

Kamil ŻYŁA<sup>1</sup>  
Jacek KEŚIK<sup>2</sup>

### **ZINTEGROWANE ŚRODOWISKO PROGRAMISTYCZNE IDE WEBRATIO 5.1 WSPOMAGAJĄCE NAUCZANIE MDE**

*Nauczanie inżynierii oprogramowania, wymaga poszukiwania nowych narzędzi, technologii i metodyk oraz sposobów ich wydajnego wykorzystania w kontekście wymagań rynku komercyjnego oraz potrzeb współczesnego społeczeństwa informacyjnego.*

*Ostatnimi czasy, w dziedzinie projektowania i tworzenia aplikacji internetowych zarządzających dużą ilością danych, coraz popularniejszą staje się inżynieria sterowana modelami (MDE). Metodyka WebML, jako jej składowa, oferuje unikalny zestaw modeli i komponentów pozwalających na ekspresję wszelkich aspektów działania takich aplikacji.*

*W artykule dokonano przedstawienia oraz oceny najpełniej implementującego WebML środowiska programistycznego, jakim jest WebRatio. Zwrócono również uwagę na przydatność w zastosowaniach innych niż edukacyjne oraz perspektywy dalszego rozwoju.*

### **AIDING MDE TEACHING PROCESS BY USAGE OF IDE WEBRATIO 5.1**

*The teaching of software engineering techniques is shaped by the needs of the information society and software producers, who want to gain well qualified programmers with appropriate knowledge regarding model driven and traditional techniques of software development.*

*Recently Model Driven Engineering is gaining popularity on the field of creating data-intensive web applications. WebML methodology, as a part of this concept, introduces a unique set of models and components useful in development process of such applications.*

*This paper evaluates from the teachers point of view, the most complete implementation of the WebML methodology, which was made in IDE WebRatio. It also describes its usefulness, possible applications (not only educational ones) and perspectives of evolution.*

---

<sup>1</sup> Politechnika Lubelska, Wydział Elektrotechniki i Informatyki; 20-618 Lublin; ul. Nadbystrzycka 36b  
tel: +48 81 525-20-46, fax: +48 81 538-43-49, e-mail: kamilz@cs.pollub.pl

<sup>2</sup> Politechnika Lubelska, Wydział Elektrotechniki i Informatyki; 20-618 Lublin; ul. Nadbystrzycka 36b  
tel: +48 81 525-20-46, fax: +48 81 538-43-49, e-mail: kesik@cs.pollub.pl

## 1. WSTĘP

Współcześnie obserwuje się duże zainteresowanie oprogramowaniem działającym w środowisku rozproszonym, w tym w pełni profesjonalnymi aplikacjami internetowymi. Oprogramowanie związane z geolokacją, kartografią internetową, zagospodarowaniem przestrzennym, sprzedażą internetową, a także zarządzające przyjęciami na studia i procesem nauczania, to tylko niektóre współczesne zastosowania technologii internetowych. Wzrasta w związku z tym potrzeba kształcenia specjalistów w zakresie szybkiego projektowania i implementacji oprogramowania z uwzględnieniem takich aspektów, jak bezpieczeństwo, skalowalność, wydajność oraz ergonomia.

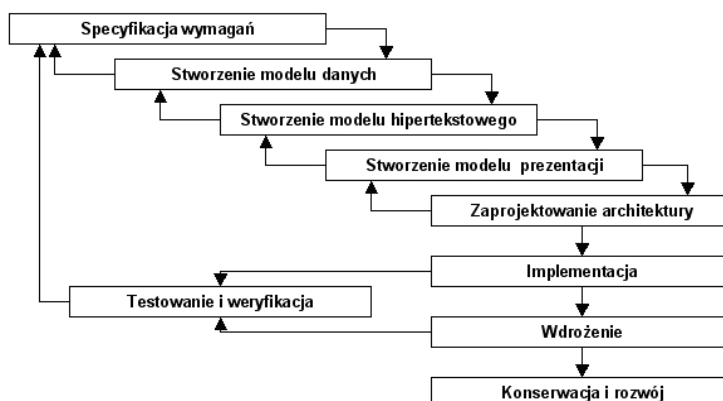
Właśnie takim potrzebom wychodzi naprzeciw Inżynieria Sterowana Modelami (ang. Model Driven Engineering) oferując podejście do tworzenia aplikacji bardziej z punktu widzenia projektanta niż programisty. Różnorakie modele, komponenty, gotowe biblioteki oraz wzorce projektowe tworzą wyższą warstwę abstrakcji, pozwalającą skupić się na zasadzie działania aplikacji, jej funkcjonalności, układzie treści, sposobie nawigacji oraz przepływie informacji pomiędzy jej modułami. Zyskuje się dzięki temu oszczędność czasu oraz zmniejsza liczbę potencjalnych błędów, jakie mogłyby się pojawić podczas kodowania funkcji typowych dla każdej aplikacji np. internetowej. Te typowe funkcje stają się komponentami modeli, które bardzo często można traktować jako „czarne skrzynki”, zwracając uwagę na udostępnianą przez nie funkcjonalność, a nie sposób jej implementacji. Ponadto model, z którego powstaje aplikacja stanowi również aktualną dokumentację.

Koncepcje MDE w dziedzinę aplikacji internetowych przenosi metodyka WebML, udostępniając zestaw modeli i komponentów typowych dla aplikacji internetowych zarządzających dużymi ilościami danych. Odpowiednia metodyka zapewnia połowiczny sukces, konieczne jest również wykorzystanie odpowiedniego środowiska programistycznego. Dotychczas najpełniejszą implementacją metodyki WebML jest środowisko programistyczne IDE WebRatio, którego dotyczy ten artykuł.

## 2. WPROWADZENIE DO WEBML

WebML (ang. Web Modeling Language) to jednocześnie nazwa metodyki i języka modelowania bazującego na UML, który wchodzi w jej skład.

Charakterystyczne dla metodyki WebML etapy cyklu życia aplikacji internetowej (rys. 1), to specyfikacja wymagań, stworzenie modelu danych, hipertekstowego oraz prezentacji. To właśnie one odróżniają ją od innych metodyk.

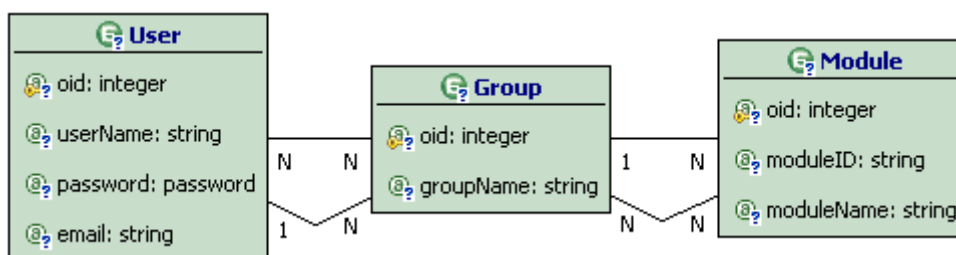


Rys.1. Cykl życia aplikacji internetowej według metodyki WebML [1]

Specyfikacja wymagań polega na zebraniu możliwie kompletnych informacji o funkcjonalności i wymaganiach stawianych projektowanej aplikacji oraz ich analizie. Produktem końcowym powinien być zestaw półformalnych dokumentów zawierający [2]:

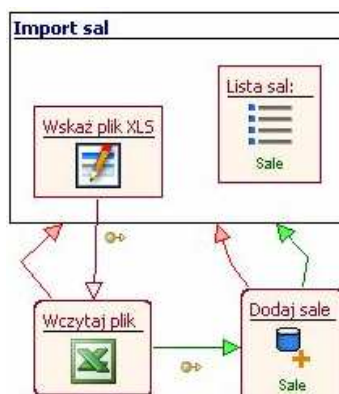
1. Specyfikację grup użytkowników aplikacji, wraz z ich hierarchią i uprawnieniami.
2. Specyfikację najważniejszych przypadków użycia.
3. Specyfikację najważniejszych obiektów w dziedzinie aplikacji.
4. Specyfikację rozmieszczenia treści i przypadków użycia w ramach stron aplikacji.
5. Szkic wyglądu interfejsu aplikacji.
6. Specyfikację wymagań нефункциональных.

Model danych to zmodyfikowany diagram ERD. Zmianie uległ sposób oznaczania liczebności, a typy pól są zbiorem możliwie ogólnych typów danych języków programowania (np. Java) i języka SQL. Jego przykładowy fragment przedstawiono na rysunku 2.



Rys.2. Podszemat personalizacji modelu danych

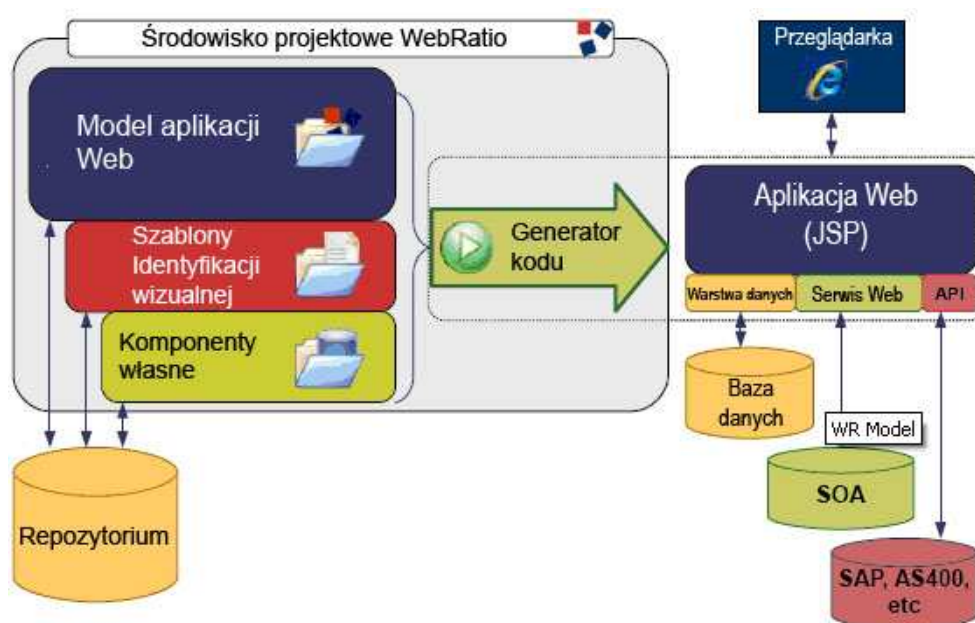
Najciekawszą składową WebML jest model hipertekstowy, który przedstawia wszystkie strony aplikacji, uwzględniając prezentowane treści, operacje na tych treściach, nawigację pomiędzy stronami aplikacji oraz obszary wymagające autoryzacji [2]. Przykładowy fragment modelu hipertekstowego przedstawiono na rysunku 3.



Rys.3. Fragment modelu umieszczający w bazie danych dane z arkusza kalkulacyjnego

### 3. WPROWADZENIE DO IDE WEBRATIO 5.1

IDE WebRatio 5.1 jest środowiskiem, opracowanym przez twórców metodyki WebML na podstawie platformy Eclipse Ganymede, określanym jako ang. Model Driven Development Enviroment. Umożliwia generowanie kodu aplikacji internetowych na podstawie modelu (projektu) stworzonego przy użyciu notacji WebML (rys. 4). Kod aplikacji jest generowany w języku Java (Java Server Pages), w związku z tym możliwe jest jego umieszczenie na dowolnym serwerze obsługującym ten język, m.in. Apache Tomcat, Jboss, Resi, IBM WebSphere. IDE WebRatio 5.1, dzięki wsparciu dla biblioteki Hibernate, pozwala na współpracę z większością oferowanych na rynku baz danych, dla których są dostępne sterowniki JDBC, włączając bazy danych Oracle 8i/9i/10g, DB2, MySQL, SQL Server i PostgreSQL [3].



Rys.4. Schemat środowiska programistycznego IDE WebRatio 5.1 [3]

Generowanie kodu aplikacji dokonuje się, w zasadzie, za jednym kliknięciem myszy. Model danych służy wygenerowaniu struktury bazy danych, przy czym możliwe jest wykorzystanie istniejącej już bazy danych i automatyczne dostosowanie do niej istniejącego modelu oraz wykorzystanie więcej niż jednej bazy danych (bazy danych mogą być różnego typu) w pojedynczej aplikacji. Kreator synchronizacji pozwala na: użycie pustych lub wypełnionych informacjami baz danych bez wprowadzania zmian, rozszerzenie ich o dodatkowe struktury danych lub całkowite przekształcenie. Na podstawie modelu hipertekstowego oraz prezentacji są generowane dynamiczne strony JSP z uwzględnieniem najbardziej powszechnych standardów HTML, CSS, Flash i AJAX. Użytkownik IDE WebRatio może wpływać na całokształt aplikacji poprzez modyfikację reguł generowania jej kodu oraz dzięki możliwości tworzenia własnych komponentów.

Warto wspomnieć, że twórcy IDE WebRatio twierdzą, że ich narzędzie można wykorzystać na każdym etapie cyklu życia aplikacji – od określenia wymagań, przez prototypowanie i wypuszczenie wersji finalnej, do korygowania błędów i rozbudowy aplikacji. Ponadto deklarują wsparcie dla interfejsu Web2.0, aplikacji działających w czasie rzeczywistym oraz możliwość użycia protokołów Web, WAP i PDA w celu tworzenia aplikacji dostępnych dla wielu platform [3].

#### 4. KOMPONENTY ŚRODOWISKA IDE WEBRATIO 5.1

Kompleksowość środowiska wynika z wykorzystania wielu komponentów wspierających jego funkcjonowanie, wśród których warto wspomnieć o:

1. Apache Ant 1.7.0 – narzędzie służące do zautomatyzowania procesu budowania oprogramowania. Zostało napisane w języku Java, co pozwoliło na jego wykorzystanie w wielu środowiskach programistycznych. Umożliwia:
  - definiowanie budowanych obszarów oraz zależności pomiędzy nimi,
  - automatyczne pozyskiwanie kodu źródłowego z repozytoriów,
  - budowanie aplikacji poprzez kompilację niezbędnych plików źródłowych w odpowiedniej kolejności.Pliki konfiguracyjne narzędzia mają postać dokumentów XML [4].
2. Apache POI 3.2-FINAL – „Apache POI – Java API To Access Microsoft Format Files”, zbiór interfejsów służących manipulowaniu plikami w formatach bazujących na Microsoft’s OLE 2 Compound Document oraz OpenXML przy użyciu czystego języka Java. Obecnie wspierane są m.in. pliki XLS (Excel 97 – 2007), DOC (Word 97 - 2007), XLSX (Excel 2007+), DOCX (Word 2007+), PPTX (Power Point 2007+) [5].
3. Apache Struts 1.3.8 – elastyczny framework bazujący na technologii Java, dokumentach XML oraz pakietach ang. Apache Commons, składający się z komponentów:
  - ang. Controller – dostarczony z frameworkiem,
  - ang. Model – współpracuje z powszechnie używanymi technologiami dostępu do danych, takimi jak JDBC, EJB, Hibernate, Object Relational Bridge,
  - ang. View – współpracuje m.in. z JSP, JSTL, JSF i XSLT [6].
4. Hibernate 3.3.1 – narzędzie przeznaczone dla środowiska Java, wykorzystujące technikę ORM (ang. Object/Relational Mapping) polegającą na mapowaniu reprezentacji danych z modelu obiektowego na relacyjny model danych. Pozwala na mapowanie klas języka Java na tabele bazy danych, typów danych języka Java na typy danych charakterystyczne dla języka SQL oraz dostarcza narzędzia do dwukierunkowej komunikacji między bazą danych i aplikacją, z automatycznym użyciem sterowników JDBC [7].
5. Java Mail 1.4 – niezależny od platformy i protokołu framework pozwalający na budowanie aplikacji w technologii Java zarządzających wiadomościami e-mail. Dostarcza narzędzia do odbierania i wysyłania wiadomości oraz definiowania dostawców usług (ang. Service Providers). Firma Sun dostarczała zbiór pakietów, w skład którego wchodzi JavaMail API oraz pakiety pozwalające na obsługę serwerów IMAP, POP3 i SMTP [8].
6. Quartz 1.6.1 – darmowy framework o otwartym kodzie, służący do tworzenia harmonogramów zadań dla systemów informatycznych zbudowanych w oparciu o J2EE lub J2SE, wspierający JTA (ang. Java Transaction API), czyli mechanizm transakcji rozproszonych oraz klastrowanie. Może być użyty w dużych aplikacjach komercyjnych, w których zadania są zdefiniowane jako EJBs (ang. Enterprise Java Beans) [9].

## 5. STRUKTURA ŚRODOWISKA IDE WEBRATIO 5.1

Instalator IDE WebRatio 5.1 podczas procesu instalacji tworzy katalog WebRatio5.1.0 o strukturze, której najważniejsze elementy to:

1. Tomcat – folder zawierający zainstalowany serwer aplikacji Apache Tomcat dołączony do IDE WebRatio 5.1. W podkatalogu webapps są umieszczane wygenerowane przez środowisko aplikacje.

2. WebRatio – katalog zawierający zainstalowane narzędzie WebRatio 5.1. Ponieważ opisywana wersja narzędzia bazuje na środowisku Eclipse 3.4, katalog ma strukturę charakterystyczną dla tego środowiska. Jego najważniejsze podkatalogi to [10]:
  - configuration – katalog zawierający informacje pozwalające na uruchomienie środowiska oraz informacje gromadzone w trakcie jego działania,
  - drivers – katalog zawierający sterowniki pozwalające na dostęp do źródeł danych w postaci baz danych lub plików o odpowiedniej strukturze,
  - features – katalog zawierający składniki środowiska, czyli informacje o wtyczkach realizujących wspólne funkcje,
  - p2 – katalog wykorzystywany przez framework Equinox/P2, który odpowiada za aktualizację środowiska,
  - plugins – katalog zawierający wtyczki, czyli programy w języku Java rozszerzające funkcjonalność środowiska Eclipse.
3. workspace – katalog zawierający katalogi z projektami tworzonymi w IDE WebRatio 5.1.

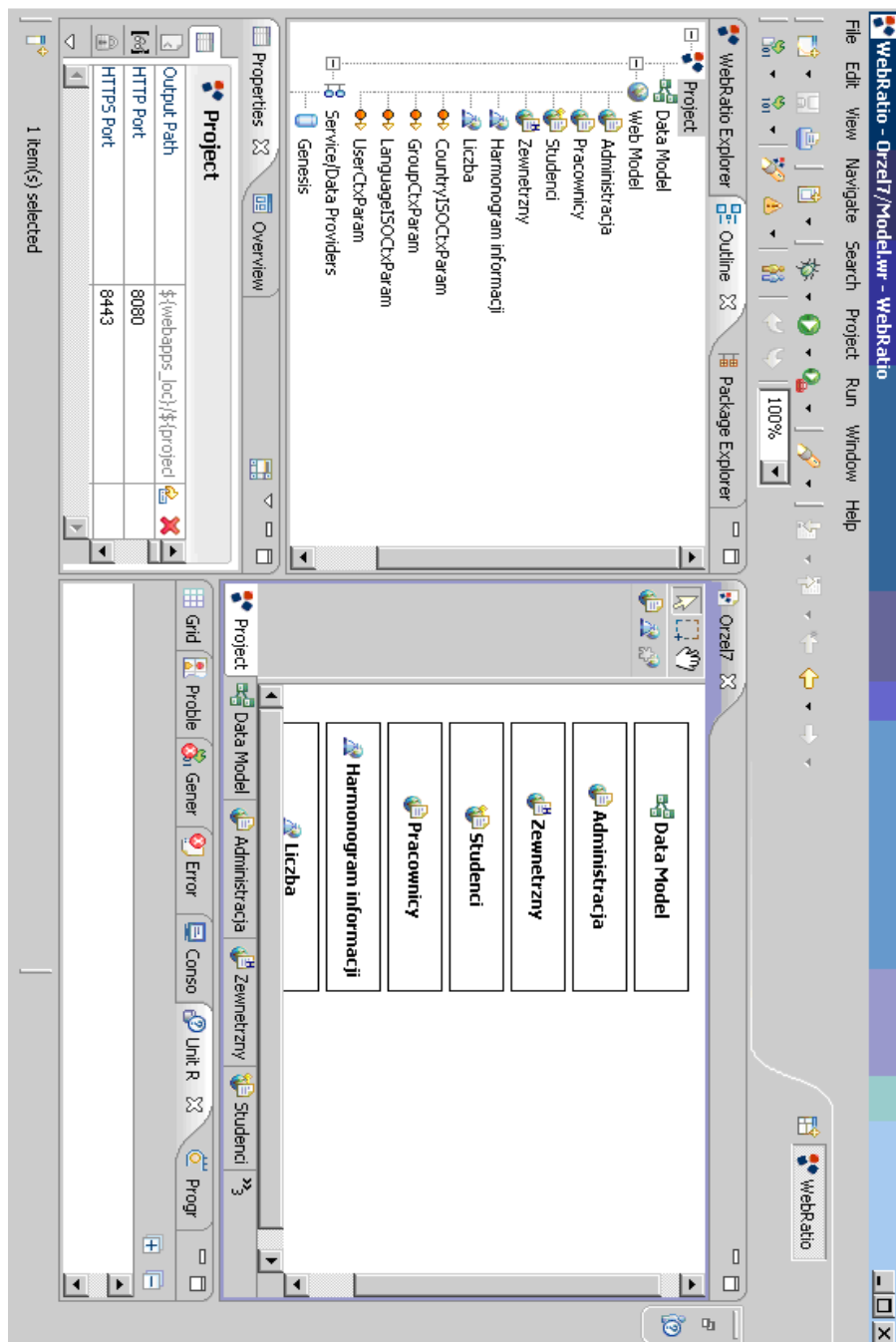
Katalog zawierający projekt tworzonej w środowisku IDE WebRatio 5.1 aplikacji ma następującą strukturę:

1. .Temp~ – katalog zawierający m.in. logi oraz informacje dotyczące synchronizacji modelu danych ze źródłami danych.
2. DBScripts – katalog zawierający skrypty wygenerowane w trakcie mapowania elementów modelu danych do źródeł danych.
3. WebContent – katalog zawierający m.in. tekst i pliki multimedialne prezentowane w ramach stron aplikacji oraz pliki szablonów np. wiadomości e-mail.
4. Model.wr – główny plik z projektem aplikacji.

Katalog z wygenerowaną na podstawie projektu i umieszczoną na serwerze aplikacją ma strukturę typową dla technologii JSP.

## 6. WPROWADZENIE DO NARZĘDZIA WEBRATIO 5.1

Graficzny interfejs użytkownika narzędzia WebRatio 5.1 (rys. 5) standardowo składa się z następujących sekcji [11]:



Rys.5. Obszar roboczy narzędzia WebRatio 5.1



1. Eksplorator WebRatio (ang. WebRatio Explorer) – pokazuje poniższe elementy:
  - aktualnie otwarte projekty aplikacji internetowych,
  - aktualnie otwarte projekty stylów,
  - predefiniowane w WebRatio, nie podlegające modyfikacji, projekty stylów,
  - aktualnie otwarte projekty komponentów,
  - predefiniowane w WebRatio, nie podlegające modyfikacji, projekty komponentów.
2. Zarys (ang. WebRatio Outline) – przedstawia strukturę aktualnie aktywnego elementu projektu.
3. Podgląd projektu (ang. Overview) – widok obszaru projektu z dalekiej perspektywy.
4. Widok właściwości (ang. Properties View) – prezentuje i umożliwia modyfikację właściwości aktualnie aktywnego elementu projektu.
5. Widok siatki (ang. Grid View) – pozwala na rozmieszczenie komponentów danych w ramach strony.
6. Lista problemów (ang. Problems View) – zawiera wykaz błędów i ostrzeżeń wykrytych w trakcie analizy poprawności stworzonego modelu.
7. Widok logów (ang. Log View) – pozwala na otwieranie plików z logami wygenerowanej aplikacji.
8. Pomoc komponentu (ang. Unit Reference) – zawiera kompletny opis właściwości komponentu modelu.
9. Obszar edytora (ang. Editor Area) – obszar, w którym budowany jest projekt, umożliwiający również wyświetlenie oraz edycję plików tekstowych wchodzących w skład aplikacji.
10. Pomoc (ang. Help) – sekcja pozwalająca na korzystanie z dokumentacji dostarczonej wraz ze środowiskiem.

Narzędzie WebRatio 5.1 pozwala na tworzenie trzech typów projektów [11]:

1. Aplikacji internetowej – projekt zawiera modele projektowanej aplikacji wykonane w notacji WebML. Model prezentacji aplikacji jest określany poprzez wybór wcześniej wykonanych projektów stylów.
2. Stylów – projekt określający warstwę prezentacji aplikacji internetowej. Zawiera kolekcję szablonów, na podstawie których są generowane strony HTML oraz grupuje zasoby takie, jak arkusze CSS, pliki multimedialne, applety Java, kontrolki ActiveX oraz instrukcje języków skryptowych wykonywane po stronie klienta.
3. Komponentów – projekt zawierający zestaw zdefiniowanych przez projektanta komponentów modelu hipertekstowego wraz z zasobami niezbędnymi w trybie wykonania.

## 7. ŚRODOWISKO PRACY GENEROWANYCH APLIKACJI

Jak już wcześniej wspomniano, w skład środowiska programistycznego wchodzi serwer aplikacji, na którym automatycznie są umieszczane aplikacje wygenerowane na podstawie opracowanych modeli. W przypadku IDE WebRatio 5.1 jest to Apache Tomcat 5.5.27, który jest implementacją technologii Java Servlet i JavaServer Pages rozpowszechnianą na licencji Apache Software License i składa się z następujących komponentów [12]:

1. Catalina – kontener servletów implementujący specyfikacje firmy Sun: Servlet 2.4 i JSP 2.0.

2. Coyote – konektor HTTP wspierający protokół HTTP 1.1. Nasłuchuje określony port TCP, przekazuje żądania do serwera Tomcat w celu ich przetworzenia oraz wysyła odpowiedzi do klientów.
3. Jasper – silnik JSP odpowiedzialny za analizę plików JSP i ich kompilację do servletów, które mogą zostać przetworzone przez Catalinę. W trakcie działania automatycznie wykrywa zmiany w plikach JSP i dokonuje ich rekompilacji.

Wygenerowane aplikacje, poprzez ORM Hibernate, mogą współpracować z wieloma bazami danych, w tym z bazą danych Apache Derby 10.4 dostarczoną w pakiecie instalacyjnym IDE WebRatio 5.1. Jest ona w całości napisana w Javie, dostępna na licencji Apache License, Version 2.0. Jej kluczowe zalety, to [13]:

- niewielki rozmiar – silnik bazodanowy i sterownik JDBC zajmują około 2 MB,
- bazuje na standardach Java, JDBC i SQL,
- jest dostarczana z wbudowanym sterownikiem JDBC,
- dwa tryby działania embedded i server.

## 8. WNIOSKI

Niestety środowisko programistyczne IDE WebRatio 5.1 nie jest pozbawione wad. Najtrudniejszą do zrozumienia jest brak implementacji procesu specyfikacji wymagań. Dołączenie modułu obsługującego tworzenie diagramów UML i dokumentów będących produktami specyfikacji nie powinno być szczególnie kłopotliwe ze względu na wykorzystanie Eclipse.

Największą wadą jest skomplikowany proces tworzenia niestandardowych rozwiązań w ramach narzędzia WebRatio. Szczególne problemy sprawia AJAX tam, gdzie braknie predefiniowanych rozwiązań oraz niestandardowe szablony wyglądu aplikacji. Nie przekreśla to jednak zastosowań dydaktycznych (kilka predefiniowanych szablonów gotowych do wybrania), a jedynie utrudnia komercyjne.

Występują sporadyczne problemy związane ze stabilnością działania oraz odświeżaniem zmian dokonywanych w projekcie.

W modelu danych opcja usuwania mapowania okazuje się nie być ułatwieniem, w dodatku edytor umożliwia tworzenie połączeń pomiędzy encjami pochodzącymi z różnych źródeł danych mimo tego, że jest to zabronione. Nieprzyjemną niespodzianką jest pokrycie się nazw encji personalizacyjnych ze słowami zastrzeżonymi w bazie danych Oracle. Rozwiązanie problemu nie jest skomplikowane, ale wprowadza niepotrzebne zamieszanie.

Komunikaty o błędach bywają nazbyt ogólnikowe, co utrudnia debugowanie wygenerowanej aplikacji. W większej mierze dotyczy to informacji o błędach wyświetlanych w oknie przeglądarki internetowej, niż w narzędziu WebRatio. Naganny jest stan dokumentacji, która jest niekiedy przestarzała, co znacząco utrudnia wykonanie najbardziej zaawansowanych zadań. Jednakże powinno to ulec poprawie dzięki pracom, w które są zaangażowani również autorzy artykułu.

Na szczęście wady są w dużej mierze rekompensowane przez zalety. Siła IDE WebRatio tkwi w integracji serwera aplikacji, narzędzia developerskiego, bazy danych i generatora kodu źródłowego aplikacji oraz darmowej licencji akademickiej. Studenci

dzięki specyfice notacji i zintegrowanym komponentom IDE WebRatio są w stanie szybko sprawdzić, jak zmiany w modelu wpływają na działanie i wygląd końcowej aplikacji.

Jest możliwe uzyskanie w pełni funkcjonalnego oprogramowania bez znajomości jakiegokolwiek języka oprogramowania. Pozwala to na umiejscowienie laboratoriów z MDE na dowolnym etapie cyklu edukacyjnego i uzyskanie podstaw do nauczania innych przedmiotów, których tematyką jest wytwarzanie oprogramowania. Nauka dobrych praktyk projektowania aplikacji internetowych, takich jak np. MVC, nie wymaga podwalin w postaci znajomości języków serwer-side. Można więc w jednakowy sposób uczyć wszystkich studentów, niezależnie od ich późniejszego wyboru specjalizacji.

Implementacja WebML w IDE WebRatio 5.1 nadaje się idealnie do nauki szybkiego tworzenia średniej wielkości aplikacji internetowych zarządzających dużą ilością danych, w których nie odgrywa istotnej roli warstwa prezentacji. W przypadku niewielkich stron internetowych, stron nastawionych na wygląd prezentowej treści i w pełni profesjonalnych aplikacji przemysłowych opartych o J2EE oraz towarzyszące jej technologie, lepiej jest skorzystać z innych dostępnych rozwiązań.

Nie bez znaczenia jest fakt, że środowisko jest cały czas rozwijane i ulepszane, pojawiają się nowe komponenty oraz modele. Poprawiane są również poważniejsze błędy.

Podsumowując, IDE WebRatio posiada duży potencjał, jest łatwe w obsłudze i pomimo pewnych braków godne polecenia. Można się posunąć do stwierdzenia, że nadaje się do zastosowań dydaktycznych, jak mało które.

## 9. BIBLIOGRAFIA

- [1] [webml.org](http://webml.org), wrzesień 2010
- [2] Ceri S., Fraternali P., Bongio A., Brambilla M., Comai S., Matera M.: *Designing Data – Intensive Web Applications*, Elsevier Science 2003
- [3] [webratio.com](http://webratio.com), wrzesień 2010
- [4] [ant.apache.org](http://ant.apache.org), wrzesień 2010
- [5] [poi.apache.org](http://poi.apache.org), wrzesień 2010
- [6] [struts.apache.org](http://struts.apache.org), wrzesień 2010
- [7] [hibernate.org](http://hibernate.org), wrzesień 2010
- [8] [java.sun.com/products/javamail/](http://java.sun.com/products/javamail/), wrzesień 2010
- [9] [opensymphony.com/quartz/](http://opensymphony.com/quartz/), wrzesień 2010
- [10] [wiki.eclipse.org](http://wiki.eclipse.org), wrzesień 2010
- [11] *WebRatio User Guide*, Web Models 2009
- [12] [tomcat.apache.org](http://tomcat.apache.org), wrzesień 2010
- [13] [db.apache.org/derby/](http://db.apache.org/derby/), wrzesień 2010