# Biter

José M. Vidal

Tue Oct 14 11:37:54 EDT 2003

This talk summarizes material from:

- Paul Buhler and José M. Vidal. Biter: A Platform for the Teaching and Research of Multiagent Systems' Design using RoboCup.[1] A. Birk and S. Coradeschi and S. Tadokoro ed.In *RoboCup 2001: Robot Soccer World Cup V. LNCS/LNAI Lecture Notes Volume 2377,*

  Springer-Verlag. 2002.

- Hrishikesh J. Goradia and José M. Vidal. Building Blocks for Agent Design.[2] Paolo Giorgini ed.In *Agent-Oriented Software Engineering,* Springer-Verlag. 2004.

- Biter Homepage[3]

- SoccerBeans Homepage[4]

## 1 History

- **Spring 2000** : First use of Robocup in Introduction to Multiagent Systems class. Students built almost everything from scratch. Most of the time was spent on building and maintaining a world model with absolute coordinates. *Paul Buhler* implements the first version of Biter for his project. This version already includes the world model.

- *Shaun Wood* works on it during the summer, adding low-level functions for finding players in view cone, intercepting ball, etc., as well as a graphical debugging tool.

- **Fall 2000** : First use of Biter in the class is successful. While the system is in use *José Vidal* releases version 2.0 which implements the Generic Agent Architecture.

[1] `http://jmvidal.cse.sc.edu/papers/biter.pdf`
[2] `http://jmvidal.cse.sc.edu/papers/goradia03a.pdf`
[3] `http://jmvidal.cse.sc.edu/biter/`
[4] `http://jmvidal.cse.sc.edu/soccerbeans/`

- During the winter break José gets rid of the multiple threads and is able to dribble the ball around the player (tight integration with with sserver's action loop).
- During the summers of 2001 and 2003 *Hrishikesh Goradia* added some more behaviors. This became the other version available for download.
- Our design was heavily influenced by The CMUnited-98 Champion Simulator Team.[5]
- **Fall 2001** : Hrishikesh Goradia has developed SoccerBeans[6] which lets you implemented new players by dragging and dropping beans.

## 2 Purpose

- For students to spend their time dealing with issues of agent coordination, cooperation, and emergent behaviors, but without completely sacrificing performance.
- Biter does not provide tournament-level performance, but it would not be hard to extend it to do so.
- To enable students to explore both reactive and goal-driven (BDI) architectures, and everything in between.
- To provide students with a basic framework and avoid the dreaded if-then-else statement with 40 else-if parts and a nesting level of up to 10.
- To provide help in debugging. This is minimal at the moment. A logging facility would be nice.
- In the future I hope to get one of you to implement Java Bean behaviors so that students can drag-and-drop their way to creating a new agent.

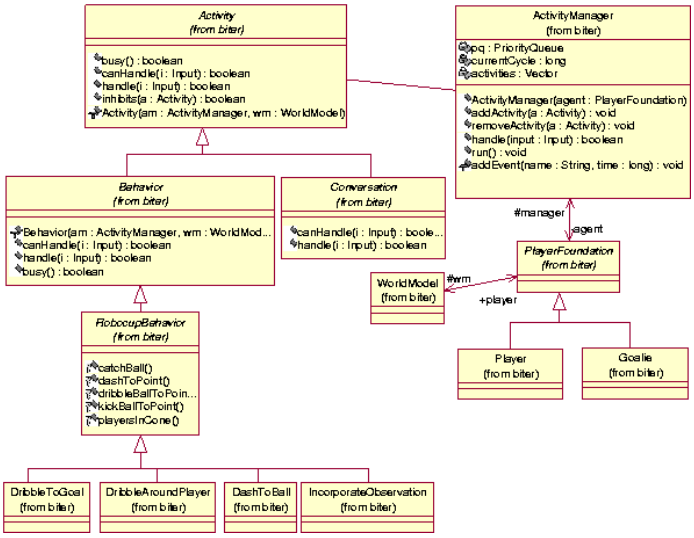
## 3 World Model

- Biter receives `see` commands from the sserver and parses them.
- Biter maintains a `WorldModel` which has the absolute location of every object on the field that the agent can or has seen recently.
- Static objects are read in at startup.
- First, the closest line and flag are used to determine the player's absolute location, then the location of all other objects is deduced from it. Their certainty is set to one.

[5] `http://jmvidal.cse.sc.edu/library/stone98a.ps`
[6] `http:/jmvidal.cse.sc.edu/soccerbeans/`

- If the ball is not seen then its updated using dead-reckoning from its previous velocity. The certainty about its location is reduced. When the certainty reaches zero it is removed.
- Player positions can be updated the same way.
- You will need to decide how fast to forget (reduce phantom objects).

# 4 Biter Architecture




- An agent is built by:
  1. Creating all the desired behaviors.
  2. Deciding which other behaviors a behaviors `inhibits`.
  3. Deciding which situations every behavior `canHandle`.
  4. Deciding how long each behavior will be `busy`. This is forever for permanent behavior.
  5. Adding all the behaviors to the `ActivityManager`.

---

This talk is available at `http://jmvidal.cse.sc.edu/talks/biter`