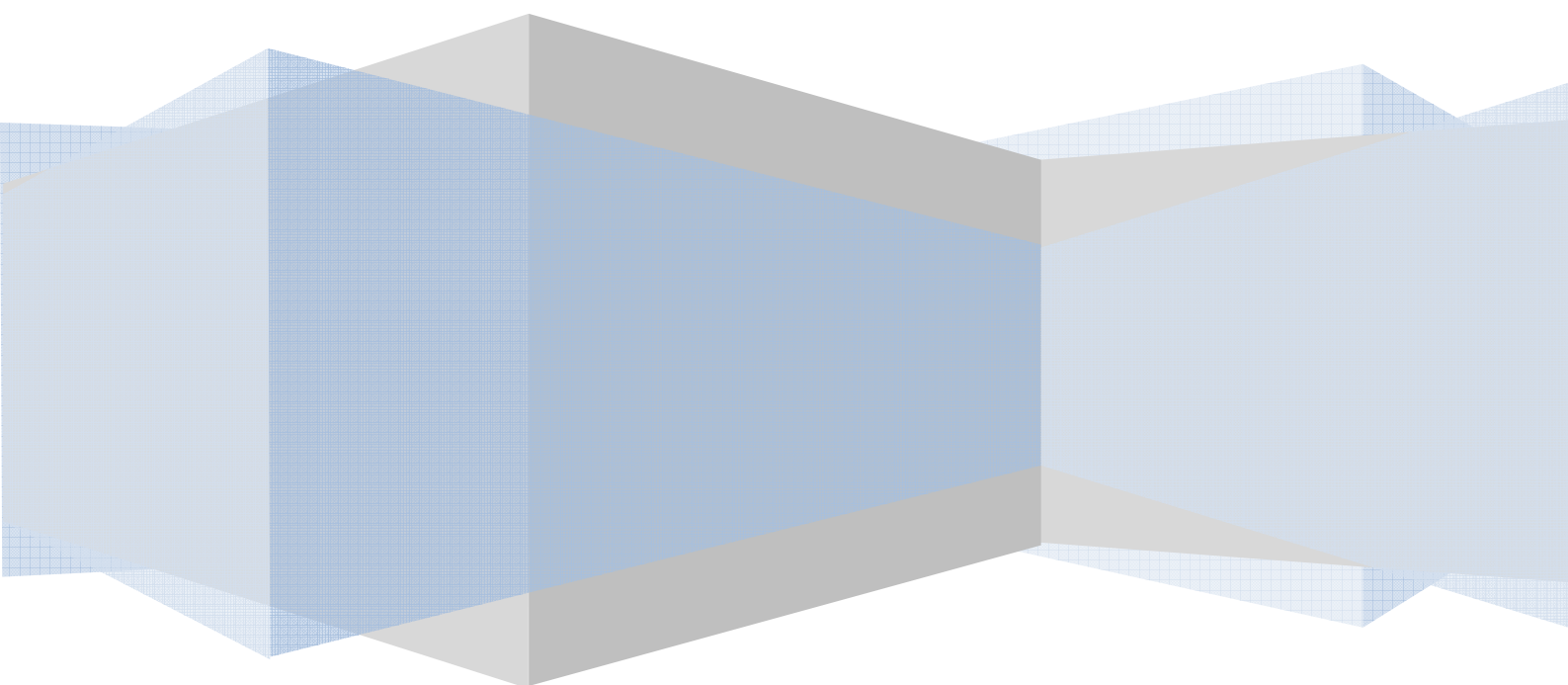


AGH, EAIE, Informatyka

Winda

Systemy czasu rzeczywistego

Mirosław Jedynak, Adam Łączyński



Spis treści

1	Opis problemu	3
1.1	Ilość pięter	3
1.2	Panel zewnętrzny	3
1.3	Panel wewnętrzny	3
2	Diagram przypadków użycia	3
2.1	Aktorzy	4
2.1.1	Potential passenger	4
2.1.2	Passenger	4
2.2	Przypadki użycia	4
2.2.1	Request ride	4
2.2.2	Request down ride	4
2.2.3	Request up ride	5
2.2.4	Request floor	5
2.2.5	Alarm	5
3	Algorytm działania	5
4	Model dziedziny	7
4.1	ExternalKeyPad	7
4.2	InternalKeyPad	7
4.3	Elevator	7
4.4	Elevator Controller	7
4.5	AbstractFactory i FactoryImpl	8
5	Diagram architektury	8
5.1	MainHardware	9
5.2	FloorListener	9
5.3	Elevator Engine	9
6	Sygnały	9
7	Diagram stanów	9
7.1	Panel wewnętrzny	9
7.2	Panel zewnętrzny	10
7.3	Winda	11
8	Diagram sekwencji	11
9	Interfejs graficzny	14
10	Bibliografia	15

1 Opis problemu

Celem naszego projektu jest zamodelowania działania systemu sterowania dwoma windami w biurowcu. Modelowany system powinien działać w taki sposób, aby użytkownik mógł dojechać na jedno z wybranych pięter biurowca. Czas oczekiwania i przejazdu powinien być jak najkrótszy.

System sterowania windami udostępnia funkcjonalność analogiczną do tej, jaką spotykamy w życiu codziennym przy korzystaniu z windy, czyli:

- System udostępnia dwa panele sterowania, jeden wewnątrz windy i drugi na każdym z pięter przed wejściem do wind (panel zewnętrzny).
- Z panelu wewnętrznego jest możliwość wybrania piętra, na które chcemy jechać oraz prośba o otwarcie lub zamknięcie drzwi.
- Z panelu zewnętrznego jest możliwość wysłania żądania z chęcią jechania na piętro niższe lub wyższe.

W realizowanym projekcie przyjęliśmy następujące założenia:

1.1 Ilość pięter

Ilość pięter w budynku wynosi 10 (parter oznacza pierwsze piętro).

1.2 Panel zewnętrzny

Panel zewnętrzny składa się z dwóch przycisków góra i dół, przy czym na piętrze najniższym (1) jest tylko przycisk góra, natomiast na piętrze najwyższym (10) jest tylko przycisk dół.

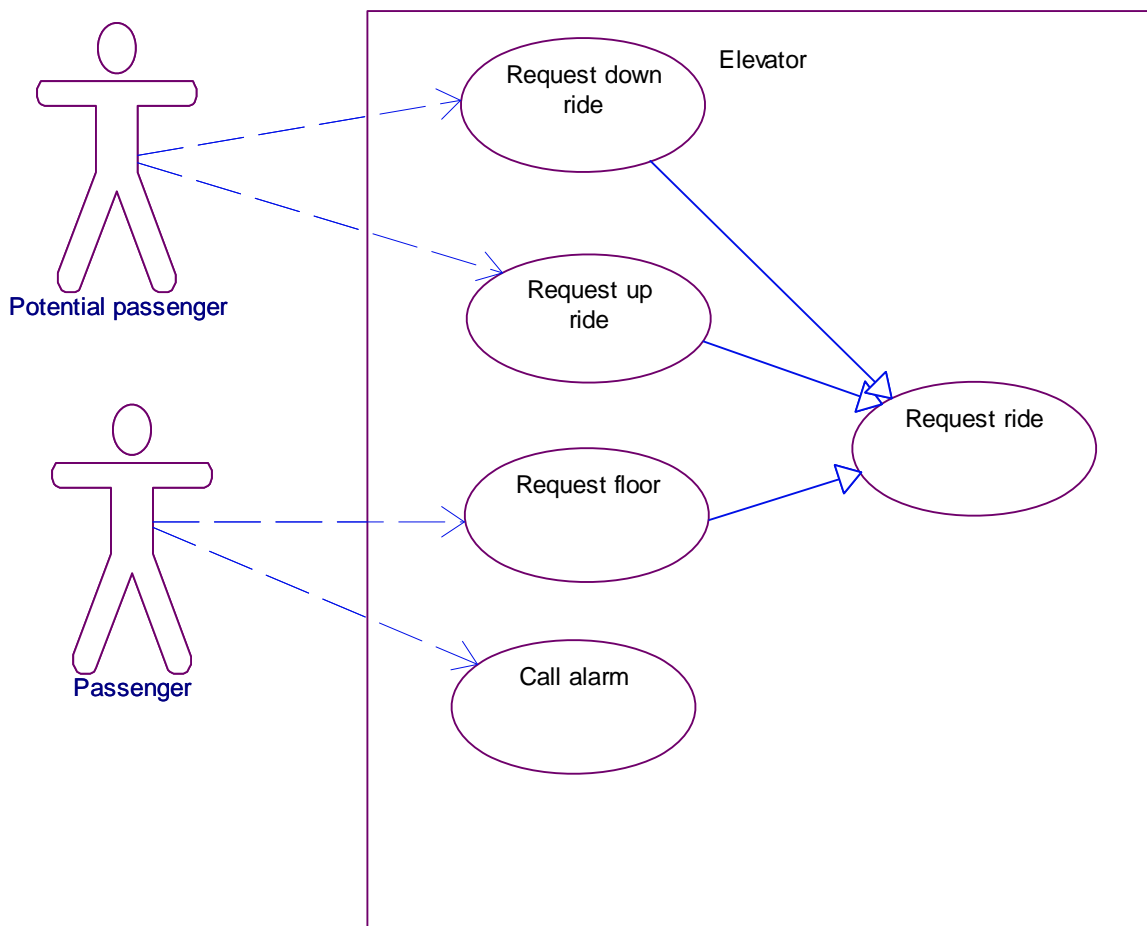
1.3 Panel wewnętrzny

Panel wewnętrzny znajdujący się w każdej z wind składa się z 11 przycisków:

- 10 przycisków z etykietą pietra od 1 do 10;
- 1 przycisk włączenia alarmu.

2 Diagram przypadków użycia

Poniżej (Rysunek 1) został przedstawiony diagram przypadków użycia dla systemu



Rysunek 1 Diagram przypadków użycia

2.1 Aktorzy

2.1.1 Potential passenger

W tego aktora utożsamiany jest osobą stojącą poza windą – na piętrze. Może ona korzystać z zewnętrznego panelu w celu wezwania windy.

2.1.2 Passenger

W momencie wejścia do windy potencjalny pasażer staje się „właściwym” pasażerem, który może korzystać z panelu wewnątrz windy.

2.2 Przypadki użycia

2.2.1 Request ride

Żądanie jazdy windy na określone piętro. Żądanie to jest realizowane na podstawie aktualnego kierunku jazdy oraz zajętości wind. Sposób wybrania piętra i argumentów żądania jest ustalany przez bardziej szczegółowe przypadki użycia.

2.2.2 Request down ride

Osoba znajdująca się na dowolnym (z wyjątkiem parteru) piętrze zgłasza żądanie jazdy na niższe piętro. Wybrana winda jest wysyłana na piętro, z którego przyszło żądanie.

2.2.3 Request up ride

Osoba znajdująca się na dowolnym (z wyjątkiem ostatniego) piętrze zgłasza żądanie jazdy na wyższe piętro. Wybrana winda jest wysyłana na piętro, z którego przyszło żądanie.

2.2.4 Request floor

Osoba znajdująca się wewnątrz windy wybiera, na które piętro chce jechać poprzez naciśnięcie przycisku z numerem piętra w panelu wewnętrznym. Dana winda wysyłana jest na wybrane piętro.

2.2.5 Alarm

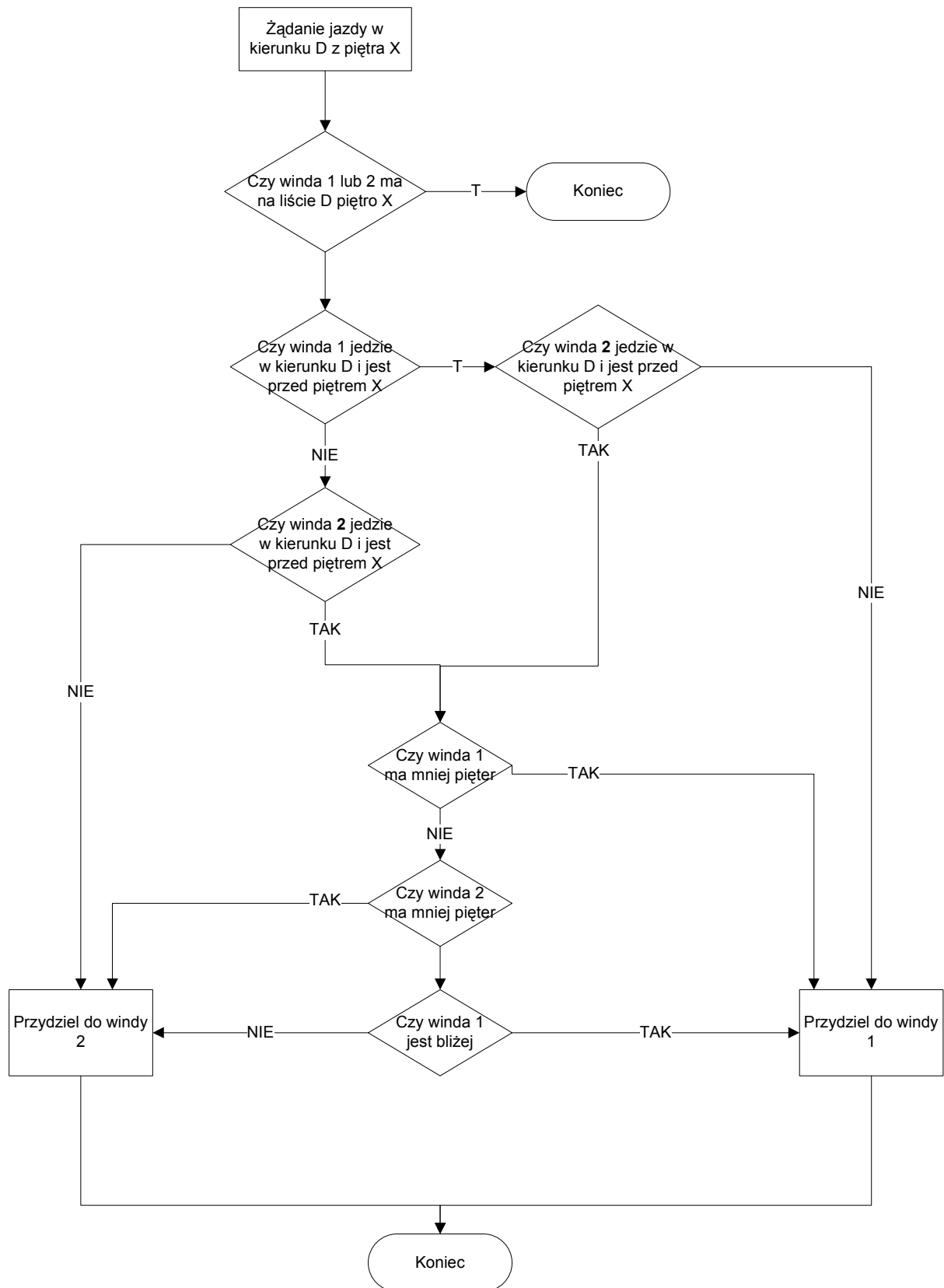
Osoba znajdująca się w windzie żąda uruchomienia alarmu. Powoduje to automatyczne uruchomienie sygnału oraz zatrzymanie się windy na najbliższym piętrze, jeśli winda jest w ruchu.

3 Algorytm działania

Cała logika związana z przydziałem zawarta jest w centralnym kontrolerze. Takie podejście umożliwia łatwiejszą dystrybucję zadań, ponieważ ten komponent ten ma dostęp do informacji na temat obydwóch wind.

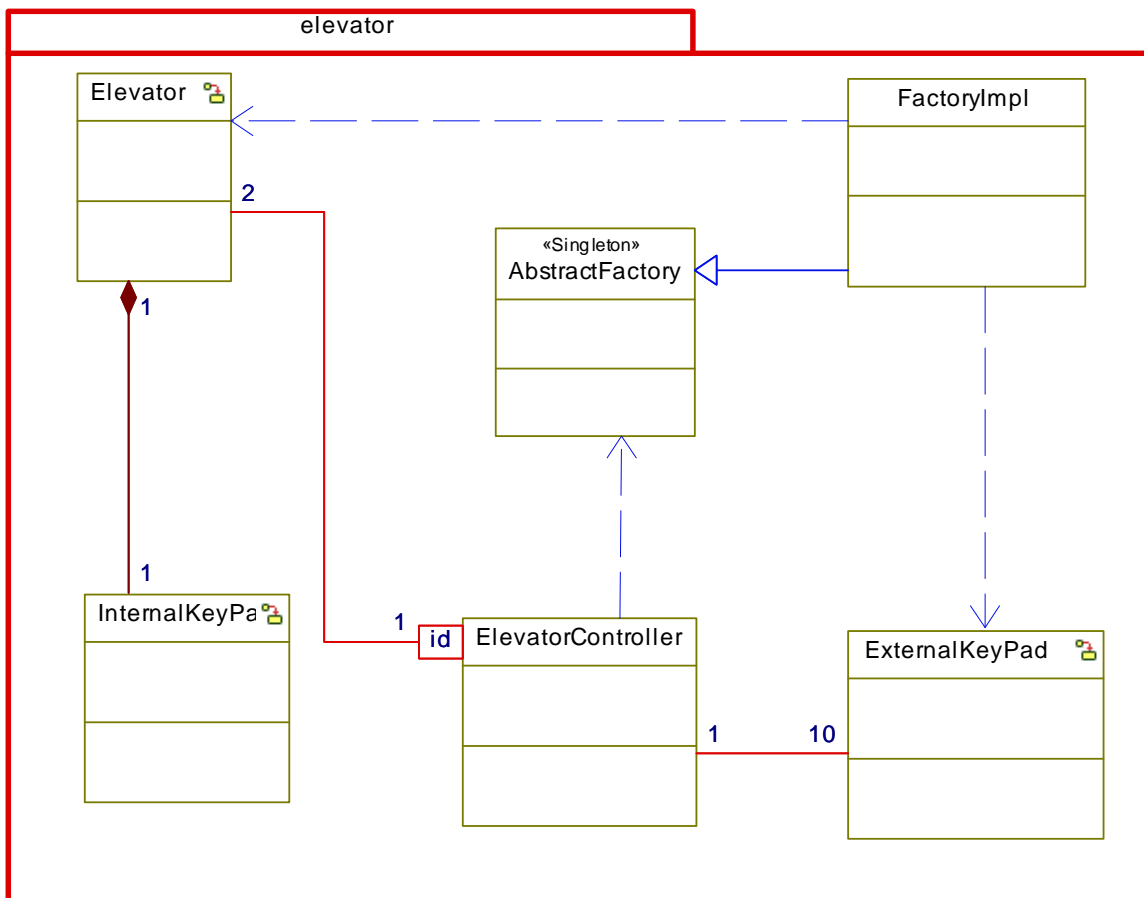
Algorytm przydziału (Rysunek 2) zakłada:

- W pierwszej kolejności przydzielana jest winda, która jedzie w tym samym kierunku, co wysłane żądanie pod warunkiem, że jest przed tym piętrem.
- Jeśli obie windy spełniają (lub obie nie spełniają) wybierana jest winda z mniejszą ilością pięter do zatrzymania (uwzględniając żądania z zewnątrz i wewnątrz) – wynika to z założenia, że czas postoju związany z otwarciem i zamknięciem drzwi jest większy niż czas przejazdu przez dane piętro.
- Jeśli obie windy mają taką samą liczbę pięter wybierana jest winda, która znajduje się bliżej piętra docelowego.



Rysunek 2 Algorytm przydzielenia żądania do windy

4 Model dziedziny



Rysunek 3 Model dziedziny

Klasy modelu (Rysunek 3) znajdują się w pakiecie „elevator” i jest w nich zawarta logika zarządzania windą. W poniższych punktach wymienione zostaną odpowiedzialności tych klas.

4.1 ExternalKeyPad

Reprezentacja panelu do przywoływania windy. Posiada możliwość przywołania windy do jazdy w dół (patrz 2.2.2) oraz w górę (patrz 2.2.3). Na każdym piętrze istnieje osobny panel, przy czym na pierwszym i ostatnim piętrze jest możliwość przywołania do jazdy tylko w jednym kierunku.

4.2 InternalKeyPad

Reprezentacja panelu sterowania wewnątrz windy. Posiada możliwość reagowania na żądanie jazdy na określone piętro oraz włączenia alarmu. Każda winda posiada jeden panel do sterowania.

4.3 Elevator

Reprezentacja windy w modelu. Winda posiada swój stan oraz zapamiętuje listę pięter, na których powinna się zatrzymać. Każda winda posiada referencje do swojego panelu wewnętrznego.

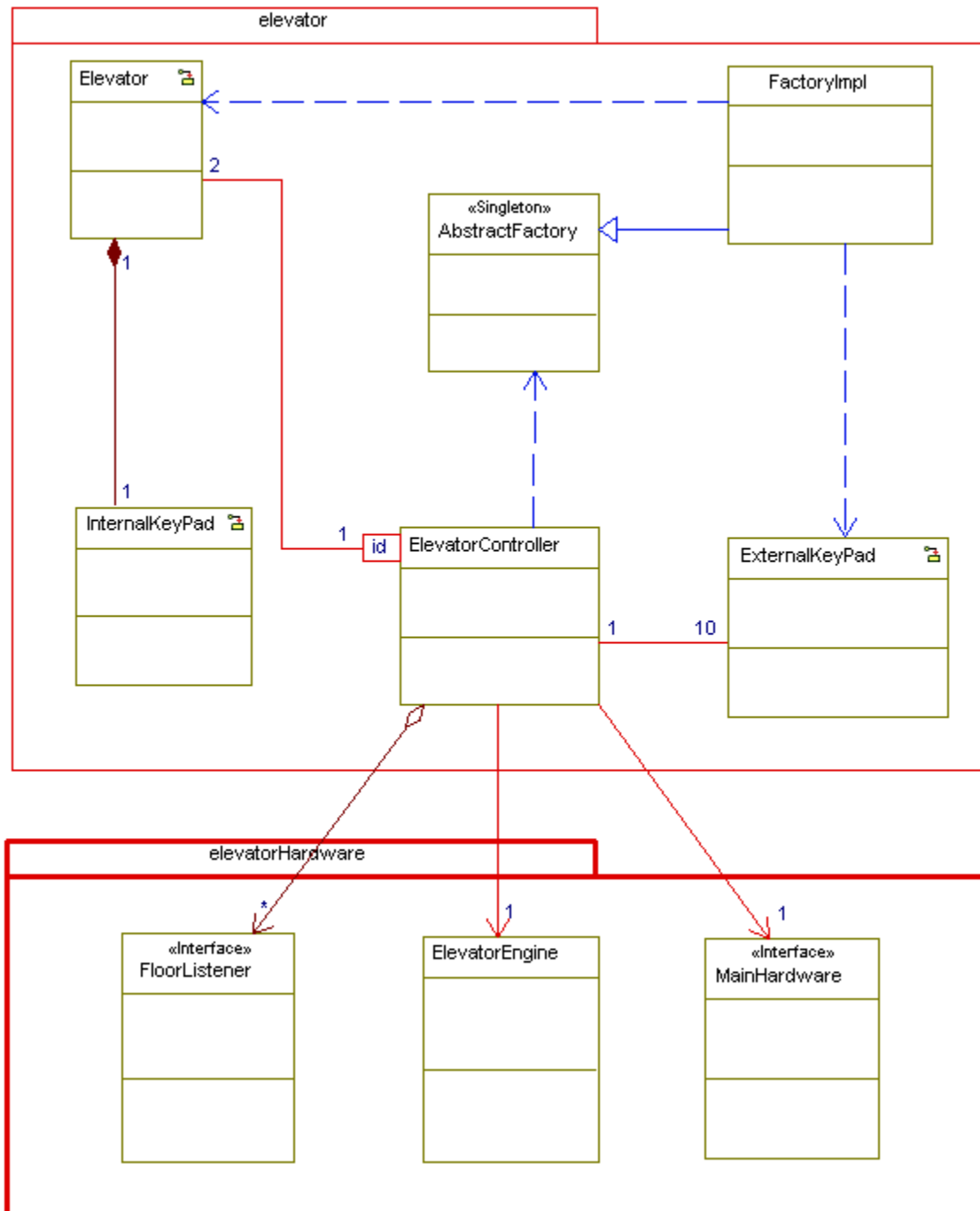
4.4 Elevator Controller

Reprezentacja centralnego modułu odpowiedzialnego za sterowanie windą. Posiada referencję do wszystkich wind, którymi zarządza. Żądania z zewnętrznych paneli (ExternalKeyPad) kierowane do tego modułu i następnie ustalana jest winda odpowiedzialna za wykonanie zadania.

4.5 AbstractFactory i FactoryImpl

AbstractFactory i FactoryImpl są to klasy stworzone w celu uniezależnienia warstwy tworzącej obiekty od samych obiektów. Dzięki takiemu podejściu stosunkowo łatwe jest dodanie klasy dziedziczącej np. z Elevator w celu dodanie nowej logiki zarządzania jazdy windy. Dodanie tej klasy nie wpływa na implementację klasy ElevatorController, która korzysta z fabryki obiektów AbstractFactory.

5 Diagram architektury



Rysunek 4 Model architektury

Model architektury dodaje klasy związane z warstwą integracji ze sprzętem. Elementy tej warstwy zawarte są w pakiecie elevatorHardware, a opisane są w punktach poniżej.

5.1 MainHardware

Interfejs definiujący centralną jednostkę wykonawczą. Udostępnia on metodę umożliwiającą włączenie i wyłączenie sygnału alarmowego.

5.2 FloorListener

Interfejs definiujący jednostki, które chcą być powiadamiane o zmianie piętra przez windę. Definiuje on metody umożliwiające:

- Powiadomienie o zmianie piętra przez konkretną windę
- Powiadomienie o zmianie stanu otwarcia drzwi.

Interfejs ten jest implementowany przez jednostki takie jak panel zewnętrzny i wewnętrzny windy, które wyświetlają aktualne położenie windy.

5.3 Elevator Engine

Klasa ta jest reprezentacją fizycznego silnika windy, który przyjmuje 3 komendy:

- Rozpocznij jazdę w dół
- Rozpocznij jazdę w górę
- Zatrzymaj silnik

6 Sygnały

Poniższa lista przedstawia wykorzystywane sygnały oraz ich zastosowanie:

- evAlarm – sygnalizuje naciśnięcie przycisku alarm w danej windzie
- evDownRequest – sygnalizuje żądanie jazdy w dół z określonego piętra
- evUpRequest – sygnalizuje żądanie jazdy w górę z określonego piętra
- evKeyPressed - sygnalizuje żądanie jazdy na konkretne piętro
- evFloorReached – sygnalizuje dotarcie windy na piętro docelowe
- evRun – sygnalizuje konieczność uruchomienia windy.

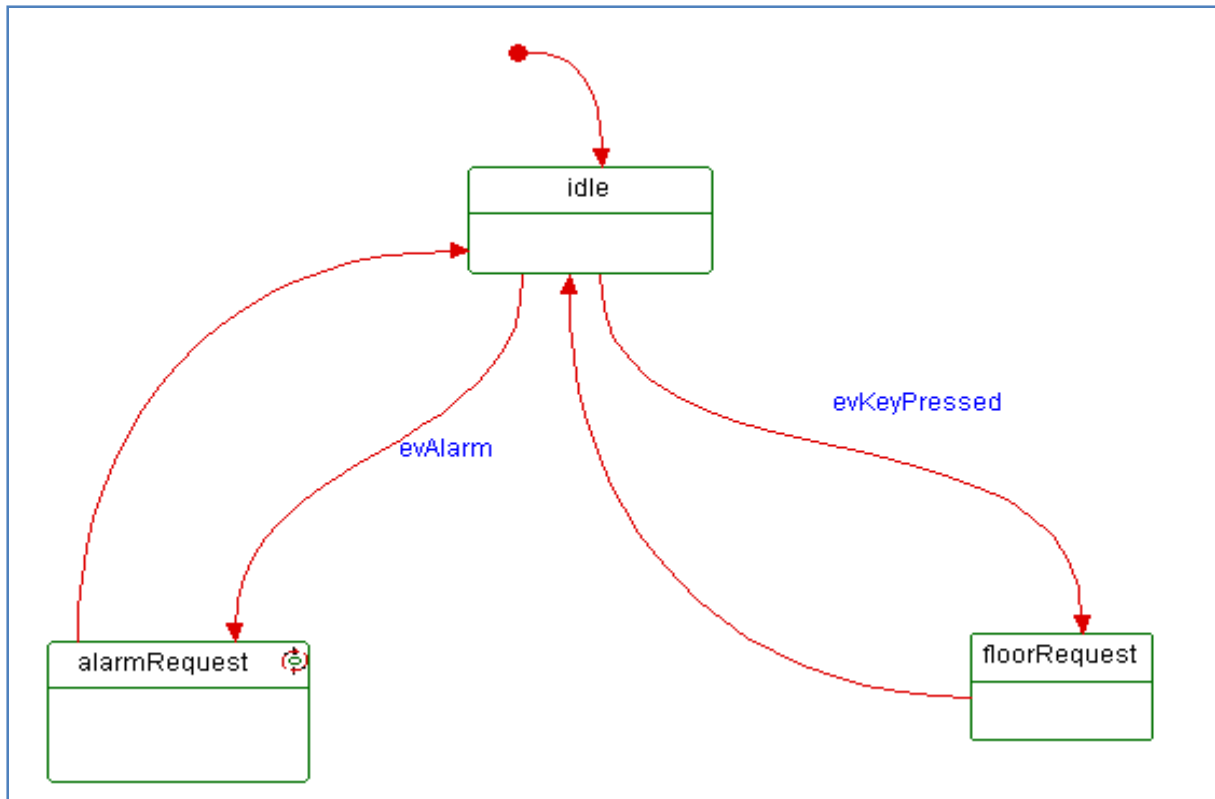
7 Diagram stanów

W poniższym rozdziale zaprezentowane zostały użyte maszyny stanowe. Maszyny stanowe zostały wykorzystane przy budowie następujących komponentów:

7.1 Panel wewnętrzny

Ma za zadanie obsługę zdarzeń związanych z naciskaniem przycisków na panelu na panelu wewnątrz windy(Rysunek 5)

- Idle - stan spoczynku
- alarmRequest– naciśnięty został przycisk alarmu
- floorRequest – naciśnięty został przycisk jazdy na wybrane piętro

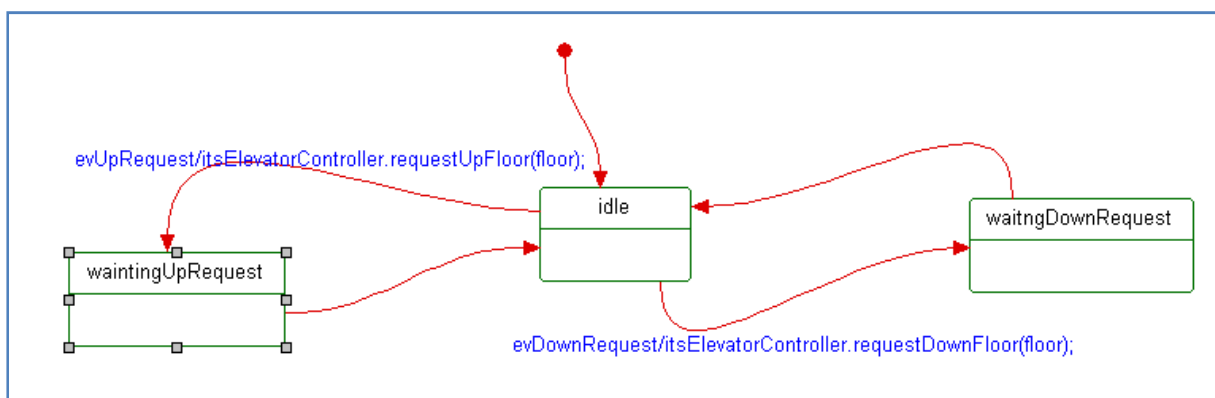


Rysunek 5 Maszyna stanowa dla InternalKeyPad

7.2 Panel zewnętrzny

Ma za zadanie obsługę zdarzeń związanych z naciskaniem przycisków na panelu na zewnątrz windy (Rysunek 6)

- Idle – stan spoczynku
- waitingUpRequest – naciśnięty został przycisk żądania jazdy w górę
- waitingDownRequest – naciśnięty został przycisk żądania jazdy w dół

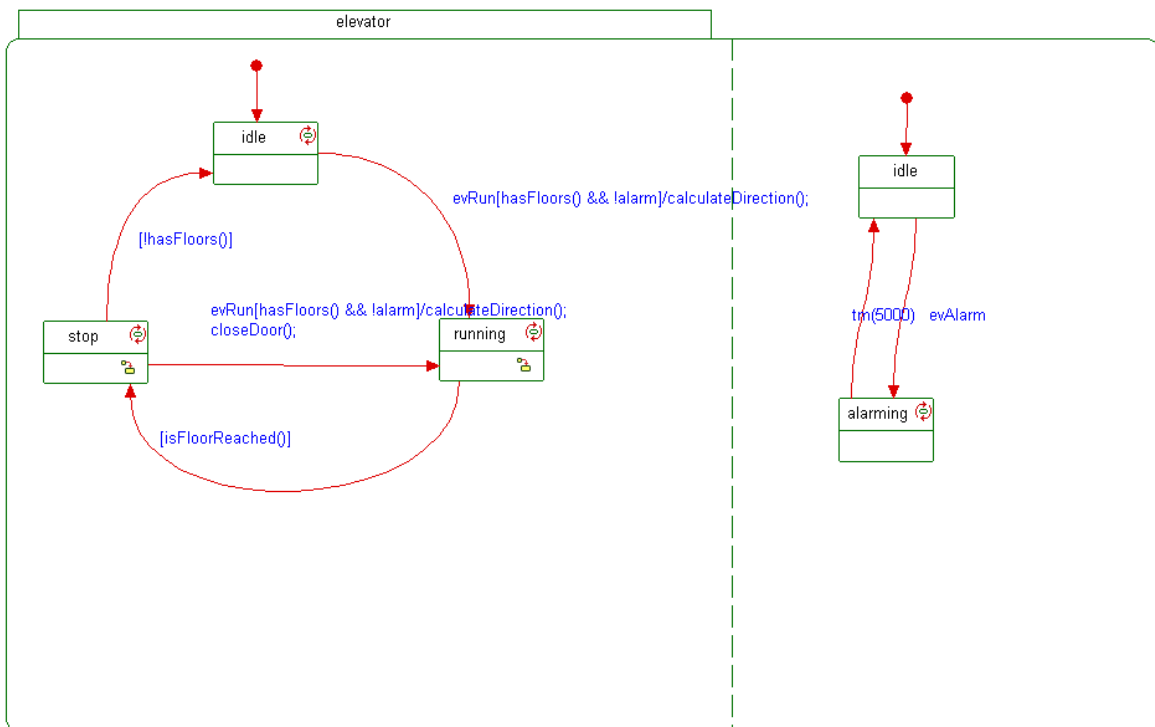


Rysunek 6 Maszyna stanowa dla ExternalKeyPad

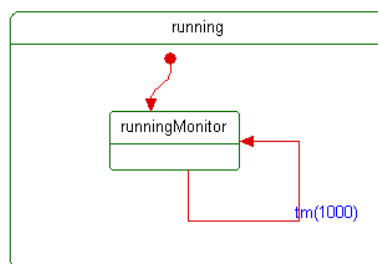
7.3 Winda

Ma za zadanie obsługę przycisku alarm oraz uruchamianie i zatrzymywanie windy(Rysunek 7 i Rysunek 8)

- Idle – stan spoczynku, brak pięter do obsłużenia
- Stop – winda zatrzymała się na piętrze, ale mogą być jeszcze oczekujące piętra do obsłużenia
- Running – winda porusza się



Rysunek 7 Maszyna stanowa dla Elevator



Rysunek 8 Wewnętrzna maszyna stanowa dla Elevator (stan running)

8 Diagram sekwencji

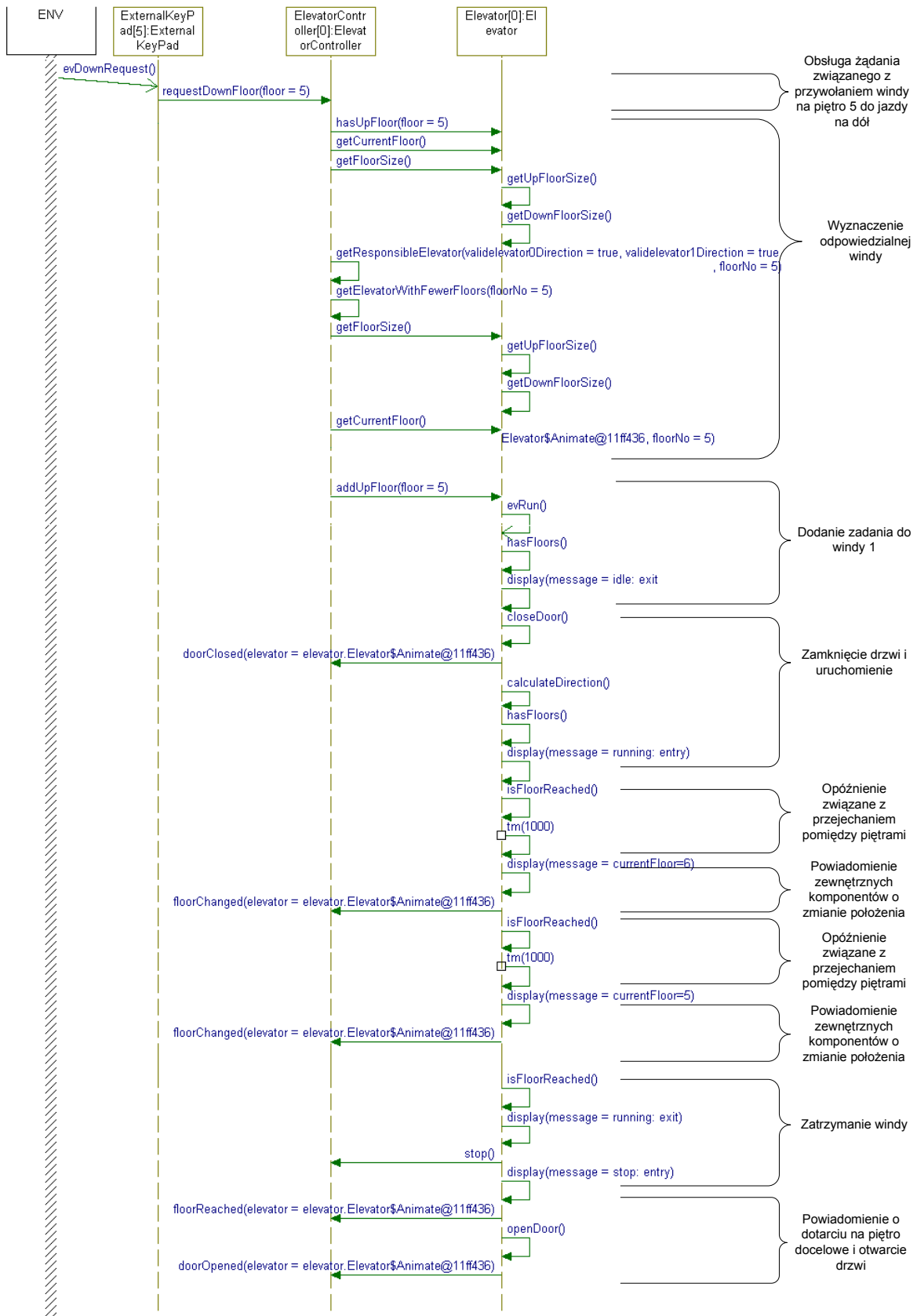
Do definicji sposobu działania systemu zostały użyte diagramy stanu, natomiast diagramy sekwencji wykorzystaliśmy do weryfikacji poprawności działania. Dokładna analiza sekwencji wywołań pozwala

śledzić kolejność wywołań w systemie, pozwalając upewnić się, że ich kolejność jest zgodna z przewidywaniami.

Analizowane były następujące przypadki:

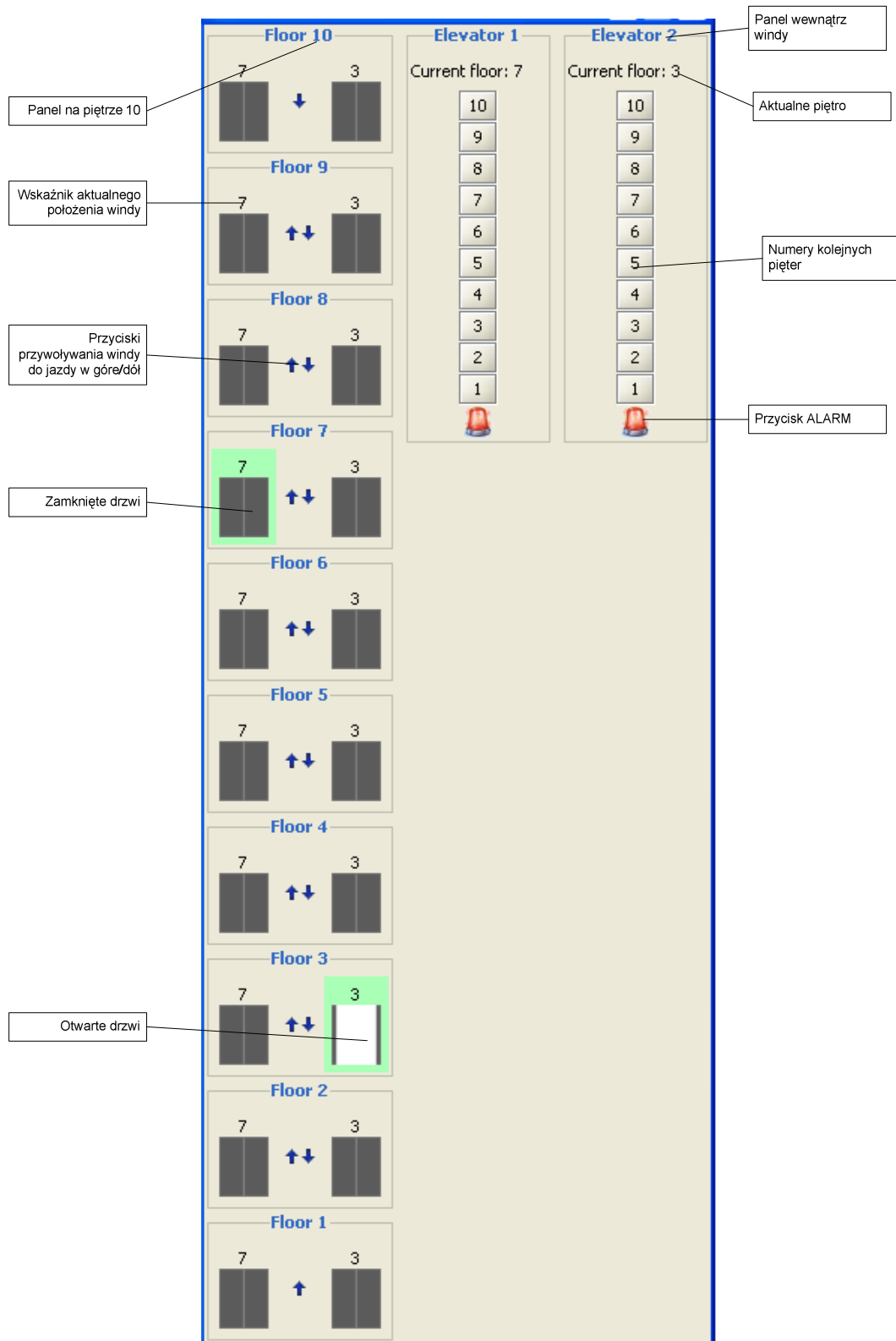
- Przywołanie windy, gdy obie są beczynne
- Przywołanie windy jadącej w tym samym kierunku, co nadchodzące żądanie
- Przywołanie windy, gdy obie jadą w przeciwnych kierunkach niż nadchodzące żądanie
- Żądanie jazdy na wybrane piętro (panel wewnętrzny), gdy winda jedzie we właściwym kierunku
- Żądanie jazdy na wybrane piętro (panel wewnętrzny), gdy winda jedzie w przeciwnym kierunku
- Równoczesne przywoływanie z różnych pięter i dystrybucja zadań między windy
- Uruchomienie alarmu powodujące zatrzymanie windy

Przykładową analiza zachowania systemu w momencie żądania jazdy w dół wysłanego z piętra 6 przedstawia Rysunek 9. W momencie wysłania żądania winda 1 znajduje się na piętrze 8 a winda 2 na 10. Przewidywane zachowanie systemu to wezwanie windy 1 (znajduje się bliżej) i jej przejazd na piętro 8, gdzie zostaną otwarte drzwi. W czasie przejazdu powinna być aktualizowana informacja o zmianie piętra.



Rysunek 9 Diagram sekwencji przedstawiający przywołanie windy

9 Interfejs graficzny



Rysunek 10 Interfejs graficzny

10 Bibliografia

- [1] Telelogic Rhapsody „Getting Started Guide”
- [2] Telelogic Rhapsody „Java Tutorial”
- [3] Telelogic Rhapsody „User Guide”
- [4] Jim Rumbaugh , “Using UML for Modeling Complex Real-Time Systems”