

Piotr JERUSZKA

Politechnika Częstochowska, Instytut Informatyki Teoretycznej i Stosowanej

## WEBMES – PROJEKT USŁUG SIECIOWYCH DO REALIZACJI SYMULACJI INŻYNIERSKICH

**Streszczenie.** W pracy zawarto projekt i opis dotychczasowej implementacji usług sieciowych, służących do przeprowadzania symulacji zagadnień początkowo-brzegowych. Opisywane usługi korzystają z formatu JSON oraz zwykłych plików tekstowych (z danymi wejściowymi). Nie jest wymagana lokalna kopia oprogramowania inżynierskiego, co upraszcza naukowcom przygotowanie danych wejściowych i analizę wyników obliczeń.

**Słowa kluczowe:** MES, usługi sieciowe, symulacje inżynierskie

## WEBMES – WEB-SERVICES PROJECT FOR THE IMPLEMENTATION OF ENGINEERS' SIMULATIONS

**Summary.** The paper contains description and the project of the implementation of web-services. The web-services simulate boundary value problems with initial conditions. The JSON and text file have been used in the described system. The request to have local instance of simulating program is not required which makes the study about phenomena less problematic.

**Keywords:** FEM, web-services, engineers' simulations

### 1. Wstęp

Symulacje inżynierskie przeprowadzane w systemach informatycznych (superkomputerach, klastrach) rozwiązują zagadnienia początkowo-brzegowe, modelujące dane zjawisko fizyczne. Takie problemy są często opisywane pod postacią równań różniczkowych (zwłaszcza równań różniczkowych cząstkowych), z zadanymi warunkami brzegowymi. W większo-

ści modelowane są zjawiska niestacjonarne, tj. poszukiwana funkcja jest funkcją zależną od czasu.

Przykładem zagadnienia początkowo-brzegowego jest problem przewodzenia ciepła w dwuwymiarowym obszarze. Problem ten jest opisany równaniem Fouriera:

$$\nabla \cdot (\lambda \nabla T) = c\rho \frac{\delta T}{\delta t}, \quad (1)$$

gdzie  $\lambda$  to współczynnik przewodzenia ciepła,  $c$  – ciepło właściwe, a  $\rho$  – gęstość materiału.

Analityczne rozwiązanie powyższego równania jest zależne od warunków początkowych i brzegowych. Dlatego istnieje potrzeba przybliżenia rozwiązania zadania, korzystając z jednej z metod numerycznych. Jako *rozwiązanie zadania* w tej pracy będzie przyjmowany zbiór wartości pewnej wielkości fizycznej, reprezentujący wartości w pewnych punktach danego obszaru. Ponieważ przedstawiona symulacja inżynierska dotyczy przewodzenia ciepła, więc poszukiwaną wielkością fizyczną jest temperatura.

### 1.1. Metoda elementów skończonych

Metoda elementów skończonych jest jedną z metod przybliżonego rozwiązywania zagadnień początkowo-brzegowych. Ciąg przekształceń równania różniczkowego cząstkowego prowadzi od postaci równania algebraicznego do postaci równania macierzowego. Obszar zadania dzielony jest na elementy skończone. Zbiór wierzchołków wszystkich elementów danego obszaru wyznacza zbiór tych węzłów, dla których rozwiązania równania macierzowego dadzą dyskretne wartości. Możliwe jest obliczenie przybliżonej wartości w dowolnym punkcie elementu z wykorzystaniem funkcji kształtu.

Równanie macierzowe doprowadzane jest do postaci:

$$\mathbf{AX} = \mathbf{B}, \quad (2)$$

gdzie  $\mathbf{A}$  jest macierzą kwadratową o rozmiarze  $n$  (liczba obliczanych węzłów),  $\mathbf{B}$  jest wektorem prawych stron, a  $\mathbf{X}$  jest wektorem niewiadomych, w tym przypadku temperatur.

Asemblacja i rozwiązywanie równań macierzowych są przedmiotami badań naukowców. Celem tych badań jest przyspieszenie rozwiązywania wspomnianych równań, często złożonych z milionów niewiadomych. Jednym z rozwiązań jest skorzystanie z technik zrównoleglenia i rozpraszania obliczeń [1]. W artykule skupiono się jedynie na rozwiązaniach, które są proste w procesie wdrożenia i nie korzystają ze zrównoleglenia obliczeń.

### 1.2. Istniejące usługi i cel pracy

Badając aktualny stan wiedzy, skupiono się na systemach (oprogramowaniu), które za pomocą MES realizują symulacje inżynierskie. W pracy [2] stworzono aplikację internetową,

która jest wdrożona po stronie serwera. Umożliwia to przeprowadzanie obliczeń bez potrzeby posiadania odpowiedniego oprogramowania (oprócz przeglądarki internetowej), ale nie udostępnia wyników w formie usług sieciowych. Podobne założenie, tj. umożliwienie przeprowadzania zdalnych obliczeń z wykorzystaniem aplikacji internetowej, realizowano podczas prac nad systemem webMES.

Artykuł [3] opisuje wykorzystanie usług sieciowych do generowania siatek elementów skończonych. System nie umożliwia dokonywanie obliczeń, ale przedstawia interesującą koncepcję wykorzystania wieloplatformowej komunikacji pomiędzy aplikacją i generatorem. Pozwala to programiście symulacji inżynierskich na delegowanie funkcjonalności generowania siatek elementów skończonych do realizacji przez zdalną aplikację.

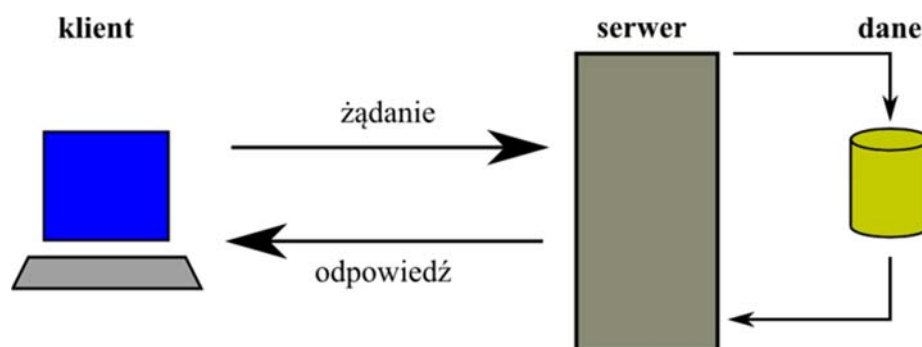
### 1.3. Cel i założenia pracy

Celem pracy jest zaprojektowanie usług sieciowych, umożliwiających udostępnianie wyników obliczeń realizowanych na zdalnej maszynie. Podczas projektowania systemu webMES założono następujące cele działającego systemu:

- możliwość uruchomienia aplikacji inżynierskiej jako procesu, z możliwością dołączenia argumentów wywołania programu;
- łatwe dla użytkownika przekazywanie danych wejściowych;
- użycie usług sieciowych do pozyskania stanów zadań i wyników obliczeń.

## 2. Usługi sieciowe

Aplikacje (grupa aplikacji, programów, systemów informatycznych itp.) komunikują się w celu wymiany informacji. W pracy założono komunikację w architekturze „klient-serwer”, tj. jedna aplikacja (klient) wysyła żądanie do drugiej (serwer) i w odpowiedzi otrzymuje dane (rys. 1).



Rys. 1. Schemat komunikacji sieciowej  
Fig. 1. Client-server communication diagram

Żądanie jest ciągiem bajtów, których format jest zależny od protokołu używanego w komunikacji. W dalszej części pracy będzie stosowany protokół HTTP (ang. *Hypertext Transfer Protocol*) z wykorzystaniem metody GET.

Żądanie z użyciem łańcucha zapytania (ang. *query string*) może wyglądać następująco:

```
http://strona/zadanie?numer=10&krok=20
```

W wyniku zostanie uzyskany zasób o nazwie `zadanie` z parametrami `numer` o wartości 10 oraz `krok` o wartości 20.

Użyta technologia tworzenia usług sieciowych pozwala na inny sposób zapisu, w którym wartości parametrów są rozdzielane znakami ukośnika. Taki format zapisu nazywa się semantycznym URL (ang. *semantic URL*). Dopasowaniem wartości do zmiennych zajmuje się kontroler, który rozdziela żądania. Przedstawiony adres, z wykorzystaniem łańcucha zapytań, w formacie semantycznego URL wygląda następująco:

```
http://strona/zadanie/10/20
```

Taki format skraca zapis (a więc i ilość danych przesyłanych do kontrolera). Wadą tego zapisu jest brak wiedzy na temat semantyki kolejnych składowych adresu.

## 2.1. Format JSON w Django

Komunikacja sieciowa musi zachodzić na odpowiednich zasadach. Protokół definiuje, *jak* dane są przesyłane, natomiast nie zdefiniowano, *jakie* dane są przesyłane. Wspólnym językiem pomiędzy usługami sieciowymi, klientem i oprogramowaniem inżynierskim (poza danymi wejściowymi) jest język wymiany danych JSON.

JSON jest formatem wziętym z notacji obiektów języka JavaScript [4]. Istotną zaletą tego formatu jest to, że jest bardzo zbliżony do definicji słowników w Pythonie, którego użyto do zaprogramowania usług sieciowych. Ze względu na rozpowszechnienie tego formatu danych (podobnie jak języka semistrukturalnego XML) praktycznie każdy język programowania posiada gotową bibliotekę do obsługi tego formatu. Dodatkowo wykorzystany format jest wspierany w technologii Django – platformie programistycznej (ang. *framework*) do budowy aplikacji internetowych w języku Python [5].

Celem użycia Django jest szybka budowa aplikacji internetowych. Framework samodzielnie buduje prosty panel administratora. W pracy panel spełnia rolę „centrum sterowania” – za jego pomocą można przesyłać pliki wejściowe, zdefiniować parametry zadania, a następnie uruchomić je oraz śledzić stan wykonywania zadania.

W dalszej części pracy zamiennie z terminem „usługi sieciowe”, w odniesieniu do opisywanego systemu, będzie stosowany skrót API (ang. *Application Programming Interface*). Usługi sieciowe można wykorzystać jako interfejs programistyczny dla programistów zewnętrznych aplikacji, korzystających z systemu webMES.

### 3. Projekt systemu webMES

W tabeli 1 umieszczono zaprojektowane metody, udostępniane przez API systemu webMES.

Tabela 1

Lista dostępnych metod za pomocą API systemu webMES

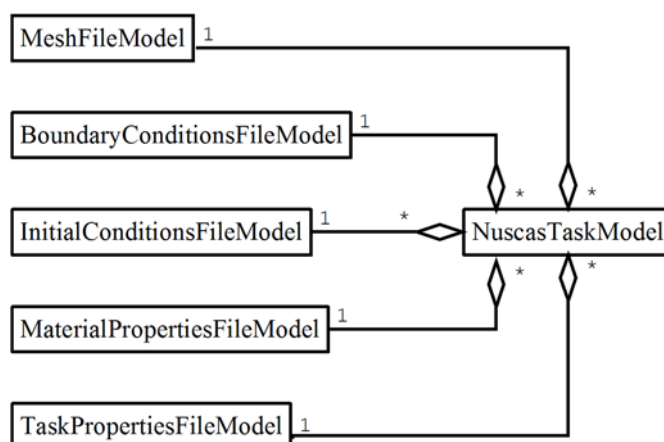
Metoda	URLPattern	Argumenty	wynik
get_tasks	tasks	-	lista zadań
get_computed_steps	tasks/<NRT>/steps	NRT – nr zadania	liczba obliczonych kroków
get_computed_result	tasks/<NRT>/<NRS>	NRT – nr zadania NRS – nr kroku	lista wartości w węzłach

Termin URLPattern oznacza wzorzec żądania w postaci adresu URL, który jest przetwarzany przez kontroler Django. Przykładowo, jeżeli adres URI serwera to `serwer`, to wywołanie poniższego żądania HTTP:

```
http://serwer/tasks/5/4
```

spowoduje wywołanie metody `get_computed_result`, która spowoduje zwrócenie, w formacie JSON, wyników obliczeń z kroku czwartego zadania o numerze 5.

#### 3.1. Modele



Rys. 2. Diagram klas, przedstawiający klasę `NuscasTaskModel` i jej związki  
 Fig. 2. `NuscasTaskModel` (and its connections) class diagram

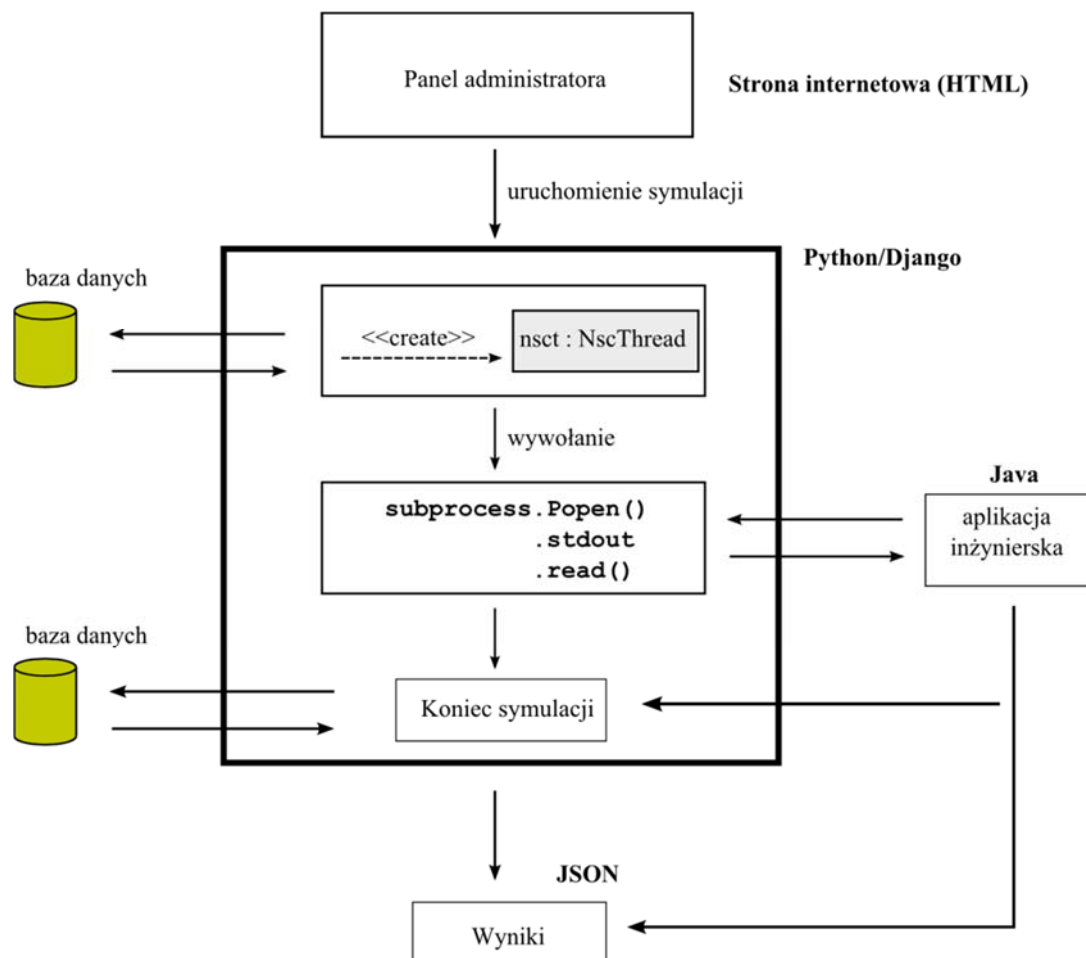
Django umożliwia mapowanie obiektowo-relacyjne (ang. *Object-Relational Mapping*, ORM) do jednej ze skonfigurowanych baz danych. Oznacza to, że programista nie musi przygotowywać schematu bazy (przez zapytania DDL) i zajmować się niskopoziomym dostępem do danych. Wykorzystano prostotę tego rozwiązania do stworzenia systemu przechowywania zadań i ich składowych. Przykładowo, możliwe jest wielokrotne wykorzystanie tej samej siatki, danych zadania itp.

Podstawową klasą jest model `NuscasTaskModel`, kapsułkujący stan pewnego zadania obliczeniowego. Klasa ta agreguje (rys. 2) dane dotyczące zadań w polach typu:

- `MeshFileModel` (plik siatki);
- `BoundaryConditionsFileModel` (plik warunków brzegowych);
- `InitialConditionsFileModel` (plik warunków początkowych);
- `MaterialPropertiesFileModel` (plik własności materiałowych);
- `TaskPropertiesFileModel` (plik własności zadania).

Wspomniane klasy dziedziczą z klasy `GeneralFileModel`, która zapewnia wspólny interfejs dla modeli korzystających z zapisu plików. Późniejszy refaktoring pozwoli na rozbudowę fizycznego modelu systemu o obsługę plików z innych narzędzi inżynierskich.

### 3.2. Warstwy systemu



Rys. 3. Warstwy wykonywania obliczeń w systemie webMES

Fig. 3. Computing layers in webMES

Użytkownik systemu tworzy nowe zadanie i przesyła pliki za pomocą panelu administratora (rys 3). Uruchomienie zadania zaimplementowano w bardzo prosty sposób: model `NuscasTaskModel` posiada pole `Start Box`, widoczne w panelu administratora. Naciśnięcie tego przycisku powoduje utworzenie instancji obiektu `NscThread`.

Klasa `NscThread` jest implementacją wątku (klasy `Thread` w Pythonie) i jego głównym zadaniem jest utworzenie obiektu wątku (`Popen`). Utworzenie wspomnianej instancji powoduje rozpoczęcie procesu, tj. wywołanie w powłoce polecenia. Polecenie składa się z wywołania pliku wykonywalnego (stąd wymóg stosowania oprogramowania inżynierskiego w formie plików wykonywalnych) oraz argumentów. Argumenty są zależne od implementacji symulatora danego zjawiska. W używanej obecnie implementacji argumentami są ścieżki do plików z danymi.

### 3.3. Dane wejściowe i schemat testów

Dane wejściowe, wymagane przez system, składają się z pięciu plików:

- pliku geometrii i siatki elementów skończonych,
- pliku z definicją warunków początkowych,
- pliku z definicją warunków brzegowych,
- pliku z definicją własności materiałowych,
- pliku z definicją zadania.

Schemat plików wejściowych jest kompatybilny z systemem NuscaS [6], ale możliwa jest rozbudowa systemu o wczytywanie danych w dowolnym, zaimplementowanym przez programistów formacie. NuscaS jest programem powstałym na Politechnice Częstochowskiej podczas prac nad numerycznym modelowaniem zjawisk termodynamicznych. Wykorzystany format plików wejściowych pozwolił na szybkie sprawdzenie wyników obliczeń (podrozdział 3.4).

Do przetestowania systemu użyto autorskiej aplikacji jMES, która obecnie potrafi symulować przewodzenie ciepła w jednorodnym obszarze dwuwymiarowym. jMES składa się z dwóch głównych bibliotek. Ogólna, `jmes_core`, zawiera klasy potrzebne do rozpatrywania zjawisk dwuwymiarowych. Szczegółowa, `jmes_tc`, złożona jest z komponentów wymaganych przy rozpatrywaniu przewodzenia ciepła, bez uwzględnienia zjawisk fazowych. Sprawdzono wyniki symulacji dla pięcioelementowego kwadratowego obszaru, w którym na jednym brzegu zadano warunek brzegowy II rodzaju (stały strumień ciepła), a na drugim – warunek brzegowy III rodzaju (wymiana ciepła z otoczeniem).

### 3.4. Przykładowe wyniki obliczeń

Rezultaty symulacji zostały sprawdzone z rezultatami generowanymi przez system NuscaS. Uznano obliczone wyniki za dostatecznie przybliżone (NuscaS generuje wyniki z dokładnością 0,001 K). Poniżej przedstawiono rezultat działania API systemu webMES przy pobraniu wyników obliczeń 10 kroku w zadaniu opisanym w podrozdziale 3.3. Odpowiedź zawiera podstawowe informacje o siatce oraz wartości temperatur w pięciu węzłach.

```
{
  "step": 10,
  "desc": "test1",
  "dimensions": 2,
  "nodes_count": 5,
  "elements_count": 4,
  "val_in_els": 0,
  "vals_count": 1,
  "vals_names": "temperatura [K]",
  "vals": [
    {
```



```
    "id": 1,  
    "val": 399.98788950235684  
  },  
  {  
    "id": 2,  
    "val": 397.6141462630295  
  },  
  {  
    "id": 3,  
    "val": 400.0321221114701  
  },  
  {  
    "id": 4,  
"val": 397.6141462630295  
  },  
  {  
    "id": 5,  
"val": 400.0321221114701  
  }  
] }  
}
```

#### 4. Podsumowanie i plan dalszych prac

Przedstawiony system pozwolił na przeprowadzenie prostej symulacji zjawiska niestacjonarnego, z wykorzystaniem autorskiego symulatora jMES. Uruchomienie symulacji wymagało jedynie stworzenia odpowiedniego obiektu i przesłania danych wejściowych, bez potrzeby uruchamiania programu po stronie użytkownika.

Można zauważyć, że zarówno usługi sieciowe, jak i strona internetowa (obecnie – panel administratora) działają na tym samym zestawie danych, dlatego kolejnym etapem prac będzie budowa aplikacji internetowej. Pozwoli ona na wizualizację wyników obliczeń, zapewnianych przez API systemu. W fazie rozwoju systemu można również zwrócić uwagę na poprawę metod przechowywania danych wejściowych. Rozbudowany system pozwoli także na łatwe wdrożenie procesu testowania własnego oprogramowania inżynierskiego.

#### BIBLIOGRAFIA

1. Wyrzykowski R., Szczygiol N., Olas T.: Parallel numerical modelling of solidification on a PC-based cluster. Archives of foundry, 2001, s. 408÷413.
2. Chen H.-M., Lin Y.-C.: Web-FEM: An internet-based finite-element analysis framework with 3D graphics and parallel computing environment. Advances in Engineering Software, Vol. 39, Issue 1, 2008, s. 55÷68.
3. Szczygiol N., Mikoda J., Wawszczak A.: Web service for finite element mesh generator. Computer Methods in Material Sciences, Vol. 10, 2010, s. 176÷180.

4. Crockford D.: The application/json Media Type for JavaScript Object Notation (JSON), <http://tools.ietf.org/html/rfc4627> (dostęp 03.02.2015).
5. Strona projektu Django, <https://www.djangoproject.com/> (dostęp 03.02.2015).
6. Sczygiol N., Nagórka A., Szwarz G.: NuscaS – autorski program komputerowy do modelowania zjawisk termomechanicznych krzepnięcia. [w:] Świątkowski R. (red.): Polska metalurgia w latach 1998-2012. Wyd. Naukowe „AKAPIT”, Kraków, t. 2, s. 243÷249.

## Abstract

This article presents the idea, project and initial implementation of the webMES system - the web-services for the execution of engineers' simulations.

Physical phenomena can be often presented in the form of the boundary value problem. That issue consists of the differential equation (frequently partial), assigned initial and boundary conditions. The paper is focused on the way to solve the problem of heat conduction by the Fourier's equation (1). Approximate solution was obtained with the Finite Element Method (FEM) using the author's engineering application (jMES) and web-services.

Web-services are based on client-server architecture (Fig. 1). HTTP protocol is used to send requests and receive responses. Web-services server responses are stored in a JSON format. Mentioned communication was developed with Python and Django framework. The framework allows to create web applications quickly. WebMES control panel allows to create tasks i.e. `NuscaSTaskModel` class object (Fig. 2). The task aggregates information about files: mesh, the initial conditions, boundary conditions, material properties, and configuration tasks. Created task is run as a thread. The thread runs engineering application executable file specifying the path to the uploaded files. The application generates results in JSON format accessible through web-services (Fig. 3 and Table 1).

The results of the sample simulation (thermal conduction in a squared two-dimensional area) were compared with the results generated by NuscaS (developed at the Technical University of Czestochowa). The same input data was used and the results are similar.

It must be emphasized that webMES may cooperate with any engineering software that accepts specific input files and generates the results in JSON format. The system is in the initial phase of implementation. Web-services may be used as the Application Programming Interface (API). Engineering software developers will be able to use the API for remote execution of calculations.

**Adres**

Piotr JERUSZKA: Politechnika Częstochowska, Instytut Informatyki Teoretycznej  
i Stosowanej, ul. Dąbrowskiego 69, 42-200 Częstochowa, Polska, piotr.jeruszka@icis.pcz.pl.