Faculty of Gina Cody School of Engineering & Computer Science
Department of Institute for Information System Engineering

INSE 6250: Quality Methodologies for Software

Project: Modelling and Verification of a Vending Machine System Using UPPAAL

Submitted to: *Dr. Jamal Bentahar*

Student name	Student ID
Syed Muhammad Irtaza Rizvi	40272071
Anik Das Gupta	40221795
Md. Ahsanul Haque Laskor	40205655
Mukul Prabhu	40257131
Hira Amir	40242225

Introduction

Today's vending machines are complex, software-driven systems instead of just basic mechanical devices, thanks to automation and pervasive computing. These devices are expected to carry out difficult jobs effectively and dependably because they are now a necessary part of many settings, including workplaces, bus stops, and educational institutions. Robust techniques to guarantee the proper functioning of these systems are becoming increasingly necessary as their complexity rises. In addition to bothering customers, a broken vending machine may result in monetary losses and harm to the device itself.

Formal approaches have become more popular in system design and verification to overcome these issues. Formal methods are procedures for the development, verification, and specification of hardware and software systems that are based on mathematical rigorousness. Among the most effective instruments in this field is UPPAAL, which offers a thorough environment for real-time system modelling, simulation, and verification. With the help of UPPAAL, designers may produce models whose attributes, such as safety (making sure nothing bad happens) and liveness (making sure something positive happens eventually), can be examined.

In-depth UPPAAL modelling and verification of a vending machine system are covered in this study. Building an accurate model of the vending machine that captures all important parts of its functioning, such as user interactions, payment processing, and product dispensing, is the main goal. Once the model is developed, the verification capabilities of UPPAAL are utilised to verify correctness against predefined criteria. Examples of these specifications include making sure the machine correctly handles concurrent processes or that a product is distributed only after the correct payment has been made.

With this project, we hope to highlight the usefulness of UPPAAL in preventing design faults and guaranteeing the resilience of automated systems by demonstrating its practical use in confirming the reliability of real-time systems. This report illustrates the wider implications of formal verification in the development of reliable software and hardware systems in diverse sectors by modelling a typical yet complex system such as a vending machine.

Key Components of UPPAAL

Modelling:

- **Templates:** To make the system easy to build and expand reusable parts are taken which define how the system works.
- **Global Declarations:** For ensuring data is used consistently variables and constants are shared across different templates.
- **System Definition:** For forming a clear and organised overall system various templates are combined.

Simulation:

- **Simulator:** Enable detailed observation and analysis of how the system behaves by allowing step-by-step execution of the model.

- **Trace:** To make it easier to understand the event sequence and spot potential issues it gives a visual display of execution paths.

Verification

- **Query Language:** Allows the specification of properties to be verified. For example, safety, liveness, and reachability conditions.
- **Model Checker:** Whether the system satisfies the specified properties, ensuring correctness and reliability is verified by the model checker.

Implementation of UPPAAL in Our Project

Our project involved the modelling and verification of a vending machine system using UPPAAL.

The vending machine system consists of three components:

1. **Client:** The user interacts with the vending machine, initiating transactions and selecting items.
2. **Vending Machine:** The central component responsible for item dispensing, inventory management, and user interface.
3. **Payment System:** Manages financial transactions, validates payments, and ensures secure processing.

Modelling Process

- We began by defining templates for each component, specifying their behaviours and interactions.
- Global declarations were used to manage shared variables such as inventory levels, transaction status, and timing constraints.
- The overall system was composed by integrating these templates, forming a complete and functional model.

Simulation Process

- The simulator allowed us to execute the model step-by-step, observing the interactions between the client, vending machine, and payment system.
- Traces were generated to visualise the execution paths, helping us identify and address any issues in the system's behaviour.

Verification Process

- We specified various properties to be verified, such as ensuring that an item is only dispensed upon successful payment and that the system correctly handles timeouts and errors.
- The model checker verified these properties, confirming the system's correctness and reliability.

Benefits of Using UPPAAL

A vital component for safety-critical industries including the automotive, aerospace, and healthcare sectors, UPPAAL is a flexible and potent tool for formal verification that guarantees that systems operate as intended in all potential circumstances. Because of its capacity for accurate timing analysis, it is especially well-suited for applications where real-time requirements are stringent including embedded systems and real-time operating environments. A major factor is the ability to streamline the modelling process in a user-friendly graphical interface, which makes it simple for users to visualise system behaviours, spot mistakes, and successfully explain complicated ideas to consumers who are not technical.

In addition to being very user-friendly, UPPAAL is also very scalable, meaning it can analyse big, complex systems with complicated timing and synchronisation requirements. Because system complexity may provide a big problem, this makes it a great option for applications involving advanced communication protocols or industrial automation. Additionally, before committing to formal verification, users can investigate various scenarios in a controlled setting with the help of UPPAAL's simulation tools. This feature offers a degree of flexibility and improves the user experience, making it a useful tool for both novices and experts in formal techniques.

Challenges Faced

Although UPPAAL provided useful tools for modelling and validating our vending machine system, there were a few obstacles we had to overcome. Model construction required careful preparation to build precise templates and global declarations, as well as a thorough grasp of the workings of the vending machine. The dependability of the entire model might be impacted by any errors in this procedure.

Another major problem was managing non-determinism in the interactions of the vending machine. Precise attention to detail was necessary to ensure that all possible states and transitions were accurately modelled, particularly those requiring user input and payment processing.

Debugging and problem detection were difficult even with UPPAAL's powerful visualisation capabilities, especially when handling complicated interactions like multiple payment insertions or simultaneous user actions. The intricate nature of these relationships occasionally made it challenging to identify and fix mistakes, highlighting the necessity of a rigorous and methodical approach to guarantee the accuracy of the vending machine system's representation and verification.

System Information

This vending machine system consists of three primary subsystems:

1. **Client:** Represents the user interacting with the machine.
2. **Vending Machine:** The physical machine that dispenses products.
3. **Payment System:** Handles financial transactions.

These subsystems interact to facilitate the purchase and delivery of products.

System Components and Interactions

Based on standard vending machine operations, here's a breakdown of components and their interactions:

Client Subsystem

- **Product_Selection:** Allows the user to choose a desired product.
- **Payment:** Starts the payment process.
- **Dispense_Product:** Collects the dispensed product.

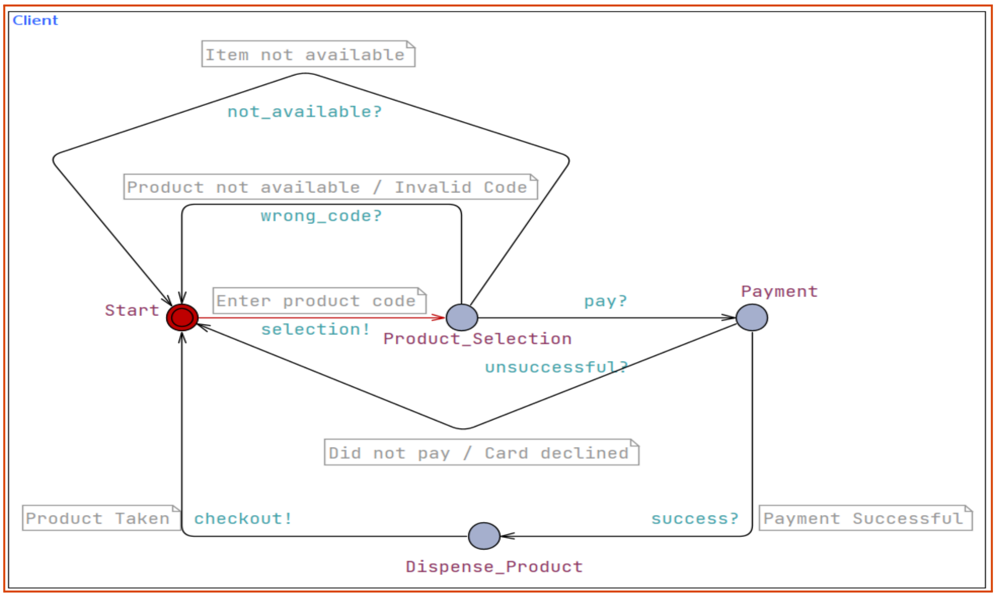


Figure 1:Client Subsystem

Vending Machine Subsystem

- **Product Inventory:** Manages product stock levels.
- **Code Verification:** Verifies the correct code.
- **Payment_Status:** Checks payment information.
- **Product_Dispense:** Dispenses the product.

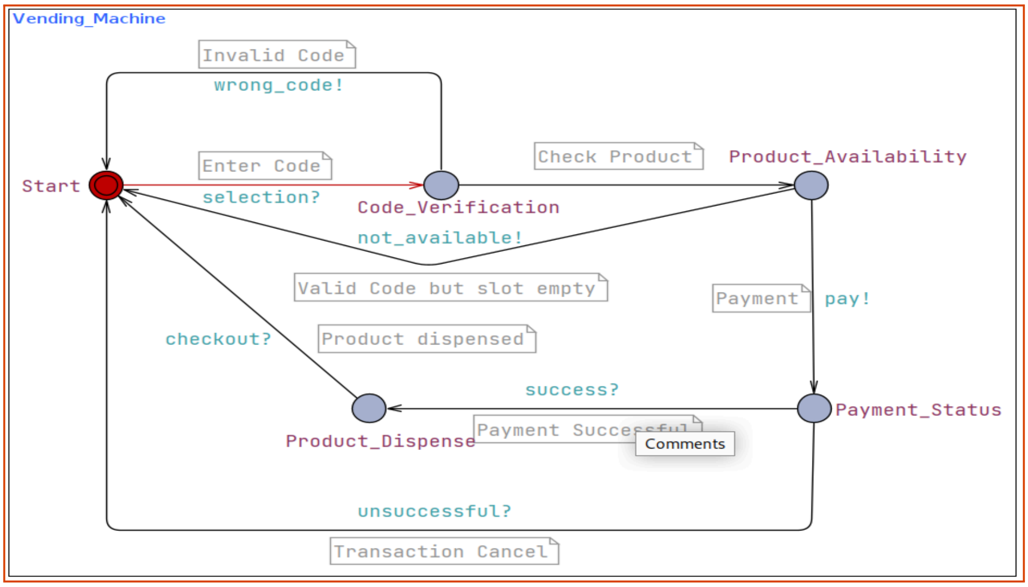


Figure 2: Vending machine Subsystem

Payment System Subsystem

- **Payment Method:** Handles various payment methods (cash, card, mobile).
- **Cash_Pay:** Payment by cash.
- **Card_Pay:** Payment by card.
- **Status:** Checks for complete or partial payment.

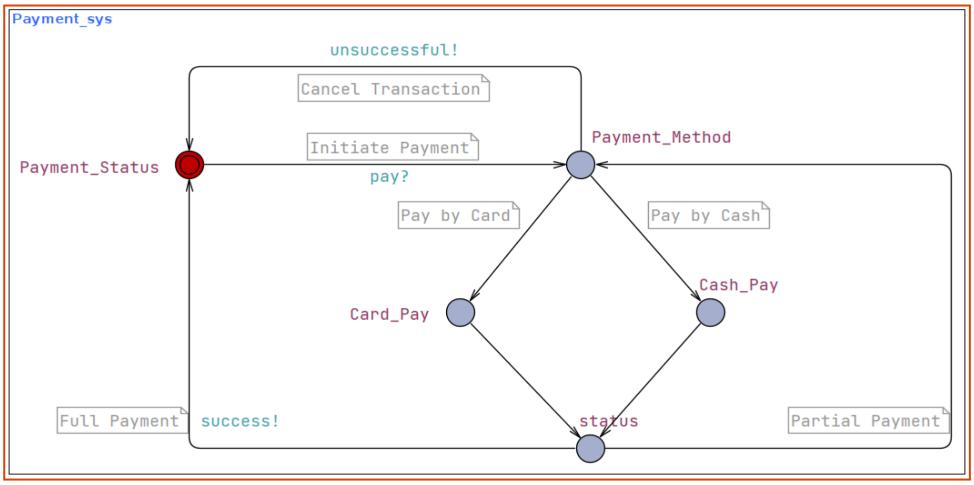


Figure 3: Payment Subsystem

System Workflow

- **Product Selection:** The client selects a product from the available options.
- **Payment Initiation:** The client initiates the payment process by inserting cash, using a card, or selecting a mobile payment option.
- **Payment Processing:** The payment system verifies the payment method and authorises the transaction.

- **Product Dispensing:** If the payment is successful, the vending machine dispenses the selected product.
- **Change Return (if applicable):** The vending machine returns any change due to the client.

Working Model

As a working example of the system, here's the workflow of one of the possible paths. The flow starts from the client side. After the client enters the product code, the control will move to the vending machine where it will check if the code is valid. After validation, it will check if the product is available. After the availability check, control will go to both the client and the payment system.

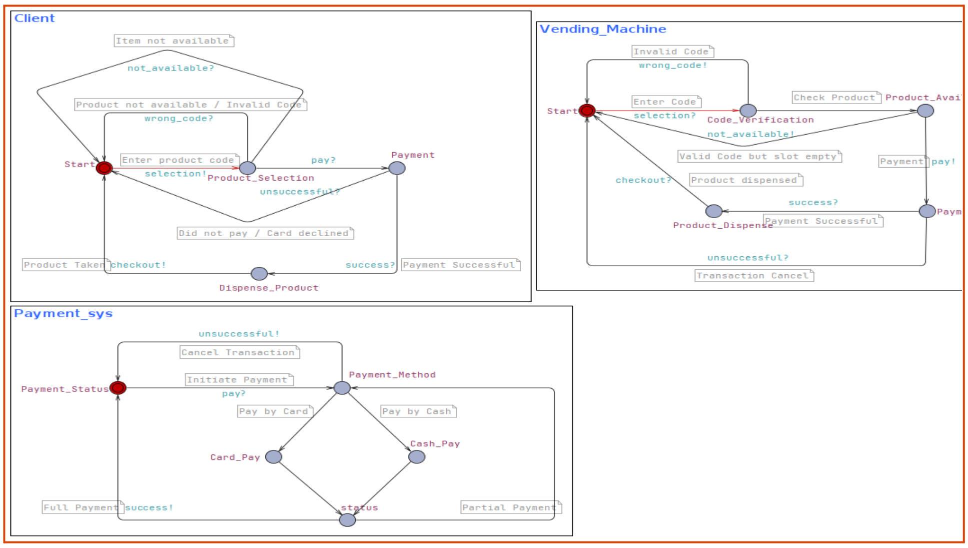
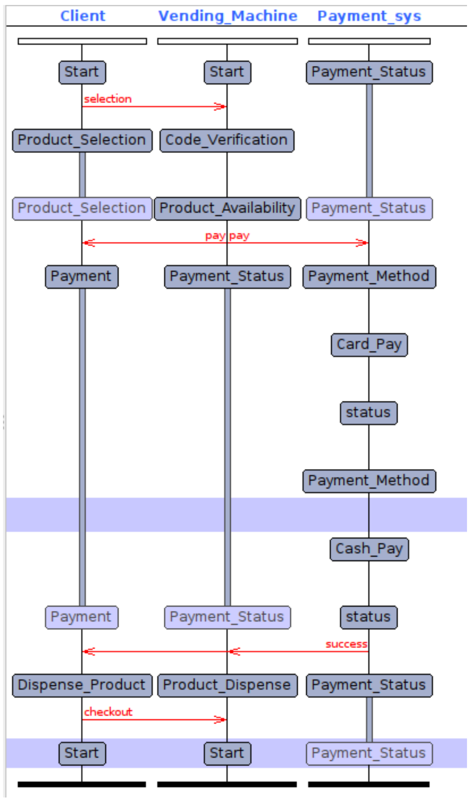


Figure 4: Working Model



At this point, the client will wait until the payment system finishes its process. In the payment system, a payment method is selected. Initially, Card payment is selected, and the control goes back to the payment method state as partial payment is made. So now cash payment is selected and the status is checked again. It is seen the payment is complete and control will be resumed at the client's side. Payment completion is confirmed at the client's side and control is given to the vending machine where the product is dispensed to the client. This ends the entire workflow.

Properties

After we had a working model of our system, we wrote some properties to verify the behaviour of the system. It is useful to check if our system performs as it is intended to do, specify safety properties and make sure the system is designed correctly. The green colour indicates that the property is verified and red means the property is not verified. The following are the properties used to verify the system:

- A[] bit deadlock
  This is to check if there's any deadlock in the system in all the states
- E<> not deadlock
  This is to check if there are no deadlocks in the system on any one path
- E<> deadlock
  This is to check if there are deadlocks in the system on any one path
- E<> Vending_Machine.Product_Dispense
  This is to check if the above state is reached or not
- A[] !(Vending_Machine.Product_Dispense && Payment_sys.Payment_Method) This is to make sure the vending machine does not give the product before payment is done
- A[] !(Vending_Machine.Product_Dispense && Client.Product_Selection) This is to make sure the vending machine does not give the product before the client selects an item
- A[] !(Vending_Machine.Product_Dispense && Vending_Machine.Product_Availability)
  This makes sure the vending machine does not dispense anything if the selected item is not available
- A[] !(Vending_Machine.Product_Dispense && Payment_sys.Status)
  This makes sure the item is not given even if partial payment is made
- E<> (Vending_Machine.Payment_Status imply Vending_Machine.Product_Dispense)
  This says that if the payment is made the product will be dispensed
- E<> (Payment_sys.Cash_Pay imply Payment_sys.status)
  This says after paying by cash the system goes to payment status state.
- E<> (Client.Product_Selection imply Client.Dispense_Product)
  This says after the product is selected there's a path that will lead to dispensing the product
- A[] (Client.Product_Selection imply Client.Dispense_Product)
  This says that after a product is selected all paths lead to the product dispenser
- E<> (Vending_Machine.Code_Verification imply Vending_Machine.Product_Availability)
  This says after the code is verified there's a path that will lead to a product availability check.
- A [] (Vending_Machine.Code_Verification imply Vending_Machine.Product_Availability)
  This says after the code is verified all paths that will lead to a product availability check.
- E<> (Vending_Machine.Code_Verification imply Vending_Machine.Payment_Status)
  This says there's a path where the code is verified, and the control will move to the payment state.
- A[] (Vending_Machine.Code_Verification imply Vending_Machine.Payment_Status)

This says if the code verification is true then a path will move to the payment status.

- E<> (Vending_Machine.Code_Verification imply Vending_Machine.Payment_Status)

  This says if the code verification is true then all paths will move to payment status.
- E<> (Vending_Machine.Code_Verification imply Vending_Machine.Product_Dispense)

  If a code is verified, then there is a path where the product will be dispensed.
- A[] (Vending_Machine.Code_Verification imply Vending_Machine.Product_Dispense)

  If a code is verified, then in all paths the product will be dispensed.
- A[] (Payment_sys.Card_Pay && Payment_sys.Cash_Pay)

  This says that in all paths users cannot pay by cash and card at the same time.
- E<> (Payment_sys.Card_Pay && Payment_sys.Cash_Pay)

  This says that there is no path where the user cannot pay by cash and card at the same time.

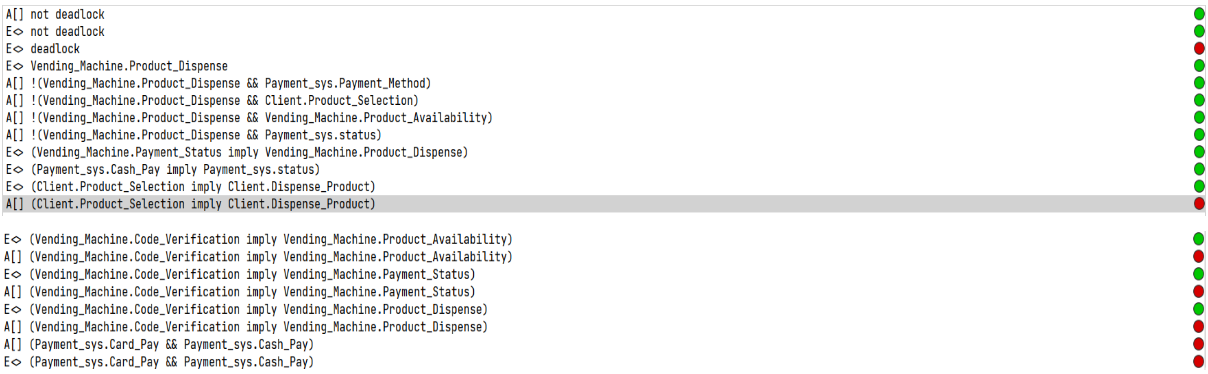


Figure5: Properties

Conclusion

We were able to ensure the accuracy and dependability of our vending machine system by using UPPAAL for modelling and verification, which yielded significant benefits. UPPAAL's formal verification and timing analysis capabilities were helpful in handling our project's real-time limitations. Notwithstanding the difficulties encountered, UPPAAL turned out to be a crucial resource for our project's successful conclusion. This experience made clear how crucial thorough modelling and verification are to creating dependable real-time systems. This report provides an overview of our project, which involved modelling and evaluating a vending machine using UPPAAL. It highlights the main features of the tool, the method of installation, its advantages, and the difficulties encountered along the way.