anomalii jest wysyłana do administratora lub do osoby prowadzącej dany kurs e-learningowy, jak i aktywne – polegająca na zablokowaniu dostępu oraz uruchomieniu dodatkowych procedur.

W celu stworzenia procedury wykrywania anomalii należy zwrócić uwagę na możliwość logowania się użytkownika z różnych adresów IP otrzymywanych dynamicznie, dlatego analiza powinna nie porównywać wyłącznie adresów IP, ale także:

- czy adresy są z tej samej puli adresowej,
- czy adresy należą do tego samego systemu autonomicznego (AS) – który w tym przypadku można porównać z jednym dostawcą internetowym,
- czy pozycja geograficzna przynależności poszczególnych adresów IP jest zbliżona do siebie.

2) **Informacje czasowe** – informacje, w jakich godzinach oraz jak długo dany użytkownik korzysta z systemu.

Dodanie do systemu takiej funkcjonalności, umożliwi analizę oraz dokonanie odpowiedniej akcji na dwóch poziomach: podczas logowania do systemu oraz podczas pracy z systemem.

Podczas logowania do systemu mogą być podjęte następujące niżej wymienione akcje.

- Podjęcie lub wysłanie do administratora informacji o logowaniu do systemu w innych niż oczekiwane godzinach, w których dany kursant wcześniej korzystał. Mechanizm taki pozwoli na wprowadzenie dodatkowej procedury zabezpieczenia systemu przed nieautoryzowanym logowaniem.
- Dostosowanie profilu wyświetlania strony zwiększającej ergonomię pracy z systemem: modyfikację wyglądu strony (np. inny wygląd w nocy i w dzień).

W trakcie pracy kursanta *informacje czasowe* umożliwią administratorowi dostosowanie systemu do warunków określonych przez liczbę spędzonych godzin oraz pory korzystania z systemu. Dzięki temu system może poinformować użytkownika, że liczba jednocześnie spędzonych godzin przekracza określone normy.

Oczywiście w przypadku aplikacji internetowych, sprawdzanie przekroczenia określonego czasu bezczynności, umożliwi automatyczne wylogowanie z systemu, co aktualnie staje się normą w działaniu systemów z wykorzystaniem przeglądarek WWW.

3) **Informacje aplikacyjne** – informacje, z jakiego systemu operacyjnego, przeglądarki korzysta użytkownik, jaką posiada wersję popularnych aplikacji, takich jak np.: *Adobe Flash* czy *Java*.

Większość użytkowników zazwyczaj korzysta z wybranego systemu operacyjnego oraz przeglądarki do pracy z systemem zdalnego nauczania. Zmiana profilu przeglądarki może być sprawdzana przez:

- sprawdzenie systemu operacyjnego,
- sprawdzenie typu przeglądarki internetowej oraz jej wersji,
- sprawdzenie wersji dodatkowych komponentów, takich jak wersja biblioteki *Macromedia Flash* czy wersja języka *Java*.

Wszystkie powyższe informacje można uzyskać w łatwy sposób, wykorzystując wyłącznie protokół *HTTP*. Zmiana profilu powinna skutkować uruchomieniem dodatkowej procedury weryfikacji użytkownika.

4) **Inne informacje** zgromadzone podczas systemu nauki zdalnej.

Bardziej wyspecjalizowane mechanizmy, wymagające zaawansowanej modyfikacji systemu e-learningu to np.:

- analiza sposobu wpisywania tekstu,
- analiza ruchu myszki nad poszczególnymi elementami strony [12].

Powyższe sposoby są możliwe do zaimplementowania przez zastosowanie specjalistycznych algorytmów, które aktualnie w większości przypadków są w fazach rozwojowych i nie stanowią podstawowych technik autoryzacyjnych.

Techniki analizy wprowadzania tekstu czy ruchów myszy polegają na biometrycznej analizie danego użytkownika podczas logowania się oraz w trakcie pracy w systemie. Zarówno w przypadku analizy wpisywanego tekstu, jak i analizy ruchów myszki do systemu na poziomie przeglądarki internetowej muszą zostać wprowadzone odpowiednie modyfikacje (np. poprzez aplikację stworzoną w *Javie*).

Przykładem zastosowania analizy wprowadzania tekstu jest rozwiązanie firmy *Biopassword*, polegające na rozpoznawaniu rytmu uderzeń w klawiaturę podczas wprowadzania tekstu. Technikę tę można wykorzystać podczas wpisywania hasła czy wprowadzania tekstu w polach tekstowych aplikacji [15].

W przypadku drugiego rozwiązania, polegającego na analizie ruchów myszki, mechanizm taki można badać na podstawie bibliotek realizujących rozpoznawanie gestów myszki (wykonywanych przez użytkownika w aplikacji). Istnieje szereg projektów realizujących to rozwiązanie [10].

Trzeba jednak zauważyć, że wykorzystanie tych sposobów analizy w systemach e-learningowych ma za zadanie wyłącznie ciągłe lub wrywkowe weryfikowanie już zalogowanego użytkownika.

Wspomniane powyżej metody autentykacji użytkownika zmniejszają prawdopodobieństwo wykorzystywania systemu nauczania przez nieuprawnione osoby lub przekazy-

wania haseł dostępu innym osobom. Należy zauważyć, że poinformowanie użytkowników o wprowadzeniu takich technik, stanowi dodatkowe, „psychologiczne” zabezpieczenie systemu nauczania.

Oczywiście nie da się z całą pewnością wykluczyć przekazywania użytkownikom nieuprawnionym swoich danych, ale w przypadku gdy użytkownicy są rozproszeni geograficznie, można w maksymalny sposób im to utrudnić.

Podobnie jak w większości tego typu projektów może zaistnieć sytuacja złej interpretacji danych, skutkującej zablokowaniem dostępu do platformy osobie uprawnionej. Każdy system powinien posiadać dodatkowe procedury rozwiązania takiego problemu.

Analizując powyższe problemy, można uznać, że weryfikacji użytkownika pod względem statystycznym powinna zostać osiągnięta wysoka skuteczność weryfikacji autentyczności pracy kursanta, bez dodatkowych sprzętowych urządzeń po stronie użytkownika.

### 2.3. Zbieranie oraz analiza informacji na temat korzystania z systemu

Oprócz wymienionej weryfikacji użytkownika, także dane zbierane podczas pracy w systemie zdalnego nauczania mogą umożliwić automatyczną adaptację procesu kształcenia, specyficzną dla danego kursanta. Proces ten, przy wykorzystaniu wcześniej wspomnianych technik, umożliwiłby dostosowanie zarówno wyglądu, jak i treści merytorycznej do aktualnej pracy danego użytkownika oraz zaistniałych warunków [2].

Głównym elementem dostarczającym podstawowe informacje na temat charakterystyki pracy danego kursanta jest analiza korzystania z systemu (wspomniane wcześniej *informacje czasowe*) – czyli czas logowania, częstotliwość logowania oraz czas pracy z systemem, w porównaniu ze zbliżającymi się terminami zaliczeń, zamknięcia projektów itp. Dzięki temu osoba nadzorująca przebieg nauki danego materiału dydaktycznego będzie miała czytelne informacje na temat konkretnego studenta: czy nauka była systematyczna, czy zwiększenie aktywności było następstwem jakiegoś określonego terminu.

Dodatkowo system analizy powinien także brać pod uwagę następujące parametry:

Związane z *kursantem* takie jak:

- wiek,
- płeć,
- dodatkowe parametry wprowadzane przez kursanta (np.: intensywność),
- czy dany kursant dodatkowo pracuje, jeżeli tak to, w jakich godzinach.

Związane z *kursem*:

- długość kursu,
- poziom przygotowania materiałów,
- stopień trudności w odniesieniu do zaawansowania kursantów,
- współczynnik informacji do czasu trwania wpływający na intensywność kursu,
- maksymalna ilość materiału przyswajanego przez kursanta.

Związane z *prowadzącym*:

- intensywność weryfikacji informacji od kursanta,
- częstotliwość weryfikacji ogólnej pracy kursantów [8].

Oprócz powyższych stałych parametrów system powinien także uwzględniać parametry zależne od bieżącego środowiska:

- okres trwania kursu (pora roku, miesiąc itp.),
- godzina korzystania z systemu (dzień, noc),
- z jakiej przeglądarki korzysta kursant (ważnym elementem dodatkowo jest rozdzielczość – oraz ewentualnie rozpoznanie urządzenia – notebook, palmtop itp.).

Zabranie wszystkich parametrów oraz informacji uzyskanych przy autentykacji pozwoli na przygotowanie sposobu działania systemu, czyli:

- Rodzaju wyświetlenia oprawy graficznej:
  - grafika bardziej lub mniej intensywna,
  - dodatkowe banery informacyjne,
  - wielkość informacji dodatkowych podług zawartości merytorycznej.
- Sposobu wyświetlenia tekstu:
  - kolor oraz rodzaj czcionki,
  - sposób automatycznego przewijania treści.
- Merytorycznej zawartości wyświetlanego materiału:
  - dokonywanie selekcji przekazywanych informacji,
  - przedstawienie propozycji powtórzenia materiału,
  - przedstawienie propozycji przejścia do prostszych lub bardziej skomplikowanych zagadnień,
  - wykonanie cząstkowych weryfikacji materiału [8].

W celu odpowiedniego przygotowania działania systemu do zastosowania powyższych elementów trzeba nauczyć system jak interpretować dane wejściowe. Można tego dokonać podczas adaptacji systemu, z wykorzystaniem kilku mechanizmów:

- przeprowadzenie badań na temat sposobu wykorzystania platformy oraz dobranie jak najbardziej optymalnych rozwiązań;
- przeprowadzenie w ramach systemu ustawienia preferencji każdego użytkownika podczas pierwszej rejestracji (system musi mieć później możliwość dodatkowej modyfikacji tych wartości);
- opracowanie możliwości modyfikacji parametrów podczas pracy przez użytkownika, dzięki czemu uzyskane zostaną informacje statystyczne dla najbardziej optymalnego rozwiązania.

Należy pamiętać, że system e-learningowy to nie tylko aplikacja, ale także sposób przekazywania wiedzy, który powinien być dostosowywany indywidualnie do każdej oso-

by. Domyślnie, w przypadku aplikacji zdalnych, osoba pełniąca rolę kursanta jest nieznaną dla systemu. Aplikacja powinna z czasem tworzyć „informatyczny portret psychologiczny” dla takiego użytkownika. Podstawowe ustawienia systemu powinny być natomiast zaimplementowane z wykorzystaniem badań psychologicznych lub przez przeprowadzenie wiarygodnych testów takiego systemu [4].

Ważnym elementem dla mechanizmu dostosowywania się systemu do potrzeb poszczególnych użytkowników jest czas korzystania przez nich z aplikacji. Należy pamiętać, że system nie dostosuje preferencji dla osoby, która korzysta z systemu przez kilka dni. Sensowność takiego rozwiązania uwidacznia się w momencie dłuższego korzystania z systemu: jak na przykład studia przeprowadzane w ramach wyższych uczelni trwających semestr lub dłużej.

Opracowanie profili użytkowników w trakcie autoryzacji i korzystania z systemu, może także posłużyć do ustawienia dodatkowych systemów m-learningu, które w łatwy sposób można zintegrować z systemami online typu *LMS*. Systemy m-learningowe zazwyczaj wykorzystywane są do wspomagania nauki na urządzeniach mobilnych. Odpowiednia analiza pracy użytkownika może umożliwić przygotowanie materiałów oraz opracowanie programu dydaktycznego do samodzielnej pracy przez kursanta, bez połączenia z głównym systemem *LMS* [1].

### 3. Realizacja zadania

#### 3.1. Gromadzenie, analiza informacji oraz modyfikacja systemu – podział zadań

Proces gromadzenia i analizy wiedzy w systemie zdalnego nauczania, powinien działać na poziomie:

- zbierania informacji (*kolektor*),
- analizy informacji (*analizator*),
- procedury modyfikacji ustawień systemu (*modyfikator*).

Podział na powyższe zadania uprości proces implementacji systemu w już istniejących aplikacjach. Elementem zwrotnym na poziomie aplikacji e-learningowej będzie wyłącznie analiza poprawności autoryzacji, okresowa weryfikacja użytkownika podczas pracy z systemem oraz modyfikacja ustawień systemu.

Mechanizm zbierania informacji polega na dodaniu w odpowiednich miejscach aplikacji procedur zbierania danych oraz przekazywania ich do bazy danych. Przed opracowaniem procedur powinna zostać określona lista użytecznych danych możliwa do uzyskania z systemu. Na podstawie parametrów: systemu, kursanta, osoby prowadzącej oraz materiału dydaktycznego zostanie utworzona procedura personalizacji działania aplikacji.

### **Kolektor**

Część systemu odpowiedzialna za zbieranie informacji odczyta aktualne informacje pracy systemu, zarówno na poziomie autoryzacji, jak i w trakcie pracy. Zbieranie informacji w trakcie pracy powinno być dokonywane okresowo – poprzez wystąpienie określonego stanu systemu lub w określonym interwale czasowy. Wszystkie dane powinny być zapisywane w centralnym magazynie (bazie danych).

### **Analizator**

Analizator zajmować się będzie interpretacją zgromadzonych danych oraz stworzeniem dla każdego użytkownika profilu. Analiza powinna odbywać się zarówno na danych globalnych, jak i na danych spersonalizowanych danego użytkownika. Poszczególne preferencje będą wykorzystane podczas kolejnej sesji pracy z systemem oraz umożliwią modyfikację pracy systemu w trakcie korzystania.

### **Modyfikator**

Po dokonaniu analizy oraz stworzenia preferencji danego użytkownika informacje powinny być przekazane do procedur modyfikacji ustawień systemu. Sposób dokonywania zmiany powinien być uzależniony od budowy aplikacji. Podobnie jak w przypadku kolektora oraz analizatora procedury powinny być uruchamiane okresowo.

Okresowość działania modułów zmniejszy niepotrzebne obciążenie systemu, co w praktyce umożliwi bardziej komfortową pracę z systemem.

## **3.2. Poziom aplikacyjny**

W celu zastosowania w praktyce zaproponowanych wyżej zadań można zaprojektować od podstaw aplikację bądź zmodyfikować aplikację istniejącą.

Rozpatrując system na etapie projektowania, wyróżnia się dwa sposoby modyfikacji istniejącej aplikacji:

- 1) dodanie do systemu e-learningowego nowych procedur, w pełni realizujących zadanie (*kolektor*);
- 2) dodanie do systemu e-learningowego wyłącznie procedur zbierania informacji oraz procedur modyfikacji systemu; za analizę, przetwarzanie oraz ewentualnie raportowanie będzie odpowiedzialna oddzielna aplikacja.

Zarówno pierwsze, jak i drugie podejście do modelu aplikacyjnego ma swoje wady i zalety. W przypadku stworzenia zewnętrznej aplikacji system może być integrowany z wieloma różnymi platformami nauczania. Jednak dodatkowo wymaga stworzenia szkieletu programistycznego (*Framework*) aplikacji. W przypadku modyfikacji istniejącej aplikacji można wykorzystać środowisko programistyczne oferowane przez dany system LMS, dzięki czemu w znaczny sposób uproszczy się praca programistyczna związana z interfejsem oraz sposobem raportowania informacji.



### **Aplikacje e-learningowe**

Rozpatrując przypadek modyfikacji istniejących aplikacji – dużo możliwości otrzymujemy przy wykorzystaniu programów z jawnym źródłem kodu programu.

Aktualnie dostępne są następujące *darmowe* produkty wykorzystywane w pracy zdalnej:

- *Moodle* [<http://moodle.org/>],
- *Atutor* [<http://www.atutor.ca/atutor/demo/login.php>],
- *Claroline* [<http://www.claroline.net/demo/demo.html>],
- *Dacebo* [<http://www.demo.docebo.org/index.php>],
- *Dokeos* [<http://campus.dokeos.com/>],
- *Mahara* [<http://demo.mahara.org/>].

W przypadku aplikacji komercyjnych dostosowanie funkcjonowania systemu jest znacznie bardziej uciążliwe, ze względu na zamknięte kody oprogramowania. Niemniej jednak, w przypadku wdrożenia aplikacji, można skorzystać z pomocy producenta – bądź opierać się wyłącznie na danych zawartych w bazach danych oraz zewnętrznych informacjach serwera WWW.

*Komercyjne platformy e-learningowe:*

- *Blackboard*,
- *Scholar360*,
- *JoomlaLMS*,
- *SharePointLMS*,
- *Lotus Learning Management System* [9].

### **Aplikacja Moodle**

Adaptacja systemu na potrzeby użytkownika zostanie przedstawiona na przykładzie popularnej, darmowej aplikacji e-learningowej – *Moodle (Modular Object-Oriented Dynamic Learning Environment)*.

*Moodle* jest typem aplikacji określanej, jako system zarządzania kursami (*Course Management System*), którego celem jest:

- gromadzenie oraz udostępnianie materiałów dydaktycznych poprzez usługę WWW,
- kreowanie for dyskusyjnych oraz czatów,
- dostarczanie oraz przeglądanie zadań z możliwością oceniania,
- opracowanie quizów dydaktycznych,
- zapisywanie oraz raportowanie na temat ocen uczestników,
- zarządzanie oraz przypisywanie praw użytkownikom.

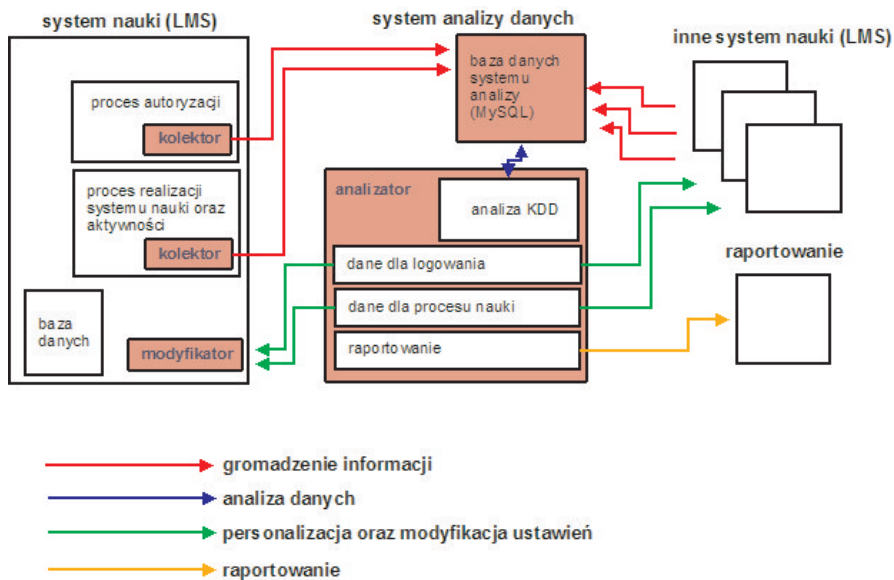
System *Moodle* gromadzi wiele informacji na temat aktywności użytkowników. Informacje takie jak adresy IP, z których następowało połączenie, czas pracy użytkownika bądź aktywność na forach lub czatach są zapisywane w bazie danych. Informacje te mogą być w prosty sposób zaimportowane do systemu analizy informacji.

### 3.3. Adaptacja systemu – założenia

Projekt adaptacji aplikacji na potrzeby użytkownika obejmuje realizację następujących założeń:

- 1) jako zasób gromadzący informacje, zostanie wykorzystana zewnętrzna baza danych systemu *MySQL*, do stworzenia oraz zarządzania bazą danych wykorzystane zostanie oprogramowanie: *PhpMyAdmin*;
- 2) w aplikacji zostaną przeprowadzone odpowiednie modyfikacje wysyłające do bazy danych informacje potrzebne w projekcie (*kolektor*):
  - na poziomie logowania,
  - na poziomie aktywności elementów/składowych kursu;
- 3) w aplikacji zostaną dodane funkcje personalizacji aplikacji w momencie logowania oraz aktywności elementów/składowych kursu (*modyfikator*);
- 4) jako *analizator* zostanie wykorzystany oddzielny program, którego działania będzie zwracał wyniki dla:
  - procedury logowania,
  - procedur aktywności elementów/składowych kursu,
  - procedur raportowania;
- 5) program będzie wykorzystany dodatkowo do tworzenia raportów, przeprowadzania badań oraz tworzenia konfiguracji systemów *LMS*.

Schemat realizacji opisanych założeń modyfikacji systemu *Moodle* przedstawia rysunek 1.



**Rys. 1.** Schemat projektu adaptacji systemu *Moodle* na potrzeby użytkownika (opracowania własne)

System *Moodle* został stworzony z wykorzystaniem języka programowania *PHP*, jako silnik bazy danych wykorzystuje zaś bazę *MySQL*. Dzięki otwartemu kodowi, w łatwy sposób można dorzucić interesujące nas biblioteki. Z kolei wykorzystanie, jako serwera *WWW* aplikacji *Apache*, pozwala zintegrować system *Moodle* z innymi aplikacjami, napisanymi nie tylko w języku *PHP*, ale także w językach *Java*, *Perl* czy *Python* [16].

Adaptacja systemu *Moodle* może odbywać się na dwa sposoby:

- 1) poprzez modyfikację kodu za pomocą odpowiednio przygotowanych tzw. łatek (*patchy*),
- 2) poprzez dodanie do systemu odpowiedniego modułu.

W przypadku modyfikacji kodu dodanie nowych funkcjonalności powinno odbywać się poprzez przygotowanie odpowiednio spreparowanych plików, modyfikujących oryginalny kod programu. Dokonywanie takich zmian w systemie *Linux* może być przygotowane przez bezpośrednią modyfikację kodu źródłowego do docelowej konfiguracji, a następnie porównanie zmodyfikowanego drzewa plików z oryginalnym drzewem kodu źródłowego programu *Moodle*. Do porównania można wykorzystać program *diff*, dostępny w większości dystrybucji systemu *Linux*. Wynikiem działania programu *diff* jest plik zawierający wszystkie różnice pomiędzy programem oryginalnym a zmodyfikowanym.

Adaptacja systemu według wskazanych potrzeb wymaga znajomości struktury programistycznej aplikacji *Moodle*. Szczegółowa dokumentacja źródeł programu jest dostępna na stronie [<http://moodle.org/development/>].

Na stronie tej można uzyskać także informacje na temat opracowania nowych modułów do programu. W ramach struktury programu należy rozróżnić:

- schemat bazy danych,
- moduły wewnętrzne systemu,
- biblioteki graficzne służące do wyświetlania elementów systemu,
- komponenty konfiguracyjne takie jak plik konfiguracyjny oraz pulpit administracyjny.

### 3.4. Przykład adaptacji mechanizmu autoryzacji *Moodle*

Przeprowadzanie modyfikacji systemu odbywa się w ramach aplikacji, poprzez dodanie odpowiedniego *kolektora* w trakcie autoryzacji. Do tego celu można wykorzystać moduł *Authentication API* wbudowany domyślnie w program *Moodle*.

Wszystkie dane powinny trafiać do centralnej bazy danych. Baza ta może być zarówno elementem struktury schematu bazy danych *Moodle*, jak i oddzielną bazą danych – służącą wyłącznie do zbierania informacji. Jednak w drugim przypadku przy połączeniach należy zapewnić możliwość nowego schematu inicjalizacji połączenia z bazą danych.

System *Moodle* tworzony był jako modułarna platforma wspomagająca naukę. Modularność w tym przypadku oznacza możliwość tworzenia dodatkowych funkcjonalności, co może zostać wykorzystane do własnej modyfikacji systemu. *Moodle* umożliwia opracowanie różnych typów rozszerzeń modułów.

Przykładowa procedura opracowania nowego modułu autoryzacyjnego:

**Krok 1:** Zdefiniowanie nazwy modułu np.: *autoryz* oraz utworzenie nowego katalogu */auth/autoryz*.

**Krok 2:** Stworzenie pliku o nazwie *auth.php* w katalogu modułu zawierającego klasy: *auth\_plugin\_autoryz* i rozszerzającą klasę podstawową *auth\_plugin\_base*.

**Krok 3:** Stworzenie funkcji *user\_login()*, która będzie nadpisywała mechanizm autoryzacji użytkownika. W funkcji zostaną umieszczone wszystkie procedury związane z wysłaniem informacji na temat logowania do zewnętrznej bazy danych.

**Krok 4:** W pulpicie administratora w sekcji „Zarządzanie autoryzacją” pojawią się informacje na temat nowego stworzonego modułu.

W ramach stworzonej funkcji lub klasy możemy dodać opisane wcześniej zachowania systemu.

### 3.5. Analizator

Najważniejszym elementem projektu jest system analizowania danych oraz interpretacji danych. Przydatne tu będą dostępne metody, wykorzystywane w eksploracji danych, tzw. *DM – Data Mining*. Jedną z nich jest dziedzina informatyki, zajmująca się tworzeniem narzędzi oraz algorytmów służących odkrywaniu wiedzy z danych – *KDD (Knowledge Discovery in Databases)* [6, 7].

W ramach *KDD* proces przygotowania działania algorytmów *analizatora*, można podzielić na następujące etapy:

1. opracowanie założeń projektu – czyli określenie dziedziny problemu,
2. stworzenie zbioru danych, jakie mają być wykorzystywane przez *analizator* (w tym przypadku będą to dane przysyłane przez *kolektor*),
3. przygotowanie danych dla etapu decyzyjnego, przekształcenie oraz oczyszczenie danych,
4. opracowanie metod eksploracji danych (*Data Mining*).

Istnieje wiele technik możliwych do wykorzystania w celu określenia parametrów wyjściowych, są nim:

- drzewo decyzyjne,
- reguły klasyfikacji,
- reguły skojarzenia,
- reguły z wyjątkami,
- reguły wymagające relacji,
- drzewa z numerycznym prognozowaniem,
- reprezentacja przypadków,
- klastry.

Najprostsze metody, czyli drzewo decyzyjne oraz reguły klasyfikacji, umożliwiają ustawienie profili użytkownika oraz personalizację ustawień systemu. Jednak w przypadku, gdy założeniem systemu ma być stopniowe uczenie się *analizatora*, należy wykorzystać bardziej zaawansowane techniki (przykładem może być dodatkowe wykorzystanie sieci neuronowych, metod ewolucyjnych czy logiki rozmytej) [14].

## 4. Wnioski

W instytucjach edukacyjnych coraz częściej wykorzystuje się Internet w celu wspomaganie procesu nauczania. Często za pomocą tego medium przeprowadzane są wszystkie zajęcia i tworzą one model studiowania przez Internet. W porównaniu z tradycyjnym nauczaniem, z systemów takich możemy uzyskać wiele pożytecznych informacji, wykorzystywanych dodatkowo do wzbogacenia oraz ukierunkowania sposobu nauczania. Proponowana adaptacja systemu nauczania (np. *Moodle*) umożliwi opracowanie mechanizmów analitycznych bez ingerencji w aktualne działania systemu. W ramach modyfikacji aplikacji tworzone są wyłącznie *kolektory*, które stopniowo mogą być wprowadzane do modelu aplikacyjnego systemu *LMS*. System analityczny, który stanowi oddzielną aplikację, może być rozwijany już w trakcie gromadzenia danych w zewnętrznej bazie danych.

## Literatura

- [1] Attewell J., *From Research and Development to Mobile Learning: Tools for Education and Training Providers and their Learners*. mLearn 2005, 4th World conference on mLearning, Cape Town 2005 [<http://www.mlearn.org.za/CD/papers/Attewell.pdf>].
- [2] Dawn Shaikh A., Chaparro B., Fox D., *The Personality of Terms and Concepts Used in Online Material*. Usability News, vol. 11 Issue 1, 2009 [<http://www.surl.org/usabilitynews/111/personality.asp>].
- [3] Garrison D.R., Anderson T., *E-learning in the 21st Century: A Framework for Research and Practice*. RoutledgeFalmer, London, 2003.
- [4] Horton W., Horton K., *E-learning Tools and Technologies*. John Wiley & Sons Inc., New York, 2003.
- [5] Hyla M., *Przewodnik po e-learningu*. Wolters Kluwer Polska, 2007.
- [6] Kozielski S., Małysiak B., Kasprowski P., Mrozek D. (Eds), *Bazy danych: Nowe technologie*. Wydawnictwa Komunikacji i Łączności, Warszawa, 2007.
- [7] Kozielski S., Małysiak B., Kasprowski P., Mrozek D. (Eds), *Bazy danych: struktury, algorytmy, metody*. Wydawnictwa Komunikacji i Łączności, Warszawa, 2006.
- [8] Krupski M., Cader A., *E-learning Tools in Distance and Stationary Education*. Journal of Applied Computer Science Methods, vol. 1, nr 1, Academy of Management (SWSPiZ), Łódź 2009, 159–178.
- [9] Marchlewska A., Paszkowski J., Cader A., Goetzen P., *E-technologie w pracy grupowej*. [w:] Wybrane zagadnienia inżynierii wiedzy, Studia i monografie, nr 21, Łódź, 2008, 77–90.
- [10] Poole A., Ball L.J., *Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future Prospects*. [w:] Claude Ghaoui (Ed.), Idea Group [<http://www.alexpoole.info/academic/Poole&Ball%20EyeTracking.pdf>].

- [11] Rogers P., Howard C., *Encyclopedia of Distance Learning*. 2nd ed., Idea Group Inc (IGI), 2009.
- [12] Sowa G., Filutowicz Z., Marchlewska A., Paszkowski J., *Interfejsy użytkownika w zdalnym nauczaniu*. [w:] Wybrane zagadnienia inżynierii wiedzy, Studia i monografie, nr 21, Łódź 2008, 77–90.
- [13] Sowa G., *User interface – Standards and Research Tools*. Journal of Applied Computer Science Methods, vol. 1, nr 1, Academy of Management (SWSPiZ), Łódź, 2009, 53–66.
- [14] Witten I., Frank E., *Data Mining – Practical Machine Learning Tools and Techniques*. 2nd ed., Morgan Kaufman Publisher, 2005.
- [15] [<http://www.admitonesecurity.com/>].
- [16] [<http://moodle.org/>].