Presented at ICTOpen 2019, https://ict-research.nl/ict-open/. 20th of March 2019

How to model soccer robot software? – A comparison of approaches

Suzana Andova, Eric Dortmans, Teade Punter
Fontys University of Applied Sciences, Lectoraat High Tech Embedded Software
{s.andova, e.dortmans, teade.punter}@fontys.nl

Abstract
We perceive soccer robots as an interesting example of cyber physical systems. In the Robocup MSL[1] a multirobot team existing of five holonomic robots (Siegwart et al, 2011) should be controlled in a way that they play soccer against a competing team of another five robots. Each of the autonomous robots has to play a match of two times fifteen minutes. Because of the autonomy of the robots, their behaviour has to be implemented beforehand of the matches; communication and interaction with the robots is not allowed. In these matches numerous game positions are taken by the robots, while almost an infinite set of possible robot (re-)actions can be thought of.

In setting up a robot soccer team –the RIF Robocup MSL team[2]- we face the challenge of programming the behaviour of the soccer robots. In our software architecture we apply C++ on the actuator and sensor (firmware) level. On the perception and action planning level we apply the ROS software framework[3] (Quigley et al. 2009). We think, that a modelling approach is required to deal with the robot behaviour. Therefore, we address the question: how to model the behaviour of a soccer robot?

Applying state machines is the usual approach for controlling robot behaviour. However, developing relatively complex (hierarchical) state machines has some disadvantages, like the difficulty of extending models when defining new behaviour, as well as of maintaining and understanding the model over time. We therefore have examined an alternative technique, called Behavior Trees (Colledanchise 2017, Marzinotto et al. 2014, Ögren, 2012), which application in the game industry is widespread. Furthermore, we also have to cope with not well understood behaviour in robots. In such case we may have to make the robot learn new behaviour. Therefore we also have looked at machine learning, in particular (deep) reinforcement learning (Sutton & Barto, 2018) to improve the learnability of the robots.

In our presentation we focus on the comparison of state machines, behavior trees and machine learning. We compare the principles by showing how to combine elementary skills to build behaviours. We do so by providing an example from robot soccer.

Keywords: autonomous robot, behavior trees, hierarchical state machines, machine learning, Robocup Soccer Robots.

1. Model as a hierarchical state machine

The example we take for the presentation is a common general soccer pattern in a Robocup match, which consists of several skills. A robot intercepts a ball. It dribbles with the ball to contain the ball and when it perceives a goal it has two options. The first is to shoot the ball directly to the goal. The second option is to pass the ball to a teammate. In a hierarchical state machine the set of robot skills can be expressed graphically like presented below.

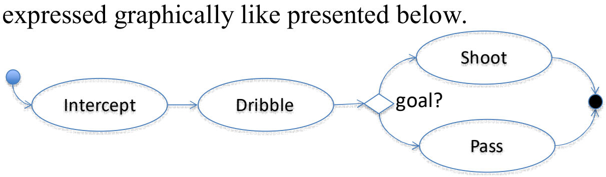

Figure 1 – General soccer situation described as state machine.

2. Model as a behavior tree

We did also model the same soccer situation as described before (in 1) according to the behavior tree format. This results in a graphical representation as presented below. In the behaviour tree approach the sequences and choices are presented as a tree. The prioritization in actions has to be read from left to right.

The benefit of the behaviour tree representation is the hierarchical approach that depicts the logic of the sequence and choices. This is represented in the tree structure in the red dotted triangle in figure 3. In fact the basic actions like intercept, dribble, etcetera are identical to the actions as depicted in the state

[1] http://wiki.robocup.org/Middle_Size_League

[2] RIF = Robot Initiative Fontys; this is a development team existing of students and lecturers of Fontys Hogeschool ICT in Eindhoven.

[3] http://www.ros.org/

machine depicted in figure 1. The tree structure in the behaviour tree representation makes it easier to change sequence, choices, that it would be to do in a state machine representation.

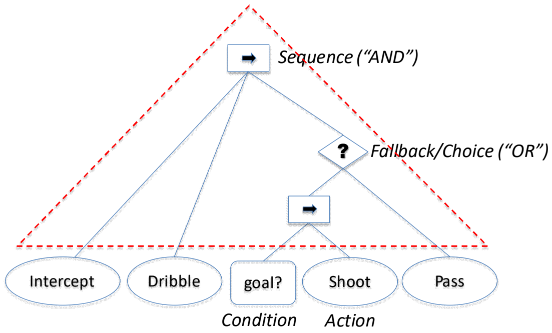

Figure 3 – Behavior tree model for soccer situation.

3. Model as machine learning
The third representation for the general soccer situation is to perceive the basic soccer actions, like intercept and dribble, but not defining their relationships and priorities beforehand. Instead we related them via a policy function that we can learn using (deep) reinforcement earning. The learning concerns the mapping between states and actions in the robot soccer match. For example, the state that a robot has already intercepted the ball, and is dribbling the ball. The robot does also perceive a goal and has to decide to shoot directly to the goal or pass to a friend robot first. The success of the choice will depend of the distance to goal as well of other robots surrounding the robot. This is a situation where learning might be useful to identify the effective action. For the model, the basic actions remain the same to previously discussed models. For the learning situation a policy function is added.

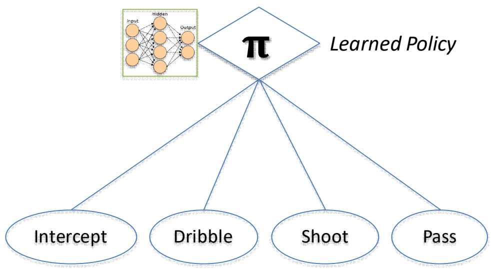

Figure 4 – Machine learning representation for soccer situation.

Acknowledgements
This work is conducted as work in the RIF[4] Robocup Soccer team of Fontysand in the ToLeRo project (Toestanden en Leren met Robots), which was funded by Centre of Expertise High Tech Systems and Materials[5].

References
- Brooks, R. (1986). A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23.
- Browning, B., J. Bruce, M Bowling, M Veloso (2004), STP: skills, tactics, and plays for multi-robot control in adversarial environments. *Proceedings IMechE. Vol. 219 Part I: J. Systems and Control Engineering*. DOI: 10.1243/095965105X9470
- Colledanchise, M. (2017). *Behavior Trees in Robotics*. PhD thesis KTH Royal Institute of Technology, Stockholm, ISBN 978-91-7729-283-8.
- Marzinotto A. et al. (2014). Towards a unified behavior trees framework for robot control. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 5420-5427
- Ögren, P. (2012). Increasing Modularity of UAV Control Systems using Computer Game Behavior Trees. *In AIAA Guidance, Navigation and Control Conference*, Minneapolis, MN, 2012.
- Quigley, M., Gerkey, B.P., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., & Ng, A. (2009). ROS : an open-source Robot Operating System. ’ in Proc. ICRA Workshop Open Source Software., 2009, p. 5.
- Siegwart, R., I.R. Nourbakhsh, D. Scaramuzza (2011). *Introduction to Autonomous Mobile Robots*. Cambridge MA: MIT press. ISBN: 978-0-262-01535-6.
- Sutton, R. S., Barto, A. G. (2018). Reinforcement Learning: An Introduction. The MIT Press.

[4] RIF: Robot Initiative Fontys or Resistance is Futile.
[5] https://fontys.nl/Innovatie-en-onderzoek/Expertisecentra/Fontys-Centre-of-Expertise-HTSM.htm