

# Aplikacje RIA w systemach CMS oraz e-commerce

Michał Łoza\*, Piotr A. Kowalski\*\*

\* Koło Naukowe IT, Katedra Automatyki i Technik Informatycznych, Politechnika Krakowska

\*\* Instytut Badań Systemowych, Polska Akademia Nauk

**Streszczenie:** Aplikacje RIA zyskują popularność w szerokim świecie Internetu. Ich główną zaletą jest bardzo duży stopień przystępności dla użytkownika, przede wszystkim ze względu na eliminację konieczności „przeładunku” strony po dokonaniu pewnych jej modyfikacji. RIA jest idealną technologią do współpracy z środowiskiem CMS. System CMS umożliwia zarządzanie stronami w bardzo łatwy sposób, nawet dla osób z relatywnie małą znajomością języka HTML. Technologia RIA jest wymarzoną rozwiązaniem dla właścicieli małych firm, które nie mogą sobie pozwolić na zatrudnienie dodatkowej osoby tylko do zarządzania stroną internetową. Skomplikowane systemy, takie jak CMS, stają się nawet prostsze niż pisanie strony w zwykłym – klasycznym kodzie HTML. Do głównych celów realizowanych w powyższym systemie można zaliczyć uproszczenie wszystkich skomplikowanych operacji oraz zapewnienie przyjaznego dla użytkownika interfejsu, który „poprowadzi” za rękę użytkowników. System CMS musi być bardzo elastyczny, aby dostosować się do użytkowników. Innym bardzo ważnym aspektem jest bieżące utrzymanie systemu, gdyż każdy błąd zagrażający bezpieczeństwu danych musi zostać wyeliminowany. Dla środowiska rozwijającego oprogramowanie ważnym wątkiem jest również rejestracja wszystkich błędów, tak aby w przyszłości móc go odtworzyć w zależności od kierunku rozwoju oprogramowania.

**Słowa kluczowe:** RIA, CMS, e-commerce, aplikacje internetowe, aplikacje webowe, PHP, JavaScript

Rich Internet Application (RIA) [1] w wolnym tłumaczeniu oznacza bogatą aplikację internetową. Pierwotnie określenie to stosowane było do stron wykonanych w technologii Flash, które zmieniały zawartość bez przeładowania całej strony. Aktualnie określenia tego używa się nie tylko do stron wykonanych w technologii Flash, ale również do tych, na których cała komunikacja z serwerem odbywa się w tle. Po załadowaniu całej strony przesyłane są tylko informacje o tym, co ma zostać zmienione. Dzięki temu ilość transferu wykorzystywanego przez aplikację drastycznie spada, a więc zmniejszają się koszty utrzymania aplikacji na serwerach. Zmniejsza się także obciążenie serwera aplikacji, który musi wygenerować zmienioną treść, ale nie całą stronę. W takim wypadku często nie wykonuje się dodatkowych zapytań do bazy danych. Oprócz wymienionych korzyści po stronie zarządcy aplikacji, sporo zyskuje także użytkownik. Nie musi on za każdym razem czekać na załadowanie się strony, tylko na

jeden element. Załadowane wcześniej arkusze stylów CSS oraz pliki ze skryptami JavaScript (JS) nie są pobierane za każdym razem. Często na czas wykonania jakiejś akcji treść jest ukrywana, a po wykonaniu akcji jest ona zamieniana z aktualnie wyświetlaną (a nie ładowana od nowa). Na przykład w przeglądarkowym kliencie pocztowym Gmail, który jest bardzo dobrym przykładem aplikacji RIA, po włączeniu podglądu wiadomości skrzynka odbiorcza jest ukrywana. Jeżeli chcemy potem powrócić do listy wiadomości, nie jest ona ładowana jeszcze raz, a jedynie zamieniana z aktualnie wyświetlaną wiadomością. Wraz z pojawieniem się standardu HTML5 otrzymaliśmy całą gamę technologii wspierających. Wcześniej do dyspozycji programisty była technologia AJAX (ang. *Asynchronous JavaScript And XML*), dzięki której można było wykonywać asynchroniczne odwołania do serwera aplikacji, podczas gdy pozostała część aplikacji pozostawała responsywna dla użytkownika. HTML5 wprowadza m.in. WebSockets, które umożliwiają przesyłanie danych z serwera bez wcześniejszego wysłania zapytania przez przeglądarkę, czy WebStorage, dzięki któremu otrzymujemy co najmniej 5 MB przestrzeni do przechowywania danych po stronie użytkownika, przy czym dokładna ilość zależy od zaimplementowanej technologii w przeglądarce.

## 1. Systemy CMS

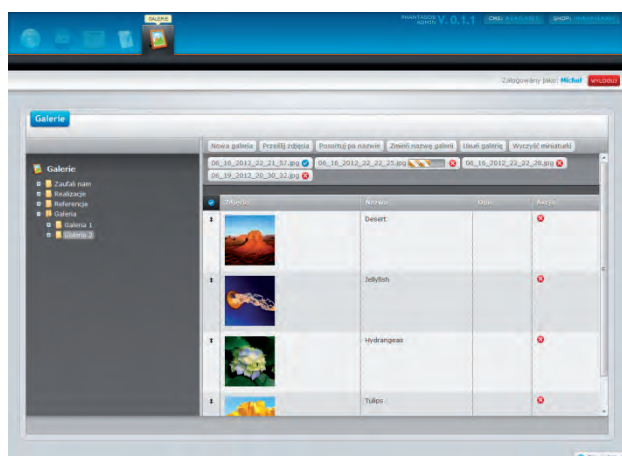
Systemy CMS są aplikacjami internetowymi, które umożliwiają zarządzanie stroną WWW bez potrzeby znajomości języka HTML. Są przygotowane tak, aby użytkownik mógł tylko wprowadzić odpowiednie treści w pola danych, z których następnie zostanie wygenerowana zawartość strony WWW. Liczba elementów zarządzanych jest zależna od wybranego systemu.

Głównym elementem w systemach CMS jest edytor tekstu. Jest to edytor typu WYSIWYG (ang. *What You See Is What You Get*). Dostępna jest duża liczba edytorów typu open source, która może być wykorzystana przez systemy CMS. Najbardziej dopracowanym i zarazem najczęściej używanym jest edytor TinyMCE. Swoim wyglądem i zachowaniem stara się on naśladować wcześniejsze wersje edytora tekstu Microsoft Word (przed wersją 2007) lub OpenOffice Writer.

Najważniejszym wymogiem jest prostota użytkownika. Należy wziąć pod uwagę, że takie systemy często są obsługiwane przez osoby niemające nic wspólnego z branżą IT. Dużą grupą docelową są właściciele małych i średnich przedsiębiorstw, którzy nie mogą sobie pozwolić na

utrzymywanie dodatkowego etatu do obsługi firmowej strony WWW. Aktualnie, w dobie Internetu, trudno znaleźć przedsiębiorstwo, które nie posiada firmowej strony internetowej. Jednak nie można wymagać od właścicielki salonu kosmetycznego biegłej znajomości HTML, CSS i JavaScript. Głównie z myślą o takich osobach tworzone są systemy CMS.

Cytując Charlesa Mingusa, „tworzenie skomplikowanym rzeczy prostych jest codziennością, tworzenie rzeczy skomplikowanych prostymi, fantastycznie prostymi, jest kreatywnością”. Dlatego twórcy systemów CMS muszą wykazać się ogromną inwencją, tak aby użytkownicy ich systemów mogli zarządzać jak największą liczbą elementów na stronie WWW w sposób jak najprostszy, nie będąc zarazem przytłoczonymi ogromem możliwości i czasochłonną konfiguracją wszystkiego.



Rys. 1. Intuicyjny interfejs zarządzania stroną  
Fig. 1. The intuitive interface site management

## 2. Sposoby uproszczenia interfejsu użytkownika

Najprostszym sposobem uniknięcia przytłoczenia użytkownika jest ukrycie wszystkich mało istotnych opcji w osobnej zakładce. Dzięki temu użytkownicy, którzy nie korzystają z tych dodatkowych opcji, będą mieli uproszczony interfejs, a ci którzy potrzebują tych opcji, będą mogli przejść do nich bez większych problemów. Często spotykanym, podobnym rozwiązaniem jest rozdzielanie dużej akcji na kilka mniejszych kroków, poprzez grupowanie powiązanych ze sobą opcji tematycznych. Najczęściej rozwiązanie to spotyka się we frontendowej części aplikacji służącej do realizacji zamówienia. Całą czynność można podzielić np. na następujące kroki:

- podawanie danych użytkownika (ewentualnie rejestracja użytkownika),
- informacje związane ze sposobem przesyłki,
- informacje związane z płatnościami,
- podsumowanie zamówienia i sprawdzenie danych.

Dzięki takiemu rozczłonkowaniu formularza użytkownik nie widzi na początku kilkudziesięciu pól, które trzeba wypełnić, i w konsekwencji nie porzuca realizacji zamówienia przez ilość danych wymaganych do przekazania. Zamiast tego jest prowadzony przez kolejne kroki,

a w każdym z nich musi podać tylko kilka danych związanych z jedną grupą działań. Oczywiście nie należy popadać w skrajność i rozdzielać czynności na zbyt wiele kroków, w każdym prezentując na przykład tylko jedno pytanie. Działając w ten sposób, otrzymamy interfejs zawierający kilkadziesiąt kroków, co przyniesie rezultat jeszcze gorszy niż pozostawienie wszystkiego na jednej stronie.

Kolejnym sposobem umoszczenia interfejsu jest usunięcie mniej ważnych opcji i pozwolenie, aby to system decydował, co dla użytkownika w danym momencie będzie najlepsze. W tym przypadku jednak należy wziąć pod uwagę, iż pomimo najlepszych algorytmów system nie zawsze może dokonywać najlepszych wyborów. Innym problemem jest sam użytkownik, który mimo że system zadecydował dobrze, będzie chciał ustawić jakąś opcję według swoich indywidualnych preferencji. Takie sytuacje nie należą do rzadkości, dlatego warto się zabezpieczyć przed taką ewentualnością, umieszczając możliwość zmiany ustawień proponowanych przez system. Może to być dobrze ukryte miejsce, tak aby osoby zadowolone z działania systemu nie trafiły tam przypadkiem i nie ustawiły czegoś, czego tak naprawdę nie chcą.

Najlepszym, aczkolwiek najtrudniejszym w realizacji rozwiązaniem jest stworzenie interfejsu, który uprości wszystkie działania administracyjne w systemie.

Dla przykładu można wskazać opcję wybierania kolejności elementów w menu, którą często realizuje się poprzez ręczne wpisanie wartości liczbowej w pole danych. Jeżeli trzeba przestawić kilkanaście elementów lub zamienić kilka miejscami, celowym jest przejść do edycji elementu, zmienić wartości pola i zapisać powyższe. Następnie wybrać kolejne elementy i powtarzać czynność. Inną metodą realizacji tego zadania jest możliwość skorzystania z pomocy technologii drag'n'drop, pozwalającej użytkownikowi przeciągnąć element w odpowiednie miejsce za pomocą myszki, a następnie obliczyć pozycję dla wszystkich elementów i zapisać do bazy danych. Jest to znaczne uproszczenie w stosunku do ręcznego wpisywania pozycji, w szczególności jeśli chcemy wstawić nowy element w środek istniejącego zestawu i trzeba zmienić pozycje wszystkich elementów następujących po wstawianym elemencie. Zwykle realizowane jest to za pomocą JavaScriptu, lecz wraz z nadejściem HTML5 [2], który dostarcza natywne API, można zrealizować wszystkie operacje dużo prościej.

Innym przykładem jest ładowanie zdjęć. W klasycznym podejściu stosujemy odpowiednie pole formularza, w którym po kliknięciu otwiera się okno wyboru zdjęcia. Następnie wybieramy zdjęcie, w polu zapisuje się ścieżka do zdjęcia. Po kliknięciu przycisku zatwierdzającego, przesyłana jest zawartość formularza, w tym zdjęcie. Przeglądarki – wyjątkiem jest Google Chrome – nie informują o postępie w wysyłaniu. Zastosowanie kilku takich pól będzie wymagało wybrania zdjęcia, osobno dla każdego z pól, czego skutkiem będzie wydłużony czas potrzebny na przesłanie całości na serwer. W tym przypadku możemy zastosować co najmniej dwa rozwiązania. Pierwszym z nich, obecnie bardzo popularnym, jest stosowanie wysyłania plików za pomocą wtyczki Adobe Flash. Dostępna jest ona w większości przeglądarek, więc możemy bez

obaw stosować takie podejście. Pozwala ona na zaznaczenie wielu plików do przesłania na raz, bez potrzeby wybierania każdego z osobna. Ważną cechą jest interakcja ze skryptami JS. Wtyczka wywołuje odpowiednie zdarzenia po dodaniu pliku, przesłaniu pliku, zmianie statusu. W tym przypadku możemy uzyskać informacje między innymi o postępie przesyłania – przykładowo dla stworzenia paska postępu – prędkości wysyłania, ewentualnych błędach w transferze.

Drugim rozwiązaniem, wspomnianym wcześniej, jest użycie HTML5 File API [3]. Dostarcza ono podobnych możliwości, jednak nie wymaga użycia wtyczki Adobe Flash. Ciekawą możliwością tego rozwiązania jest przeciągnięcie plików na odpowiednie pola, a po upuszczeniu przesłanie plików na serwer. Niestety, metoda ta będzie działać tylko z najnowszymi przeglądarkami, które wspierają standard HTML5. Aktualnie w pełni HTML5 File API obsługują tylko Google Chrome i Firefox. Przeglądarki Safari, Android Mobile 3.0 i Opera oferują tylko częściowe wsparcie, zaś Opera Mini, iOS Safari, Android 2.3 i Internet Explorer 9 nie zapewniają go wcale [4].

### 3. Elastyczność systemów CMS i e-commerce

Nieważne, jak dobry i przemyślany będzie system oraz jak wiele będzie posiadał funkcji – zawsze znajdą się osoby, które będą potrzebowały dodatkowych możliwości. Z drugiej strony, jeżeli będziemy chcieli zaimplementować wszystkie te funkcje, nawet te których potrzebować będą pojedynczy użytkownicy, okaże się, że nasz system jest zasobochłonny i powolny, oferujący zbyt wiele możliwości, w których użytkownicy się zgubią. W dodatku tylko wąskie grono korzysta z wielu funkcji, które reszcie użytkowników nie tylko nie są potrzebne, ale wręcz im przeszkadzają. Co można zrobić w takim wypadku? Bardzo często uważa się, że doskonałym sposobem rozwiązania tego problemu jest implementacja mechanizmu wtyczek. Pozwala ona na rozszerzanie możliwości systemu o nowe opcje tylko tym użytkownikom, którzy tego potrzebują. Każdy użytkownik będzie mógł spersonalizować swój system według osobistych potrzeb, nie obciążając przy tym pozostałych kopii systemu używanych przez innych. Dodatkowo, stworzenie nowej funkcjonalności na życzenie grupy użytkowników będzie dużo szybsze i prostsze.

### 4. Aktualizacje

Nie trzeba specjalnie mówić o potrzebie aktualizowania oprogramowania na komputerze. Te same zasady dotyczą aplikacji internetowych. Są one narażone na atak bardziej niż komputery osobiste, ponieważ dostępne są bezpośrednio z Internetu, np. na ataki typu SQL injection, gdzie braki w filtracji danych pochodzących od użytkowników mogą spowodować wstrzyknięcie złośliwego kodu do zapytania do bazy danych. Możemy przez to stracić całą zawartość bazy lub ktoś może zalogować się do panelu administracyjnego bez znajomości loginu i hasła. Innym popularnym typem ataku jest atak XSS. Polega on na

wprowadzeniu treści aktywnej do kodu atakowanej strony. Dla przykładu, jest to wprowadzenie skryptu JavaScript do komentarzy. Jeżeli kod nie zostanie odfiltrowany przez serwer, może on przekierować użytkowników na inną, niebezpieczną dla nich stronę. W innym przypadku „wstrzyknięty” kod może obejść niektóre mechanizmy kontroli dostępu do danych użytkownika. Może to być bardzo niebezpieczne dla użytkowników strony korzystającej z systemu CMS. Wszystkie wykryte luki należy bardzo szybko załatać i umożliwić użytkownikom aktualizację. Najlepiej jeżeli można przeprowadzić tę aktualizację automatycznie z poziomu panelu administratora bez potrzeby wgrzywania plików przez FTP i ręcznej aktualizacji struktury bazy danych.

W systemach opracowanych przez autora artykułu wdrażany jest system aktualizacji oparty na systemie wersjonowania kodu GIT. Pobiera on najnowsze aktualizacje z serwera GIT, w plikach zawarte są skrypty do aktualizacji struktury bazy danych. Instalacja nowej wersji oprogramowania sprowadza się do kliknięcia przycisku aktualizuj. Następnie można obserwować pasek postępu pobierania aktualizacji. Po ukończeniu pobierania użytkownik prowadzony jest przez proces aktualizacji, który zwykle sprowadza się to do kliknięcia przycisku „Dalej” (w niektórych przypadkach konieczne może być wybranie ustawień nowej funkcji, zaakceptowanie licencji lub zaakceptowanie licencji zewnętrznych aplikacji, jeśli wprowadzone zmiany zawierają możliwość komunikacji z oprogramowaniem osób trzecich).

### 5. Logowanie błędów

Mechanizm tworzenia raportów nie jest zbyt istotny z punktu widzenia użytkownika. Jednak może okazać się bezcenny dla twórcy systemu, który musi zdiagnozować błąd występujący w aplikacji. Użytkownik, który go znajduje, często nie potrafi dostarczyć wystarczających informacji do odtworzenia wystąpienia błędu. Wtedy raporty logujące błędy mogą być niezastąpionym źródłem informacji. Mechanizm powinien zapewniać również możliwość archiwizacji starych informacji o zdarzeniach (logów), aby nie zajmowały cennej przestrzeni dyskowej.

Aktualnie systemy do tworzenia tzw. logów wbudowane są we frameworki. W opisywanych – autorskich – systemach wykorzystywany jest system udostępniany przez framework, wraz z rozbudową o archiwizację starych logów. Pliki raportów są przechowywane oddzielnie dla każdego dnia, a następnie kompresowane razem z pozostałymi plikami z całego miesiąca. Starsze niż rok są kasowane.

### 6. Systemy multidomenowe

Dla specjalnych wymagań klienta został stworzony system CMS, który obsługuje wiele domen. Oznacza to, iż jedna kopia systemu obsługuje wiele domen internetowych, a użytkownik z jednego wspólnego panelu administracyjnego może zarządzać poszczególnymi domenami niezależnie od pozostałych. Główną zaletą takiego rozwiązania jest szybkość dodawania kolejnych stron internetowych. Stworzony system szablonów pozwala konfigurować głów-

ne elementy, takie jak podstawowa kolorystyka, logo, nazwa i hasło reklamowe w górnej części strony. Pozwala to na szybkie tworzenie pozornie różnych stron obsługiwanych przez jeden szablon. Szybkość wdrażania kolejnych stron umożliwia oferowanie budżetowych stron internetowych, tak zwanych internetowych wizytówek, firmom, które nie potrzebują pełnej strony internetowej, lub jako tymczasowe strony internetowe na czas tworzenia rozbudowanych stron informacyjnych. Odpowiedni system uprawnień daje możliwość podziału systemu na trzy główne kategorie uprawnień, gdzie każda kategoria miała dostęp do wydzielonej części stron obsługiwanych przez system. Najniższą możliwą grupą uprawnień były uprawnienia użytkownika, umożliwiające zarządzanie stronami przypisanymi do konta. Kolejną była kategoria menadżera, pozwalająca na zarządzanie wszystkimi stronami należącymi do klientów, którymi się opiekował oraz tworzenie nowych klientów, których strony były przypisane do menadżera tworzącego. Następną kategorią był administrator, który mógł zarządzać wszystkimi stronami, kontami menadżerów, nie mógł tylko zarządzać innymi administratorami – do tego zadania należało użyć wbudowanego konta superadministratora.

## 7. Panel administracyjny wbudowany w stronę WWW

Ostatnim bardzo ciekawym rozwiązaniem jest zintegrowanie panelu zarządzania z frontendem strony internetowej. Daje to możliwość natychmiastowego zobaczenia wszystkich zmian wprowadzonych w stronie WWW. W tradycyjnym podejściu po zapisaniu zmian w panelu administracyjnym musimy przejść do właściwej strony WWW, odświeżyć ją i dopiero wtedy można zobaczyć wprowadzone zmiany. W tym podejściu, po przesunięciu elementu od razu widoczne jest, jak to wygląda, ponieważ operacje wykonywane są na stronie WWW. Takie podejście wymaga zastosowania zaawansowanych skryptów JavaScript, które muszą być dopracowane, w przeciwnym przypadku będą generować błędy oraz wpływać destrukcyjnie na wygląd strony. Z drugiej jednak strony dobrze wykonany panel jest o wiele bardziej intuicyjny i łatwy w użyciu przez zwykłego użytkownika niż najlepszy panel wykonany tradycyjnymi metodami.



Rys. 2. Panel wbudowany w stronę  
Fig. 2. The panel built into the website

## 8. Podsumowanie

Mimo że język HTML nie jest skomplikowany, nadal występuje potrzeba istnienia i rozwijania systemów CMS, tak aby możliwe było zarządzanie stroną WWW przez niemal każdego użytkownika Internetu. Bardzo szybki rozwój technologii pozwala na tworzenie stron WWW



Rys. 3. Przykład strony www wykorzystującej technologię CMS  
Fig. 3. An example of a website using CMS technology

wyglądających i zachowujących się jak natywne programy komputerowe. Budowanie takich aplikacji w konwencji RIA czyni je jeszcze łatwiejszymi i przyjemniejszymi w obsłudze, a każdy nowy użytkownik może się szybko odnaleźć w gąszczu oferowanych przez aplikację opcji.

W ostatnim czasie można zauważyć dość znaczne zainteresowanie opisywanymi technologiami w środowisku firm zajmujących się szeroko pojętą automatyką. Szczególnie jest to zauważalne na rynku e-commerce, który rozwija się w gwałtownym tempie, a dzięki znacznej przewadze połączenia systemów CMS i RIA, w zakresie elastyczności, skalowalności, szybkości działania oraz niebagatelnej łatwości obsługi, znajdują one zastosowanie zarówno w małych – kilkusobowych – firmach, jak i w większych przedsiębiorstwach. Na rynku firm specjalizujących się w zakresie automatyki można znaleźć proste i przejrzyste strony informujące o charakterze, oferowanym zakresie usług oraz kontakcie z firmą (rys. 3), jak również znacznie rozbudowane strony, na których poza podstawowymi informacjami znaleźć można także informacje dotyczące procesów technologicznych, realizowanych projektów czy rozbudowanego help-desku (rys. 4).



Rys. 4. Przykład zaawansowanej strony internetowej zarządzanej z użyciem systemu CMS  
Fig. 4. An example of a website managed with CMS technology

## Bibliografia

1. Frankowski P.: *CMS. Jak szybko i łatwo stworzyć stronę WWW i zarządzać nią*, Helion, 2007.
2. [www.whatwg.org/specs/web-apps/currentwork/], *Dokumentacja HTML5*, dostęp 25.05.2012.
3. [www.w3.org/TR/FileAPI/, W3C 2011], *Dokumentacja File API*, dostęp 25.05.2012.
4. [caniuse.com], *Tabele wsparcia dla poszczególnych funkcji HTML5 przez przeglądarki*, dostęp 25.05.2012.



## RIA applications based on CMS and e-commerce systems

**Abstract:** RIA applications are gaining popularity over the internet. They are very user friendly because we are eliminating page reloading after making actions. It's ideal technology for making backend for CMS. A CMS system makes managing of website very easy for people with zero HTML skills. It's perfect for owners of small business which can't afford for hire another person, just for managing website. But such complex systems as CMS are very complicated, even if it's simpler than writing website in plain HTML it still may be too hard to understand for people. And it is developer's work to simplify all complex operation, provide user friendly interface which will take users hand and bring him safely to the other side. CMS systems must be also very flexible to adjust themselves to users. We must also remember about keeping system up to date, because every bug which is threaten for data safety must be eliminated. For developer it's also important to log all errors, so in future he or she can recreate the error on develop environment.

**Keywords:** RIA, CMS, e-commerce, internet applications, web applications, PHP, JavaScript

### Michał Łoza

Student na kierunku Informatyka na Wydziale Inżynierii Elektrycznej i Komputerowej Politechniki Krakowskiej. Przewodniczący Koła Naukowego IT. Zainteresowania naukowe: programowanie, elektronika, robotyka, technologie internetowe.

*e-mail: [michal@mloza.pl](mailto:michal@mloza.pl)*



### dr inż. Piotr Andrzej Kowalski

Studia wyższe ukończył na Politechnice Krakowskiej, gdzie pracuje od 2003 r., obecnie na Wydziale Inżynierii Elektrycznej i Komputerowej na stanowisku adiunkta. Jest również adiunktem w Instytucie Badań Systemowych PAN, gdzie w 2009 roku obronił doktorat z zakresu analizy danych i sztucznej inteligencji. Jest członkiem Polskiego Towarzystwa Informatycznego.

*e-mail: [pkowal@pk.edu.pl](mailto:pkowal@pk.edu.pl)*



## Koło Naukowe IT (KNIT) Politechniki Krakowskiej

**Koło Naukowe IT (KNIT) Politechniki Krakowskiej** działa na Wydziale Inżynierii Elektrycznej i Komputerowej przy Katedrze Automatyki i Technik Informatycznych (E-7).

Misją Koła jest stworzenie studentom Wydziału warunków sprzyjających zdobywaniu wiedzy z dziedziny informatyki poprzez wymianę materiałów dydaktycznych i naukowych, jak również umiejętności i doświadczeń. Celem funkcjonowania Koła jest także wzmocnienie i poszerzenie znaczenia informatyki na Wydziale oraz Uczelni.

Koło zajmuje się szeroko pojętymi zagadnieniami informatyki, technik informacyjnych automatyki oraz robotyki. Organizacja zrzesza ludzi chcących przede wszystkim rozwijać swoje pasje oraz uczestniczyć w ciekawych projektach badawczych. KNIT co roku bierze udział w Uczelnianej Sesji Kół Naukowych organizowanej na Politechnice Krakowskiej. Warto również wspomnieć o cotygodniowych seminariach oraz forum koła, dzięki którym można na bieżąco dyskutować o pracy oraz wymieniać się uwagami na różne tematy. Działalność koła nie ogranicza się wyłąc-

nie do regularnych posiedzeń. Jego członkowie uczestniczą w różnych spotkaniach i prelekcjach związanych z popularyzacją szeroko pojętej wiedzy informatycznej. Corocznie Koło wystawia kilka swoich projektów na Festiwalu Nauki na Rynku Głównym w Krakowie.

### Dane kontaktowe:

dr inż. Piotr A. Kowalski – opiekun koła KNIT  
Katedra Automatyki i Technik Informatycznych,  
Wydział Inżynierii Elektrycznej i Komputerowej,  
Politechnika Krakowska  
ul. Warszawska 24, 30-155 Kraków  
budynek WIEiK, pokój 200  
e-mail: [pkowal@pk.edu.pl](mailto:pkowal@pk.edu.pl)  
[www.knit.pk.edu.pl](http://www.knit.pk.edu.pl)  
<http://www.facebook.com/wieik.knit>