IJEMD-CSAI, 1 (2) (2022), 26 – 34 https://doi.org/10.54938/ijemdcsai.2022.01.2.89

International Journal of Emerging Multidisciplinaries:
Computer Science & Artificial Intelligence

Research Paper
Journal Homepage: www.ojs.ijemd.com
ISSN (print): 2791- 0164 ISSN (online): 2957-5036

Modeling and Verification of Trainify

Kainat Fatima[1,*], Jawad Ali Abbasi[1], Muhammad Talha Khalid[1], and Sidra Sultana[1]

[1] *School of Electrical Engineering and Computer Science, National University of Sciences and Technology, (NUST) Islamabad, Pakistan*
**Corresponding author*

Abstract

Safety and correctness can be assured in a system by applying numerous testing techniques. We have used formal modeling and verification for our work given it allows us to thoroughly test the app by going through all its real time aspects. In this paper we have modelled and verified a tennis training app, Trainify, to prevent wrong play in tennis and to also ensure that our tennis athletes or beginners do not hurt themselves in the process of learning or playing tennis. We found several modules which could compromise the safety pf the players and created their modules in Uppaal by modelling and then verified them using Uppaal's verifier feeding it the different properties to be tested to ensure no such bad state occurs for our users. A detailed simulation of the model is presented and described.

Keywords: Uppaal; system verification; system modeling; Tennis training; time and speed; Safety.

1. Introduction

In a world of technology and sedentary lifestyle, sports and people interested in physical activities are rare to find. [1] Tennis being a sport that is played worldwide, interest of youth plus other age spans, learning it to perfect it to the extent that one does not injure himself/herself is crucial. The goal is to provide such excellent training to the users so that more people are attracted to this sport while aiming to excel in it.

Tennis is such a sport in which the player can be easily injured by a small angle mistake while service or playing, or by not registering the ball at the right time. To develop a system that allows tennis players to learn various types of services at their right angles [2]. Also, it allows the user to practice their game with a ball machine. The safety, timing plus speed is very critical for the user. So before deploying such an app/system into the world it needs to be modeled and verified to avoid any mishaps [3]. The ball machine can only be integrated into the system after having a complete idea on the required speed, angle and the time interval between shots, integration before any idea or testing of above mention essential aspects can only lead to wastage of time, money, and resources.

Email addresses: Kfatima.bese19seecs@seecs.edu.pk (K. Fatima), Jabbasi.bese19seecs@seecs.edu.pk (J. A. Abbasi), Mkhalid.bese19seecs@seecs.edu.pk (M. K. Khalid), sidra.sultana@seecs.edu.pk (S. Sultana)

Hence the system is to be properly and thoroughly verified using software tools before progressing with its hardware implementation or construction in real world. In this paper we will3 model and verify our Trainify system using Uppaal model checker. After it the timed automata and information will be used to implement the system into the real world considering it to be fault free. We will verify various properties that are crucial to declare a system safe like the safety, deadlock freeness, reachability, liveness, mutual exclusion, utility and fairness, and bounded response, which Uppaal's inbuilt verification module will allow us to do easily [4].

2. Literature Review

The model for this training app has not been implemented. Before working on the formal verification of this system and its timed constrained modules, we did a thorough study on the tennis game. We studied about the speed and various angles required to classify a service [5]. We also studied and did interview of the company we were dealing with to get maximum intel on the working of the project [6]. How the ball machine works, the live recording module. We gathered enough knowledge to let us know how to work and what to do. We also did some research on how to use the Uppaal tool and did the best we could [7].

3. System Modeling [8]

In this section we give a system overview. Details on the Uppaal model checker are given and the timed automata of Trainify system is detailed.

A. System Overview

In our system we have 2 modules that we are dealing with. One is the Live Recorder and the other one is the Ball Machine. Here when the phone is set and the recording is started, the Ball Machine starts the practice session, and it consists of 5 shots. After which the session is ended, and the recorder classifies the shoot as types of services. Then the result is given. A new session can be started after this.

B. Uppaal

[9] We used the tool Uppaal Model Checker, which is an integrated tool environment that gives us the ability to model and verify the behavior of a system, to verify and check our system to our best of ability. Uppaal can handle real time constraints of a system which makes it very useful and powerful. Various properties like deadlock, safety, bounded response, reachability, and liveness can be checked which is a unique feature of Uppaal. Also, Uppaal has an easy-to-use interface which makes it easier to model and execute a system for a new user as well.

Moreover, Uppaal displays the sequence diagram of the system as it is being executed so every state can be checked and the user can see of the states are being changed in the same sequence as intended.

There are also a lot of projects which have been tested by Uppaal which further provided me with the confidence to use it as the model checker and verifier for this project.

C. Timed Automata

We also used the Uppaal model checker to make the timed automata for our system. We made a Live Recorder and Ball Machine. The Fig. 1 is of Recorder and Fig.2 is of Ball Machine. Here we have a recorder that when starts recording, movement monitoring is started and along with that practice is started, after 5 shots, the recorder monitors and classifies the services into its various types, after that, the result signal is sent from the ball machine to the recorder and the results are given. Then after that a new session can be started and a new practice can be started. Then a new session can be started by initializing the clocks.

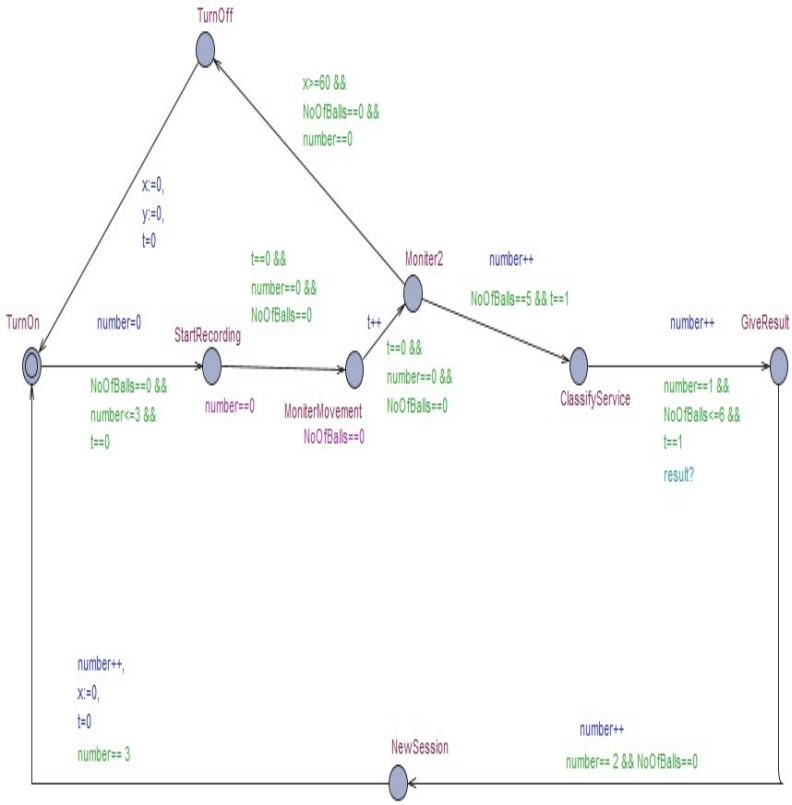

FIGURE 1 RECORDER

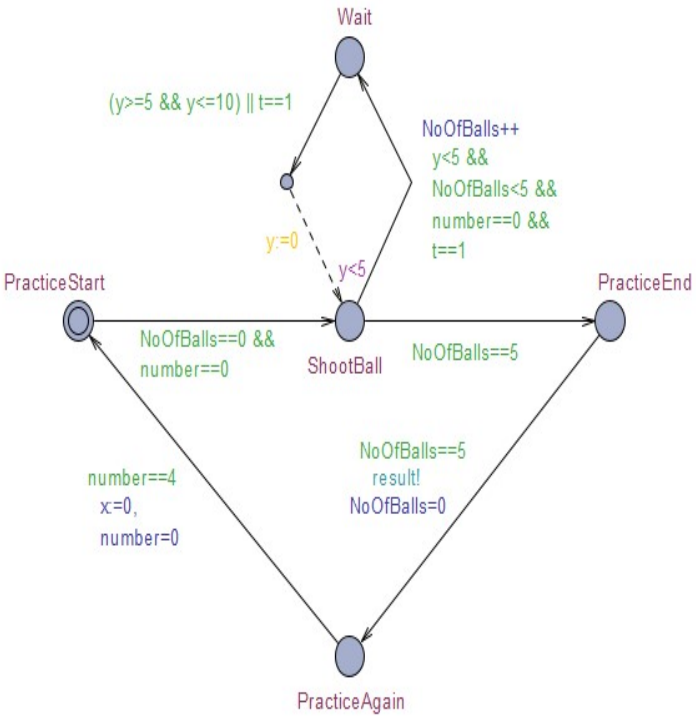

FIGURE 2 BALL MACHINE

The sequence diagram attached below in Fig.3 and Fig.4 display the sequence with which our system is supposed to work. The states and transitions of the whole system can also be seen on the Uppal Model Checker. The sequence diagram can also be used to verify the correctness of a system also to verify that no deadlock is being reached.

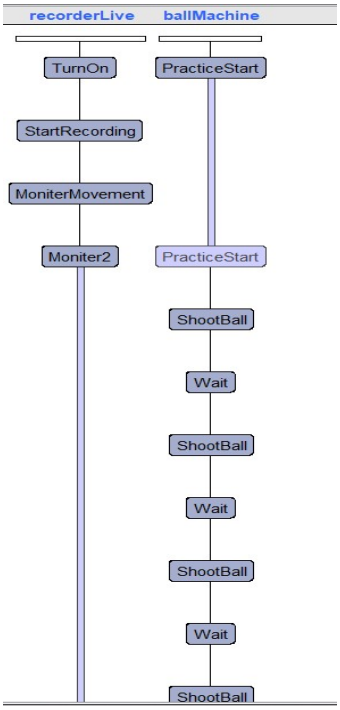

FIGURE 3 SEQUENCE DIAGRAM 1

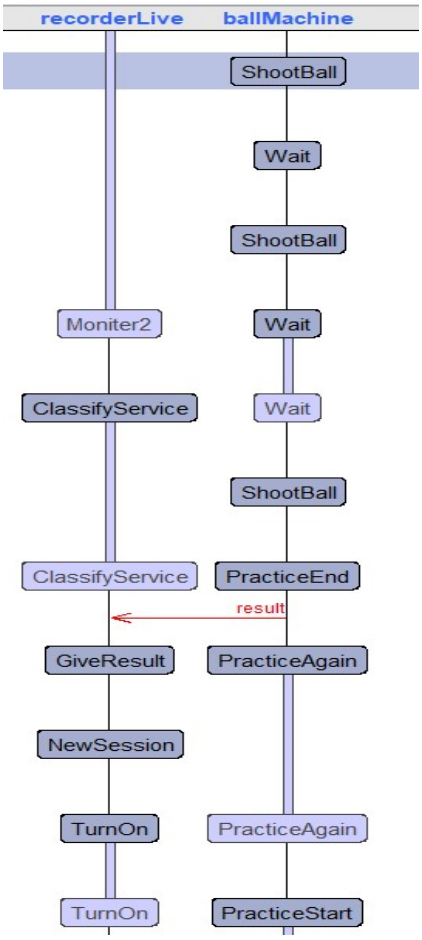

FIGURE 4 SEQUENCE DIAGRAM 2

4. System Verification

[10] The system was verified using the Uppaal model checker to ensure the safety, reachability, liveness, partial deadlock, and Bounded Response of the system.

A. Deadlock freeness

- There are no deadlocks in the system at all. A[] not deadlock
- Eventually there would not be a deadlock.

  E<> not deadlock

B. Safety

- If the recorder has been on for less than 60 seconds, Turn off state should not be reached.

  A[] !((x<60) && recorderLive.TurnOff)
- The automata can never reach the Shoot ball state as long as y is greater than 5 seconds. It only reaches it when y becomes less than 5.

  A[] !((y>5) && ballMachine.ShootBall)

C. Reachability

- The recorder is eventually turned On.

  E<>(recorderLive.TurnOn)
- The Recorder is eventually Turned Off.

  E<>(recorderLive.TurnOff)
- The Recorder will eventually Start Recording.

  E<>(recorderLive.StartRecording)
- The Recorder will Monitor Movement Eventually.

  E<>(recorderLive.MoniterMovement)
- The Recorder will eventually monitor again.

  E<>(recorderLive.Moniter2)

- The Recorder will eventually classify the service into its types.

  E<>(recorderLive.ClassifyService)

- The Recorder Will Eventually Give Results.

  E<>(recorderLive.GiveResult)

- The Recorder Will Eventually start a new session.

  E<>(recorderLive.NewSession)

- The Ball Machine will eventually start practice.

  E<>(ballMachine.PracticeStart)

- The Ball Machine will eventually practice again.

  E<>(ballMachine.PracticeAgain)

- The Ball Machine will eventually reach a state where practice ends.

  E<>(ballMachine.PracticeEnd)

- The Ball Machine will eventually reache a state for wait.
  E<>(ballMachine.Wait)
- The Ball Machine will eventually reach the state to shoot the ball.
  E<>(ballMachine.ShootBall)

D. Liveness

- There will eventually be a state where the result would be given.
  E<>(recorderLive.GiveResult)
- There will eventually be a time in automata where the service will be classified.
  E<>(recorderLive.ClassifyService)

E. Bounded Response

- The recorder will turn off after the time interval of (0,60) second has passed.
  E<>((x<=0 && x>=60) imply recorderLive.TurnOff)
- A new ball should be shot from the ball machine within a time interval of [05,10] seconds.
  E<>((y>=5&&y<=10)imply ballMachine.ShootBall)

All the above-mentioned properties are verified especially that of safety and there is no deadlock in our system. Some of the verified properties are displayed in Fig.5. [11] [12] [13]

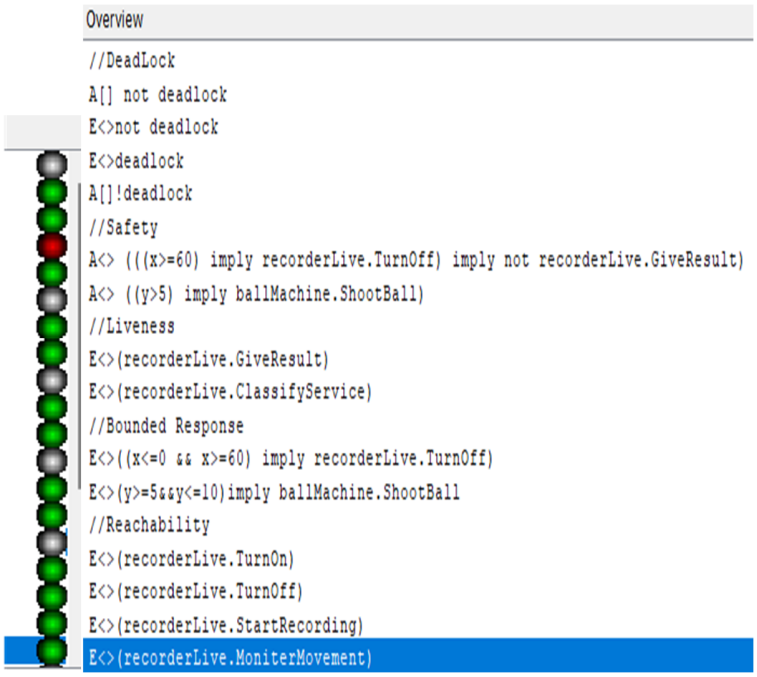

FIGURE 5 VERIFICATION PROPERTIES

5. Conclusion and Future Work

In this paper, we have modelled and verified our Trainify system to our best. The safety properties were verified ensuring that there would not be a time constraint violation in our system and ensures the reliability of our real timed automata. The result channel is used to enable both timed automata to communicate with one another safely and efficiently. Here it is also ensured that the reachability property have been satisfied for all the states. ensuring that there is no such state in the system that is not being accessed.

There were many difficulties faced by us in understanding the system and giving our understanding a shape of automata. As our future work, we will increase our work on this system, fulfill our lacking and any remaining errors and include more modules into Uppaal.

References

[1] Wikipedia,[Online]. Available: https://en.wikipedia.org/wiki/Tennis

[2] Tennis Companion, [Online]. Available: https://tenniscompanion.org/types-of-serves/

[3] P. U. The Role Of Verification and Validation, *IOSR Journal of Computer Engineering.* **5**, 17-20 (2012).

[4] G. B. A. D. K. L. P. P. and W. Y. Developing UPPAAL over 15 years, *Software: Practice and Experience.* **41**, 133-142 (2011).

[5] Feel Tennis, [Online]. Available: https://www.feeltennis.net/tennis-serve-contact/

[6] Alfa Bolt, [Online]. Available: https://www.alfabolt.com/

[7] A. David, G. Behrmann and K. G. Larsen. A tutorial on uppaal., in *Formal methods for the design of real-time systems.* 200-236 (2004).

[8] A. D. K. G. A. L. M. M. and D. Poulsen Uppaal. SMC Tutorial, *International Journal on Software Tools for Technology Transfer (STTT).* **17**, 1433-2779 (2018).

[9] Formal Methods in the Teaching Lab Examples, Cases, Assignments and Projects Enhancing Formal Methods, ON, Canada. 61-62 (2006).

[10] Julian. The Uppaal Model Checker-Julian, [Online]. Available: http://ppedreiras.av.it.pt/resources/empse0809/slides/TheUppaalModelChecker-Julian.pdf.

[11] M. P. J. and G. V. A. A Study Towards the Application of UPPAAL Model Checker, in *3rd Workshop School on Theoretical Computer Science.* 5-8 (2015).

[12] A. H. K. L. M. M. B. N. P. P. and A. S. Testing real-time systems using UPPAAL, in *Formal Methods and Testing*. 77-117 (2018).

[13] B. Nielsen, M. M. and K. G. L. Online Testing of Real-time Systems Using UPPAAL, International Workshop on Formal Approaches to Software Testing. 79-94 (2004).