

Lift verification

Sakevych Ruslan

Постановка задачі

Формалізація та верифікація алгоритму керування ліфтом.

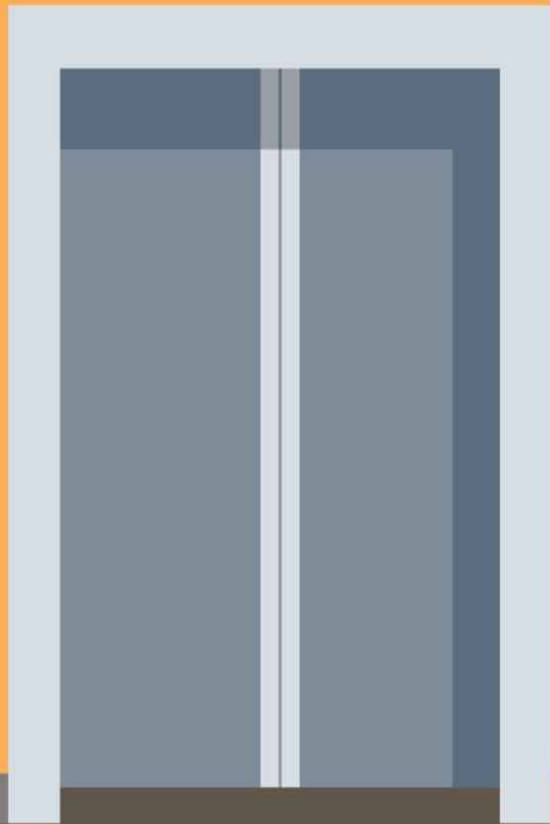
Ліфт не є багатозадачним, іншими словами він не відслідковує натискання кнопки під час того як він везе інших пасажирів (або його вже визвали).

Ми також розглядаємо можливість переходу ліфта в аварійний стан (наприклад при виникненні пожежі). В цьому стані ліфт має за першої можливості зупинитися і відкрити двері, дозволивши пасажирам вийти.

1



7



B method

Це набір математичних технологій для специфікації, проектування та реалізації компонент програмного забезпечення. Системи моделюються як сукупності незалежних Абстрактних Машин, для яких на всіх стадіях розробки застосовується об'єктно-орієнтований підхід.

Абстрактна Машина описується з використанням Abstract Machine Notation (AMN). Стандартна нотація використовується на всіх рівнях опису, від специфікації до реалізації.

Abstract Machine Notation

AMN — мова формальної специфікації, що базується на станах. Вона вийшла з тієї ж школи, що і VDM та Z. Абстрактна машина включає стан разом з операціями на тому стані. Ми оперуємо такими поняттями як множина, відношення, функція, послідовність та подібних. Оператори моделюються з використанням перед- та післяумов.

У реалізації абстрактної машини стан знову моделюється з використанням теоретико-множинної моделі, але цього разу ми вже маємо реалізацію цієї моделі.

Atelier B

Мова програмування: Event-B

- Автоматичне уточнення
- Синтаксичний аналізатор
- Перевірка типів
- Генерація вихідного коду мовою C
- Графічне відображення
- Автоматичне доведення корекції



SETS

DOORS_STATE = {OPEN, CLOSE};

MOVE_STATE = {UP, DOWN, STOP};

BUTTON_STATE = {BUTTON_UP, NONE, BUTTON_DOWN};

OCCUPANCY_STATE = {FREE, FULL};

EMERGENCY_STATE = {YES, NO}

VARIABLES

DoorsState,

MoveState,

ButtonState,

OccupancyState,

EmergencyState

INVARIANT

DoorsState : DOORS_STATE &

MoveState : MOVE_STATE &

ButtonState : BUTTON_STATE &

OccupancyState : OCCUPANCY_STATE &

EmergencyState : EMERGENCY_STATE &

INVARIANT

<code>(DoorsState = OPEN)</code>	<code>=> (MoveState = STOP)</code>
<code>(MoveState = STOP)</code>	<code>=> (ButtonState = NONE)</code>
<code>(EmergencyState = YES)</code>	<code>=> (DoorsState = OPEN)</code>
<code>(OccupancyState = FULL & DoorsState = CLOSE)</code>	<code>=> (MoveState /= STOP)</code>

INITIALISATION

DoorsState := CLOSE ||

MoveState := STOP ||

ButtonState := NONE ||

OccupancyState := FREE ||

EmergencyState := NO

OPERATIONS

Lift_arrive = **PRE**

MoveState = DOWN & DoorsState = CLOSE

THEN

MoveState := STOP ||

DoorsState := OPEN ||

ButtonState := NONE

END;

OPERATIONS

Lift_call_free_lift_up = **PRE**

MoveState = STOP & OccupancyState = FREE &

ButtonState = BUTTON_UP & DoorsState = CLOSE

THEN

MoveState := UP

END;

OPERATIONS

Lift_call_free_lift_down = **PRE**

MoveState = STOP & OccupancyState = FREE &

ButtonState = BUTTON_DOWN & DoorsState = CLOSE

THEN

MoveState := DOWN

END;

OPERATIONS

Lift_emergency = **PRE**

EmergencyState = NO

THEN

EmergencyState := YES ||

MoveState := STOP ||

DoorsState := OPEN ||

ButtonState := NONE

END;

- local
 - Elevator (OK|OK|-|10|0|100%)
 - Components
 - Lift
 - Definitions
 - Libraries
 - source WD lemmas








Top-Bottom Graphical view ▼

Top-Bottom Graphical view ▼



Clear

☐ Hide Finished tasks

Project	Component	Action	Status	Messages	Server
Elevator	Lift		Finished	B0 check finished	localhost
Elevator	Lift		Finished	End of Proof	localhost
Elevator	Lift		Finished	End of Proof	localhost
Elevator	Lift		Finished	Nothing to prove in Lift	localhost
Elevator	Lift		Finished	Nothing to prove in Lift	localhost
Elevator	Lift		Finished	Nothing to prove in Lift	localhost
Elevator	Lift		Finished	Nothing to prove in Lift	localhost



☒ Errors (0)
 ☒ Warnings (0)

☐ Multi-Line messages

Message	Location	Component

No errors

9:58 AM
4/14/2019