MIDDLE EAST TECHNICAL UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT

Software Requirements Specification

YOLO

Mustafa Sezgin
2380863

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose of the System

YOLO is a social robot that encourages children's creativity. This robot was designed to be utilized by kids during free play, when they may use it as a character in their stories. YOLO employs creative tactics during play to encourage the invention of new storylines. As a result, throughout the engagement, the robot functions as a tool that has the ability to promote creativity in youngsters. YOLO, in particular, can encourage divergent and convergent thinking when it comes to tale development. YOLO may also take on numerous personas, allowing it to engage in socially sophisticated and engaging actions.

1.2 Scope

1.3 System Overview

1.3.1 System Perspective

YOLO is not a part of a larger system. YOLO Agent is in contact with some external entities, which are AI system, sensors and actuators.

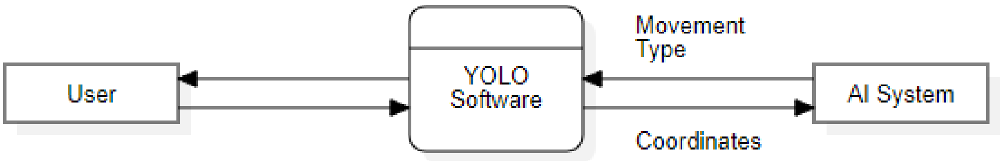

Figure 1.1: *Context Diagram*

1.3.2 System Functions

1.3.3 Stakeholder Characteristics

1.3.4 Limitations

1.4 Definitions

2 References

[1] Patrícia Alves-Oliveira, Patrícia Arriaga, Ana Paiva, and Guy Hofman. 2021. Children as Robot Designers. In *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction (HRI '21), March 8–11, 2021, Boulder, CO, USA*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3434073.3444650

3 Specific Requirements

3.1 External Interfaces

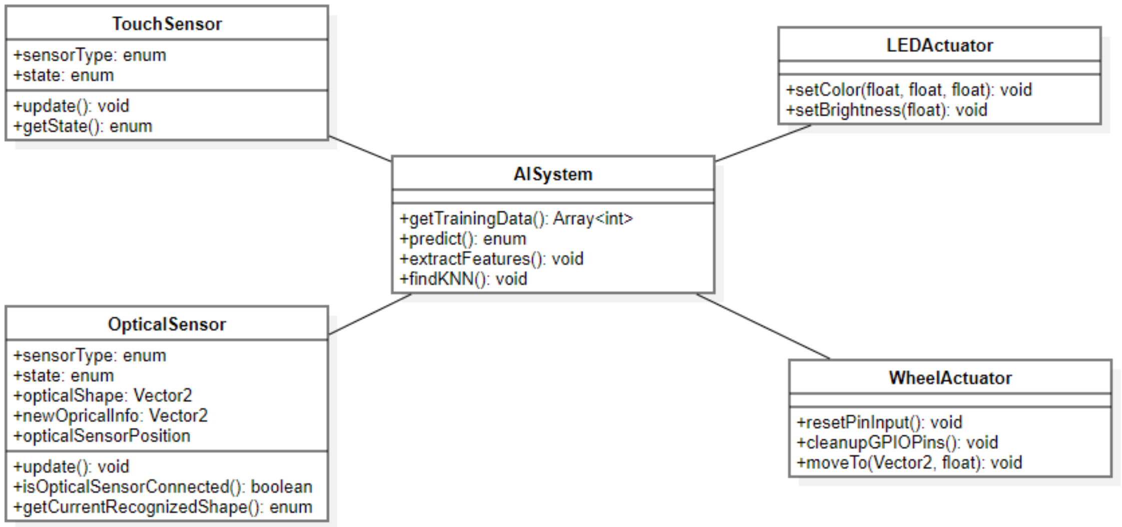

Figure 3.1: *External Interfaces Class Diagram*

3.2 Functions

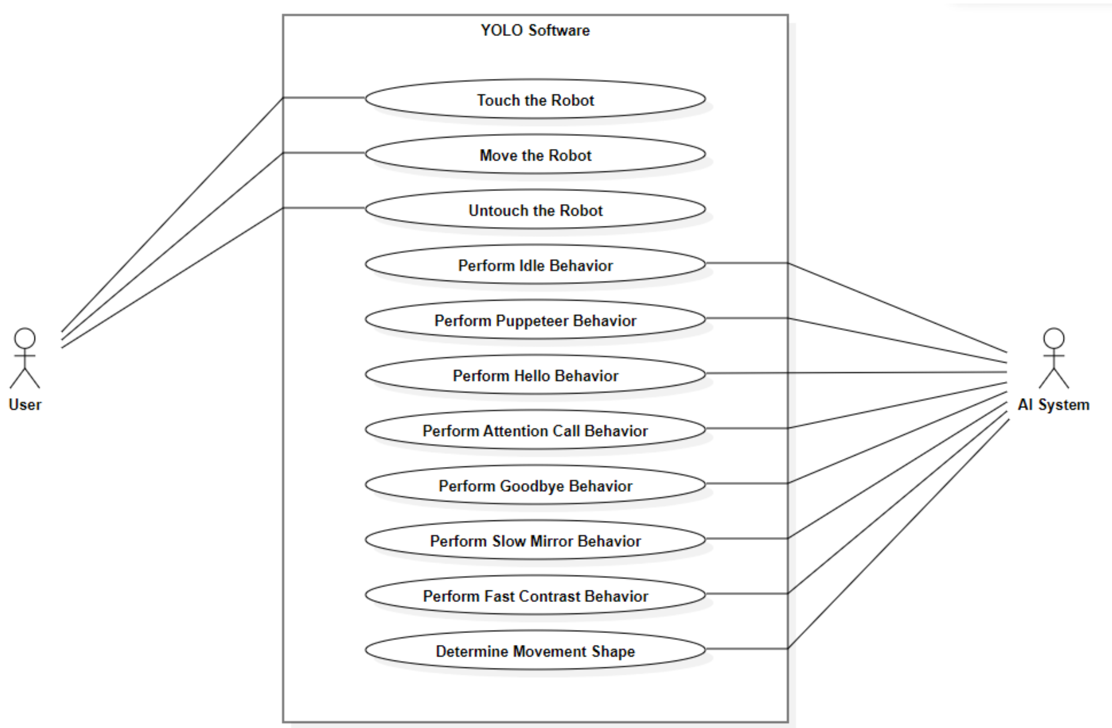

Figure 3.2: *Use-Case Diagram*

Use-case name	Touch the Robot
Actors	User
Description	The touch sensor on the robot detects that the user is touching the robot.
Data	Touch information
Preconditions	The robot must be in idle mode.
Stimulus	The touch sensor senses the robot is being touched.
Basic flow	Step 1: The user makes physical contact with the touch sensor. Step 2: The touch sensor gets stimulated.
Alternative flow	-
Exception flow	When the maximum interaction time is achieved, the interaction terminates and the robot turns off.
Postconditions	The robot performs hello behavior if the user touches the robot for the first time. Otherwise, the robot enters puppeteering mode and does not try to perform any behavior.

Table 3.1: *Touch the Robot Function*

Use-case name	Move the Robot
Actors	User
Description	The optical sensor on the robot detects that the robot is being moved by the user and collects the coordinate data.
Data	Movement information
Preconditions	The user must be touching the robot.

Stimulus	The optical sensor senses changes in the position.
Basic flow	Step 1: The user touches the robot. Step 2: The user applies force on the robot to change its position. Step 3: The optical sensor senses changes in the position and detects the speed and the direction of movement.
Alternative flow	-
Exception flow	When the maximum interaction time is achieved, the interaction terminates and the robot turns off.
Postconditions	The collected movement information is sent to the agent to determine the shape of movement.

Table 3.2: *Move the Robot Function*

Use-case name	Untouch the Robot
Actors	User
Description	The touch sensor on the robot detects that the user stops touching the robot.
Data	Touch information
Preconditions	The user must be touching the robot.
Stimulus	The touch sensor senses the robot is not being touched.
Basic flow	Step 1: The user touches the robot. Step 2: The user cuts off the physical contact with the touch sensor. Step 3: The touch sensor gets stimulated.
Alternative flow	-
Exception flow	When the maximum interaction time is achieved, the interaction terminates and the robot turns off.

Postconditions	The planning module calculates touch duration and determines the robot's behavior depending on the profile of the robot and the current phase of the storytelling arc.

Table 3.3: *Untouch the Robot Function*

Use-case name	Perform Idle Behavior
Actors	AI System
Description	While the robot is not being touched, the robot enters in idle mode where it expects an input from the user.
Data	Touch information
Preconditions	The user must not be touching the robot.
Stimulus	Any behavior execution is finished, or touch duration is smaller than the shape threshold.
Basic flow	Step 1: A behavior action has come to the end. Step 2: The robot goes into idle mode.
Alternative flow	Step 1: The user touches the robot. Step 2: The user moves the robot. Step 3: The user stops touching the robot. Step 4: The touch duration is smaller than the shape threshold. Step 5: The robot goes into idle mode.
Exception flow	When the maximum interaction time is achieved, the interaction terminates and the robot turns off.
Postconditions	-

Table 3.4: *Perform Idle Behavior Function*

Use-case name	Perform Puppeteer Behavior
Actors	AI System
Description	While being touched, the robot enters in puppeteering mode where it collects the movement data and does not try to perform any behavior.
Data	Touch information
Preconditions	The user must be touching the robot, and this must not be the first touch.
Stimulus	The user touches the robot.
Basic flow	Step 1: The user touches the robot. Step 2: The robot goes into puppeteering mode.
Alternative flow	Step 1: If the user touched the robot for the first time, then the hello behavior is performed.
Exception flow	-
Postconditions	The planning module determines the next behavior depending on the touch duration and the current profile of the robot.

Table 3.5: *Perform Puppeteer Behavior Function*

Use-case name	Perform Hello Behavior
Actors	AI System
Description	The interaction starts when the robot is activated. The planning module determines that the user touched the robot for the first time and performs hello behavior.
Data	Touch information, movement speed and direction, LED color and brightness

Preconditions	The user must be touching the robot, and this must be the first touch.
Stimulus	The user touches the robot for the first time.
Basic flow	Step 1: The user touches the robot for the first time. Step 2: The robot performs hello behavior.
Alternative flow	Step 1: If this is not the first time the user touched the robot, then the puppeteer behavior is performed.
Exception flow	-
Postconditions	Idle behavior is performed.

Table 3.6: *Perform Hello Behavior Function*

Use-case name	Perform Attention Call Behavior
Actors	AI System
Description	YOLO is a socially interactive robot. Therefore, it alerts the user by performing attention call behavior if the user is not in contact with it for a while. The minimum time required for this behavior is called attention call threshold.
Data	Touch information, movement speed and direction, LED color and brightness
Preconditions	The user must not be touching the robot for some time that is greater than the attention call threshold.
Stimulus	The time since the last touch passes the attention call threshold.

Basic flow	Step 1: The user stops touching the robot. Step 2: The user leaves the robot untouched for a while. Step 3: If the time without touching goes beyond attention call threshold, attention call behavior is performed.
Alternative flow	-
Exception flow	When the maximum interaction time is achieved, the interaction terminates and the robot turns off.
Postconditions	Idle behavior is performed.

Table 3.7: *Perform Attention Call Behavior Function*

Use-case name	Perform Goodbye Behavior
Actors	AI System
Description	The interaction terminates when the maximum interaction time is achieved. The planning module detects that the robot is active for the maximum time allowed and decides the robot to turn off. Before that, the goodbye behavior is performed.
Data	Touch information, movement speed and direction, LED color and brightness
Preconditions	Any other behavior must finish.
Stimulus	The time the robot is active passes the maximum interaction time.

Basic flow	Step 1: The robot is activated. Step 2: The user interacts with the robot for a while. Step 3: If the time with interaction goes beyond maximum interaction time, attention call behavior is performed. Step 4: The robot is inactivated.
Alternative flow	-
Exception flow	-
Postconditions	The robot is inactivated.

Table 3.8: *Perform Goodbye Behavior Function*

Use-case name	Perform Slow Mirror Behavior
Actors	AI System
Description	The robot predicts the shape of movement the user makes and imitates it if the current storytelling phase is rising action or falling action. This behavior stimulates convergent thinking.
Data	Movement shape, speed and direction
Preconditions	The robot must be in the rising action phase or the falling action phase of the storytelling arc, and the touch duration must be greater than the shape threshold.
Stimulus	The user stops touching the robot after moving it for a longer period of time than the shape threshold.

Basic flow	Step 1: The user touches the robot. Step 2: The user moves the robot. Step 3: The user stops touching the robot. Step 4: If the touch duration is greater than the shape threshold, AI System predicts the movement shape. Step 5: If the robot is in the rising action or falling action stage of the storytelling arc, then the mirroring behavior is performed. Step 6: The agent activates actuators via control module to imitate the shape.
Alternative flow #1	Step 4: If the touch duration is smaller than the shape threshold, Idle Behavior is performed.
Alternative flow #2	Step 5: If the robot is in the climax stage of the storytelling arc, then the contrasting behavior is performed.
Exception flow	Step 4: If the shape is not recognized by the AI System, the exception is logged and the default shape is returned.
Postconditions	Idle behavior is performed.

Table 3.9: *Perform Slow Mirror Behavior Function*

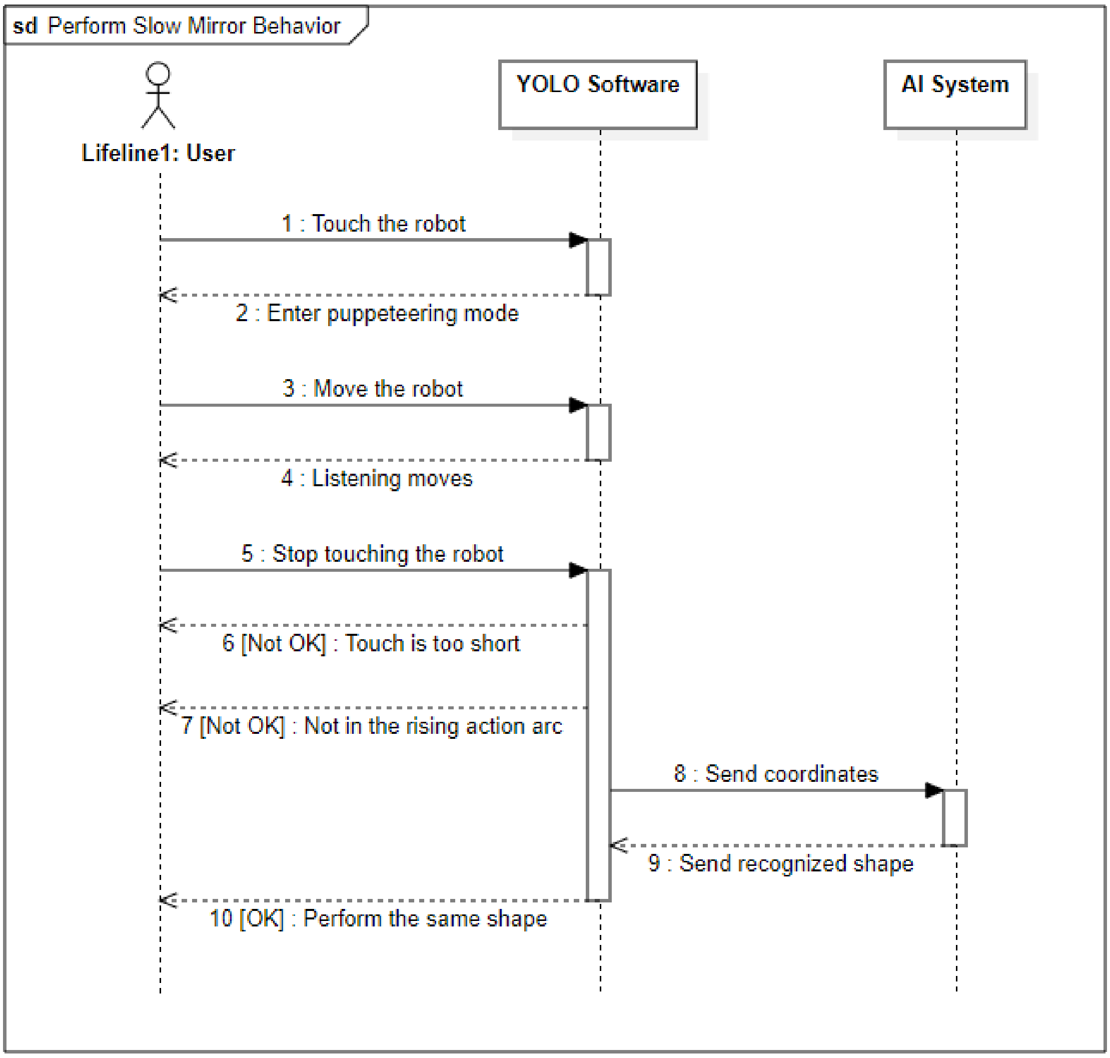

Figure 3.3: *Sequence Diagram for Perform Slow Mirror Behavior Function*

Use-case name	Perform Fast Contrast Behavior
Actors	AI System
Description	The robot predicts the shape of movement the user makes and imitates it if the current storytelling phase is climax. This behavior stimulates divergent thinking.
Data	Movement shape, speed and direction

Preconditions	The robot must be in the climax phase of the storytelling arc, and the touch duration must be greater than the shape threshold.
Stimulus	The user stops touching the robot after touching it for a longer time than the shape threshold.
Basic flow	Step 1: The user touches the robot. Step 2: The user moves the robot. Step 3: The user stops touching the robot. Step 4: If the touch duration is greater than the shape threshold, AI System predicts the movement shape. Step 5: If the robot is in the climax stage of the storytelling arc, then the contrasting behavior is performed. Step 6: The agent activates actuators via control module to imitate the shape.
Alternative flow #1	Step 4: If the touch duration is smaller than the shape threshold, Idle Behavior is performed.
Alternative flow #2	Step 5: If the robot is in the rising action or falling action stage of the storytelling arc, then the mirroring behavior is performed.
Exception flow	Step 4: If the shape is not recognized by the AI System, the exception is logged and the default shape is returned.
Postconditions	Idle behavior is performed.

Table 3.10: *Perform Fast Contrast Behavior Function*

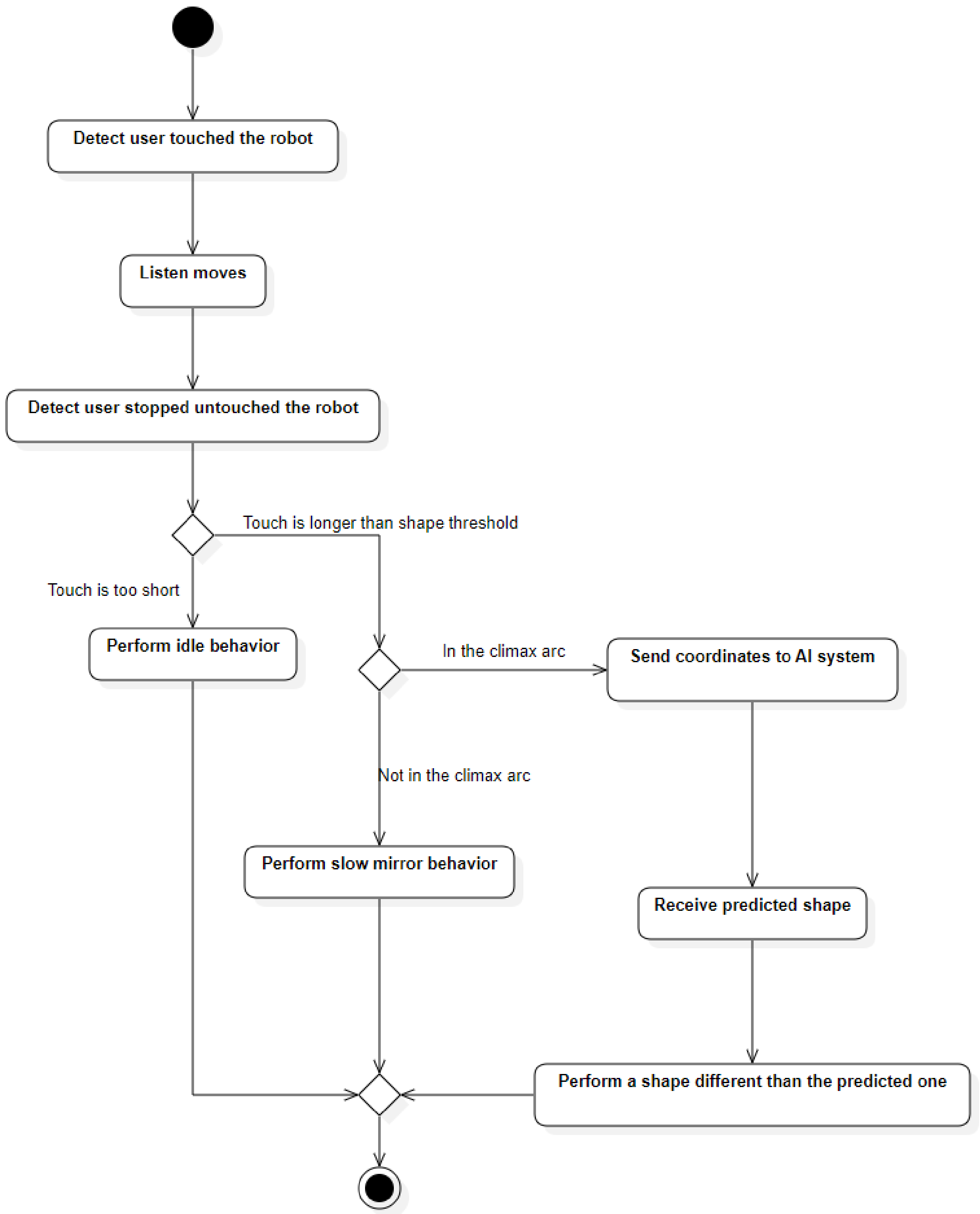

Figure 3.4: *Activity Diagram for Perform Fast Contrast Behavior Function*

Use-case name	Determine Movement Shape
Actors	AI System

Description	Movement data collected by the optical sensor are sent to the AI System. The shape of movement is predicted using machine learning algorithms.
Data	Movement information, movement shape
Preconditions	The user must move the robot and the optical sensor must collect the movement information.
Stimulus	AI System predicts the shape of movement.
Basic flow	Step 1: The user touches the robot. Step 2: The user applies force on the robot to change its position. Step 3: The optical sensor senses changes in the position and detects the speed and the direction of movement. Step 4: Collected movement information is sent to the AI system by the agent. Step 5: AI System runs machine learning algorithms to predict the movement shape. Step 6: Predicted shape is returned to the agent to be used by the planning module to determine the behavior of the robot.
Alternative flow	-
Exception flow	Step 5: If the shape is not recognized by the AI System, the exception is logged and the default shape is returned.
Postconditions	The planning module determines the robot's behavior depending on the movement shape and the current profile of the robot.

Table 3.11: *Determine Movement Shape Function*

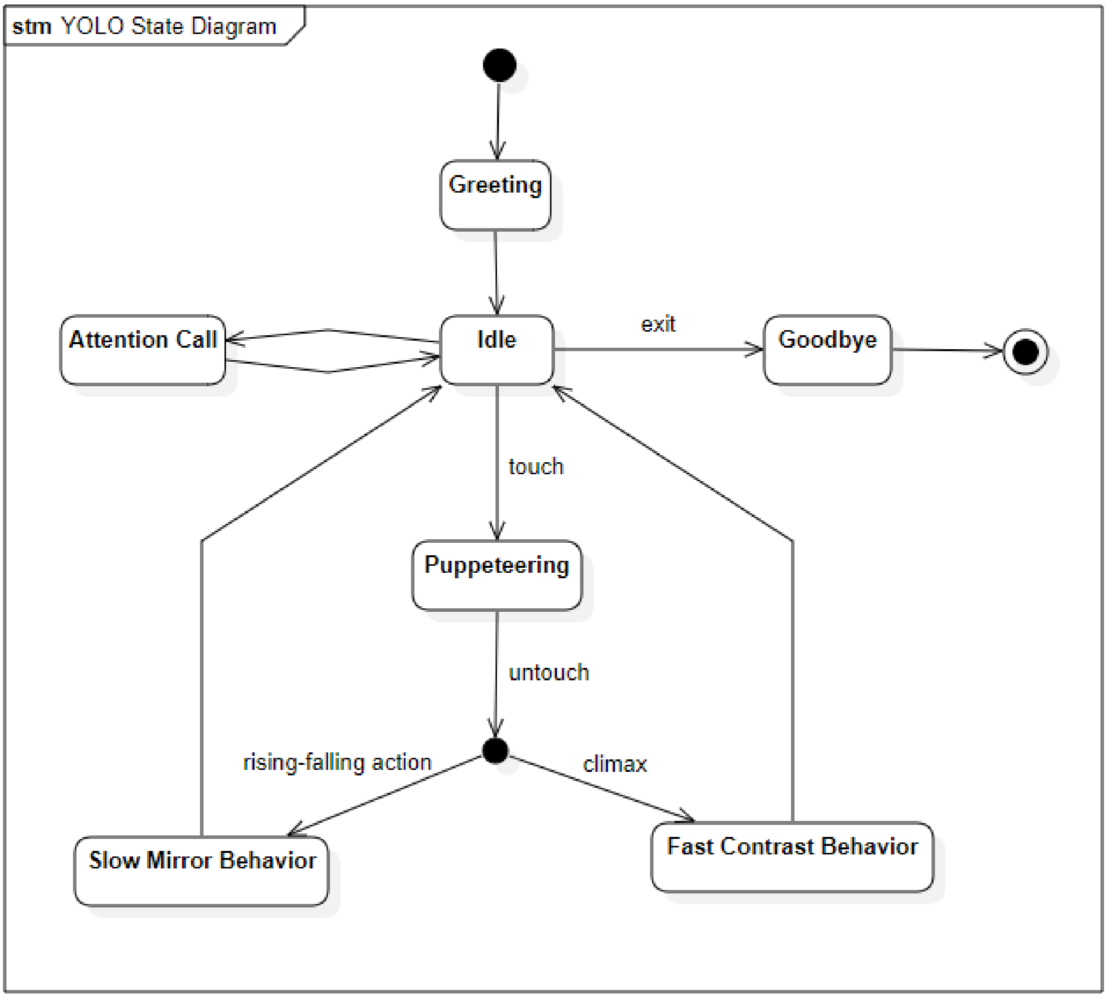

Figure 3.5: *State-chart Diagram*

3.3 Usability Requirements

3.4 Performance Requirements

3.5 Logical Database Requirements

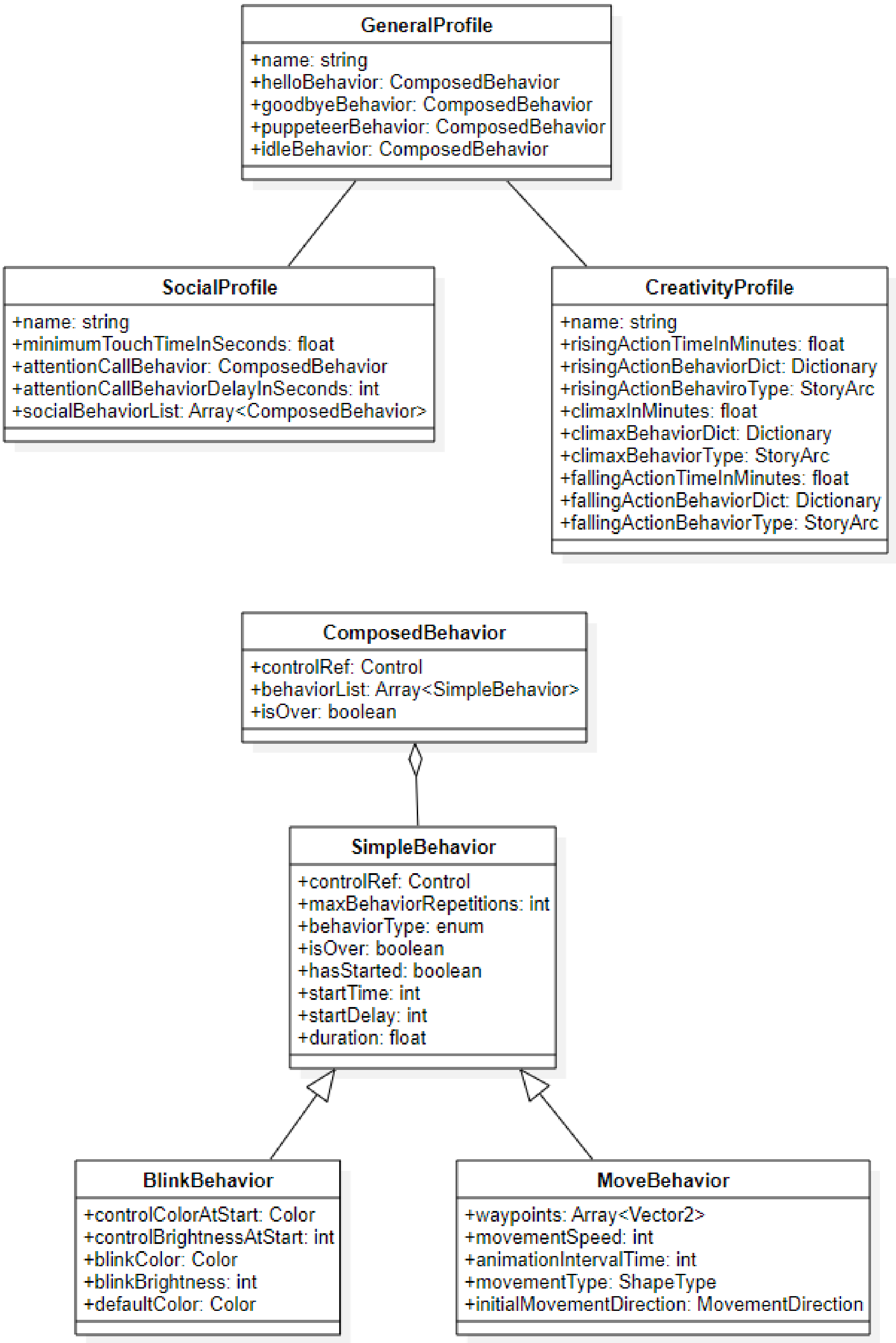

Figure 3.6: *Logical Database Class Diagram*

3.6 Design Constraints

3.7 System Attributes

3.8 Supporting Information

4 Suggestions