Formal verification of a UPPAAL-based garage control system

Bilal Abdullateef Kareem[1] Electrical Engineering Department, College of Engineering, Salahaddin Universty-Erbil, Iraq
Email: [1]belalengineer53@gmail.com

Dr. Ahmad Senjari[2] Electrical Engineering Department, College of Engineering, Salahaddin Universty-Erbil, Iraq
[2] ahmadsinjari@gmail.com

ABSTRACT

One of our society's major issues is citizen insecurity. Many homes currently lack a control system that allows them to operate an automatic garage door. Liftgate automation raises the issue of access security and system control. Because an intruder can corrupt the system and gain access to it. In other cases, opening and closing the door can result in an accident. Faced with this challenge, the goal is to create and test a system with formal specifications for controlling and accessing automatic sliding doors using UPPAAL. This system aims to ensure the system's proper operation and the safety of the environment that surrounds the sliding doors.

Keywords: Verification, UPPAAL, Garage, Automatic System

Introduction

With today's technological advances, various systems are developed in which people seek to automate most of their functions, allowing them to simplify some daily tasks; however, very few of the developed systems have a formal review, which ensures the system's quality and the complete safety of the people who interact or have direct contact with it.

Every day, formal specifications play an important role in critical project development because they prevent errors that could lead to significant losses, as previously demonstrated on several occasions. Currvarious systems perform functions, but for the most part, there is no revision of the formal specifications, which is due to a variety of factors such as the cost of this investment(Ikpeze, Uwaezuoke et al. 2021).

Various systems perform critical functions today, but the majority of them do not include formal specification revision due to various reasons such as the cost of this investment. Our work presents formal verification for the development of an automatic gate control system using UPPAAL. Our main goal is to demonstrate the importance of formal verification in the development of critical systems, allowing us to reduce the risk of errors and ambiguities during the development of various systems.

The remainder of the article is structured as follows: Section 2 refers to the various works dealing with door automation and applications. Section 3 defines the various concepts relevant to the development of an automatic garage door modeling system and provides information about the UPPAAL tool.
In Section 4, we define the solution proposal, requirements discovered, and model in the tool used. Finally, we will present our findings regarding the formal verification modeling used in the automated parking control system.

2. Related Works

Automation with the help of technology emerging from different manual actions has allowed the development time of these to be improved, the following works explain the different investigations to automate doors for garages or similar topics:

In (Marinescu 2014) performs the collection of cryptographic information and flow ciphers for the design and implementation of remote control for a garage liftgate. This design specifies the mathematical operation of the liftgate. Benefiting the greater control of the door, in front of intersections, which appear in different cases of theft. In(Han and Lim 2010) proposes a security and surveillance system that informs the client via email every time there has been a change in the state of the sensors installed in the residence, allowing the owner to access a web page that will allow you to view the status, history and captured videos.

The system incorporates an Arduino Mega 2560 card, an Ethernet shield card, a web camera, a servo motor, a relay card, and five sensors: a magnetic sensor, a movement sensor, a vibration sensor, a gas sensor, and a temperature sensor.

Similarly,(Gota, Puscasiu et al. 2020) proposes using arduino, but with specialized determination for garage door automation. That is, when a green button is pressed, the door opens, and when a red button is pressed, it closes. The door will have two limit switches, one for total opening and one for total closure.

Taking a different angle, consider(Ducreux, Guyon-Gardeux et al. 2015), which proposes a parking control system based on the recognition of vehicles in the parking lot. The recognition is accomplished through the use of a camera that stores the number of placards and the location of the vehicle. This system attempts to automate vehicle recognition through the use of a camera that detects the presence of vehicles. This could be used to detect the presence of a vehicle in a home's garage.

Our work is unique in that we demonstrate the development of an Automatic Garage Control System. Using UPPAAL, our system will have a verification that guarantees the quality and safety of the system, increasing the system's reliability significantly

3. Methodology: Model Verification

A. Model Validation

Model verification is a technique for validating concurrent finite-state systems such as sequential circuit design and communication protocols. Have several advantages over traditional approaches based on simulation, testing, and deductive reasoning. The model's verification, in particular, is automatic and usually quite fast. Furthermore, if the design contains an error, checking the model will generate a counterexample that can be used to pinpoint the source of the error(Clarke, Henzinger et al. 2018).

B. Model of State Transition

A state transition diagram depicts an information system's time-dependent behavior. Realizing the representation of the states that a system can take in order to later show the events that imply the change of various states(Clarke, Henzinger et al. 2018)

4. The UPPAAL Tool

It is a model verification tool used for system modeling, simulation, and verification(Truscan, Ahmad et al. 2015). UPPAAL is made up of three major components: a description language, a simulator, and a verifier model. As shown in Figure 1, the description language or editor is a command language with real-time clock variables and simple data types. It describes the system behavior as a network of automata with clock and data variables.

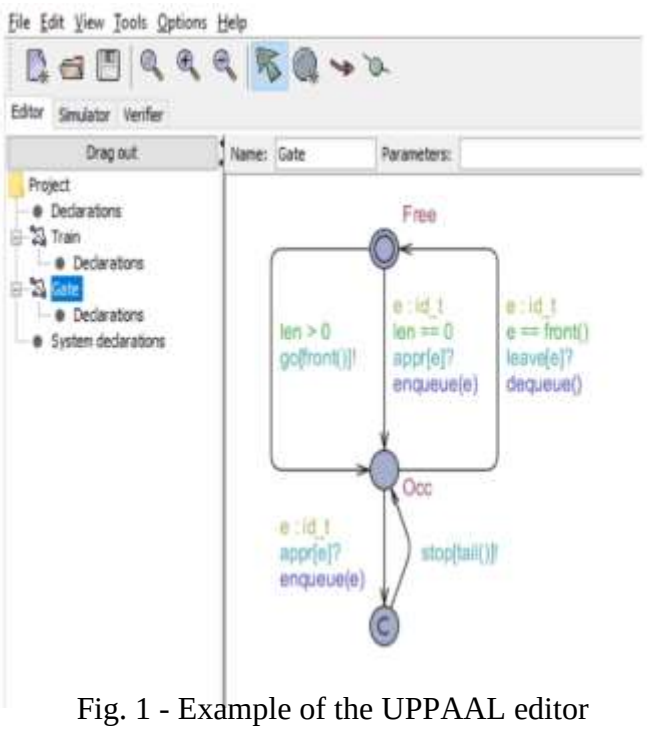

Fig. 1 - Example of the UPPAAL editor

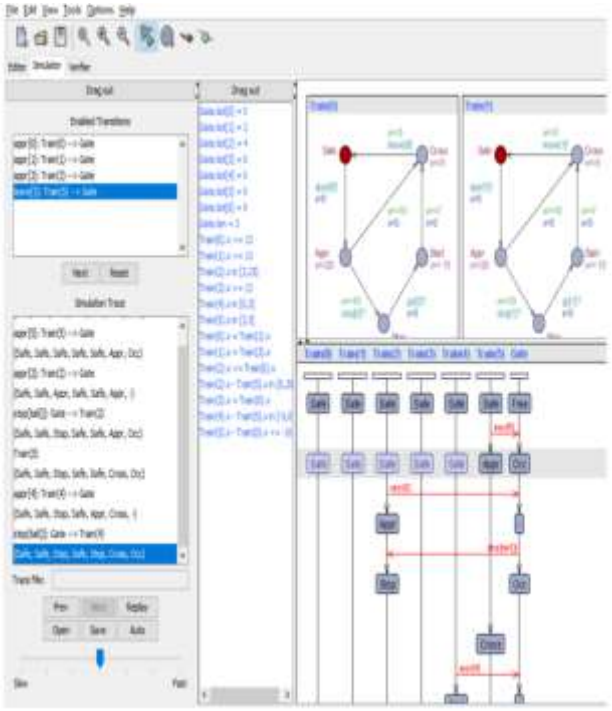

Fig. 2 - Example of the UPPAAL simulator environment.

The simulator, as shown in Figure 2, is a validation tool that allows the examination of possible dynamic executions of a system during the early stages of design, allowing the detection of faults prior to verification, because the model checker covers the system's exhaustive dynamic behavior.

Figure 3 shows a pattern checker for accessibility analysis in terms of state symbols represented by constraints(Marinescu 2014). This will allow us to see if it meets the system's requirements.

The model can be validated using UPPAAL. They are capable. The model can be queried in a variety of ways, including:

A []: Can be used for safe accessibility properties.

A<>: Can be used for simple accessibility properties.

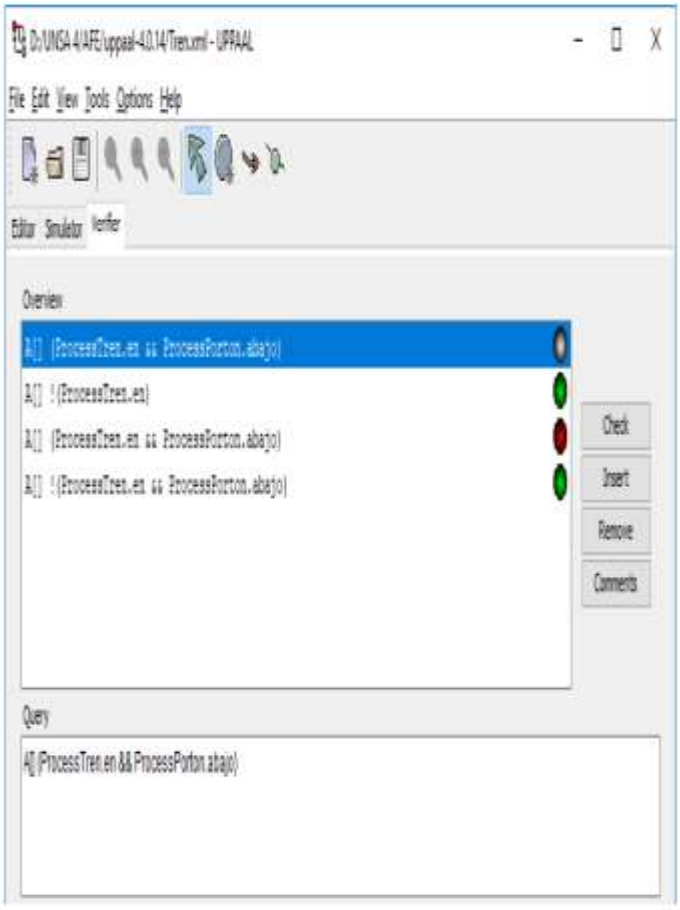

Fig. 3 - Example of the UPPAAL verifier environment.

The UPPAAL tool presents us with three states that we can assign to each based on their function.

-Initial (initial): Each schema can only have one initial state. This is the first state of the schema.

-Urgent (urgent): Declaring the state as urgent halts the passage of time while a process is in one of them.

-Committed (committed): Committed states, like urgent ones, stop time and also force the next transition chosen to be executed to be some outgoing enabled transition from (originating in) a committed state (establishes a priority in choosing transitions).

Edges connect the states and establish the transition relationship between them. Selectors, guards, synchronizations, and modifications can be associated with edges.

– Selectors: Selectors return a value in a specified range in a non-deterministic manner.

– Guards: The guards create the conditions for an edge to be enabled. A guard is specified using logical expressions that adhere to the same constraints as invariant specifications.

– Synchronizations: Shared channels can be used to synchronize processes. Channels allow the following actions: write (!) and read (?). Each edge can only point to one channel.

– Changes: When the edges are executed, the expressions specified in this section are evaluated, and their effect changes the state of the system.

5. CASE STUDY: AUTOMATIC GARAGE CONTROL SYSTEM

A .description of the issue

As we know, the security situation in the capital is bad and unstable and is considered among the top five cities in terms of citizens' insecurity, according to various studies conducted in recent years; Theft is one of the most common crimes, as we see, the insecurity of citizens is a big problem, at the moment, many homes do not have a control system that allows them to control the automatic garage door. And in our region, the culture of creating a private garage for the car is very little, as most people use the method of random parking on the roads and in front of houses, which leads to accidents hitting the car while it is parked.

The automation of the tailgate raises the issue of security of access and control of the system. Since the system can be damaged and accessed by an intruder, if not properly checked, it can expose these shortcomings, causing people inconvenience and financial losses. In other cases, automatic garage doors can cause an accident when opening and closing said door. There are a variety of control systems, but most of the systems do not have formal verification, so the control system becomes critical because it can cause serious accidents to people if it fails in any job.

Accidents due to system failures occur on a large scale on a global scale, where the stats are important because there are Thousands of accidents, errors and injuries and are directly related to automatic garage doors, a very high number when one takes into account that there are safety regulations in this The connection warns of provisions to ensure the safe use of these devices(https://www.thespruce.com/troubleshooting-common-garage-door-problems-1398186).

Fig. 4 - Garage door

B. Proposal

In response to the problem, our proposal is to create an automatic garage control system, with the goal of verifying the control system that allows a garage door to be governed automatically. When a green button is pressed, the door opens; when a red button is pressed, it closes. The door will have two sensors: one for total opening and one for total closure.

The third signal from an optical sensor parallel to the door that detects the presence or absence of an object may also be considered as an option. When the sensor signal is activated, the door should return to its previous state, and if an object is detected on both sides, the door should automatically stop.

C. Requirements

In order to develop the formal verification of the system, the following requirements must be identified:

RQ1. The system will be developed on a computer, allowing for accurate simulation of the operation.

RQ2. There are two green and two red buttons on the system.

RQ3.The green button indicates that the door is about to open.

RQ4. The red button indicates that the door is about to close.

RQ5. There will be only one time mode in the system.

RQ7. True and false are the two states of the sensor.

RQ8. The sensor's true state indicates full opening.

RQ9. The sensor's false state indicates total closure.

RQ10. It will also have an optical sensor parallel to the door to detect whether or not an object is present.

RQ11. True and false are the two states of the optical sensor.

RQ12 The presence of an object in the optical sensor indicates the true state.

RQ13: The optical sensor's false state indicates the absence of objects.

D. UPPAAL Model

The control system will be modeled as shown below. Figure 5 depicts the remote control used to operate the door. Simulate the states of the buttons as needed: green for open and red for close.

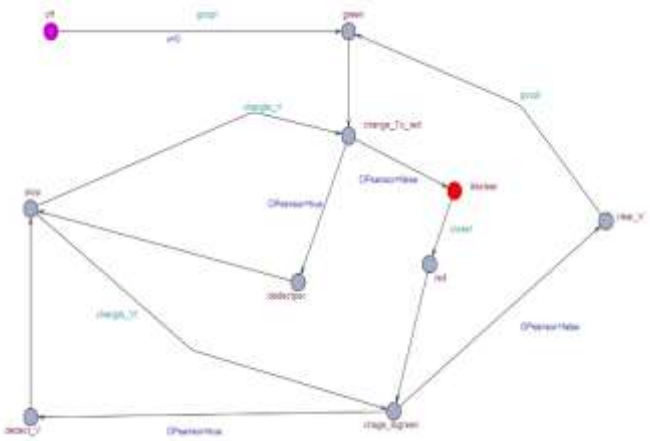

Fig. 5 - Garage System

Figure 6 depicts the simulation of the system controller's states, which allows us to identify the synchronized processes gate and remote control.

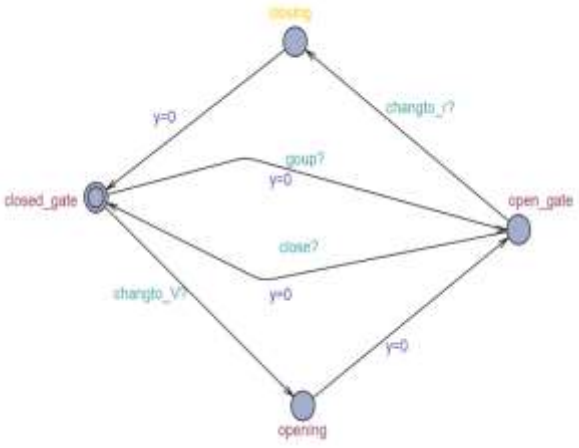

Fig. 6 - System Controller

In figure 7, it is possible to observe the simulation of the behavior of the garage system.

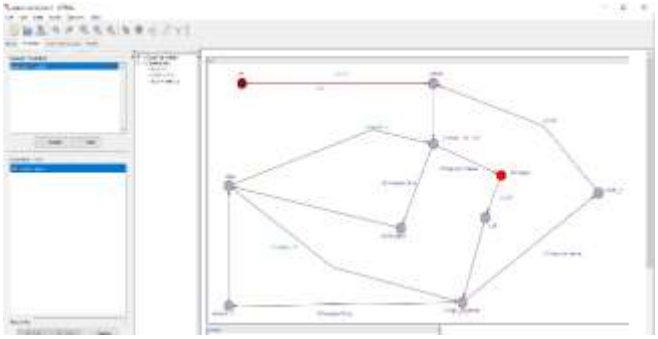

Fig. 7 - Simulator

In figure 8, we see the complete system of the controller simulation and the garage system.

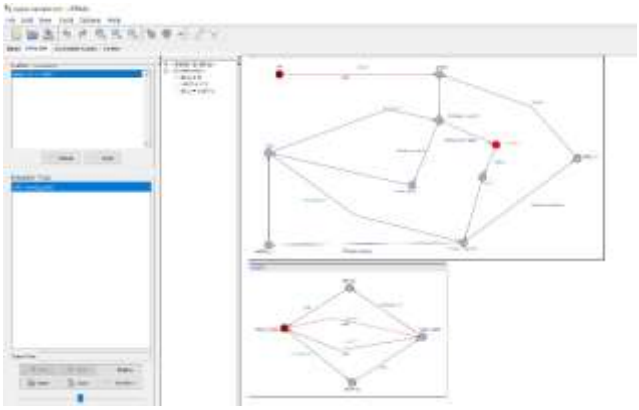

Fig. 8 - Controller simulation system

Fig 9 depicts the behavior of the simulated system when run in random mode.

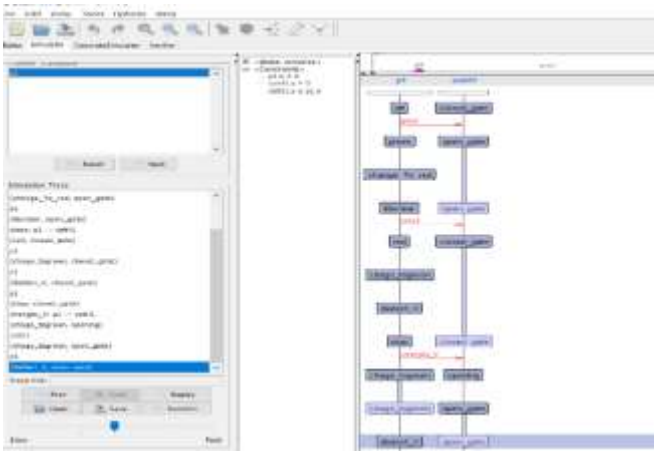

Fig 9 depicts a simulation of the entire garage door system.

E. Verification

The automatic verification made to the garage control system, we have:

A[]!(p1.dedectper&&cotrl1.opening),which means that if the system detects an object in red to close, it cannot perform the open, as shown in figure 12. We appreciate your confirmation and approval.

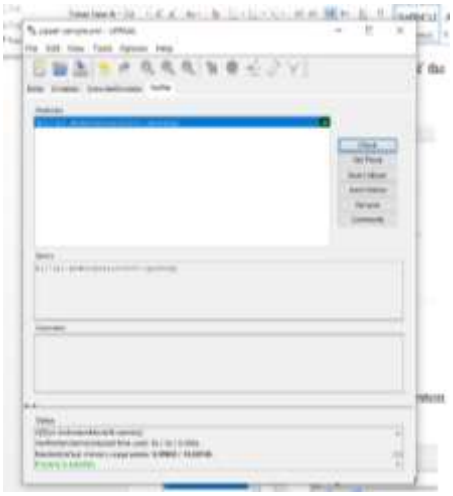

Fig. 10 Verification for contradictions

The second verification shows the opposite part of the object in recognition, which was given to determine the object to open in green, so the door cannot be closed, as shown in figure 10.

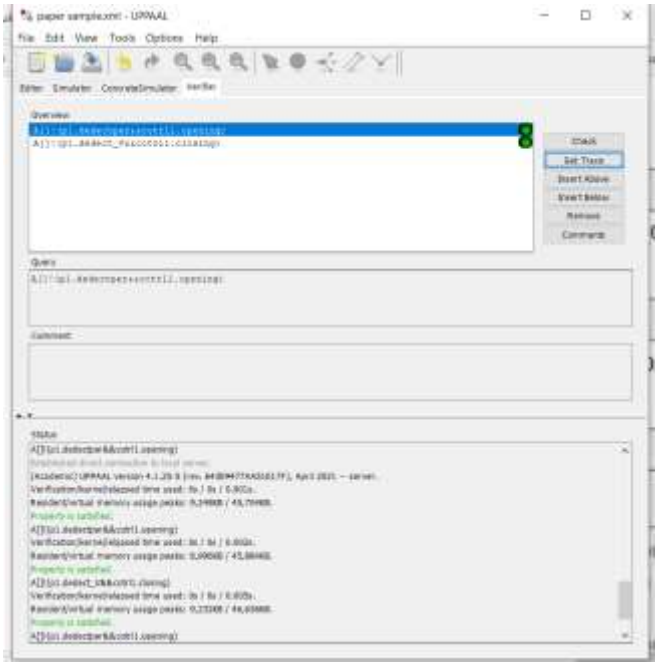

Fig. 10 Verification for contradictions

F. Analysis of Results

The automatic garage door control system is a solution to a daily problem presented by different people in our city, which could be applied to any other city in the world; our work is a simple sample that addresses a specific topic taking into account only automatic garage doors are available.

Model Checking was used in this system to formally verify correct operation, along with the UPPAAL tool, which assisted us in simulating it. As a result, we can conclude that formal verification gives us a better perspective on different systems and eliminates ambiguities in the requirements according to the proposed system, resulting in a more reliable system.

Declaration and system declaration

References

Bengtsson, J., et al. (1995). UPPAAL—a tool suite for automatic verification of real-time systems. International hybrid systems workshop, Springer.

Clarke, E. M., et al. (2018). Handbook of model checking, Springer.

Ducreux, L. F., et al. (2015). Rapid prototyping of complete systems, the case study of a smart parking. 2015 International Symposium on Rapid System Prototyping (RSP).

Gota, D. I., et al. (2020). Smart home automation system using Arduino microcontrollers. 2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR).

Han, D. M. and J. H. Lim (2010). "Design and implementation of smart home energy management systems based on zigbee." IEEE Transactions on Consumer Electronics **56**(3): 1417-1425.

https://www.thespruce.com/troubleshooting-common-garage-door-problems-1398186. "Common Garage Door Opener Problems."

Ikpeze, O., et al. (2021). "Design and Construction of an Automatic Gate." **2**: 123-131.

Marinescu, R. (2014). Model-checking and model-based testing of automotive embedded systems: Starting from the system architecture, Mälardalen University.

Truscan, D., et al. (2015). A Practical Application of UPPAAL and DTRON for Runtime Verification. 2015 IEEE/ACM 2nd International Workshop on Software Engineering Research and Industrial Practice.