

*modelowanie układów elektromechanicznych,  
algorytmy numeryczne, aplikacja internetowa,  
system bazy danych, język programowania Python,  
rama programowa Django*

Krystyna MACEK-KAMIŃSKA\*, Marcin KAMIŃSKI\*

## **METODY REALIZACJI WYBRANYCH ALGORYTMÓW NUMERYCZNYCH DO MODELOWANIA UKŁADÓW ELEKTROMECHANICZNYCH ZA POMOCĄ APLIKACJI INTERNETOWEJ**

Celem artykułu jest próba zbudowania internetowej aplikacji realizującej wybrane algorytmy numeryczne przydatne w modelowaniu i symulacji układów elektromechanicznych. Aplikacja będzie umożliwiała przeprowadzenie obliczeń symulacyjnych w czasie rzeczywistym i przedstawienie ich wyników na dynamicznie generowanych stronach internetowych. Wszystkie dane niezbędne do obliczeń będą pochodzić z odpowiednich tabel bazy danych. Również wyniki obliczeń będą mogły być zachowywane w bazie. Proponowane rozwiązanie daje dwie nowe możliwości, niedostępne dla tradycyjnych metod prowadzenia obliczeń: będzie umożliwiać przeprowadzanie obliczeń z wykorzystaniem dowolnego komputera posiadającego dostęp do Internetu oraz udostępni narzędzia do łatwej archiwizacji danych w odpowiednio zaprojektowanej bazie danych.

### 1. WSTĘP

W roku 2013 minie dwadzieścia lat od udostępnienia aplikacji *Mosaic*, która pozwalała w sposób graficzny przedstawiać treść stron internetowych i od której wywodzą się wszystkie aktualnie stosowane przeglądarki. Kolejnym krokiem pozwalającym Internetowi osiągnąć aktualny stan rozwoju, było wprowadzenie do użytku dynamicznych serwisów internetowych. Wcześniej wykorzystywane statyczne strony internetowe generowały za każdym żądaniem taką samą treść, a nawet najmniejsza zmiana ich zawartości wymagała ingerencji twórcy w jej kod. Z kolei dynamiczne strony internetowe potrafią dostosować swoją treść do konkretnych wymagań użytkownika zdefiniowanych w żądaniu.

---

\* Politechnika Opolska, Wydział Elektrotechniki, Automatyki i Informatyki, Instytut Układów Elektromechanicznych i Elektroniki Przemysłowej ul. Prószkowska 76, 45-758 Opole, k.macek-kaminska@po.opole.pl

Działają one najczęściej w oparciu o szablony, które zawierają zarówno stałe elementy witryny jak i elementy zmienne, które są pobierane z bazy danych. Wszystkie popularne obecnie zastosowania aplikacji internetowych, takich jak systemy zarządzania treścią (ang. *Content Management System, CMS*), sklepy internetowe, fora internetowe, portale społecznościowe i inne, opierają się o ten rodzaj tworzenia stron internetowych. Najbardziej popularne technologie używane do tworzenia dynamicznych stron internetowych wykorzystują przede wszystkim narzędzia oparte o paradygmat wolnego oprogramowania (*Open Source*): serwer WWW *Apache*, język *PHP* i baza danych *MySQL*. Istnieje też wiele innych produktów przeznaczonych do tych celów – najbardziej znane z nich oparte są o technologię Microsoft .NET (*ASP.NET*) oraz o technologię Java (*JSP, Servlets Java*). Wybór konkretnej technologii jest zazwyczaj kwestią osobistych preferencji, ale możliwe jest określenie cech, które mogą być przydatne w procesie tworzenia aplikacji.

### 1.1. WYBÓR TECHNOLOGII TWORZENIA APLIKACJI INTERNETOWEJ

Wspomniane w poprzednim paragrafie produkty do tworzenia serwisów internetowych są specjalizowanymi narzędziami do szybkiego i optymalnego przygotowania aplikacji internetowej. Przede wszystkim pozwalają one wykorzystać metody protokołu HTTP wykorzystywanego w sieci WWW oraz umożliwiają dostęp do informacji przechowywanych w wybranym systemie bazy danych. Niestety, założona przez ich twórców specjalizacja na obsługę tych funkcjonalności nie idzie w parze z łatwością ich stosowania w innych dziedzinach techniki komputerowej. Istnieje więc wyraźny podział na narzędzia wykorzystywane w technice internetowej oraz na języki programowania i specjalizowane środowiska obliczeniowe wykorzystywane w obliczeniach naukowych i inżynierskich.

Dynamiczny rozwój Internetu powoduje, że dla wielu użytkowników komputera staje się on podstawowym medium prezentacji i wymiany informacji. Wynika to między innymi z faktu, że korzystanie z jego zasobów jest łatwe i nie wymaga żadnej specjalistycznej wiedzy. Biorąc również pod uwagę fakt, że powstaje coraz więcej narzędzi służących do tych celów, pozwalającym nawet osobom nie związanym zawodowo z tworzeniem stron internetowych tworzyć nawet rozbudowane serwisy. Narzędzia te – *Joomla, WordPress, Mambo* – oparte o język *PHP* i bazę *MySQL*, zapewniają typowe funkcjonalności wymagane od systemów zarządzania treścią (ang. *Content Management System – CMS*). Jednak trudne jest ich wykorzystanie w mniej typowych zastosowaniach. Język *PHP*, jako specjalizowany skryptowy język programowania, był głównie zaprojektowany do generowania stron internetowych w czasie rzeczywistym. Wykorzystanie tego języka jako narzędzia do programowania algorytmów numerycznych wymagałoby od programisty przyswojenia zasad jego działania.

Technologia internetowa oparta na dynamicznych stronach WWW rozwinęła narzędzia wykorzystujące systemy baz danych. Ciekawym wyzwaniem byłoby zastosowanie systemu zarządzania bazą danych w obliczeniach naukowych i inżynierskich.

Mając za cel opracowanie narzędzia pozwalającego prezentować wyniki obliczeń numerycznych w Internecie można było albo poszukać sposobu automatycznego ich formatowania do postaci strony internetowej albo zastanowić się nad sposobem przeprowadzania obliczeń na serwerze i generowania wyników od razu w formie pliku *.html*. Podstawowym problemem był wybór odpowiedniej technologii. W niniejszym artykule zaprezentowano doświadczenia autorów w tworzeniu aplikacji internetowej z wykorzystaniem ramy programowej (ang. *framework*) *Django* [3, 4].

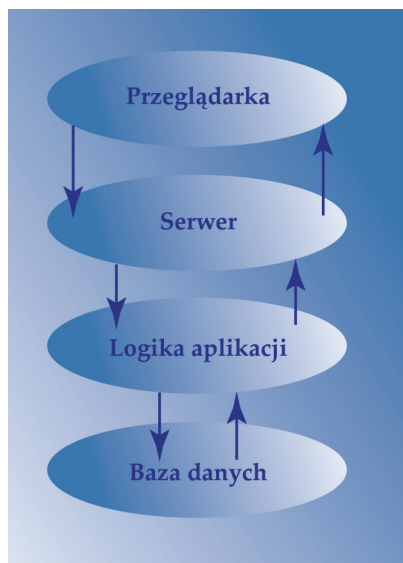
*Django* jest narzędziem napisanym w języku programowania *Python*, który realizuje powszechnie stosowany w tego typu aplikacjach wzorzec programowy *Model-Widok-Kontroler* (ang. *Model-View-Controller – MVC*). Zasadniczą cechą tego narzędzia jest to, że wszystkie etapy tworzenia aplikacji, łącznie z projektem systemu bazodanowego, metodami dostępu do bazy danych i generowaniem dynamicznych stron internetowych, są programowane z poziomu języka *Python* [6].

Język *Python* jest obecnie jednym z najszybciej rozwijanych języków programowania. Charakteryzuje się on jasną i dobrze przemyślaną składnią, szerokim zakresem stosowania, wieloma rozszerzeniami, pozwalając zrealizować różnorodne zadania, od zadań grafiki trójwymiarowej (biblioteka *Visual Python* [8], moduł game engine w programie *Blender* [1]) poprzez wykorzystanie w systemach baz danych (w dystrybucji języka *Python* znajduje się silnik bazodanowy *SQLite3*), do obliczeń naukowo-inżynierskich (biblioteka *Matplotlib* [5, 9], biblioteki *NumPy*, *SciPy* [7]).

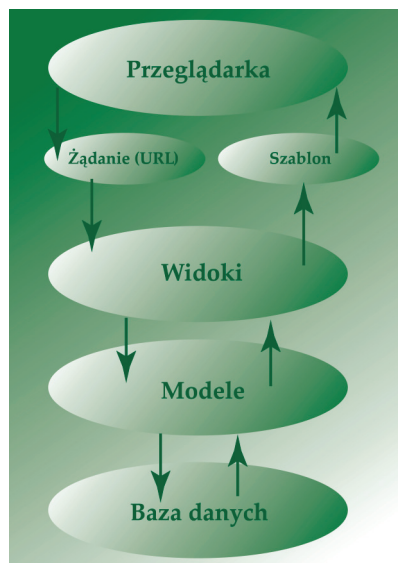
## 2. DJANGO JAKO NARZĘDZIE DO TWORZENIA APLIKACJI INTERNETOWEJ

Ogólna struktura aplikacji internetowej ukazująca logiczny podział jej elementów na osobne warstwy, które komunikują się ze sobą, przedstawiono na rysunku 1. Warstwy te są związane z logicznym podziałem komponentów i funkcjonalności. Żądanie od użytkownika trafia do serwera. Serwer przetwarza żądanie przy użyciu algorytmów wdrożonych w warstwie logiki aplikacji. W celu realizacji tych algorytmów, może być konieczne sięgnięcie do danych przechowywanych w bazie danych. Przygotowana odpowiedź jest wyświetlana jako strona internetowa.

Schemat przedstawiony na rysunku 2. przedstawia strukturę aplikacji internetowej przygotowanej z wykorzystaniem ramy projektowej *Django*. W stosowanym w niej wzorcu *Model-View-Controller* następuje modyfikacja: *Widok (View)* jest zastąpiony przez *Szablon (Template)*, a *Kontroler* przez *Widok*. Ten rodzaj wzorca nosi nazwę *Model-Template-View (MTV)*. W *Django* zasadniczą rolę odgrywa właśnie element *Widoku* – on to bowiem, na podstawie pobranego z przeglądarki żądania, przygotowuje treść tworzonej dynamicznie strony internetowej. W tym celu korzysta z określonych w elemencie *Model* definicji struktur bazodanowych i poprzez ten element pobiera odpowiednie dane z bazy danych. Przygotowane do opublikowania przez element *Widok* dane są renderowane do postaci strony WWW za pomocą odpowiednio zdefiniowanego *Szablonu*.



Rys. 1. Architektura typowej aplikacji internetowej  
Fig. 1. Architecture of a typical Web application

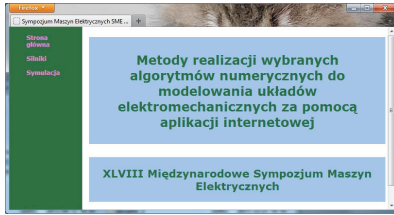


Rys. 2. Architektura aplikacji internetowej opartej na framework Django  
Fig. 2. Architecture of the Django application

*Django* obsługuje kilka popularnych silników baz danych (*MySQL*, *SQLite*, *PostgreSQL*, *Oracle*) i daje możliwość stworzenia systemu baz danych w aplikacji internetowej niezależnie od wybranego silnika i realizacji dostępu do danych przechowywanych w takiej bazie w języku programowania *Python*, co zwalnia użytkownika od posiadania doświadczenia w języku *SQL*. Odbývá się to dzięki temu, że w *Django* jest zawarty moduł mapera obiektowo-relacyjnego (ang. *Object-Relational Mapper – ORM*), czyli narzędzia, które odwzorowuje obiektową architekturę aplikacji w *Django* na wybraną w projekcie konkretną relacyjną bazę danych.

### 3. PRZYKŁADY REALIZACJI APLIKACJI INTERNETOWEJ

Proponowana aplikacja internetowa realizuje typowy schemat obliczeń symulacyjnych. Na podstawie wybranego modelu matematycznego maszyny indukcyjnej został przygotowany model obliczeniowy w postaci układu równań różniczkowych. Wartości liczbowe parametrów modelu potrzebne do obliczeń będą pobrane z bazy danych. Aplikacja, uruchomiona na serwerze, realizuje typowy algorytm Rungego–Kutty 4. rzędu w celu iteracyjnego rozwiązywania układu równań różniczkowych [2]. Uzyskane rozwiązania układu równań różniczkowych dla poszczególnych chwil czasu pozwalają obliczyć wartości zdefiniowanych w modelu obliczeniowym wartości



Rys. 3. Strona główna projektowanej aplikacji internetowej

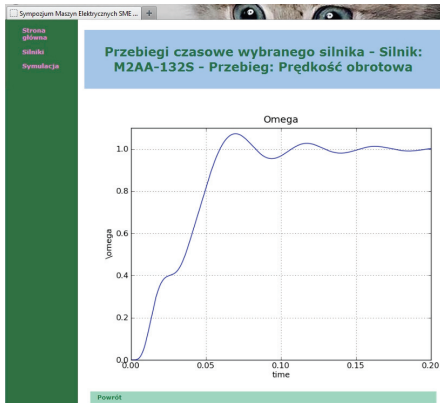
Fig. 3. Home page of the proposed application internet site

No	Typ	Producent	Rok	Kategoria
1	SZJre-134IE	DOLMEL	1985	A
2	M2AA-132S	ABB	1997	D

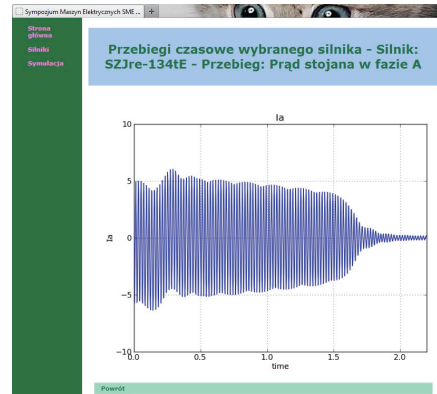
Rys. 4. Fragment listy silników z bazy danych  
Fig. 4. Part of a motor list from database

Parametr	Wartość
Napięcie znamionowe	380.0
Prąd znamionowy	6.7
Moment znamionowy	36.2
częstotliwość sieci	50.0
Liczba par biegunów	2.0
Moment bezwładności	0.031
Rezystancja fazy stojana	2.9
Reaktancja magnesowania	143.5
Kąt fazowy	-1.5728
Rezystancja 1. klatki	3.66322
Rezystancja 2. klatki	2.48496
Reaktancja 1. klatki	8.37725
Reaktancja 2. klatki	18.0526
Reaktancja stojana	1.6528

Rys. 5. Strona z parametrami wybranego silnika  
Fig. 5. Page of parameters of selected motor



Rys. 6. Wykres wybranego przebiegu symulacyjnego  
Fig. 6. The graph of the selected simulation run



Rys. 7. Wykres wybranego przebiegu symulacyjnego  
Fig. 7. The graph of the selected simulation run

funkcji wyjścia. Na ich podstawie będzie można przygotować ich wykresy w wygodnym formacie graficznym. Sposób przeglądania tych wykresów będzie podobny do przeglądania internetowej galerii zdjęć. Wykresy są tworzone z wykorzystaniem biblioteki *Matplotlib*, która umożliwia programową obsługę tworzenia wykresów. Również z tej biblioteki zaczerpnięta została metoda odwracania macierzy wykorzystana w obliczeniach. Rysunki od 3. do 7. pokazują różne etapy działania aplikacji.

#### 4. WNIOSKI

Język programowania *Python* pozwolił po pierwsze na łatwą i szybką adaptację kodu procedur numerycznych opracowanych wcześniej w językach *FORTRAN* i *C++* oraz na eleganckie przygotowanie obsługi całej zaplanowanej funkcjonalności aplikacji internetowej. Czas wykonania na serwerze obliczeń symulacyjnych dla 1000. chwil czasowych trwa na tyle krótko, że użytkownik nie odczuwa żadnego przestoju w działaniu aplikacji. Doskonale przemyślana konstrukcja ramy projektowej *Django* sprawia, że budowa aplikacji internetowej z jej użyciem jest prawdziwą przyjemnością dla programisty.

Zebrane doświadczenia potwierdzają, że możliwe jest przygotowanie aplikacji internetowej realizującej nawet bardzo skomplikowane algorytmy numeryczne. Dodatkowo wykorzystanie systemów bazodanowych w obliczeniach symulacyjnych pozwala rozszerzyć funkcjonalność programów obliczeniowych.

#### LITERATURA

- [1] ANDERS M., *Blender 2.49 Scripting*, Pact Publishing, Birmingham 2010.
- [2] BARON B., *Metody numeryczne w Turbo Pascalu*, Helion, Gliwice 1994.
- [3] FORCIER J., BISSEX P., CHUN W., *Python i Django Programowanie aplikacji webowych*, Helion, Gliwice 2009.
- [4] HOLOVATY A., KAPLAN-MOSS J., *The Definitive Guide to Django: Web Development Done Right (Expert's Voice in Web Development)*, Apress, New York 2009.
- [5] <http://matplotlib.sourceforge.net/> – *Matplotlib official website*
- [6] <http://python.org/> – *Python official website*
- [7] <http://scipy.org/> – *SciPy/NumPy official website*
- [8] <http://vpython.org/> – *Visual Python official website*
- [9] TOSI S., *Matplotlib for Python Developers*, Pact Publishing, Birmingham 2009.

#### METHODS OF IMPLEMENTATION OF SELECTED NUMERICAL ALGORITHMS FOR MODELING OF ELECTROMECHANICAL SYSTEMS USING WEB APPLICATIONS

Article presents the authors experiences in creating a web-based application which is able to perform selected numerical calculations useful in electromechanical systems modeling. This application will enable to carry out the simulation calculations in real time and present their results on dynamic created web pages. All data required for calculations will be derived from the corresponding database tables. The proposed solution will give two new possibilities not available in traditional methods of calculation: it will allow to carry out calculations using any computer with Internet access and tools for easy data storage in a properly designed database.