

SYSTEM INFORMATYCZNY DO ZARZĄDZANIA BADANIAM PROWADZONYMI W OBSZARZE INŻYNIERII ROLNICZEJ, BAZUJĄCY NA TECHNOLOGII .NET ORAZ XML

Streszczenie

Racjonalne zarządzanie badaniami prowadzonymi przez ośrodki badawcze, z uwagi na konieczność przetwarzania znacznej ilości niejednorodnych informacji, nie jest praktycznie możliwe bez wspomagającego go systemu informatycznego. Proponowany przez autorów system informatyczny, pozwalający na zarządzanie tymi danymi, to aplikacja internetowa bazująca na SZBD SQL Server 2000 oraz wykorzystująca technologie ASP .NET i XML. Standaryzacja danych, osiągnięta przy wykorzystaniu technologii XML, ma umożliwić dalsze zindywidualizowane ich przetwarzanie przez użytkowników. Na etapie projektowania systemu, w celu przyspieszenia jego realizacji oraz uniknięcia błędów, zostały w pełni wykorzystane narzędzia inżynierii oprogramowania.

Słowa kluczowe: zarządzanie badaniami, system informatyczny, ASP.NET, XML

Wprowadzenie

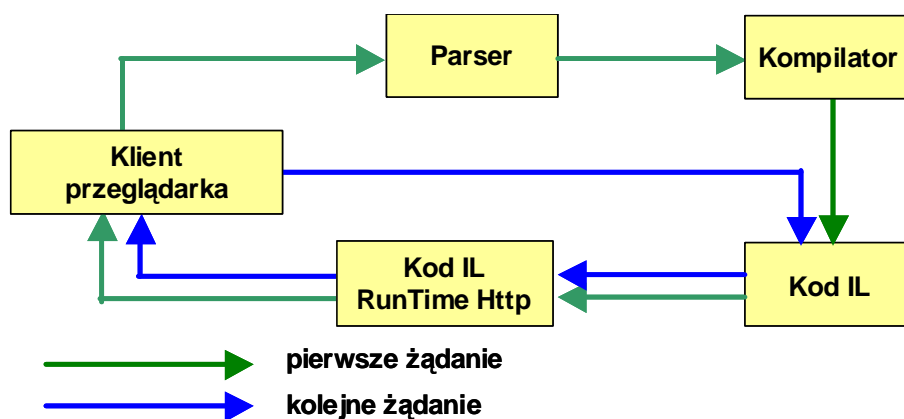
Funkcjonowanie przedsiębiorstw i ośrodków badawczych jest nierozdzielnie sprzężone z prowadzeniem i przepływem dokumentów zawierających różnorodne informacje. Wzrastająca liczba dokumentów i związanych z nimi informacjami podlegającymi przetwarzaniu, wymusza stosowanie specjalistycznych systemów informatycznych zorientowanych na określone grupy użytkowników. Niewątpliwie ośrodki badawcze, działające na potrzeby sektora rolniczego, z uwagi na charakter prowadzonych działań jak i dużą mobilność pracowników, stanowią specyficzną grupę. Brak wspomnianego systemu komplikuje i generuje dodatkowe nakłady pracy w tych jednostkach organizacyjnych.

Celem publikacji jest zaprojektowanie, połączone z wieloaspektowym i wielopoziomowym modelowaniem, oraz wytworzenie rozproszonego systemu informatycznego, wspomagającego proces zarządzania badaniami prowadzonymi w Instytucie Inżynierii Rolniczej. Złożoność danych podlegających przetwarzaniu, w połączeniu z różnorodnością

wymagań użytkowników zdecydowały o przyjęciu założenia, iż tworzony system będzie miał budowę modułową. Pierwszy z nich będzie odpowiedzialny za dokumentowanie całościowej działalności badawczej, prowadzonej w jednostce, natomiast drugi będzie wykorzystany do zarządzania zasobami finansowymi, przeznaczonymi do realizacji badań. Jego funkcjonalność zdecydowała o rozszerzeniu obszaru jego działania o sferę dydaktyki. Do wytworzenia systemu wykorzystano najnowsze technologie, takie jak: XML, .NET w tym ASP.NET

Technologia ASP .NET i XML

Jedną z najnowszych technologii do tworzenia internetowych systemów informatycznych, szczególnie przydatnych w sektorze rolniczym, jest ASP .NET, które pomimo prawie identycznej nazwy nie stanowi kontynuacji ASP. Ta całkiem nowa idea budowy programowalnych stron WWW stanowi integralną część platformy .NET [Worley 2003]. Istotną zmianą, eliminującą ograniczenia poprzedniczki, jest możliwość tworzenia aplikacji przy wykorzystaniu pełnej gamy języków programowania, takich jak; C#, Visual Basic .NET, C++, Jscript .NET i inne. Stało się to możliwe między innymi dzięki wprowadzeniu w platformie .NET typów danych zgodnych ze specyfikacją CLS (*Common Language Specification*) [Worley 2003].

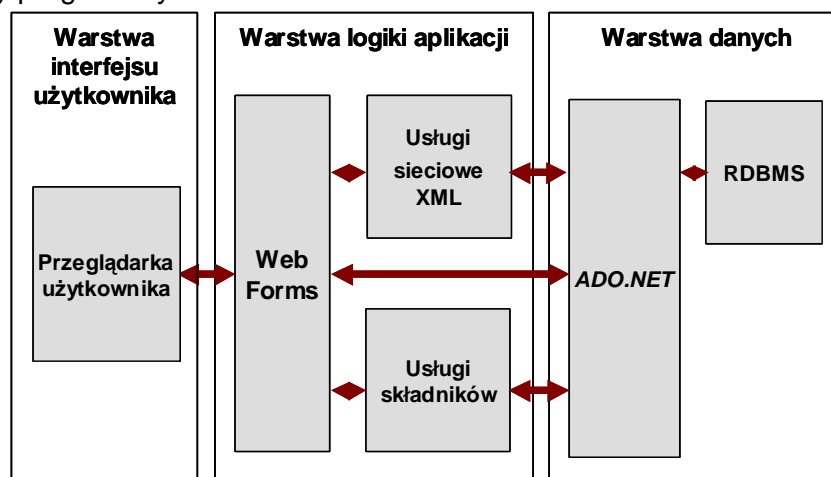


Rys. 1. Przebieg uruchamiania aplikacji ASP.NET
Fig. 1. Starting the ASP.NET application

Tworzenie aplikacji rozproszonych w klasycznym ASP miało jeszcze jedną istotną wadę, polegającą na umieszczaniu kodu HTML oraz fragmentów programu, napisanych w języku skryptowym (np. Visual Basic), akceptowalnym przez to środowisko, w jednym wykonywalnym pliku .asp. Brak możliwości dekompozycji kodu był szczególnie odczuwalny w przypadku złożonych systemów informatycznych, podlegających częstym

modyfikacjom. Aktualnie bazując na nowej technologii ASP.NET, programista uzyskał możliwość oddzielania kodu od interfejsu użytkownika. Istotną zauważalną różnicą jest też sposób przetwarzania, który aktualnie odbywa się w ramach jednej strony, a nie kilku jak było dotychczas, gdzie każda z nich wykonywała część określonego zadania. Wpływa to korzystnie na szybkość działania systemu informatycznego, podobnie jak zmiana organizacji wykonywania operacji po stronie serwera, co graficznie obrazuje (rys. 1).

Funkcjonowanie aplikacji ASP.NET, oparte na różnych technikach wbudowanych w platformę .NET, z rozbiciem na poszczególne warstwy, dostępne dla programisty przedstawia rysunek 2. Kluczowym elementem prezentowanego schematu, a zatem i aplikacji, jest Web Forms [Payne 2002] osadzony w warstwie środkowej. Odpowiedzialny on jest za prezentację danych i ich przetwarzanie sterowane zdarzeniami. Kontrolki, tworzące jego biblioteki, będące podstawowymi elementami składowymi aplikacji, mając możliwość generowania interfejsu użytkownika dopasowanego do jego przeglądarki. Uzyskana tym sposobem kompatybilność systemu informatycznego eliminuje dodatkowe nakłady pracy programisty.



Rys. 2. Możliwa architektura aplikacji ASP.NET [Worley 2003]
 Fig. 2. A possible architecture of the ASP.NET application [Worley 2003]

Analiza obiektowa zorientowana na ASP.NET pozwala opisać etapy przetwarzania pliku .aspx oraz określić jego strukturę. Każda strona tego typu dziedziczy metody i właściwości klasy *Page*, która jest zlokalizowana w przestrzeni nazw *System.Web.UI* – traktowanej jako kontener. Sytuacja nieco się komplikuje, gdy programista umieszcza kod w osobnym pliku (np.

.vb). Zawarte w nim zdefiniowane klasy stanowią rozszerzenie klasy *Page*, co jest zasygnalizowane w kodzie przez określenie związku dziedziczenia. Natomiast strona .aspx, wyposażona w niezbędne kontrolki interfejsu użytkownika, stanowi z kolei rozszerzenie klasy zdefiniowanej w tym zewnętrznym pliku, na co wskazują wartości odpowiednich atrybutów *Src* i *CodeBehind*.

Podstawowy budulec wspomnianych stron stanowią już sygnalizowane kontrolki serwera oraz kontrolki sprawdzania poprawności. Na etapie implementacji z reguły najpierw wykorzystujemy kontrolki pierwszej grupy, tworzące interfejs graficzny użytkownika, które możemy podzielić na dwa rozłączne podzbiory – kontrolki HTML i kontrolki sieci WWW.

Nowszą biblioteką, o znacznie większych możliwościach, wynikających z szerszego wachlarza właściwości oraz automatycznego rozpoznawania przeglądarki są kontrolki sieci WWW. W przypadku programowalnych stron WWW, które przetwarzają dane pochodzące z SZBD, do ich udostępnienia niewątpliwie wykorzystamy kontrolki *DataGrid* oraz *DataList* i interfejs programistyczny dostępu do danych ADO lub zalecaną jego nowszą wersję ADO.NET.

Model dostępu do danych. ADO.NET [Worley 2003] to całkowicie przeprojektowana i przebudowana technologia, która oferuje szerszą paletę obiektów, w efekcie czego uzyskujemy nowe możliwości. Szczególną rolę odgrywa obiekt *DataSet* o potężnych możliwościach, który pozwala na jednoczesne manipulowanie danymi pochodzącymi z wielu zestawów rekordów. Upraszczając problem, obiekt *DataSet* można traktować jako obraz bazy danych, odzwierciedlony w pamięci operacyjnej serwera.

Kolejną nowością, mającą prawdopodobnie charakter długofalowy, jest orientacja w technologii ASP.NET na język XML. Język ten stanowi wewnętrzną podstawę tej technologii, która z kolei dostarcza nam biblioteki klas pozwalające na przetwarzanie i zwracanie dokumentów XML. Predefiniowane klasy do wykorzystania w aplikacji zlokalizowane są w przestrzeni nazw *System.XML*. Szczególnie przydatna przy tworzeniu aplikacji może się okazać klasa *XmlDataDocument*, zwłaszcza w sytuacji, gdy dane pochodzące z SZBD gromadzone są w obiekcie *DataSet*. Współpraca owych obiektów pozwala na dokonywanie transferu danych w obie strony, czyli z schematu relacyjnego w dokument XML i na odwrót.

Projektowanie

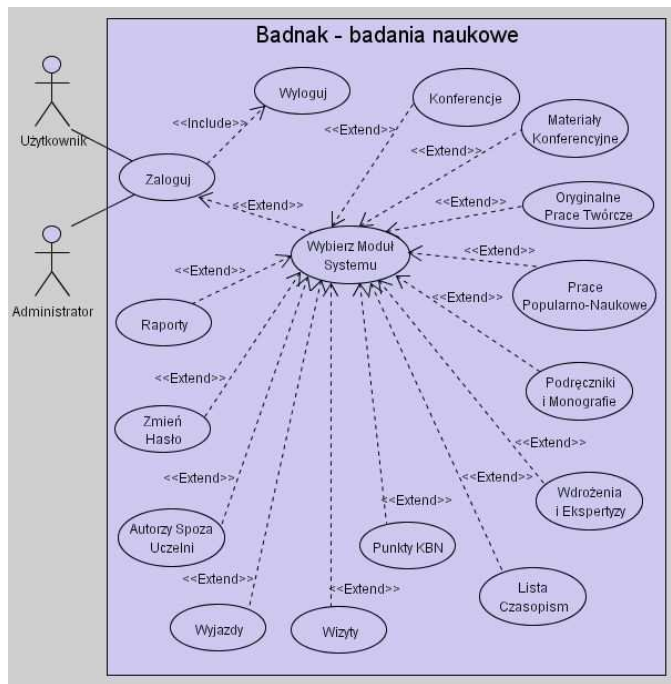
Tworzone oprogramowanie, zgodnie z przeznaczeniem ma wspomagać proces dokumentowania i zarządzania pracami badawczymi, prowadzonymi w obszarze inżynierii rolniczej. Z uwagi na specyfikę

funkcjonowania pracowników nauki, czyli użytkowników systemu, zdecydowano się na rozproszony charakter aplikacji. Proces wytwarzania systemu przebiegał, zgodnie z założeniami autorów, według etapów przewidzianych w modelu kaskadowym, aczkolwiek złożoność problematyki, trudności w uchwyceniu wszystkich szczegółów, wymusiły występowanie iteracji. Oznaczało to, iż występowała również częściowa zgodność postępowania z modelem „realizacja przyrostowa”.

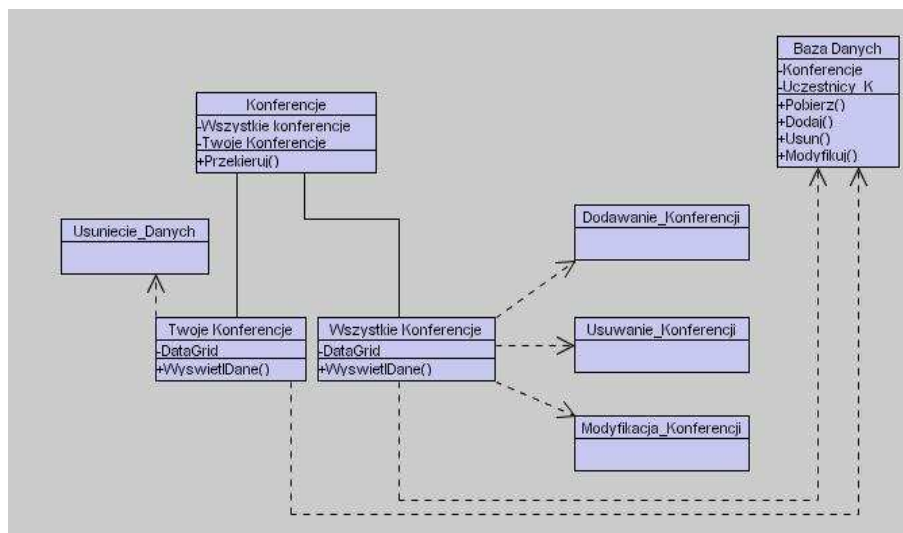
Projektowanie przebiegło przy pełnym wykorzystaniu notacji graficznej języka modelowania UML [Booch i in. 2000] i z jednoczesnym stosowaniem zasad inżynierii oprogramowania. Jako narzędzie do budowy diagramów przypadków użycia oraz klas, które odwzorowują zarówno dynamikę, jak i strukturę systemu, użyto środowisko Visual Paradigm. Diagramy przypadków użycia reprezentują sobą wymagania funkcjonalne systemu. Odwzorowanie zidentyfikowanych wymagań osiągamy przez wprowadzenie aktorów oraz zbioru sprecyzowanych dla nich scenariuszy. Z uwagi na złożoność dziedziny problemowej zastosowano procedurę etapowego, wielopoziomowego tworzenia diagramów. Wytworzony diagram przypadków użycia, z uwagi na poziom ogólności, zlokalizowany najwyżej w hierarchii ilustruje rysunek 3.

Modelowanie struktury systemu informatycznego, na którą składają się klasy, interfejsy, obiekty oraz wzajemne powiązania między nimi przeprowadzono za pomocą wielopoziomowych diagramów klas. Jeden z poziomów zrealizowanego modelowania przedstawia rysunek 4.

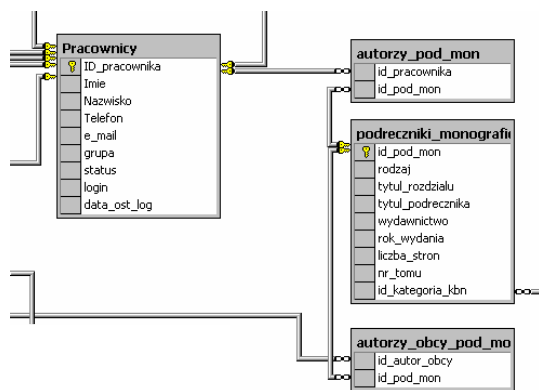
Wytworzona aplikacja, zgodnie z przyjętymi założeniami współpracuje z SZBD. Charakter przechowywanych danych, wynikający z przeprowadzonego modelowania, wskazywał, iż zalecanym obszarem wyboru powinny być bazy relacyjne. Z uwagi na silne wsparcie technologii XML podjęto decyzję, iż wytworzona aplikacja będzie bazowała na SQL Serwer 2000 [Griffin 2002]. Kolejnym etapem projektowania było modelowanie struktur relacyjnych przy użyciu narzędzi graficznych wbudowanych w SQL Serwer 2000 [Waymire, Sawtell 2002]. Efekty modelowania logicznego i fizycznego bazy danych przedstawia rysunek 5.



Rys. 3. Ogólny diagram przypadków użycia
 Fig. 3. The general use case diagram



Rys. 4. Szczegółowy diagram klas związany z konferencjami
 Fig. 4. The specific class diagram corresponding to conferences



Rys. 5. Fragment struktury baz danych – tabele i powiązania
 Fig. 5. A fragment of the database structure – tables and links

Narzędzia programistyczne

Omówione wyżej technologie zostały z powodzeniem zastosowane w wytworzonym systemie informatycznym. Jego kodowanie, z uwagi na przyjęte rozwiązania techniczne, zrealizowano w zintegrowanym środowisku programistycznym Visual Studio .Net, dedykowanym między innymi tego typu aplikacjom. Operacje budowania aplikacji ASP.NET wykonujemy w głównym oknie projektu, które staje się dostępne po określeniu typu tworzonego programu. Otoczone jest ono elementami graficznymi, reprezentującymi moduły narzędziowe, których stosowanie ułatwia pracę, skraca czas tworzenia oraz pozwala poprawnie zarządzać projektem. Obszar zasadniczy oferuje szereg widoków, wynikających z przyjętej idei, iż kod programu powinien być oddzielony od interfejsu użytkownika. W związku z tym, ma do dyspozycji dwie najczęściej wykorzystywane perspektywy, pierwsza z nich umożliwia płynne graficzne projektowanie formularzy - stron, druga służy do wprowadzania kodu.

Poprawne funkcjonowanie zbudowanego systemu wymaga następujących składników, komponentów informatycznych zlokalizowanych po stronie serwera:

- Windows Serwer.
- Serwer WWW – IIS.
- Platforma FrameWork.
- SQL Serwer 2000.

Natomiast po stronie klienta powinna być dostępna przeglądarka.

System informatyczny

Złożoność wytworzonego systemu informatycznego nie implikuje trudności w jego użytkowaniu. Przemysłane, uporządkowane hierarchicznie oraz zunifikowane formularze i podformularze pozwalają w sposób płynny i zrozumiały przemieszczać się po aplikacji. Etap poprzedzający korzystanie z programu, będący konsekwencją poufności danych, to identyfikacja użytkownika na podstawie nazwy i hasła. Pomyślnie jego przejście udostępnia formularz (rys. 6), pełniący rolę panelu sterowania, wyposażonego w pogrupowane przyciski akcji, które pozwalają na przeglądanie i edycję danych związanych z prowadzonymi badaniami oraz formami udostępniania ich rezultatów. Wybranie przycisku o nazwie „Konferencja” udostępnia użytkownikowi listę wszystkich konferencji zarejestrowanych w bazie danych (rys. 7), ale ponadto oferuje kompletny zestaw operacji możliwych do wykonania na nich, oczywiście zależny od posiadanych uprawnień.



Rys. 6. Strona główna – Badnak

Fig. 6. The main page – Badnak

Podobny sposób przemieszczania się po pogrupowanych formularzach towarzyszy rejestracji, modyfikacji danych dotyczących względnie trwałych form prezentacji wyników badań, jakimi są publikacje. Duża różnorodność form ich udostępniania implikuje zróżnicowane pakiety informacji do zapisania lub modyfikacji, co z kolei wymusza szeroką gamę formularzy.

ID	Rodzaj	Nazwa	Data	Miejsce	Organizator	L. ref.	L. wykł.	L. przew. sesji	L. osób	Uczestnicy	Uczestnicy
Edytuj 13	krajowa druga		2002-02-14	Poznan	AR Poznan	3	1	1		Pokaż	Dodaj Mnie
Edytuj 17	krajowa szósta		2002-06-23	Leszno	AR Poznan	2	1	1		Pokaż	Dodaj Mnie
Edytuj 22	krajowa siedemnasta		2003-10-10	Olsztyn	UWM				1	Pokaż	Dodaj Mnie
Edytuj 23	krajowa dwunasta		2001-07-15	Lublin	AR Lublin					Pokaż	Dodaj Mnie
Edytuj 24	krajowa Dziewiąta		1999-06-19	Radom	AR Lublin	1	1	1	1	Pokaż	Dodaj Mnie
Edytuj 25	krajowa trzydziesta		2004-01-05	Warszawa	SGGW	1	1	1	1	Pokaż	Dodaj Mnie
Edytuj 26	Krajowa Jedenasta		2004-02-12	Poznan	AR Poznan				3	Pokaż	Dodaj Mnie

Użytkownik: Adam Adamowski (administrator) Ostatnio zalogowany: 2004-01-06 23:45:47 Wyloguj

Rys. 7. Formularz „Wszystkie Konferencje”
Fig. 7. The „All Conferences” form

Niektóre z tych formularzy pozwalają na prezentację danych w postaci dokumentów XML, stwarzając użytkownikowi możliwość ich dalszego zindywidualizowanego przetwarzania, włącznie z zapisem do innej bazy danych. Równoległym sposobem uzyskiwania zbiorczych informacji, dotyczących zarówno publikacji, udziału w konferencjach itp., jest korzystanie z zaprojektowanych raportów. Porcja informacji uzyskiwana tą drogą jest zróżnicowana i wynika wprost z posiadanych uprawnień.

Wnioski

1. Użycie wizualnego środowiska programistycznego Visual Studio .NET ułatwia i zwiększa wydajność wytwarzania złożonych, specjalistycznych systemów informatycznych.
2. Zastosowanie rozproszonego modelu aplikacji zmniejsza ograniczenia dostępu do systemu oraz ułatwia jego pielęgnację.
3. Wykorzystanie technologii XML do prezentacji danych umożliwia ich dalsze zindywidualizowane przetwarzanie.
4. Modułowa budowa systemu umożliwia jego dalszą rozbudowę.
5. Zastosowanie technologii ASP.NET w tym kontrolek sieci WWW do budowy interfejsu użytkownika, dostępnego przez dowolną przeglądarkę, minimalizuje wymagania niefunkcjonalne, stawiane przez system.

6. Wytworzona aplikacja, ze względu na swoje możliwości może być z powodzeniem stosowana w innych ośrodkach i placówkach naukowych.

Bibliografia

Booch G., Rumbaugh J., Jacobson I. 2000. UML przewodnik użytkownika. WNT, Warszawa

Griffin J. 2002. XML i SQL Server 2000. Helion, Gliwice

Payne Ch. 2002. ASP.NET dla każdego. Helion, Gliwice

Waymire R., Sawtell R. 2002. MS SQL Server 2000 dla każdego. Helion, Gliwice

Worley S. 2003. ASP.NET Vademecum profesjonalisty. Helion, Gliwice

INFORMATION SYSTEM BASED ON .NET AND XML TECHNOLOGIES FOR MANAGING THE RESEARCH IN AGRICULTURAL ENGINEERING

Summary

Rational management of the research performed in research centers is practically impossible without supporting information system due to necessity of processing a huge amount of heterogeneous information. The information system proposed by the authors and allowing to manage such data is a form of internet application based on the SQL Server 2000 database management system and the ASP.NET and XML technologies. Standardization of data achieved with the use of the XML technology enables further data processing by individual users of the system. The system development was fully grounded on the software engineering tools to eliminate defects and to deploy the system faster.

Key words: research management, information system, ASP.NET, XML

Recenzent – Jerzy Dąbkowski