

**Akademia Górniczo-Hutnicza  
im. Stanisława Staszica w Krakowie**

---

Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki



**PRACA INŻYNIERSKA**

**DOROTA WOJTAŁOW, JACEK ZŁYDACH**

**SYMULACJA ROZPRZESTRZENIANIA SIĘ DYMU I OGNIA W  
OPARCIU O NIEHOMOGENICZNE AUTOMATY KOMÓRKOWE**

PROMOTOR:  
dr inż. Jarosław Wąs

Kraków 2010

### **OŚWIADCZENIE AUTORA PRACY**

OŚWIADCZAM, ŚWIADOMY ODPOWIEDZIALNOŚCI KARNEJ ZA POŚWIADCZENIE NIEPRAWDY, ŻE NINIEJSZĄ PRACĘ DYPLOMOWĄ WYKONAŁEM OSOBIŚCIE I SAMODZIELNIE, I NIE KORZYSTAŁEM ZE ŹRÓDEŁ INNYCH NIŻ WYMNIENIONE W PRACY.

.....

PODPIS



**DOROTA WOJTAŁOW, JACEK ZŁYDACH**

**SIMULATION OF FIRE AND SMOKE BY USING  
NON-HOMOGENEOUS CELLULAR AUTOMATA**

**SUPERVISOR:**  
**Jarosław Wąs Ph.D**

**Krakow 2010**

Serdecznie dziękujemy ...

## Spis treści

<b>1. Wstęp</b> .....	6
1.1. Temat pracy.....	6
1.2. Geneza tematu.....	6
1.3. Realizacja projektu .....	6
1.4. Struktura pracy.....	6
<b>2. Projekt</b> .....	8
2.1. Główne założenia.....	8
2.2. Architektura aplikacji .....	8
2.2.1. Moduł kontroler .....	9
2.2.2. Widok .....	9
2.2.3. Model .....	9
2.3. Moduły .....	9
2.4. Obiekty .....	9

# 1. Wstęp

## 1.1. Temat pracy

Tematem pracy jest stworzenie symulacji rozprzestrzeniania się dymu i ognia w oparciu o niehomogeniczne automaty komórkowe. Zakres pracy obejmuje stworzenie symulacji rozchodzenia się dymu i ognia na podstawie automatów komórkowych wraz z jej wizualizacją, a także walidację stworzonego modelu. Celem pracy jest pokazanie możliwości niehomogenicznych automatów komórkowych jako narzędzia umożliwiającego odzwierciedlenie rzeczywistego rozprzestrzeniania się dymu i ognia podczas pożaru.

## 1.2. Geneza tematu

W dobie wszechobecnej urbanizacji i ciągłego budownictwa, wraz ze wzrostem świadomości dotyczącej bezpieczeństwa pożarowego oraz zaangażowania w jego zagwarantowaniu pojawiła się potrzeba możliwości modelowania i obserwacji rozprzestrzeniania się ognia w zamkniętych budynkach. Wspomniane symulacje pożarów wykazują szereg zastosowań. Są z powodzeniem wykorzystywane w śledztwach. Dają możliwość odtworzenia przebiegu zdarzeń i porównania z wynikami oględzin. Umożliwiają zbadanie prototypu budynku pod kątem gwarancji bezpieczeństwa pożarowego. Ułatwiają projektowanie systemów oddymiania. W połączeniu z modelami ewakuacji ludzi stanowią kompleksowy system ułatwiający tworzenie bezpiecznych budowli.

W ostatnich latach powstał szereg programów umożliwiających wizualizację symulacji rozchodzenia ognia. Opracowane dotychczas rozwiązania swoje działanie opierają na metodach numerycznej dynamiki płynów (ang. Computational Fluid Dynamics). Niewątpliwą zaletą numerycznego podejścia jest dokładność wyników. Głównymi wadami jest złożoność obliczeń i stopień komplikacji modelu. Niehomogeniczne automaty komórkowe umożliwiają znaczne uproszczenie modelu. Uproszczenie modelu powoduje z kolei redukcję złożoności obliczeń czyniąc automaty komórkowe szczególnie dogodną metodą w przypadku tworzenia prototypów oraz symulacji czasu rzeczywistego.

## 1.3. Realizacja projektu

Praca została zrealizowana jako wolnostojąca aplikacja komputerowa napisana w języku Java. Do renderowania grafiki trójwymiarowej została użyta biblioteka graficzna Java3D. Aplikacja została przetestowana z wykorzystaniem biblioteki JUnit4.

## 1.4. Struktura pracy

Praca składa się z [ilusi] rozdziałów. W pierwszym rozdziale znajdują się podstawy teoretyczne, związane zarówno z modelowanymi zjawiskami fizycznymi jak i użytym algorytmem. Rozdział Modele symulacji zawiera propozycje zweryfikowanych modeli rozprzestrzeniania się dymu i ognia zaprojektowanych w oparciu o niehomogeniczne automaty komórkowe. Rozdział Implementacja przedstawia

sposób realizacji projektu, napotkane problemy oraz ich rozwiązania. Opisuje możliwości graficznego interfejsu użytkownika oraz sposób korzystania z niego.

## 2. Projekt

Głównym celem projektu jest stworzenie i weryfikacja modelu rozprzestrzeniania się ognia wykorzystując niehomogeniczne automaty komórkowe. Nacisk z pracy został położony na opracowanie algorytmu najdokładniej oddającego rzeczywistość. Aplikacja, nazwa Sparkle została zaprojektowana tak, aby zapewnić użytkownikowi wysoką ergonomię pracy i łatwość nauki. Podczas projektowania i implementacji szczególna uwaga została poświęcona dalszym możliwościom rozbudowy programu.

### 2.1. Główne założenia

- Projekt obejmuje zarówno stworzenie modelu rozprzestrzeniania pożaru jak i uproszczonej wizualizacji oraz graficznego interfejsu użytkownika (GUI).
- Interfejs aplikacji powinien umożliwiać edycję budynku w którym przeprowadzana jest symulacja: dodawanie elementów konstrukcji, określanie materiałów z których zostały stworzone.
- Użytkownik powinien mieć możliwość określenia źródła ognia: zarówno jego miejsca jak i temperatury początkowej.
- Aplikacja powinna umożliwiać także kontrolę nad symulacją: możliwość zatrzymania symulacji, wznowienia, rozpoczęcia od początku, a także dostosowanie tempa symulacji umożliwiającego obserwację zjawisk fizycznych.
- Dodatkowym elementem jest zapis wyników w postaci rozkładu temperatury do pliku, umożliwiający dogłębną analizę rezultatów.
- Wizualizacja powinna obejmować zarówno rozkład temperatury jak i rozprzestrzenianie się dymu.

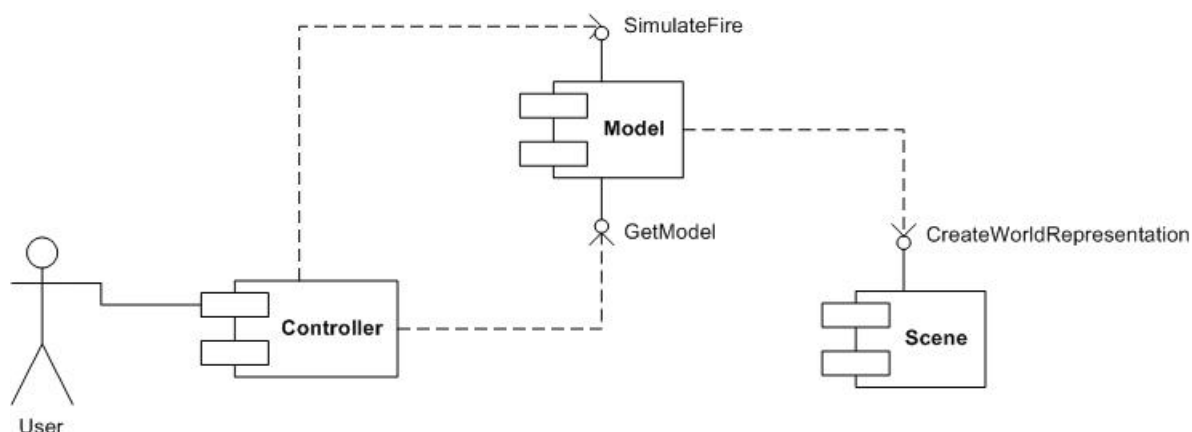
### 2.2. Architektura aplikacji

Aplikacja została podzielona na trzy główne moduły:

- Controller - odpowiada za interakcję z użytkownikiem, dostarcza GUI umożliwiające kontrolę symulacji
- Model - przechowuje model symulacji, realizuje algorytmy rozprzestrzeniania ognia
- Scene - odpowiada za wizualizację wyników

Zależności pomiędzy poszczególnymi komponentami przedstawia diagram 2.1 Zapewnienie bardzo prostych zależności między modułami pozwala niezależnie rozwijać kolejne części aplikacji, w łatwy sposób podmieniać i modyfikować ich zachowanie. Inną zaletą zastosowanego modelu jest łatwość testowania poszczególnych części aplikacji niezależnie.





Rysunek 2.1: Architektura aplikacji

Przedstawiony model powstał na bazie jednego z najpopularniejszych modeli tworzenia aplikacji wykorzystujących graficzny interfejs użytkownika: Model-View-Controller. Elementem różniącym przedstawiony powyżej model od tradycyjnej architektury Model-View-Controller jest rozdzielenie elementów GUI pomiędzy dwa moduły:

- Scene przedstawiającą wyniki aplikacji oraz
- Controller posiadający zestaw narzędzi dostarczających kontrolę użytkownikowi.

Dodatkowo w module Controller zostały połączone elementy widoku wraz z obsługującymi je listenerami.

### 2.2.1. Moduł kontroler

Moduł kontroler odpowiada za komunikację między użytkownikiem a silnikiem aplikacji. Jego podstawowym zadaniem jest dostarczenie łatwego w obsłudze, graficznego interfejsu użytkownika, oraz obsługi akcji użytkownika. Wspomniana obsługa akcji może obejmować zarówno zebranie danych ich przetworzenie i dostarczenie do modelu, pobranie z modelu danych jak i zapis lub kontrolę symulacji. Funkcjonalności dostarczane przez moduł Controller przedstawia diagram ??

### 2.2.2. Widok

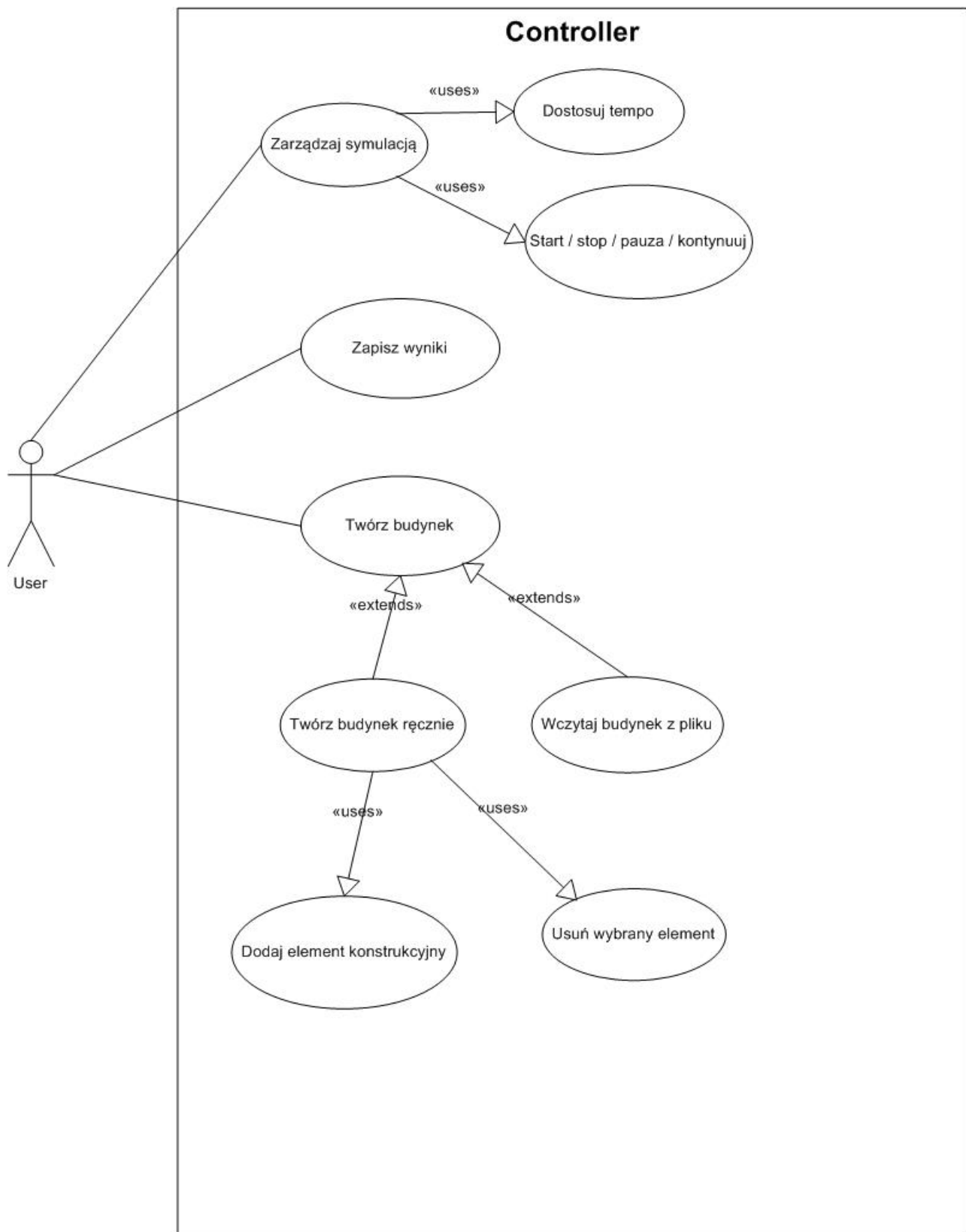
Głównym i w szczególności jedynym zadaniem widoku jest graficzne przedstawienie dostarczonego modelu. Całkowite uniezależnienie widoku od innych modułów umożliwia późniejszą jego modyfikację.

### 2.2.3. Model

Został zastosowany przypadek aktywnego modelu, który potrafi zmieniać swój stan nie tylko w wyniku akcji użytkownika ale także samoczynnie. Aktywność modelu w przypadku symulacji polega na automatycznym uaktualnianiu swojego stanu co pewien, określony okres czasu, a także powiadamianie widoku o zachodzących zmianach.

## 2.3. Moduły

## 2.4. Obiekty



Rysunek 2.2: Przypadki użycia

## **Bibliografia**