

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики

Кафедра теорії та технології програмування



**Звіт**

з дисципліни

«Формальні методи розробки програмних систем»

на тему

**«СПЕЦИФІКАЦІЇ СИСТЕМИ РОБОТИ ПРАЛЬНОЇ МАШИНИ  
НА ПРИКЛАДІ SIEMENS iQ800»**

Виконала

студентка 1-го курсу магістратури

спеціальність «Інформатика»

Київ — 2019

## ЗМІСТ

РОЗДІЛ 1. Опис методу	3
1.1 В-метод	3
1.2 Основні компоненти	3
1.3 В (мова програмування)	4
1.4 Програмне забезпечення	5
РОЗДІЛ 2. Опис системи «Пральна машина» на прикладі Siemens iQ800	8
2.1 Стани системи «Пральна машина»	8
2.2 UML-діаграми	10
РОЗДІЛ 3. Верифікація за допомогою Atelier V	13
3.1 Код загального опису пральної машини	13
3.2 Код імплементації пральної машини	20
3.3 Верифікація за допомогою системи Atelier V	27
ВИСНОВКИ	28
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	29

## **РОЗДІЛ 1. Опис методу**

### **1.1 В-метод**

**Метод В** — це метод розробки програмного забезпечення на основі мови В, формальний метод, що підтримується інструментами, заснований на абстрактній машинній нотації, використовуваної при розробці комп'ютерного програмного забезпечення. Спочатку вона була розроблена в 1980-х роках Жаном-Раймоном Абріалом [1] у Франції та Великобританії. Метод В пов'язаний з Z-позначенням (також заснованим Абріалом) і підтримує розробку коду мови програмування зі специфікаціями. Метод В використовувався у найважливіших системах безпеки в Європі (наприклад, автоматичні лінії Paris Metro 14 і 1 і ракета Ariane 5). Він має надійну, комерційно доступну підтримку інструменту для специфікації, дизайну, доказування та генерації коду. [3]

У порівнянні з Z, мова В має більш низький рівень і більш орієнтована на уточнення коду, а не просто на формальну специфікацію — отже, легше правильно реалізувати специфікацію, написану на В, ніж одну в Z. Ця ж мова використовується в специфікації, розробці та програмуванні. Механізми включають інкапсуляцію та локалізацію даних. [2]

### **1.2 Основні компоненти**

Нотація мови В залежить від теорії множин і логіки першого порядку, щоб визначити різні версії програмного забезпечення, що охоплює повний цикл розробки проекту. [2]

#### **Анотація машини**

У першій і найбільш абстрактній версії, яка називається «абстрактною машиною» потрібно вказати мету системи. [2]

#### **Уточнення**

- Потім, під час етапу доопрацювання, можна додати специфікацію, щоб уточнити мету або перетворити абстрактну машину на більш конкретну, додавши деталі про структури даних і алгоритми, які визначають, яким чином досягнута мета. [2]
- Нова версія, яка називається «Уточнення», повинна бути доведена, щоб бути когерентною і включати всі властивості абстрактної машини. [2]
- Конструктор може використовувати бібліотеки мови В для моделювання структур даних або для включення або імпортування існуючих компонентів. [2]

### **Реалізація**

- Уточнення продовжується, поки не буде досягнута детермінована версія: «Реалізація». [2]
- Під час всіх кроків розвитку використовується одна і та ж нотація, і остання версія може бути перекладена на мову програмування для компіляції. [2]

## **1.3 В (мова програмування)**

**В (Бі)** — інтерпретована мова програмування, розроблена в AT&T Bell Telephone Laboratories. Вона є нащадком мови BCPL та попередником C (Сі). Мова В була розроблена Кеном Томпсоном та Деннісом Рітчі і вперше презентована в 1969 році.

Кен Томпсон та Денніс Рітчі розробили Бі ґрунтуючись головним чином на мові BCPL, яку Томпсон використовував в проєкті Multics. По суті, Бі була версією BCPL, позбавленою деяких компонентів, без яких, як вважав Томпсон, можна не виходити за об'єм пам'яті, встановлений на мінікомп'ютерах того часу. Перехід від BCPL до Бі також містив зміни у напрямку зменшення кількості непробільних символів в типовій програмі. Велика частина типового ALGOL-подібного синтаксису BCPL була вагомо змінена протягом цього процесу. Наприклад, оператор `:=` був замінений на `=`, `,` `=` для порівняння тексту був

замінений на `==`. Пізніше оператори `&` та `|` в BCPL було змінено на `&&` та `||` відповідно, при переході до мови програмування, яка зараз відома як C (Cі). Оператор присвоювання знову став `=`.

Томпсон винайшов арифметичні оператори присвоювання для Бі, використовуючи `x =+ y`, щоб додати «у» до «х» (в мові програмування C (Cі) оператор пишеться `+=`). Бі також представила оператори інкременту та декременту (`++` та `--`). Їх префіксне або постфіксне положення визначає, чи приймаються значення до або після зміни операнда. Ці нововведення були запроваджені не в перших версіях Бі. Дехто вважає, що вони були створені для автоінкрементуючих та автодекрементуючих адресних типів DEC PDP-11, хоча це історично неможливо, оскільки ніякого PDP-11 в той час не було.

В Бі не існує типів даних, або, точніше, він лише один: комп'ютерне слово. Більшість операторів (наприклад, `«.», «+», «-», «*», «/»`) обробляли його як ціле число, а інші як посилання для розіменування посилання. У багатьох інших напрямках вона виглядала, як рання версія C (Cі). Існує кілька бібліотечних функцій, в тому числі ті, які віддалено нагадують функції з бібліотеки вводу-виводу в C (Cі).

Бі продовжують використовувати на GCOS мейнфреймах і на деяких вбудованих системах, зважаючи на ряд причин, у тому числі через обмежену апаратну частину в невеликих системах; великі бібліотеки, інструменти, питання вартості ліцензій; та просто через достатньо хорошу роботу.<sup>[6]</sup> Вельми впливова AberMUD була спочатку написана на Бі. Зараз Бі майже зникла через витіснення її наступником — мовою C (Cі).

## 1.4 Програмне забезпечення

### B-Toolkit

B-Toolkit [4], розроблений Іб Холмом Серенсенем, являє собою набір інструментів програмування, призначених для підтримки використання B-Tool,

математичного інтерпретатора теорії множин, для цілей формальної методології інженерії програмного забезпечення, відомий як метод В. [2]

Інструментарій використовує користувальницький інтерфейс X Window Motif для керування графічним інтерфейсом і виконується в основному на операційних системах Linux, Mac OS X і Solaris. Вона була розроблена британською компанією B-Core (UK) Limited. [2]

## **Atelier B**

Розроблений компанією ClearSy, Atelier B є промисловим інструментом, який дозволяє оперативно використовувати метод В для розробки програмного забезпечення, яке не має дефектів (формальне програмне забезпечення). У цілісному середовищі він забезпечує безліч функцій для управління проектами на мові В. Ці функції можуть бути розділені на три категорії:

- підтримка доказу, щоб продемонструвати доказ зобов'язання, використовуючи відповідні інструменти
- підтримка розвитку: автоматичне керування залежності між В компонентами,
- інструменти для покращення використання системи користувачем: графічне представлення проектів, відображення стану проекту і статистики, архівування проекту.

Atelier B використовується або через інтерфейс людинамашина в QT форматі або за допомогою команди безпосередньо (командний режим). Atelier B багатокористувацький. Завдання, які можуть бути автоматизовані при розробці проекту:

- синтаксична перевірка компонентів
- автоматична генерація обов'язкового доведення
- автоматична трансляція інсталяції В в С або Ada мови

Доступні дві версії: Community Edition, доступна для всіх без обмежень, видання Maintenance Edition для власників контрактів на технічне обслуговування. [2]

Вона використовується для розробки автоматизмів безпеки для різних метрополітенів, встановлених в усьому світі компанією Alstom і Siemens, а також для сертифікації Common Criteria і розробки моделей систем ATMEL і STMicroelectronics. [2]

## **РОЗДІЛ 2. Опис системи «Пральна машина» на прикладі Siemens iQ800**

У даній роботі було формалізовано роботу системи «Пральна машина». Це електропобутова машина для прання речей. Дана машина працює, якщо вона підключена до електропостачання та подачі води, додано миючий засіб — пральний порошок, кондиціонер для пом'якшення білизни та обрана програма/режим для прання. Якщо немає електропостачання, подачі води, не додано пральний порошок або відкриті дверці люку/барабану — машина не працює.

### **2.1 Стани системи «Пральна машина»**

#### ***Стани пральної машини:***

- працює (машина пере речі)
- стоїть на паузі
- не працює (одна із умов не виконується)

#### ***Електропостачання:***

- є електропостачання (пральну машину підключено до електромережі)
- немає електропостачання

#### ***Подача води:***

- є подача води (пральну машину підключено до водопостачання)
- немає подачі води або пральну машину не підключено до водопостачання

#### ***Миючий засіб:***

- додано пральний порошок або рідкій пральний засіб
- немає (не додали)

#### ***Кондиціонер для пом'якшення білизни:***



- додано кондиціонер
- немає (не додали)

***Порошок для пом'якшення води:***

- додано
- немає (не додали)

***Дверці люку/барабану:***

- відкриті
- зачинені

***Вибір користувача:***

- вибір режиму/програми прання
- вибір температури прання
- вибір типу плям (кров, вино, ягоди, трава, кава, чай...)
- вибір режиму полоскання (з додатковим чи без)
- вибір кількості обертів віджимання наприкінці

***Результат прання:***

- білизна поправа
- сталася помилка

***Для забезпечення безпечної та коректної роботи системи необхідно, щоб виконувалися наступні умови:***

- ➡ Якщо немає електропостачання — пральна машина не працює
- ➡ Якщо немає водопостачання — пральна машина не працює
- ➡ Якщо немає миючого засобу — пральна машина не працює
- ➡ Якщо користувач не обрав режим/програму прання — пральна машина не працює і подає звуковий сигнал

- ➔ Якщо користувач обрав режим програми ополаскування білизни з кондиціонером, а засіб для ополаскування відсутній у відсіку пральної машини — пральна машина не буде працювати
- ➔ Якщо дверцята люку/барабану не зачинені — пральна машина не працює та подає звуковий сигнал
- ➔ Якщо усі вищенаведені умови було дотримано — пральна машина пере білизну
- ➔ Якщо під час ополаскування припинилося водопостачання — пральна машина зупиняється, не працює та подає звуковий сигнал на табло про відсутність води

## 2.1 UML-діаграми

Розробка програмного забезпечення проходила відповідно до наступної V-моделі (Рисунок 2.1.1):

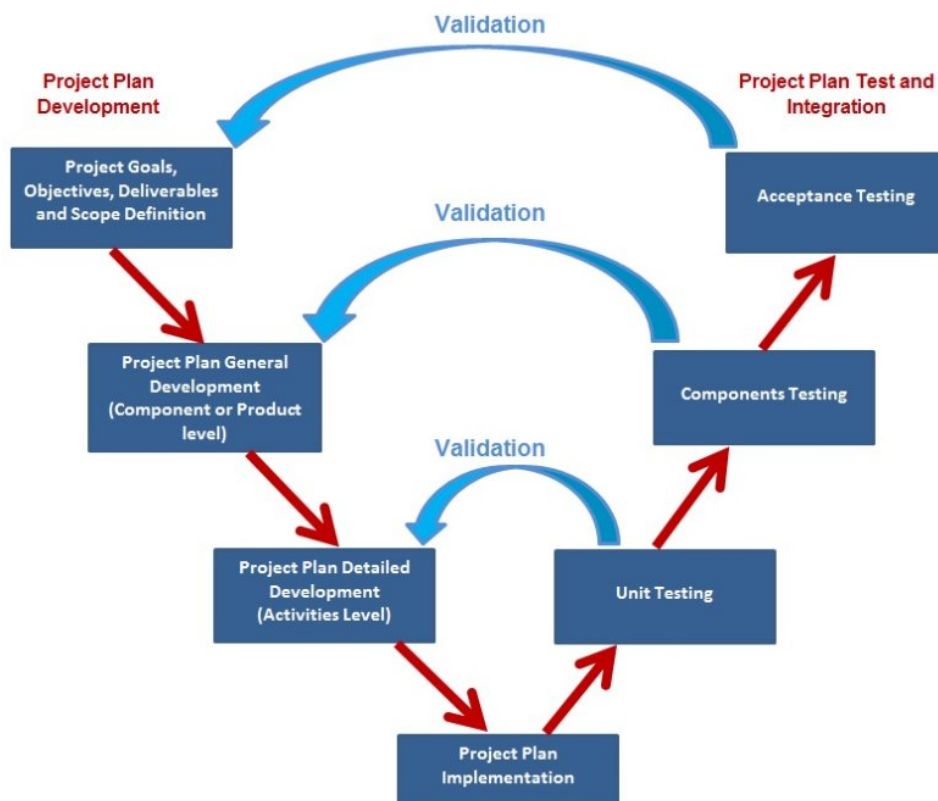


Рисунок 2.1.1 — V-модель розробки програмного забезпечення

На першому етапі було визначено концепцію майбутньої системи пральної машини, після чого була створена загальна архітектура проекту за допомогою UML-діаграм. При детальному проектуванні розширено та уточнено деякі аспекти системи. Нижче наведені діаграми роботи системи.

UML-діаграма роботи пральної машини (Рисунок 2.1.2):

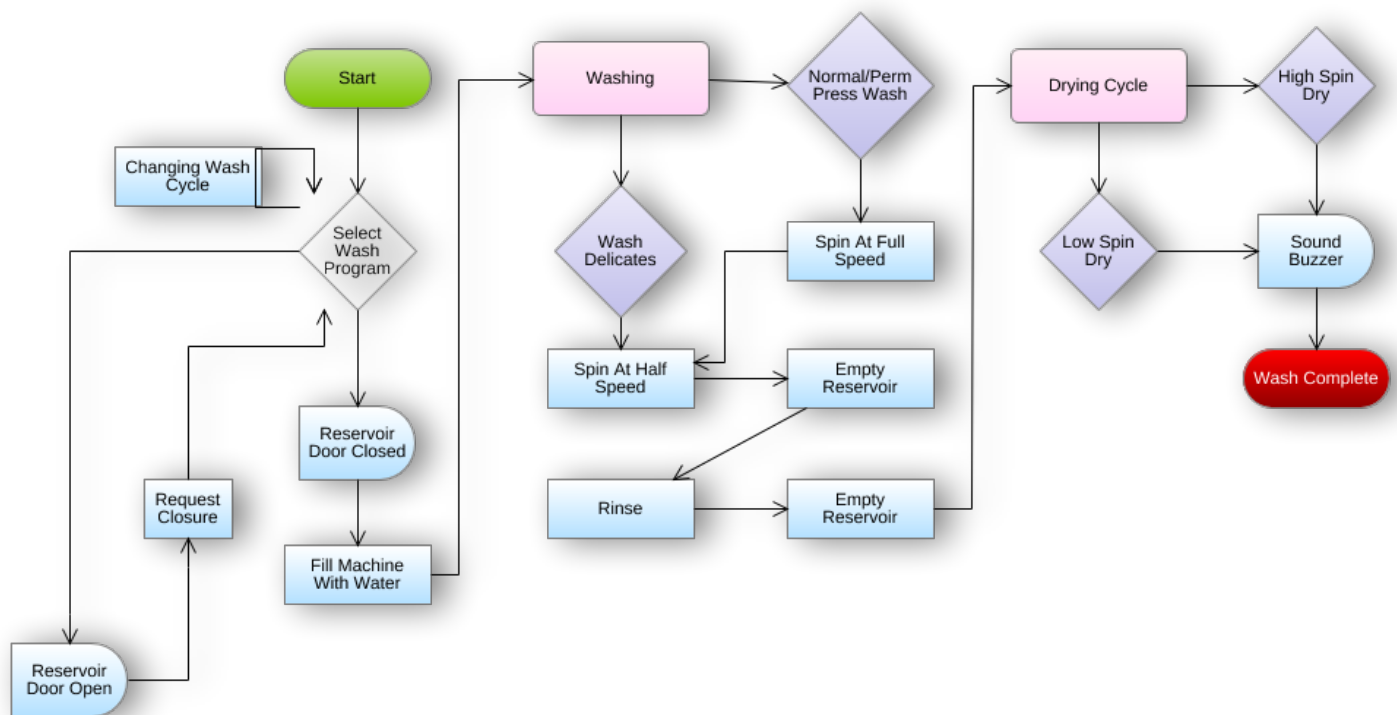


Рисунок 2.1.2 — UML-діаграма роботи пральної машини

Use-case діаграма пральної машини (Рисунок 2.1.3):

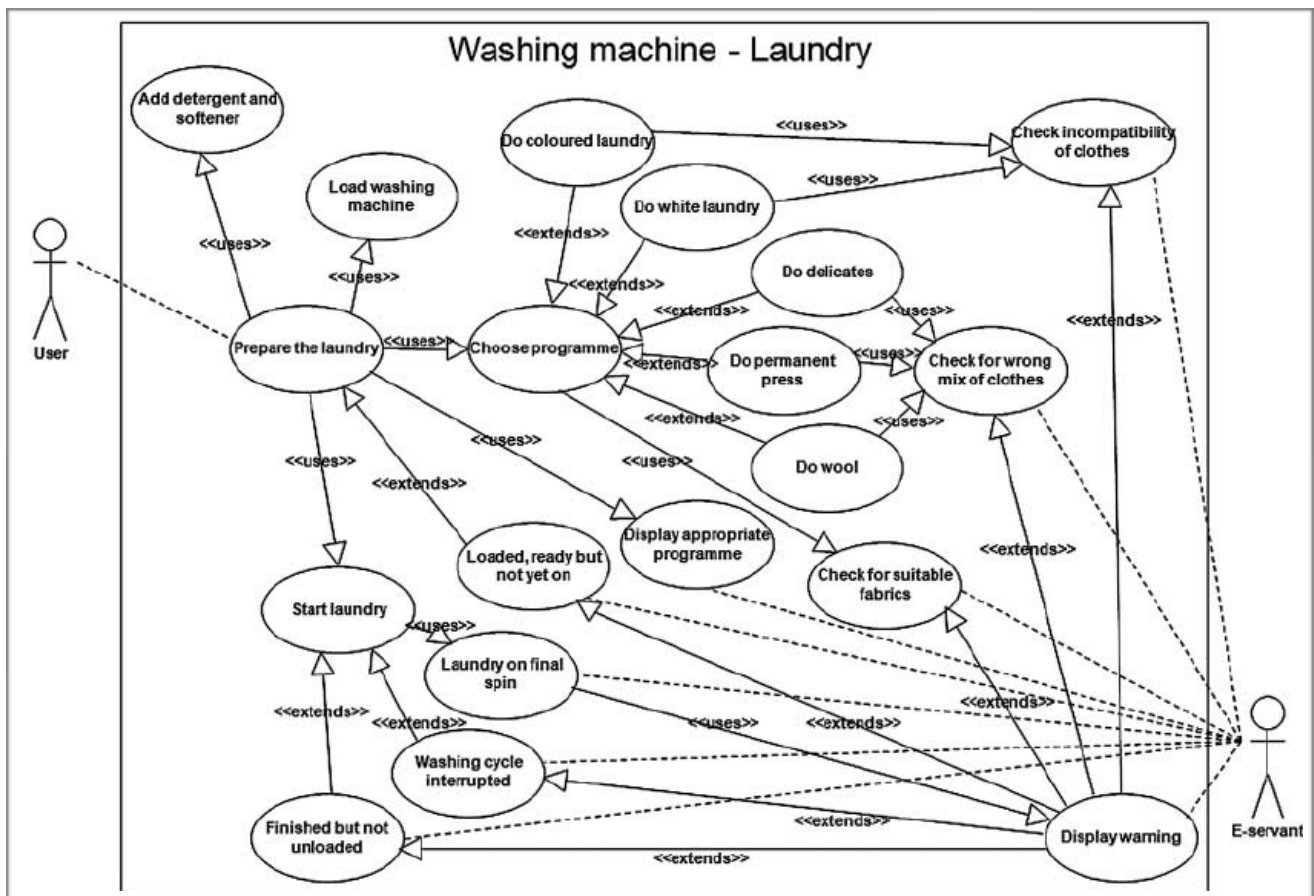


Рисунок 2.1.3 — Use-case діаграма пральної машини

Діаграма класів пральної машини (Рисунок 2.1.4):

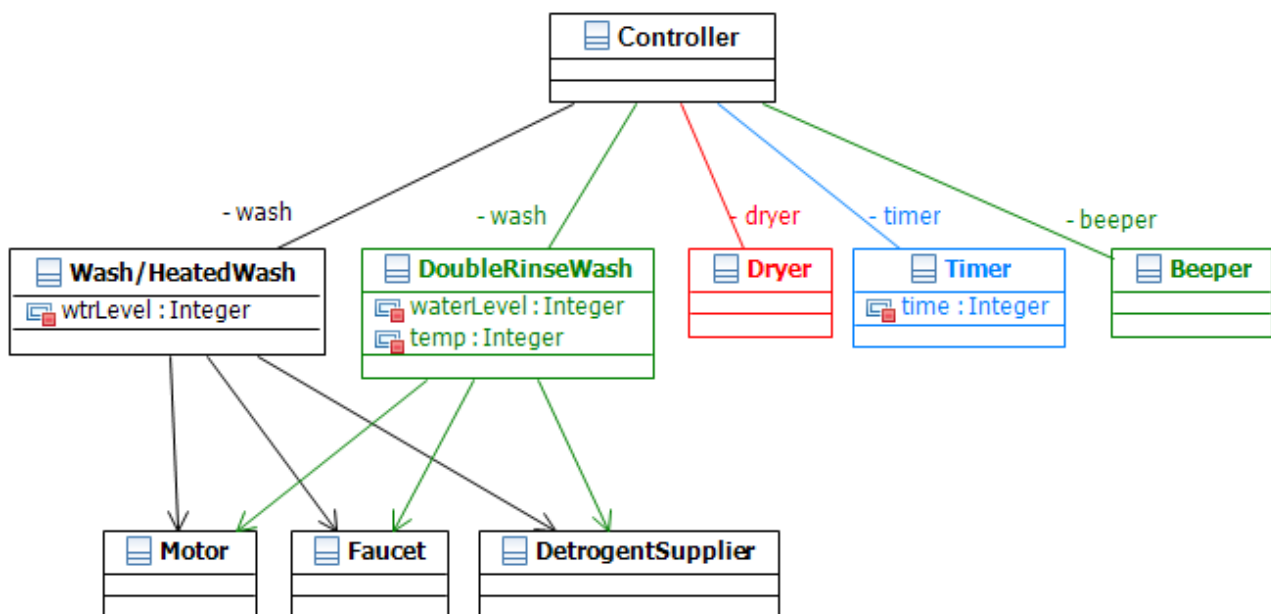


Рисунок 2.1.4 — Діаграма класів пральної машини

## РОЗДІЛ 3. Верифікація за допомогою Atelier B

Враховуючи вище описані вимоги для системи, її імплементацію було формально описано та верифіковано за допомогою формальної мови B та системи Atelier B.

### 3.1 Код загального опису пральної машини

*Опис пральної машини*

*WashingMashine.mch*

```
/* WashingMashine
 * Author: ElizaLo
 * Creation date: 5/10/19
 */

MACHINE
    WashingMashine
SETS
    WASHING_MACHINE = {WORKING, PAUSE, ALERT};
    ELECTRICITY = {ELEC_YES, ELEC_NONE};
    WATER = {WATER_YES, WATER_NONE};
    SOAP_POWDER = {S_P_YES, S_P_NONE};
    CONDITIONER = {COND_YES, COND_NONE};
    SOAP_POWDER_WATER = {SPW_YES, SPW_NONE};
    DOORS = {OPENED, CLOSED};
    PROGRAMME = {PROGPAMME_YES, PROGPAMME_NONE};
    OPTIONS = {OP_NONE, TEMP_YES, TEMP_NONE, BLUR,
RINSE, RINSE_WITH_COND, PRESS_UP};
```

LAUNDER = {LAUNDER\_DONE, ERROR}

#### VARIABLES

Washing\_Machine,  
Electricity,  
Water,  
Soap\_Powder,  
Conditioner,  
Soap\_Powder\_Water,  
Doors,  
Programme,  
Options,  
Launder

#### INVARIANT

Washing\_Machine : WASHING\_MACHINE &  
Electricity : ELECTRICITY &  
Water : WATER &  
Soap\_Powder : SOAP\_POWDER &  
Conditioner : CONDITIONER &  
Soap\_Powder\_Water : SOAP\_POWDER\_WATER &  
Doors : DOORS &  
Programme : PROGRAMME &  
Options : OPTIONS &  
Launder : LAUNDER &

((Electricity = ELEC\_NONE) => (Washing\_Machine =  
ALERT)) &

```

        ((Water = WATER_NONE) => (Washing_Machine =
ALERT)) &
        ((Soap_Powder = S_P_NONE) => (Washing_Machine =
ALERT)) &
        ((Programme = PROGPAMME_NONE) => (Washing_Machine
= ALERT)) &
        ((Options = RINSE & Conditioner = COND_NONE &
Water = WATER_YES) => (Washing_Machine = ALERT)) &
        ((Doors = OPENED) => (Washing_Machine = ALERT)) &
        ((Options = RINSE & Conditioner = COND_YES &
Water = WATER_NONE) => (Washing_Machine = ALERT)) &
        ((Washing_Machine = WORKING & Electricity =
ELEC_YES & Water = WATER_YES & Soap_Powder = S_P_YES &
        Programme = PROGPAMME_YES) => (Launder =
LAUNDER_DONE)) &
        ((Washing_Machine = WORKING & Options = RINSE &
Electricity = ELEC_YES & Water = WATER_YES & Conditioner
= COND_YES &
        Programme = PROGPAMME_YES & Soap_Powder =
S_P_YES) => (Launder = LAUNDER_DONE)) &
        ((Washing_Machine = WORKING & Electricity =
ELEC_NONE & Water = WATER_YES & Soap_Powder = S_P_YES &
        Programme = PROGPAMME_YES) => (Launder =
ERROR)) &
        ((Washing_Machine = WORKING & Electricity =
ELEC_YES & Water = WATER_NONE & Soap_Powder = S_P_YES &
        Programme = PROGPAMME_YES) => (Launder =
ERROR))

```

## INITIALISATION

```
Washing_Machine := WORKING ||
Electricity := ELEC_YES ||
Water := WATER_YES ||
Soap_Powder := S_P_YES ||
Conditioner := COND_YES ||
Soap_Powder_Water := SPW_YES ||
Doors := CLOSED ||
Programme := PROGPAEME_YES ||
Options := OP_NONE ||
Launder := LAUNDER_DONE
```

## OPERATIONS

```
autAlertElectricity =
PRE Electricity = ELEC_NONE
THEN
    Washing_Machine := ALERT
END;
```

```
autAlertWater =
PRE Water = WATER_NONE
THEN
    Washing_Machine := ALERT
END;
```

```
autAlertSoapPowder =
PRE Soap_Powder = S_P_NONE
```



THEN

Washing\_Machine := ALERT

END;

autAlertDoors =

PRE Doors = OPENED

THEN

Washing\_Machine := ALERT

END;

autAlertProgramme =

PRE Programme = PROGPAMME\_NONE

THEN

Washing\_Machine := ALERT

END;

washingNormal =

PRE Washing\_Machine = WORKING & Electricity =  
ELEC\_YES & Water = WATER\_YES & Soap\_Powder = S\_P\_YES &  
Programme = PROGPAMME\_YES & Doors = CLOSED

THEN

Launder := LAUNDER\_DONE

END;

washingNormalError\_1 =

PRE Washing\_Machine = WORKING & Electricity =  
ELEC\_NONE & Water = WATER\_YES & Soap\_Powder = S\_P\_YES &  
Programme = PROGPAMME\_YES & Doors = CLOSED

THEN

Laundry := ERROR

END;

washingNormalError\_2 =

PRE Washing\_Machine = WORKING & Electricity =  
ELEC\_YES & Water = WATER\_NONE & Soap\_Powder = S\_P\_YES &  
Programme = PROGPAMME\_YES & Doors = CLOSED

THEN

Laundry := ERROR

END;

washingNormalError\_3 =

PRE Washing\_Machine = WORKING & Electricity =  
ELEC\_YES & Water = WATER\_NONE & Soap\_Powder = S\_P\_NONE &  
Programme = PROGPAMME\_YES & Doors = CLOSED

THEN

Laundry := ERROR

END;

washingNormalError\_4 =

PRE Washing\_Machine = WORKING & Electricity =  
ELEC\_YES & Water = WATER\_YES & Soap\_Powder = S\_P\_YES &  
Programme = PROGPAMME\_NONE & Doors = CLOSED

THEN

Laundry := ERROR

END;

```

washingNormalError_5 =
    PRE Washing_Machine = WORKING & Electricity =
ELEC_YES & Water = WATER_YES & Soap_Powder = S_P_YES &
    Programme = PROGPAMME_YES & Doors = OPENED
THEN
    Launder := ERROR
END;

```

```

washingWithCond =
    PRE Washing_Machine = WORKING & Electricity =
ELEC_YES & Water = WATER_YES & Soap_Powder = S_P_YES &
    Programme = PROGPAMME_YES & Doors = CLOSED &
Options = RINSE_WITH_COND & Conditioner = COND_YES
THEN
    Launder := LAUNDER_DONE
END;

```

```

washingWithCondError_1 =
    PRE Washing_Machine = WORKING & Electricity =
ELEC_YES & Water = WATER_YES & Soap_Powder = S_P_YES &
    Programme = PROGPAMME_YES & Doors = CLOSED &
Options = RINSE_WITH_COND & Conditioner = COND_NONE
THEN
    Launder := ERROR
END;

```

```

washingWithCondError =

```

```

        PRE Washing_Machine = WORKING & Electricity =
ELEC_YES & Water = WATER_NONE & Soap_Powder = S_P_YES &
        Programme = PROGPAMME_YES & Doors = CLOSED &
Options = RINSE_WITH_COND & Conditioner = COND_YES
    THEN
        Launder := ERROR
    END;

washingWithPressUp =
    PRE Washing_Machine = WORKING & Electricity =
ELEC_YES & Water = WATER_YES & Soap_Powder = S_P_YES &
        Programme = PROGPAMME_YES & Doors = CLOSED &
Options = PRESS_UP
    THEN
        Launder := LAUNDER_DONE
    END
END

```

### 3.2 Код імплементації пральної машини

*Імплементація*

***WashingMashine\_i.imp***

```

/* WashingMashine_i
 * Author: ElizaLo
 * Creation date: 5/10/19
 */

```

*IMPLEMENTATION WashingMashine\_i*  
*REFINES WashingMashine*

*CONCRETE\_VARIABLES*

*Washing\_Machine,*  
*Electricity,*  
*Water,*  
*Soap\_Powder,*  
*Conditioner,*  
*Soap\_Powder\_Water,*  
*Doors,*  
*Programme,*  
*Options,*  
*Launder*

*INITIALISATION*

*Washing\_Machine := WORKING;*  
*Electricity := ELEC\_YES;*  
*Water := WATER\_YES;*  
*Soap\_Powder := S\_P\_YES;*  
*Conditioner := COND\_YES;*  
*Soap\_Powder\_Water := SPW\_YES;*  
*Doors := CLOSED;*  
*Programme := PROGPAMME\_YES;*  
*Options := OP\_NONE;*  
*Launder := LAUNDER\_DONE*

*OPERATIONS*

```
autAlertElectricity =  
BEGIN  
    IF (Electricity = ELEC_NONE)  
    THEN  
        Washing_Machine := ALERT  
    END  
END;
```

```
autAlertWater =  
BEGIN  
    IF (Water = WATER_NONE)  
    THEN  
        Washing_Machine := ALERT  
    END  
END;
```

```
autAlertSoapPowder =  
BEGIN  
    IF (Soap_Powder = S_P_NONE)  
    THEN  
        Washing_Machine := ALERT  
    END  
END;
```

```
autAlertDoors =  
BEGIN  
    IF (Doors = OPENED)  
    THEN
```

```

        Washing_Machine := ALERT
    END
END;

autAlertProgramme =
BEGIN
    IF (Programme = PROGPAMME_NONE)
    THEN
        Washing_Machine := ALERT
    END
END;

washingNormal =
BEGIN
    IF (Washing_Machine = WORKING & Electricity =
ELEC_YES & Water = WATER_YES & Soap_Powder = S_P_YES &
    Programme = PROGPAMME_YES & Doors = CLOSED)
    THEN
        Launder := LAUNDER_DONE
    END
END;

washingNormalError_1 =
BEGIN
    IF Washing_Machine = WORKING & Electricity =
ELEC_NONE & Water = WATER_YES & Soap_Powder = S_P_YES &
    Programme = PROGPAMME_YES & Doors = CLOSED
    THEN

```

```
        Launder := ERROR
    END
END;
```

```
washingNormalError_2 =
BEGIN
    IF Washing_Machine = WORKING & Electricity =
ELEC_YES & Water = WATER_NONE & Soap_Powder = S_P_YES &
    Programme = PROGPAMME_YES & Doors = CLOSED
    THEN
        Launder := ERROR
    END
END;
```

```
washingNormalError_3 =
BEGIN
    IF Washing_Machine = WORKING & Electricity =
ELEC_YES & Water = WATER_NONE & Soap_Powder = S_P_NONE &
    Programme = PROGPAMME_YES & Doors = CLOSED
    THEN
        Launder := ERROR
    END
END;
```

```
washingNormalError_4 =
BEGIN
    IF Washing_Machine = WORKING & Electricity =
ELEC_YES & Water = WATER_YES & Soap_Powder = S_P_YES &
```



```

        Programme = PROGPAMME_NONE & Doors = CLOSED
    THEN
        Launder := ERROR
    END
END;

```

```

washingNormalError_5 =
BEGIN
    IF Washing_Machine = WORKING & Electricity =
ELEC_YES & Water = WATER_YES & Soap_Powder = S_P_YES &
        Programme = PROGPAMME_YES & Doors = OPENED
    THEN
        Launder := ERROR
    END
END;

```

```

washingWithCond =
BEGIN
    IF Washing_Machine = WORKING & Electricity =
ELEC_YES & Water = WATER_YES & Soap_Powder = S_P_YES &
        Programme = PROGPAMME_YES & Doors = CLOSED &
Options = RINSE_WITH_COND & Conditioner = COND_YES
    THEN
        Launder := LAUNDER_DONE
    END
END;

```

```

washingWithCondError_1 =

```

```

BEGIN
    IF Washing_Machine = WORKING & Electricity =
ELEC_YES & Water = WATER_YES & Soap_Powder = S_P_YES &
    Programme = PROGPAMME_YES & Doors = CLOSED &
Options = RINSE_WITH_COND & Conditioner = COND_NONE
    THEN
        Launder := ERROR
    END
END;

```

```

washingWithCondError =
BEGIN
    IF Washing_Machine = WORKING & Electricity =
ELEC_YES & Water = WATER_NONE & Soap_Powder = S_P_YES &
    Programme = PROGPAMME_YES & Doors = CLOSED &
Options = RINSE_WITH_COND & Conditioner = COND_YES
    THEN
        Launder := ERROR
    END
END;

```

```

washingWithPressUp =
BEGIN
    IF Washing_Machine = WORKING & Electricity =
ELEC_YES & Water = WATER_YES & Soap_Powder = S_P_YES &
    Programme = PROGPAMME_YES & Doors = CLOSED &
Options = PRESS_UP
    THEN

```

*Launder* := LAUNDER\_DONE

END

END

END

### 3.3 Верифікація за допомогою системи Atelier B

Було успішно проведено верифікацію за допомогою формальної мови B та системи Atelier B (Рисунок 3.3.1):

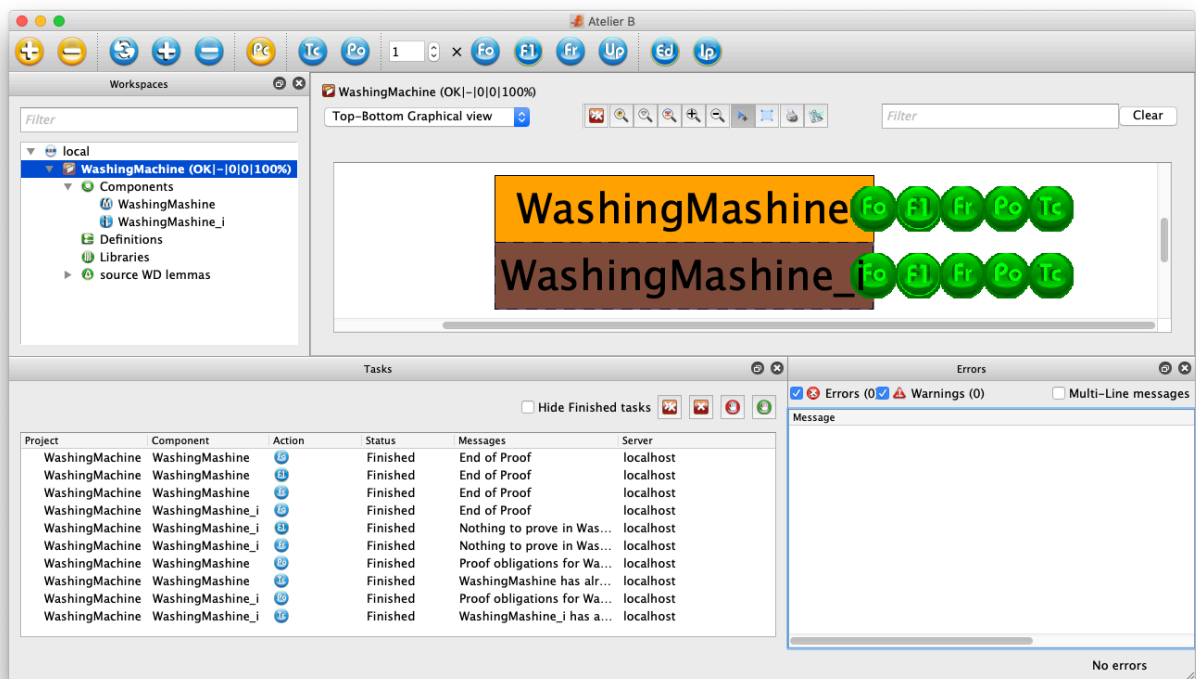


Рисунок 3.3.1 — Верифікація за допомогою системи Atelier B

## ВИСНОВКИ

Використання методів формальної специфікації та верифікації для створення системи за допомогою формальної мови програмування веде до суттєвого зменшення часу тестування та верифікації кінцевого програмного продукту чи системи.

Розроблена система на прикладі пральної машини Siemens iQ800 ілюструє перевагу використання специфікацій в мові В, яке полягає в математичному доказі виконання тих чи інших властивостей і збереження їх на наступних кроках деталізації. Таким чином, результатом такого підходу є математично доведена реалізація системи, яка не потребує подальшого тестування.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Jean-Raymond Abrial. "The B Tool (Abstract)" / Jean-Raymond Abrial. // VDM — The Way Ahead, Proc. 2nd VDM-Europe Symposium. Lecture Notes in Computer Science. Springer.. – 1988. – №328. – С. 86–87.
2. B-Method [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/B-Method>.
3. B-метод [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/B-метод>.
4. The B-Toolkit [Електронний ресурс]. – 2004. – Режим доступу до ресурсу: <https://web.archive.org/web/20041012141220/http://www.b-core.com/ONLINEDOC/BToolkit.html>.
5. B (мова програмування) [Електронний ресурс] – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/B\\_\(мова\\_програмування\)](https://uk.wikipedia.org/wiki/B_(мова_програмування)).
6. PRESENTATION OF THE B METHOD [Електронний ресурс] – Режим доступу до ресурсу: <https://www.methode-b.com/en/b-method/>.
7. THE INDUSTRIAL TOOL TO EFFICIENTLY DEPLOY THE B METHOD [Електронний ресурс]. – 2014. – Режим доступу до ресурсу: <https://www.atelierb.eu/en>.
8. Atelier B — Version 4.2 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.atelierb.eu/wp-content/uploads/sites/3/2014/12/release-notes-atelier-b-4.2-en.pdf>.
9. Bert D. Adaptable Translator of B Specifications to Embedded C Programs/ D Bert, S. Boulmé, M.L. Potet, A. Requet, L. Voisin1 // FME 2003: Formal Methods. – 2003. – Vol. 2805. – P. 94113.
10. Schneider S. The BMethod: An Introduction / S. Schneider // Palgrave, Cornerstones of Computing series, 2001.
11. Sørensen Ib Holm. Using B to specify, verify and design hardware circuits / Ib Holm Sørensen //ZUM '98: The Z Formal Specification Notation. Lecture Notes in Computer Science. – 1998. – Vol. 1493. – P. 6065.

12. Yang L. Automatic Translation from Combined B and CSP specification to Java Programs. / L. Yang, M.R. Poppleton // 7th International B Conference. 1719 January 2007. Besancon, France.

13. Johnson and Kernighan. "THE PROGRAMMING LANGUAGE B". Bell Laboratories. Archived from the original on 11 June 2015. Retrieved 21 March 2014.

14. Ritchie, Dennis M. "The Development of the C Language". Bell Labs/Lucent Technologies. Archived from the original on 11 June 2015.