RoboCup 3D Framework for Robot Behaviour

Peter NOSKO, Dušan RODINA, Daniel SLAMKA, Peter SMOLINSKÝ, Ondrej ŠEVCE, Ivan TOMOVIČ*

Slovak University of Technology
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia
tim007tp2008@gmail.com

The aim of the RoboCup project is to create a team of robots, which could play football so well, that they would beat the best human football team [2]. A minor step towards achievement of this ambitious goal is to design a usable, extensible and robust framework, which can be used for modelling of robot's behaviour. The main goal of our work is to develop such framework. The framework allows defining the behaviour of a robot with a set of actions. These actions are composed of robot's limbs motions. The main advantage of our solution is that it can be used by future RoboCup developers to help develop more complex solutions in the future.

In the original version of the robot, adding a new action to the robot behaviour, which is generally one of the first tasks of every new RoboCup team, was a difficult, tedious task susceptible to many errors and involving too much of duplicate code. For these reasons we decided to create the framework, whose main purpose is to make creation and modification of various actions of the robot easier. This is achieved by providing a mechanism to treat all such actions similarly.

In the framework (Figure 1) we implemented, every action, the robot might perform (whether it is walking, passing a ball, dribbling, or shooting at the goal) is represented by an instance of a class that describes the action. The main advantage of this concept is that a user of the framework can easily define an action in the interactive editor, which is developed in parallel with the framework, and later use it in the robot simulation even with very little knowledge about how the limb movements are executed in the SimSpark simulator [1].

The behaviour of the robot can be represented as a sequence of actions that the robot has to perform. Each action has to meet its required preconditions in order to be scheduled and executed. An action itself can be further decomposed into separate motions of limbs, i.e. rotations of hinge joints. Let such motion be defined as the following quadruple: *(J,T,D,A)*, where *J* is the identifier of a joint which is to be bent;

---

* Master degree study programme in field: Software Engineering/Information Systems Supervisor: Ing. Marián Lekavý, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava.

T is the time elapsed since the execution of the action before the move is started; *D* represents the duration of the entire move and *A* stands for the angle of rotation. With motions defined this way, it is possible to represent an arbitrary action as a set of groups of motions, ordered according to their start time. A motion group can contain motions that are to be synchronized with respect to each other.

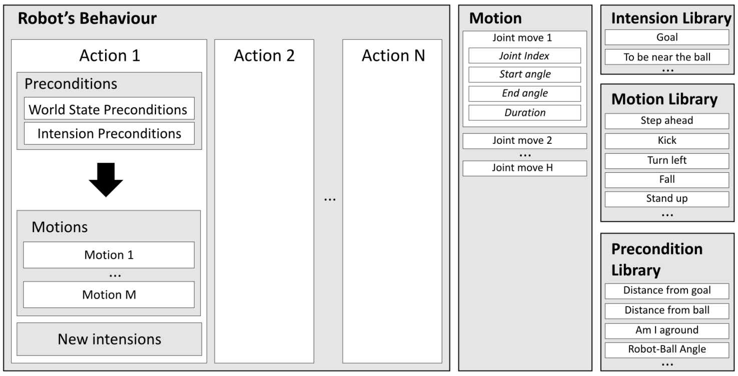

Figure 1. RoboCup 3D Framework for Robot Behaviour

Yet another of the most important features is that the actions that compose behaviours can be stored into a file and loaded by the robot when the need arises. The type of behaviour that the robot will conform to will then be selected by the higher logic on the basis of the robot's intentions, represented by predefined constants, and the fulfilment of behaviours'/actions' preconditions.

One of the problems, robot developers often have to cope with, is how to maintain the stability of the robot. It would be very frustrating if the robot kept falling down only because of small collisions or not-so-precise movements. That is why we propose a robot stability module, whose task is to balance the robot at the event that would otherwise lead to its fall. The idea is that the module will keep track of the momentum and the relative position against the stabilization points (usually the feet of the robot) of the robot's centre of mass, and via preventive movements and action correction attempt to regain stability.

Acknowledgement: This work was partially supported by the Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava. We would like to thank Lukáš Dvonč for his valuable consultations regarding the robot stability physics.

References

[1] Boedecker, J., *SimSpark User's Manual* (version 1.1), 2008, [Online; accessed November 8th, 2008]. Available at: http://labss2.fiit.stuba.sk/TeamProject/2007/team11is-si/download.html

[2] The RoboCup Federation, Robocup, [Online; accessed February 20th, 2009], Available at: http://www.robocup.org/