

SYSTEMOWE WSPOMAGANIE ZAJĘĆ PROJEKTOWO-LABORATORYJNYCH W ZAKRESIE KSZTAŁCENIA PROBLEMOWEGO INŻYNIERÓW

Streszczenie

W artykule przedstawiono wybrane elementy wspomaganie zajęć dydaktycznych w zakresie zagadnień inżynierskich realizowanych w zespołach projektowych. Szczególną uwagę zwrócono na elementy kształcenia problemowego oraz metodologię opracowywania scenariuszy dydaktycznych i przypisywania właściwych ról członkom zespołu projektowego. Przedstawiono pakiet programów Pulpit nauczyciela oraz Pulpit studenta wspomagających możliwości prezentacyjne prowadzącego zajęcia, między innymi w zakresie omawiania przedmiotowych aplikacji wykorzystywanych podczas projektowania, wykonywania obliczeń numerycznych oraz programowania. Stwierdzono, że synergia kształcenia problemowego, grupowego rozwiązywania problemów oraz właściwie umocowanych indywidualnych kompetencji jest najlepszą podstawą opracowywania kursów wykorzystujących nowoczesne metody i technologie edukacyjne.

WSTĘP

W artykule przedstawiono procedurę wspomaganie zajęć dydaktycznych w zakresie projektów inżynierskich realizowanych w formach kształcenia tradycyjnego oraz zdalnego, ze szczególnym uwzględnieniem zagadnień projektowania zespołowego. Zaproponowano połączenie metody prezentacji rozgłoszeniowej pulpitu nauczyciela z wydziałową platformą edukacyjną. Szczególną uwagę zwrócono na kursy komplementarne, w których część zajęć dotąd realizowanych w salach i laboratoriach dydaktycznych przy bezpośrednim udziale nauczyciela akademickiego, może być wydzielona i prowadzona w formie sesji zdalnych. Zilustrowano możliwości dostosowywania komponentów platformy edukacyjnej w kontekście planowanych interakcji, nie tylko w relacjach nauczyciel-student, ale również student-student. Dla ustalenia uwagi, w artykule ogólne sformułowania i sugestie wyprowadzono z przykładowego zagadnienia problemowego, tematycznie związanego z planowaniem i organizacją eksperymentu pomiarowo-obliczeniowego oraz przetwarzania rozproszonego w lokalnych sieciach komputerowych.

1. KSZTAŁCENIE PROBLEMOWE W SZKOLE WYŻSZEJ

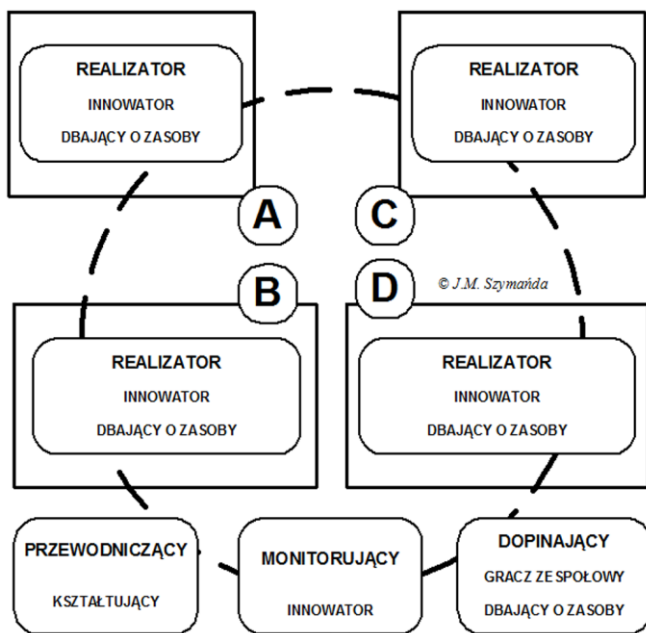
1.1. Wybrane elementy propedeutyki rozwiązywania problemów

Kształcenie problemowe i jego szerokie spektrum kształcenia wariantywnego jest przedmiotem wielu publikacji [1,2,8]. W zależności od treści i form realizacji oraz kategorii problemów, można w nich wyróżnić następujące istotne procesy, na przykład takie jak: wytwarzanie sytuacji problemowej (zadania projektowe), formułowanie celów szczegółowych (*podproblemy* i zdania), określanie warunków koniecznych i wystarczających rozwiązania, zgłaszanie propozycji rozwiązań (argumenty uzasadniające), weryfikacja zgłoszonych propozycji (kontrargumenty) oraz ocena uzyskanych rozwiązań w aspektach praktycznych i teoretycznych. Cechą charakterystyczną tej metody jest samokształcenie, a w połączeniu z pracą w zespole projektowym, pozytywna rywalizacja. W większości przypadków tworzy się zespoły (grupy) wirtualne do wykonania konkretnego zadania dydaktycznego, a po jego wykonaniu grupy ulegają rozwiązaniu. W praktyce, w zależności od tematu zadania, zespół

projektowy powinien liczyć od 5 do 7 studentów [2,6,8]. W celu zapewnienia odpowiedzialnej koordynacji działań podczas realizacji zadania, każdemu studentowi przydzielane są role (funkcje) – odpowiednio do indywidualnych predyspozycji i zainteresowań. Kluczowe dla nauczyciela jest właściwe rozpoznanie możliwości grupy. Z prac Meredith Belbina [1], twórcy koncepcji opartej na samoocenie, wynika, że specyfika poszczególnych ról zespołowych wymaga zawsze akceptacji przez wszystkich członków zespołu projektowego. Narzucanie obligatoryjnych ról, w większości przypadków nie prowadzi do oczekiwanego rozwiązania problemu (np: dla badanych 100 zespołów, 3/4 potwierdziło taką implikację). Poszczególne kategorie ról, w najbardziej ogólnym ujęciu, mogą być następujące [2,8,9]:

- przewodniczący (wyjaśnia cele i priorytety, motywuje kolegów)
- kształtujący (stawia wyzwania, wywiera naciski, poszukuje alternatywnych rozwiązań)
- realizator (przekształca pomysły i plany w działanie)
- innowator (tworzy oryginalne koncepcje, rozwiązuje trudne problemy)
- dbający o zasoby (rozpatruje nowe możliwości, nawiązuje kontakty, negocjuje)
- monitorujący/oceniający (dostrzega wszystkie opcje, ocenia prawdopodobne wyniki)
- gracz zespołowy (zapobiega tarciom, radzi sobie z konfliktami w zespole)
- dopinający (wyszukuje błędy i słabe miejsca rozwiązania, dopilnowuje terminów itp.)

Przykładowy model organizacji pracy zespołu projektowego oraz przypisywania kompetencji przedstawiono na rysunku 1. Dodatkowym walorem kształcenia problemowego jest wymagana i nieunikniona ciągła interakcja nie tylko między nauczycielem i zespołem projektowym, ale również wśród wszystkich uczestników procesu projektowania i to na każdym jego etapie. Efektywność samokształcenia może być zwiększana poprzez odpowiednio dobrane elementy z innych metod, takich jak: metoda przypadków (*case study*), giełda pomysłów (*brainstorming*), czy gry symulacyjne (*simulation game, mixed game*) [1,2,8].



Rys. 1. Scenariusz selektywnych podziałów i przypisywania kompetencji (ról) w zespole projektowym

1.2. Dydaktyczne aspekty projektów inżynierskich i obliczeń numerycznych

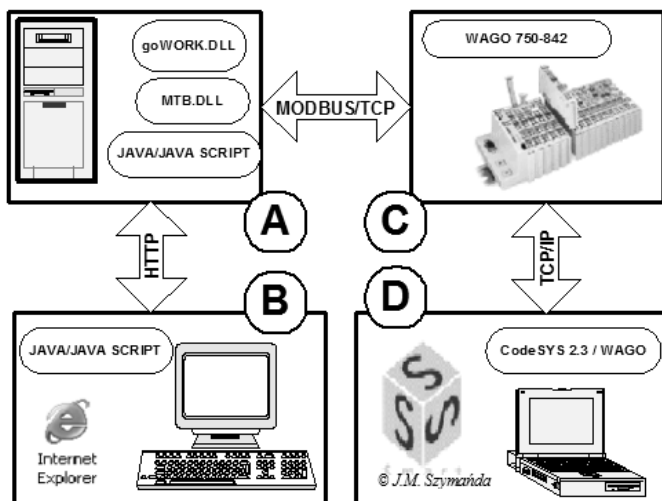
Współczesna technologia informatyczna (IT) oraz dostępne techniki programowania z jednej strony umożliwiają na bardzo kreatywne i sugestywne rozwiązywanie problemów, ale z drugiej ograniczają możliwość oceny wykorzystywanych algorytmów i zagrożeń w obliczeniach numerycznych. Jako przykład mogą być wymienione bardzo zaawansowane systemy typu CAD – czyli systemy wspomagające kolejne etapy projektowania, symulacji i eksperymentów numerycznych. Wysoka jakość większości z tych produktów wyznacza również uznawane standardy wymiany informacji oraz odniesień referencyjnych. Niestety w pewnych sytuacjach ich stosowanie może prowadzić do poważnych błędów metodologicznych i merytorycznych. Poziom zagrożenia jest tym większy im mniejsze jest rozpoznanie i wiedza w danym zakresie. Czasami nawet w opracowaniach naukowych można dostrzec bezkrytyczne stosowanie pakietów obliczeniowych, efektem czego błędy obliczeniowe wynikające z arytmetyki liczb zmiennopozycyjnych o skończonej rozdzielczości (epsilon maszynowy) stają się przedmiotem wnioskowania merytorycznego o występujących korelacjach i obserwacjach eksperymentu symulacyjnego (!). Nawet najlepsze systemy, pakiety komercyjne i programy obliczeniowe nie zwalniają od istotnego rozpoznania zagadnienia w ramach którego zmierzamy do rozwiązania zadanego problemu. Podobny kontekst można także zaobserwować podczas opracowywania własnych programów obliczeniowych, w których poszczególne etapy kodowania algorytmu są lub nie są (!) oceniane pod względem stabilności obliczeń numerycznych. Często odwołujemy się do zewnętrznych bibliotek ułatwiających programowanie np. z zakresu algebry (rozwiązywanie układów równań, poszukiwanie wartości własnych, itp.), rachunku różniczkowo-całkowego, statystyki czy ukierunkowanych zagadnień technicznych jak optymalizacja, przetwarzanie sygnałów i wielu innych. Warto tutaj podkreślić, iż profesjonalnie opracowane oraz dostępne na rynku produkty z tego zakresu (np. biblioteki numerycznej NAG [4]) są bardzo cenne i korzystanie z nich jest wysoce zalecane. Zawsze jednak po uprzednim dokładnym zapoznaniu się z odpowiednią dokumentacją, która powinna stanowić nieodłączny element zestawów dystrybucyjnych. Bez rozpoznania zastosowanych w procedurach algorytmów obliczeniowych i właściwej interpretacji parametrów wejściowo-wyjściowych włączenie ich

do opracowywanego programu a często nawet do całego projektu, może być całkowicie błędnym podejściem. Należy także zauważyć, że prawidłowe działanie procedury i wykonane obliczenia nie oznaczają z *definicji* prawidłowego rozwiązania postawionego przez nas problemu. Z teorii algorytmów wynika, że algorytm numerycznie stabilny wcale nie musi być algorytmem numerycznie poprawnym, innymi słowy, nasze rozwiązanie może być powtarzalne podczas każdej symulacji, ale powtarzane ze skorelowanym błędem numerycznym (!). Jest to sytuacja o największym stopniu zagrożenia, ponieważ bez świadomej analizy może doprowadzić do fałszywych wniosków i ocen. Nie można oczekiwać prawidłowego przybliżonego rozwiązania danego zagadnienia jeśli znany jest tylko jeden algorytm postępowania. W związku z powyższym, jeżeli jest to tylko możliwe, weryfikację obliczeń należy zawsze przeprowadzać wykorzystując inny *niezależny* algorytm. Z oceną niezależności algorytmów i zagadnieniami pokrewnymi można zapoznać się między innymi we wskazanej bibliografii [3,5]. Znacznie „bezpieczniejsze” błędy to błędy krytyczne, które pojawiają się wtedy kiedy w pierwszej fazie programowania wprowadzony algorytm będzie algorytmem numerycznie niestabilnym i podczas jego uruchomienia zostanie zgłoszony wyjątek z zatrzymaniem programu włącznie. Tak wyraźnie zaznaczone zdarzenia oczywiście nie zafałszują w tle przeprowadzanych obliczeń i symulacji. Podsumowując, rozpoczynając naukę w kontekście implementacji algorytmów obliczeniowych preferowane jest zatem programowanie podstawowe, ze szczególnym uwzględnieniem weryfikacji algorytmów pod względem ich stabilności numerycznych. Nabywane w tym zakresie doświadczenia powinny pomóc studentom w podejmowaniu decyzji projektowych i we właściwej interpretacji otrzymywanych wyników.

1.3. Przykładowa realizacja projektu inżynierskiego w zespole projektowym

W celu przybliżenia zakresu tematycznego projektów inżynierskich w artykule przedstawiono przykładowy scenariusz uwzględniający podział na *podproblemy* i przypisujący kompetencje członkom zespołu projektowego (wykonawcom). Założenia merytoryczne projektu określały warunki opracowania pakietu aplikacyjno-sprzętowego *INTERBUS* obejmującego między innymi: program *HTTPSerwer*, biblioteki dynamiczne *goWORK* i *MBT* oraz swobodnie programowalne sterowniki *PLC*. Aplikacja *HTTPSerwer* oraz biblioteka *goWORK* powinna zostać opracowana przy wykorzystaniu dowolnego środowiska programistycznego jako moduły działające samodzielnie pod kontrolą 32 bitowej wersji systemu operacyjnego *Microsoft Windows*. Otwarta architektura pakietu miała uwzględniać model komunikacji *klient-serwer*, przy czym *klient* mógł być niededykowaną aplikacją tzw. „cienkiego klienta”. Do sesji komunikacyjnych końcowy użytkownik systemu *INTERBUS* mógł zastosować także każde oprogramowanie zaliczane do tzw. przeglądarek internetowych; zainstalowanych na komputerach z dowolnym systemem operacyjnym. To rozwiązanie jest cenne, szczególnie podczas korzystania z ogólnodostępnych laboratoriów komputerowych; nie wymaga się od użytkownika, jakichkolwiek działań instalacyjnych na komputerze lokalnym. Opcjonalnie w projekcie można było założyć możliwość integracji systemu z wybranymi kursami realizowanymi w ramach internetowej platformy edukacyjnej Wydziału Elektrycznego Politechniki Wrocławskiej (<http://eportal.eny.pwr.edu.pl>). Poglądowy schemat funkcjonalny opracowanego systemu przedstawiono na rysunku 2. W szczególności wyróżnione zostały cztery główne grupy pakietu *INTERBUS*: A- aplikacja serwera komunikacyjno-interpretującego, B- interfejs komunikacyjny przeznaczony do ciągłej obserwacji i sterowania, C- komponenty swobodnie programowalnego sterownika *PLC WAGO 750-842* firmy *WAGO Corporation*, D- aplikacje producenta *WAGO*

przeznaczone do bezpośrednich (sprzętowych) konfiguracji sterownika. Podstawowym protokołem komunikacyjnym warstwy aplikacyjnej obejmującej przepływ danych pomiędzy modułami A-B jest protokół *HTTP*. Umożliwia on bezpośrednie wykorzystanie standardowych programów użytkownika mającego dostęp do Internetu. W zakresie komunikacji pomiędzy modułami A-C, zastosowano enkapsulację protokołów *MODBUS/TCP*. Specyfikacja zależności między protokołami *MODBUS/TCP*, *MODBUS* i sterownikami *PLC* została określona w dokumentacji firmy *WAGO* [10]. Niezbędną podstawową konfiguracją m.in. w zakresie określania prawidłowych parametrów komunikacji sieciowej można przeprowadzać poprzez albo interfejs szeregowy *RS232C* albo połączenie *TCP/IP* (moduły C-D). W projekcie jedno z ważniejszych założeń dotyczyło aspektu dydaktycznego w zakresie szybkiego projektowania algorytmów sterowania i diagnozowania symulowanych zagadnień technologicznych. W szczególności umożliwienie zespołom projektowym realizującym zadania podczas sesji wirtualnych na dostęp do urządzeń laboratoryjnych sterowanych poprzez programowalne sterowniki *PLC*.



Rys. 2. Przykładowe moduły funkcjonalne projektu INTERBUS

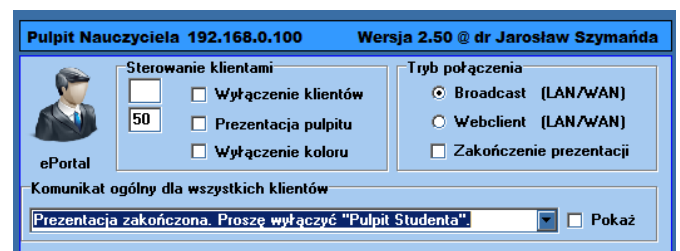
Wszystkie złożone moduły projektu zostały podzielone i przypisane do wykonania podgrupom laboratoryjnym wraz z określeniem indywidualnych kompetencji – pełniących ról w zespole projektowym. Przeprowadzona ocena realizacji projektu potwierdziła poprawność metodologiczną zajęć w zaproponowanej formie. Odpowiednie rozpoznanie zespołu projektowego oraz nieprzypadkowy rozdział kompetencji wyraźnie poprawiają zdolność nabywania umiejętności naukowo umotywowanej diagnozy problemu. Z uogólnieniem i podsumowaniem doświadczeń z tego zakresu można zapoznać się w opracowaniu [2,7,8]

2. WSPOMAGANIE ZAJĘĆ DYDAKTYCZNYCH

2.1. Pulpit nauczyciela

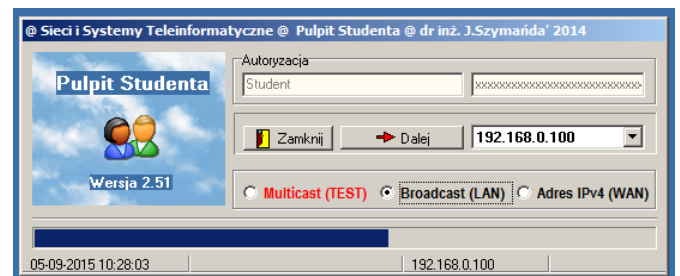
Duża różnorodność zagadnień realizowanych w zakresie zajęć projektowych oraz laboratoryjnych realizowanych na Wydziale Elektrycznym Politechniki Wrocławskiej, przyczyniła się do opracowania zintegrowanego systemu komunikacyjno-prezentacyjnego w relacjach nauczyciel–studenci oraz studenci–studenci. Założono implementację dwóch trybów komunikacyjnych: bezpośredniego (*online*) oraz ogłoszeniowego (*offline*). Tryb bezpośredni to przede wszystkim możliwość interaktywnej wymiany informacji nauczyciel–studenci podczas zajęć w laboratoriach. Podstawowym elementem tego trybu jest *Pulpit nauczyciela* będący aplikacją rozgłoszeniową (*broadcast*) uruchamianą w 32/64 bitowych wersjach systemu ope-

racyjnego *MS Windows* (w wersji *XP* i nowszy). Aplikacja ta udostępnia w lokalnych sieciach komputerowych (*LAN*), między innymi, zawartość konsoli/pulpitu graficznego komputera nauczyciela, informacje tekstowe (komunikaty), polecenia systemowe oraz dedykowane sygnały sterujące wybranymi aplikacjami użytkowników. Studenci indywidualnie lub w grupach projektowych przy swoich stanowiskach laboratoryjnych, komunikują się z nauczycielem poprzez oprogramowanie klienckie (do 30 stanowisk). W systemie zaprojektowano możliwość wykorzystywania albo aplikacji dedykowanej z rozbudowanym interfejsem komunikacji dwustronnej nauczyciel–student–nauczyciel, albo tak zwanego „cienkiego klienta” w ramach zainstalowanych w danym systemie operacyjnym komputerów *przeglądarek internetowych*. Każda z wymienionych metod podłączania klienta ma swoje wady i zalety. W pierwszym przypadku program klienta zaprojektowany jest wyłącznie do uruchamiania w systemie operacyjnym *MS Windows* (w wersji *XP* lub nowszym), umożliwia jednakże na bardziej rozbudowaną komunikację dwustronną oraz współpracę z dwoma monitorami. Drugie rozwiązanie przygotowane jest przede wszystkim do prezentacji graficznej konsoli nauczyciela z ograniczonymi możliwościami sterowania zwrótnego za to z możliwością wykorzystania na dowolnych platformach graficznych systemów operacyjnych (*MS Windows*, *Linuks*, *Apple*, *Android*). Przykładowe interfejsy *Pulpitu nauczyciela* oraz aplikacji klienckich przedstawiono odpowiednio na rysunkach 3-7.

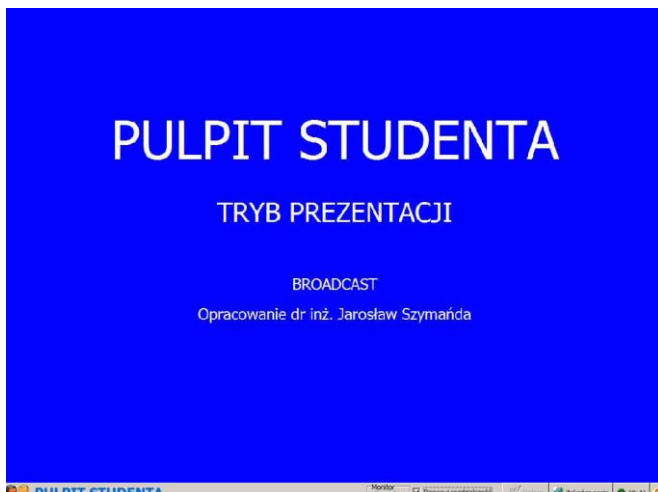


Rys. 3. Aplikacja wspomagająca prowadzenie zajęć dydaktycznych m.in. w zakresie publikacji ekranu graficznego nauczyciela w czasie rzeczywistym w lokalnej sieci komputerowej

Udostępnianie wybranych zasobów komputera prezentacyjnego nauczyciela, w znacznym stopniu ułatwia prowadzenie zajęć projektowych oraz laboratoryjnych między innymi w zakresie omawiania przedmiotowych aplikacji wykorzystywanych podczas projektowania (*CADs*), wykonywania obliczeń (*Matlab*, *SciLab*, *Statistica*, itp.), programowania (*Visual C++*, *Delphi*, *Fortran*, itp.).



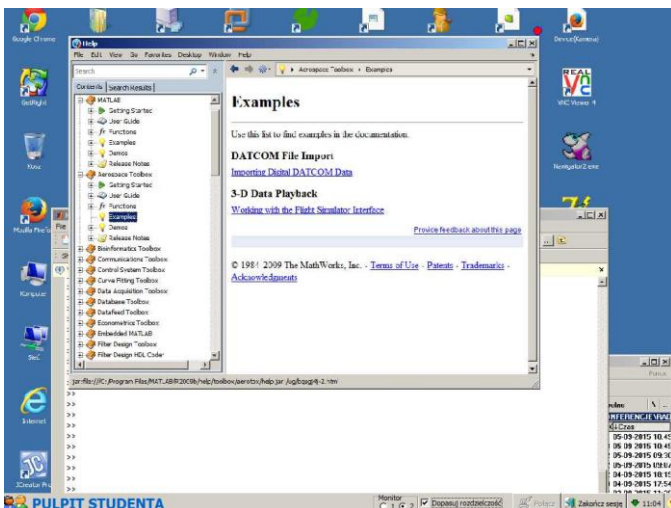
Rys. 4. Interfejs podłączania (rejestracji) aplikacji tzw. „grubego klienta” umożliwiającej podgląd ekranu graficznego komputera nauczyciela oraz na dwustronną komunikację w lokalnych sieciach komputerowych



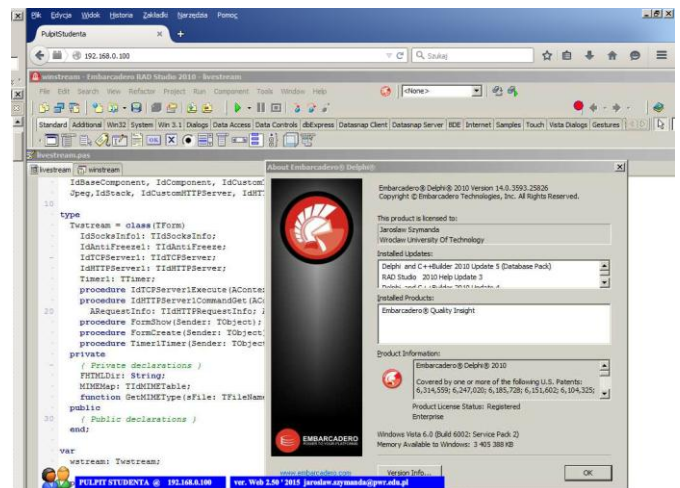
Rys. 5. Ekran informacyjny aplikacji Pulpitu studenta („gruby klient”) po pozytywnej autoryzacji dostępu do Pulpitu nauczyciela

2.2. Platforma kształcenia na odległość (e-learning i m-learning)

Platforma kształcenia na odległość (e-learning) znakomicie uzupełnia zestaw narzędzi wspomagających realizację zajęć dydaktycznych. Wykorzystywana jest przede wszystkim w zakresie trybu komunikacji ogłoszeniowej umożliwiającej studentom na dostęp do materiałów dydaktycznych oraz konspektów prowadzących zajęcia. Istotną jednakże funkcję pełni także w kontekście szeroko rozumianego samokształcenia poprzez różnorodne moduły quizów, lekcji i warsztatów. Szczegółowe odniesienie się do rozwiązań i doświadczeń autora w zakresie planowania oraz opracowywania scenariuszy kursów komplementarnych (m-learning) można odszukać m.in. w artykułach [6,7,8,9].



Rys. 6. Ekran aplikacji Pulpitu studenta („gruby klient”) po udostępnieniu przez prowadzącego zajęcia „swojego” ekranu graficznego komputera poprzez aplikację Pulpit nauczyciela



Rys. 7. Ekran aplikacji Pulpitu studenta („cienki klient” – tutaj w oknie przeglądarki internetowej Firefox) po udostępnieniu przez prowadzącego zajęcia „swojego” ekranu graficznego komputera poprzez aplikację Pulpit nauczyciela. Przykładowy ekran wykorzystywany podczas zajęć z programowania w środowisku Embarcadero RAD Studio 2010

WNIOSKI

Zaprezentowany scenariusz zajęć dydaktycznych i zaproponowane systemowe wspomaganie zajęć projektowo-laboratoryjnych, wprowadzono do realizacji od roku akademickiego 2012/2013 w ramach kursu „Metody numeryczne w technice” oraz grupy kursów z zakresu systemów identyfikacji i akwizycji danych dla studentów studiów stacjonarnych na kierunkach „Elektrotechnika” oraz „Automatyka i robotyka”. Studenci podczas zajęć z projektowania w 3-4 osobowych grupach, z odpowiednio określonymi zadaniami i rozdziałem kompetencji, byli oceniani w kontekście założonych w „Karcie przedmiotu” tzw. „Przedmiotowych efektów kształcenia”, w tym między innymi efektu: „PEK_W02 - jest w stanie zaproponować odpowiedni algorytm numeryczny do rozwiązania zadania inżynierskiego”. Przeprowadzona po zakończeniu danego semestru analiza wszystkich efektów kształcenia potwierdziła istotną poprawę nabywania przez studentów umiejętności umotywowanej diagnozy problemu w odniesieniu do zajęć realizowanych bez wymienionych innowacji.

BIBLIOGRAFIA

1. Belbin R.M.: Management team: why they succeed or fail: London: Heimann 1981.
2. Denek K.: Pomiar efektywności kształcenia w szkole wyższej: Warszawa: PWN 1980
3. Góralski A.: Algorytmy i programy statystyki jakościowej: WNT: Warszawa '79
4. NAG - Numerical Algorithms Group (<http://www.nag.co.uk/>) Hyperlink z dnia: 08.09.2015
5. Sysło M. M., Deo N, Lowalik S.J. :Algorytmy optymalizacji dyskretnej: PWN '93
6. Szymańda J.M.: Wyznaczniki kształcenia problemowego inżynierów elektryków: XXXVI Międzynarodowa Konferencja z Podstaw Elektrotechniki i Teorii Obwodów, IC-SPETO2013, Gliwice-Ustroń, 22-25.05.2013. [Gliwice : Politechnika Śląska, 2013], s.137-138
7. Szymańda J.M.: Didactic aspects of using virtual laboratories in blended learning: 2012 11th International Conference on Environment and Electrical Engineering: Venice, Italy, 18-25 May '12

8. Szymańda J.M.: Problemy kształcenia z wykorzystaniem nowych mediów, Monografia ,Rozdział 12, 2011 s.113-121. Oficyna Wydawnicza Politechniki Wrocławskiej, ISBN 978-83-7493-720-7
9. Szymańda J.M., Rezmer J.: Platforma zdalnej edukacji „Moodle” – projektowanie kursu komplementarnego, Nowe Media w Edukacji 2008 - Zastosowanie technik informacyjnych i komunikacyjnych, Oficyna Wydawnicza Politechniki Wrocławskiej: Wrocław 2008: s.131-138, ISBN 978-83-7493-404-6
10. WAGO Sp.z o.o.: Modular I/O System ETHERNET TCP/IP, 750-342, 750-842. Manual, nr 750-129/000-002 Version 2.0.0.

SYSTEM-AIDED OF DESIGN OF LABORATORY ACTIVITIES IN THE FIELD OF EDUCATION PROBLEMATIC ENGINEERS

Abstract

The article presents selected elements of the support of engineering problems carried out in the project teams. Particular attention was paid to the elements of the problem of education and teaching methods in order to develop scenarios and assign appropriate roles to the teams. Desktop software package presents the teacher and the student panel presentation capabilities to support the teacher, inter alia, the discussion of these applications used during the design, construction and programming of numerical calculations. It was found that the synergy of education problem, group problem solving and duly authorized individual competence is the best basis for the development of courses, using modern educational methods and technologies.

Autor:

Szymańda Jarosław M. - Politechnika Wroclawska, Wydział Elektryczny, 50-370 Wrocław, Wybrzeże Wyspiańskiego 27, e-mail: jaroslaw.szymanda@pwr.edu.pl