From Low Level to High Level Humanoid's Behaviors

Nicolas Jouandeau[1], Vincent Hugel[2], Thomas Da Costa[3]

[1] Université de Paris 8, LIASD, France, n@ai.univ-paris8.fr

[2]Université de Versailles, LISV, France, hugel@lisv.uvsq.fr

[3]Université de Paris 8, France, thomas.da-costa@etud.univ-paris8.fr

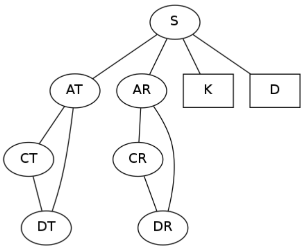

Figure 1: Moving Humanoid Phases

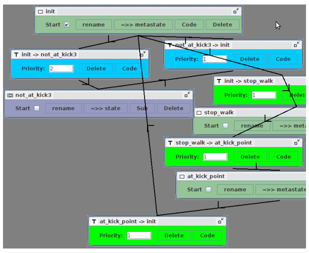

Figure 2: Single Player Hierarchical Behavior

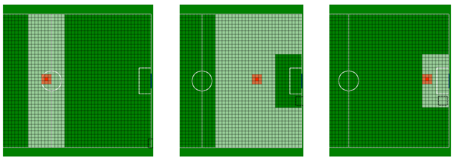

Figure 3: Case Based Penalty Kick Actions

The software architecture of Humanoid Behaviors is constrained due to the walking patterns that imply non uniform progress on the field and latency in coordinate team-play moves. The software architecture has to consider three viewpoints that result from different levels of thinking the moves in a team: the first viewpoint fixes the main constraints of single humanoid's moves, the second viewpoint sets single player's behaviors, and the third one sets multiple player's coordination. We present here a combination of three tools that tends to optimize the interactions of levels.

The first level depends on walking primitive moves. Main constraints tend to minimize pauses between phases and to maximize the efficiency of each phase. Each phase has to be as quick as possible while keeping balance. Although this efficiency depends on primitive functions, such a parameterization can be achieved automatically according to the humanoid's model used [2, 3]. Figure 1 shows the issue of phases transition in a state diagram, where S stands for Stand, A for Accelerate, D for Decelerate, C for Cruise, T for Translate, R for Rotate, K for Kick and D for Dive. Reentrant moves are represented with circles and non interruptible moves with boxes. When the robot is falling, reentrant moves are interrupted and the resulting state will be S. For the other moves, reactivity is slower due to the fact that such moves are not interruptible.

The second viewpoint can be achieved with Hierarchical Finite State Machine (HFSM). Single Player's Behaviors are built using the graphical HFSM tool [1]. Figure 2 shows a screenshot of our tool that allows to generate behaviors. States are green, Meta-states are purple, Transitions are fluorescent-green and Meta-transitions are blue.

The third viewpoint is a grid map that combines positioning and behaviors by case based reasoning. A set of cases depending on ball positions is assumed to complete all game situations. The resulting coordination is achieved without communication between players. The simplicity of this approach focuses on an abstract synthesis of possible situations and their matches with coordinate actions of humanoid players. Figure 3 shows the penalty kick sequence, divided into 3 steps : At first (a), a simple kicking behavior; Followed by (b), a dribbling behavior toward the goal; Finalized by (c), a kicking behavior toward a free space inside the goal area according to the goalie's position.

References

[1] V. Hugel, G. Amouroux, T. Costis, P. Bonnin and P. Blazevic. Specifications and Design of Graphical Interface for Hierarchical Finite State Machines. Lecture Notes in Computer Science. RoboCup 2005: Robot Soccer World Cup IX. Volume 4020/2006. 648–655.

[2] V. Hugel and N. Jouandeau. Walking Patterns for Real Time Path Planning Simulation of Humanoids. IEEE RO-MAN 2012. 424-430.

[3] V. Hugel and N. Jouandeau. Automatic generation of humanoid's geometric model parameters. Lecture Notes in Computer Science. RoboCup 2013 (to appear).