Eagle Knights AIBO Team

Alfredo Weitzenfeld , Jorge Adrián Martínez Gómez, Antonio Medrano Legorreta and Alejandro Chávez Bustamante

Robotics Laboratory
Instituto Tecnologico Autónomo de México
Río Hondo # 1, San Ángel Tizapan
Mexico City, MEXICO, 01000
email: alfredo@itam.mx
http://robotica.itam.mx

***Abstract*- This document describes our Four Legged team – Eagle Knights, a relatively new team with its first competition at US Open 2004.**

We describe the system architecture: Vision, Localization, Sensors, Kinematics, Wireless Communication and Behaviors, with special emphasis on our Localization system.

I. INTRODUCTION

RoboCup [1] is an international effort to promote AI, robotics and related field primarily in the context of soccer playing robots. In the Four Legged League, two teams of four robots play soccer on a relatively small carpeted soccer field.

We began this project in January 2004 and our first participation in an official competition was at the U.S. Open 2004 celebrated in New Orleans.

Our initial system architecture originated from basic ideas taken from other teams, where the architecture itself was relatively simple with no collaboration among team members. Localization itself was relatively simple, where robots were able to shoot in the right direction.

Since mid 2004, we extended our work into more efficient vision processing, better and more precise self-localization and more robust high-level behavior algorithms.

In addition to soccer playing, we have used our robots in other research projects, including biorobotics modeling and experimentation [8].

II. SYSTEM ARCHITECTURE

Our system architecture is shown in figure 1.

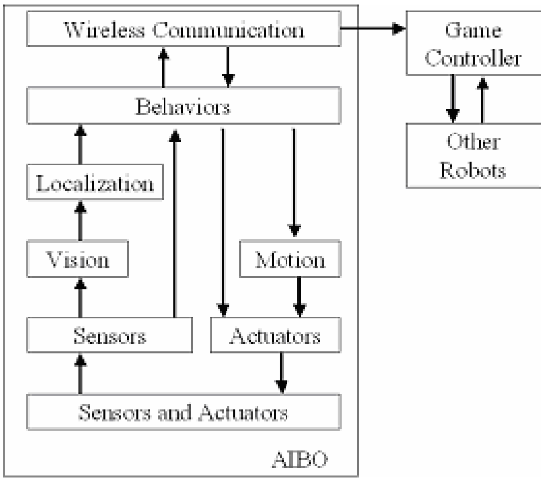

Figure 1: The system architecture includes the Sensors, Vision, Motion, Localization, Behavior and Wireless Communication modules.

The architecture includes the following modules:

1 **Sensors.** This module receives information from the sensors. We are particularly interested in vision and motor position feedback.
2 **Vision.** The vision module receives a raw image from the camera, the main system sensor, and performs segmentation over the image. It then recognizes objects in the field, such as goals, ball, landmarks and other players.
3 **Motion.** This module control robot movements, such as walk, run, throw the ball, turn to the right or left, move the head, etc. It receives commands from the behavior module with output sent to the corresponding actuators,

representing individual leg and head motor control.

4 **Actuators.** In addition to individual head and leg motor control, the system includes actuators for turning off and on head LEDs.

5 **Localization.** This module makes all the processing necessary to obtain a reliable localization of the robot in the field. In order to obtain a good localization it is necessary for the robot to perceive at least two marks, either goals or landmarks.

6 **Behaviors.** This module takes all decisions. affecting higher level robot actions. It takes input from the sensors and localization systems and sends commands to the motion and actuators module.

7 **Wireless Communication.** This module receives all commands from the external Game Controller.

In the following sections, we will explain in detail each one of these modules. If all the modules work correctly, the robot will play soccer.

III. SENSORS

This module receives sensory information from the color camera and motor position feedback. The raw camera image is passed directly to the vision module while information received from motor positions are used in making certain movement decisions at the behavior module, such as when the robot is on its back.

IV. VISION

Our original vision architecture used image segmentation from the AIBO built-in Color Detection Table. While this segmentation approach brought relatively good results, we have started to use the raw image of the camera instead, doing complete software segmentation, where our calibration method can be used in conjunction with either the CDT or our new software segmentation approach. We use external segmentation configuration files for this purpose.

The vision system we use in the AIBOs builds from our existing Small Size team vision system [9], which is relatively efficient and reliable, doing real time color and object recognition in the field.

Segmentation is done by using different color thresholds, with a calibration system allowing fast color selection. We take initial photos of objects of interest and then manually select colors that we want to distinguish. At the end of this task, we test the segmentation on real images seeing if our calibration is working properly. Figure 2 shows sample output of the segmentation calibration process.

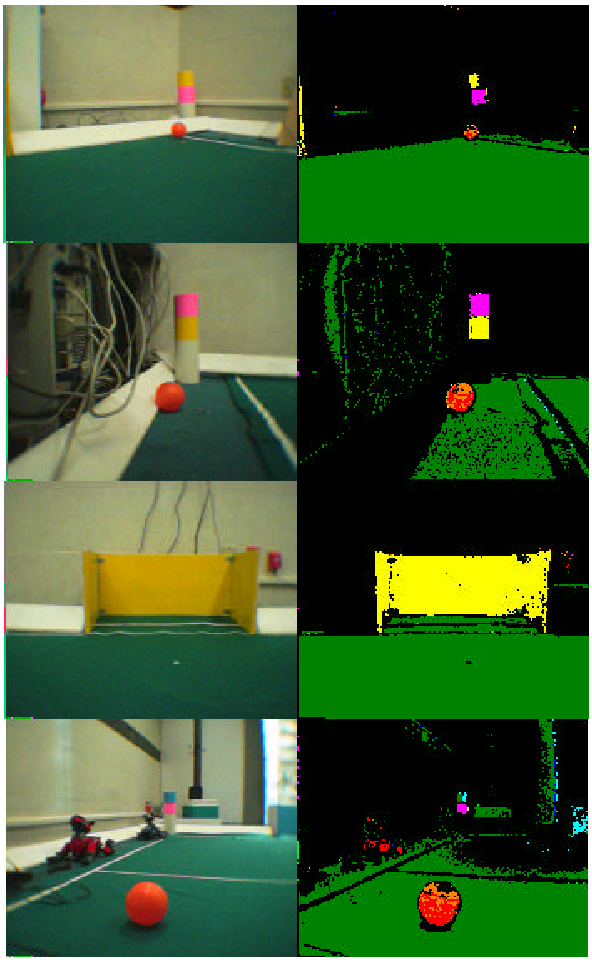

Figure 2: A sample image classified using our calibration system. Real object images are shown on the left column, while classified images are shown the on right column.

When segmentation is complete, we process same color region formations adapted from the vision system of our Small Size team [9]. Then, a Run Length Encode (RLE) algorithm is developed, compressing the image without information lost allowing to generate contiguous color regions.

After color regions are obtained, it is already possible to recognize objects. Objects in the field must fulfill certain requirements in order to allow some confidence that the region we are comparing is really the object that we want to identify. For example, the ball must have green in some adjacent area. A similar criteria is used to identify goals. The identification of the landmarks is a little more complex, nevertheless after a bit more elaborated comparison process it is possible to identify landmarks.

After objects have been identified, a data structure is generated containing objects positions as well as some additional objects characteristics, including corresponding region area and coordinates in the image.

V. MOTION

This module is responsible for robot motion control, including, walking and kicking among other set of movements. We have developed a number of different routines depending on team roles. For example, the goalie has different motions in contrast to other team players. This also applies to different head kicks and movements in general.

VI. ACTUATORS

This module is responsible for turning off and on the head and tail LEDs. This module receives commands from the behavior module to indicate the particular action being currently performing the robot. Also this module shows the state of the Game Controller.

VII. LOCALIZATION

Localization is essential in having complex behaviors and dynamic player roles. In the last months we have concentrated in developing an efficient and reliable localization system.

We have started by producing a map storing distances between field landmarks and different AIBO locations around the field. In this measurements, it is important that the robot can distinguish at least among two marks.

Our measurement algorithm uses an inverse exponential function that takes as input the visual field landmark region and throws as result the distance to the object. In order to obtain this relation, we took thousands of measurements at different distances of the area of the object that we are interested in. In order to make the system more reliable, we took measures each five centimeters. The distance rank we use runs from 15 centimeters to 4 meters; after four meters it is very difficult to notice a difference between the object and the noise.

By use of interpolation we calculated in Matlab the coefficients for a third degree equation matching the measured data. We tested these results using different distances obtaining results with an average error of less than 5 centimeters.

After the measurement curve has been obtained we applied a triangulation algorithm, based on only two landmarks, to obtain the actual robot position on the field on a two dimensions plane. Finally, to obtain the angle of the robot direction we used two vectors whose origin is at the robot current location and pointing to the coordinates of the reference marks center. Considering that the distance between each marks is identical, it is likely that the resulting vector of the sum of both previous vectors is the vector that is in the middle of both marks and the angle of this vector is the angle of the robot in the field. After making several tests, we obtained that the error for this calculation was insignificant.

When testing our algorithm in real time with the robot walking, we watched that in many occasions we collected discontinuous data between successive frames. To resolve this problem, we developed a probability algorithm considering historical data from previous robot position calculations in order to obtain average measurements. This algorithm is based in the principles of Monte Carlo, assuming that robot positions represent particles in a probability distribution.

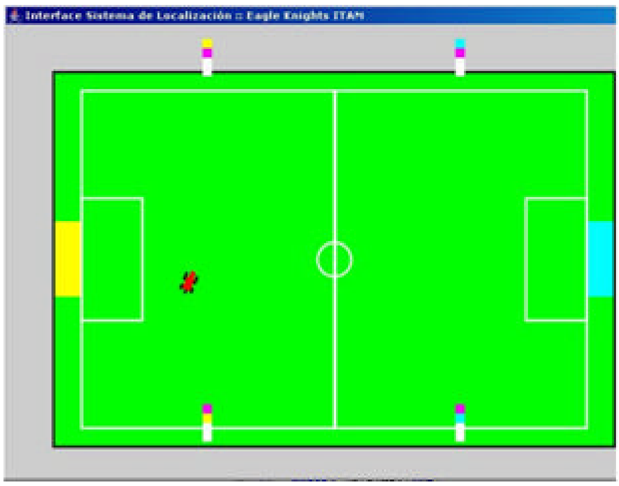

Figure 3. The graphic interface of the localization system. With this interface we can compare real with estimated positions.

The results obtained show an average mean error of less than 10 square centimeters, a small error considering that the field has approximately 19,000 square centimeters.. Although we can lower the average mean error beyond 10 centimeters, such an improvement will considerably affect the performance of the computation.

VIII. Behaviors

This module receives information from motion sensors and the Localization system sending its output back to actuators in the AIBO.

In defining our team robot behaviors, we specified two types of players: Attacker and Goalie. Each one has a different behavior that depends on the ball position and Game Controller, as described next:

Goalie

The goalie is a state machine that is explain beneath this lines. In the figure 4 we shows the diagram.

1 Ready_to_Start: When the AIBO is switched on waits the referee signal to start the game.
  - a Initial_position: The player Stands up and look for the ball with the head.

2 Playing: When the referee box sends the Playing message or the head switch is touched.
  - a See_the_ball: If the player sees the ball
    - i Look_the_ball: If the ball is far from him, he follows the ball with the head.
    - ii Go_to_ball: If the ball is not far from it, the goalie tries to approach to the ball.
    - iii Throw: If the ball is near the goalie, he throws to the ball
    - iv Go_back: The goalie walks back some steps.
    - v Search_the_ball: The goalie searches the ball. If found then he jumps to the "See_the_ball" state, otherwise, he jumps to the "cant_see_ball" state.
  - b Cant_see_ball: If the player can't see the ball
    - i Find_goal: The goalie searches the goal. If found and near the goal, then he jumps to the "Defending_goalie" state; if found and far from the goal, then he jumps to the "Go_to_goal" state, otherwise, he jumps to the Cant_see_goal_and_ball state.
    - ii Defending_goalie: The goalie walks on the goal line. The goalie tries to defend the goal.
    - iii Go_to_goal: The goalie walks to the goal.
    - iv Search_the_ball: The goalie searches the ball. If found he jumps to the "See_the_ball" state; otherwise, he jumps to the "Cant_see_ball" state.
  - c Cant_see_goal_and_ball: If the player can't see the goal and the ball.
    - i Search: The goalie moves to all directions searching the goal and the ball. If he sees the ball jumps to "See_the_ball" state. If he sees the goal jumps to the "Cant_see_ball" state.

3 Penalized: When the referee sends the Penalized message, the player jumps to the "initial_position" state.

4 Finished: When the referee sends the Finished message, the player jumps to the "initial_position" state.

Attacker

The attacker is a state machine that is explained beneath these lines. In the figure 5 we show the diagram.

1 Ready_to_Start: When the AIBO is switched on waits the referee signal to start the game.
  - a Initial_position: The AIBO Stands up and look for the ball with the head.

2 Playing: When the referee box sends the Playing message or the head switch is touched.
  - a If the AIBO see the ball:
    - i Go_to_ball: the player approaches to the ball.
    - ii Prepare_to_Kick_the_ball: If the player is near the ball, jumps to "Localization" state.
    - iii Localization: Looks for the markers and the goal and choose a kicker type.
    - iv Kick_the_Ball: The player has two types of kicker:
    - v Kick: If the player has to move to any angle except 90 degrees, the player kicks with the front legs. Kick_with_head: If the player has to move 90 degrees, the player kicks with the head.
  - b If the AIBO doesn't see the ball:
    - i Search_the_ball: The player walks forward moving the head to all directions until he finds the ball. If the player makes many steps, jumps to explore the field state. If the player sees the ball, jumps to "Go_to_Ball" state.
    - ii Explore_the_field: The player turns 45 degrees and jumps to the "Search_the_ball_state".

3 Penalized: When the referee sends the Penalized message, the player jumps to “initial_position” state.
4 Finished: When the referee sends the Finished message, the player jumps to “initial_position” state.

IX. Wireless Communication

This module receives commands from the Game Controller and passes them to the Behaviors module accepting connections to the computer using either TCP or UDP protocol.

X. Conclusions

We have presented the system architecture for the Eagle Knights AIBO team.

We have concentrated primarily in the vision module, building from our existing small size architecture and developed a localization system based on a Monte Carlo algorithm.

Our team development initiated in 2004, having competed at that year’s US Open. We expect better results with our new localization system.

XI. References

[1] RoboCup Technical Committee. “Sony Four Legged Robot Football League Rule Book.” Mayo 2004.
RoboCup. Official Web Site URL:http://www.robocup.org

[2] AIBO. Official Web Site. URL: www.aibo.com

[3] Bruce, J., Tucker B, y Veloso. M. “Fast and Inexpensive Color Image Segmentation for Interactive Robots”. 2002.

[4] Chen. J, and others. “A Description of the rUNSWift 2003 Legged Robot Soccer Team”. 2003.

[5] Sony Corporation. “OPEN R SDK – Model Information for ERS-210”. 2003. URL: http://www.openr.org/

[6] Martínez Gómez., Jorge A. “Diseño del Sistema de Localización para robots Autónomos”. ITAM, México, 2005.

[7] Official Eagle Knights Web Site. URL: http://robotica.itam.mx

[8] Flores Ando, F., and Weitzenfeld, A., 2004, Alternatives for the Implementation of the Crowley-Arbib Saccade Model in a Robotic System, Proc. 1st IEEE-RAS Latin American Robotics Symposium, ITAM, Mexico City, October 28-29.

[9] Martínez-Gómez, L.A., and Weitzenfeld, A., 2004, Real Time Vision System for a Small Size League Team, Proc. 1st IEEE-RAS Latin American Robotics Symposium, ITAM, Mexico City, October 28-29.

Goalie

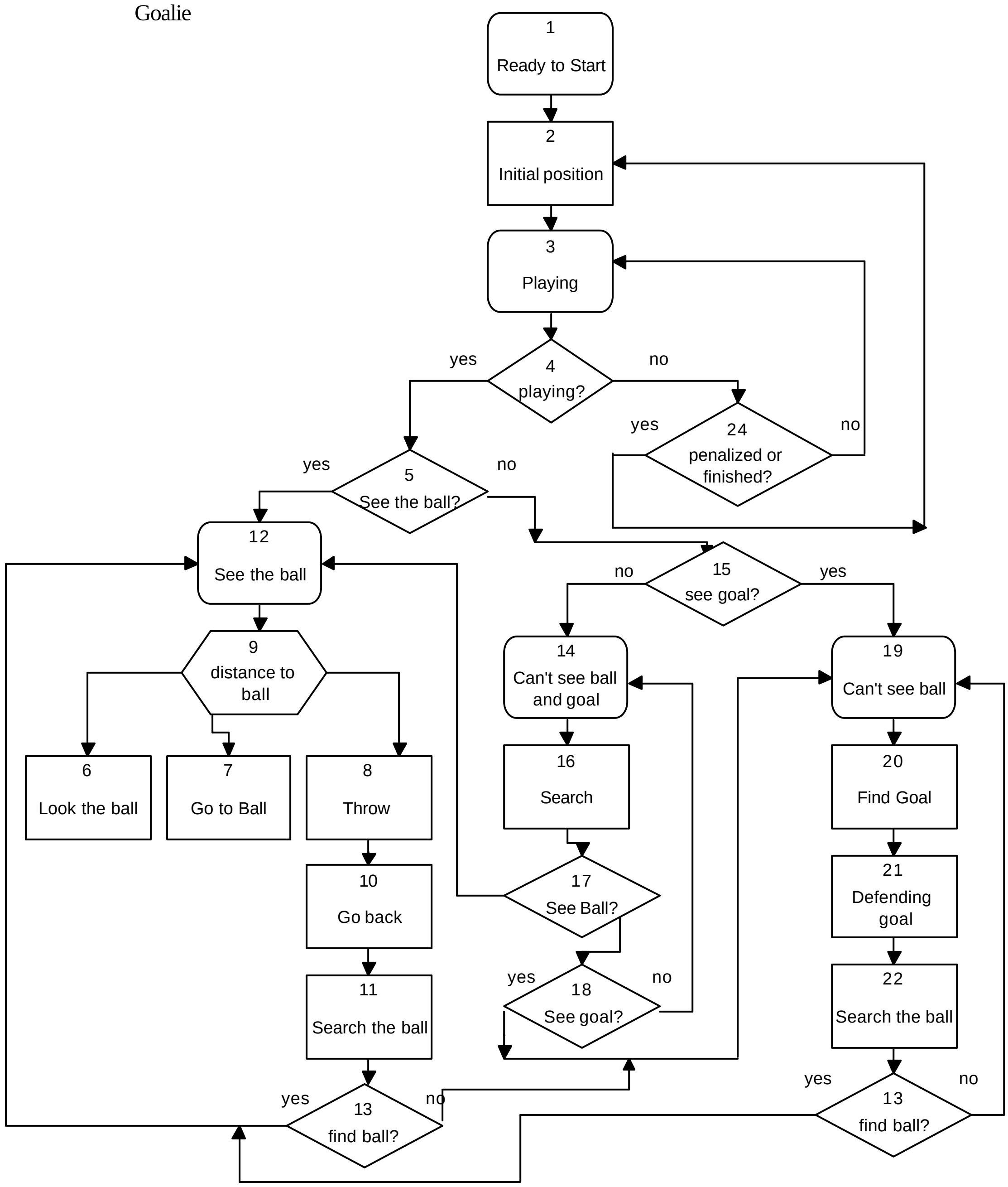

Figure 4: The goalie state machine. This diagram shows how the goalie behave in a competition.

Attacker:

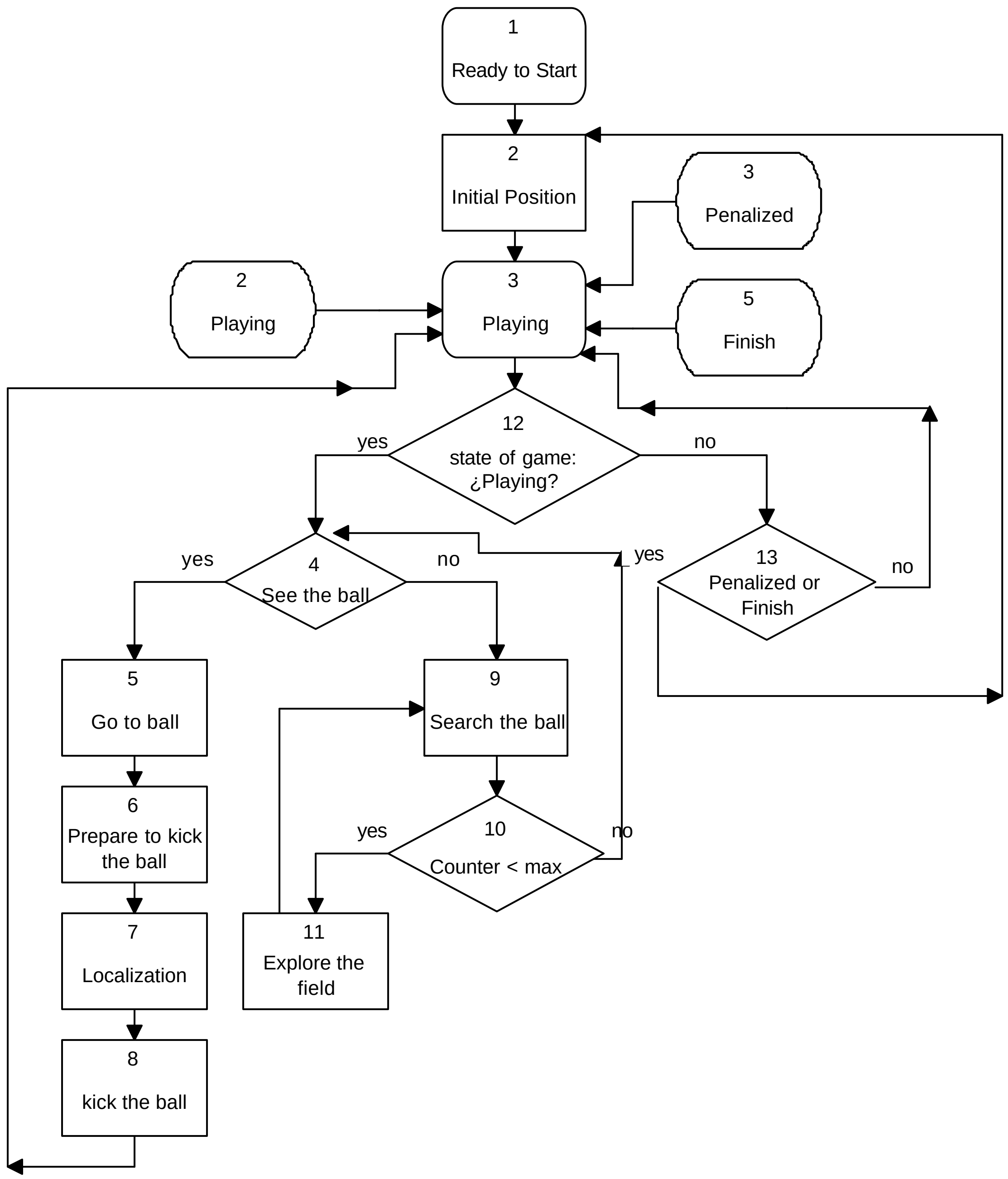

Figure 5: The Attacker state machine. This diagram shows how the attackers behave in a competition.