Jason BDI Agent in RoboCup
SYSC 5103 Final Project

Department of Systems and Computer Engineering, Carleton University

Horner, James 101067094
Marsland, Micheal 101042414
Menard, Jon 101086242
Govindasamy, Hari 101039846

***Abstract*—Jason BDI agents can be situated in the RoboCup environment by utilizing Krislet's code as an interface between the RoboCup player and the Jason Engine. A discretization of the RoboCup environment can be used to pass perceptions to the Jason Agent and intentions returned from the agent can be transformed into actions sent to the RoboCup Server. Jason's ASL file format can be used to establish plan sets and desires for each of the agents to allow them to derive intentions from a combination of perceptions and persistent beliefs. A team was developed comprising a goalie, defender, midfielder and two attackers each with their own sets of plans and desires. The agents were tested to ensure they execute the correct actions in a number of situations. Having agents with persistent beliefs as well as distinct plan sets allowed for different behaviours on the field and an inherent collaboration that resulted in vastly superior performance to the purely reactionary Krislet team.**

***Index Terms*—BDI Agents, RoboCup, Jason, Agent-Speak**

I. Introduction

This paper discusses the design and creation of several BDI agents to compete in the RoboCup [1] virtual environment through the use of Jason [2], an open-source AgentSpeak interpreter for Java. We discuss our method for utilizing Jason in the RoboCup environment and how the BDI agents receive beliefs then perform their intentions through RoboCup players. The paper will be broken down into the implementation required to link a RoboCup player and Jason, the methodology behind each Jason BDI agent, results from individual and team testing, the improvements demonstrated over Krislet's code, and instructions to run the provided code.

II. Model

Linking a RoboCup Player and Jason BDI Agent

To situate a Jason BDI agent in the RoboCup environment, the basic Krislet implementation - as discussed in the course material of SYSC 5103 [3] - was adapted into an interface between the Jason BDI engine and RoboCup. By extending Jason's AgArch class, a method was created to pass in the current perceptions of the player and synchronously receive an action from the BDI agent. The Krislet's Brain.java file was used as this interface to discretize the RoboCup environment into Beliefs and translate Intents from the Jason Engine into actions to send to the RoboCup server.

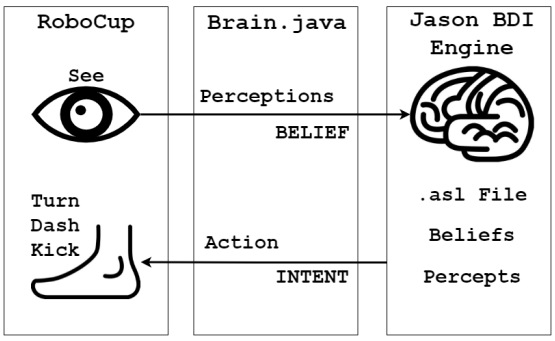

Fig. 1. Linking a Jason BDI Agent to the RoboCup Server

Beliefs: To properly link a RoboCup player to a Jason BDI agent, a method of converting perceptions from the server into BDI beliefs was needed. An enumeration of perceptions was created, which could be added to the agent's belief state and used in plans. During each cycle of the player, information about the environment, received from the server through Krislet, could be added based on what was seen in the form of atomic propositions, e.g. `ball_seen` or `enemy_goal_seen`.

Desires: Each Jason BDI agent was initialized with a specific Jason ASL file to describe the agent's set of available plans and the agent's initial desire. New desires and intentions for the agent were derived through the execution of these plans.

Intentions: A means of executing an action through the RoboCup player based on the current intention was still required. In order to fulfil an intention, a mapping was needed between intentions in the plan and those available to the RoboCup player. An enumeration of simple actions was defined and incorporated into the agent's ASL plans, so when the Jason agent returned an intention from the plan the corresponding player action could be recognized and performed in RoboCup through the Krislet interface, e.g. `turn`, `kick`, or `run`.

Jason BDI Agent

With RoboCup connected to Jason, development could begin on the specification of each of the agents using the ASL file format of the Jason framework. At the start of the project it was determined that four types of agents should

be specified: attackers, midfielders, defenders, and goalies. Each of the agents rely on the same set of perceptions from the environment and a set of possible actions that can be performed. In the specification of the agents a combination of persistent and temporary beliefs (percepts) was used to give the agents some state based behaviour, rather than being entirely reactionary. Most beliefs of the BDI agent only remain true temporarily as they are based on what the player can currently see, however persistent beliefs about the environment can be derived from the perceptions and remain in the belief base until proven otherwise. An example of this can be see with the `ball_was_left` belief, which is added when the angle of the ball relative to the player is less than 0. This belief will continue to persist, until the agent sees the ball on the right and removes the `ball_was_left` belief. Having persistent beliefs, allows the agent to remember the state of the environment, and have a better chance at choosing the best action in the future, the results of which are discussed in the following sections.

Team: To compete in the RoboCup environment, we developed a team comprising a goalie, defender, midfielder, and two attackers. We determined that this combination of agent behaviours produces the best results when playing against a team of default Krislet players. By utilizing these different agents we were able to break up the field into sections where each agent performs a distinct role. This implies some collaboration between the agents which greatly increases their effectiveness over a team of identical agents. Figure 2 below shows the default configuration of the team when each agent is located at their desired positions on the field as they perform their own plans as written in their respective ASL files.

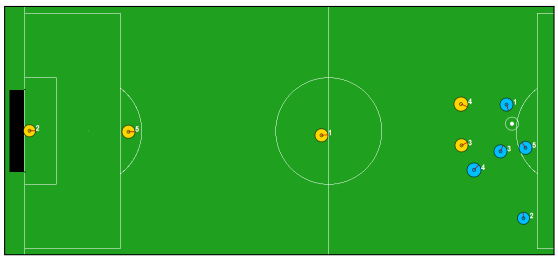

Fig. 2. Default configuration of 5v5 team

Attacker: The attacker's behaviour focuses on getting the agent to kick the ball closer to the opposing team's goal. The agent will always start by locating the ball. If the agent is the closest teammate to the ball, it will run to it. Once at the ball, the agent will kick the ball towards the enemy net if it has a clear shot or kick the ball up the side of the field if an enemy is blocking the path to the net. In the case of a teammate being closer to the ball, the agent will run towards the enemy net in anticipation of a pass rather than running to the ball.

Midfielder: The midfielder exists to pass the ball down the field from the defenders to the attackers. The midfielder begins by finding the ball, then, if it can see a teammate closer to the ball than itself, it will simply find the center point of the field man make its way there while always keeping an eye on the ball and ensuring there is a closer teammate. If it cannot see a teammate closer to the ball then it will run to the ball and attempt to kick it towards the opposing goal where it expects the attacker agents to be. The midfielder makes use of its persistent beliefs to remember the last seen direction of the ball, opposing goal, and center point.

Defender: The aim of the defender is to react to the opposing team's behaviour and reverse any progress they might make towards the agent's goal. The defender starts by moving into a strategic position where it can better intercept the ball if kicked towards the agent's goal. Once in position, the defender finds the ball and monitors its position. Once the ball is kicked onto the agent's side of the pitch, it reacts by running to the ball and kicking it either towards the opponents goal or away from the agent's goal, depending on which side the agent is facing. The defender relies on a mixture of perceptions and persistent beliefs to track it's position as well as the position of the ball to inform whether it should be actively defending or not.

Goalie: The primary role of the goalie is to stop opponents from scoring when an opponent is successfully able to get the ball past the team's defences. The goalie is stationary at the middle of their net and continuously observes the position of the ball. If the ball is observed to be close the net, the goalie will run towards the ball, in an attempt to kick it away from their net. If the goalie is facing the ball or can see the centre of the penalty box the goalie will continuously run towards it to kick it, otherwise it will return to the goal.

III. Testing

Testing began by confirming that the designed model architecture properly connected the RoboCup environment to Jason's AgentSpeak interpreter. A simple test plan was created in the Jason ASL files to confirm that beliefs added from the RoboCup player were also being added to the Jason BDI agent. The intent returned to the RoboCup player was printed and compared to the expected value to confirm the model was performing as expected.

After confirming the model was working correctly, testing continued by analyzing logs that captured the agent's beliefs, desires, and chosen intention. By manually comparing the agent's beliefs and desires to the plans available to the agent, it was confirmed that the agent was choosing the right intention and behaving as expected. Shown in Figure 3 are the logs of an example where the agent chose three different plans sequentially.

Each intention performed by the agent resulted in new perceptions being added to the system, allowing new plans to be executed. Furthermore, this demonstrates that while each plan is executed individually, the combined outcome results in the agent participating in the RoboCup environment by shooting the ball towards the net.

```
Starting Reasoning:
[BALL_SEEN, BALL_TO_LEFT, ENEMY_GOAL_SEEN, ENEMY_GOAL_TO_RIGHT ]
EXECUTING PLAN:  +!findball: ball_seen & not at_ball & ball_to_left
<- turn_to_ball +ball_was_left !movetoball
Got Intent: TURN TO BALL

Starting Reasoning:
[BALL_SEEN, FACING_BALL, ENEMY_GOAL_SEEN, ENEMY_GOAL_TO_RIGHT]
EXECUTING PLAN:  +!movetoball: ball_seen & facing_ball
<- run_to_ball !findball
Got Intent: RUN TO BALL

Starting Reasoning:
[BALL_SEEN, AT_BALL, FACING_BALL, BALL_TO_LEFT, ENEMY_GOAL_SEEN, ENEMY_GOAL_TO_RIGHT]
Got Intent: KICK AT NET
EXECUTING PLAN:  +!findball: ball_seen & facing_ball & enemy_goal_seen
<- kick_at_net !findball
```

Fig. 3. Sample log of BDI agent turning to ball, running to ball, and then kicking ball

Additional testing was completed by having several initial starting configurations at the beginning of the game to force the use of different plans by the agents. For every plan tested, the agent performed as expected by choosing the right action and performing correctly in the RoboCup environment.

IV. Results and Discussion

When comparing the BDI agent team to a team of default Krislet agents, the BDI agent team significantly out performed Krislet with an average score of 10 to 1. The BDI players are able to robustly select and execute plans based on their percepts and internal beliefs. The specific desires and plan set of the agent determine its observed behaviour and therefore its role within the RoboCup team.

BDI agents may receive the same perceptions, but based on their desires and belief set, a different plan may be selected for execution. Figure 5 and 4 show the reasoning of the goalie and attacker. Both the attacker and goalie perceive the same percepts, but different intentions are selected based on their desires, internal beliefs and available plans.

```
Starting Reasoning:
[OWN_GOAL_SEEN, ON_OWN_SIDE, FACING_OWN_GOAL, GOAL_LINE_SEEN]
Got Intent:
LOOK_RIGHT
```

Fig. 4. Intent selection for attacker

Since the goalie's desire is `+!gotogoal`, the selected intent is `RUN_TO_OWN_GOAL`, however since the attacker's desire is to `+!findball` in order to score, the selected intent is `LOOK RIGHT` to locate the ball on the field. This behaviour closely models a human soccer team where different players may play differently based on their inherent plans and desires.

```
Starting Reasoning:
[OWN_GOAL_SEEN, ON_OWN_SIDE, FACING_OWN_GOAL, GOAL_LINE_SEEN]
Got Intent:
RUN_TO_OWN_GOAL
```

Fig. 5. Intent selection for goalie

V. Conclusion

This paper has demonstrated how a Jason BDI agent can be situated in the RoboCup environment through a Krislet-based RoboCup player. By using enumerations of perceptions and actions to communicate with the BDI agent, the agent could receive information about the environment, form beliefs based on the perceptions, consider each available plan, make a rational decision about which plan to execute, and make it's intentions known to the server in the form of actions. The testing conducted on our BDI agents verified that the behaviour aligned with the specified plans in a variety of situations. We also showed how the use of multiple agent specifications can increase the efficacy of the overall team by partitioning the problem between the agents. Overall, this paper has endeavoured to show the possibility and performance benefits of using a BDI agent based player in the RoboCup environment.

VI. Running Locally

To experiment for yourself or to watch our BDI agents play against regular Krislet agents, complete the following steps:

- Download the ZIP file from the submission or the project GitHub
- Extract the files into a folder
- Read, Understand, and run compileCode.bat located in the top folder.
- Read, Understand, and run playBDIvsKrislet.bat located in the top folder.

NOTE: Do not run BAT files without fully understanding what they are doing. If you wish to use the model with your own ASL agents, they can be started in a similar manner to Krislet agents using:

```
start java -cp .;jason-2.3.jar Krislet -team
BDI -asl aslfilenamehere.asl
```

For more information, see the project's Github repository.

References

[1] *Robocup federation official website*, Nov. 2021. [Online]. Available: http://www.robocup.org/.

[2] *Jason*. [Online]. Available: http://jason.sourceforge.net/wp/.

[3] B. Esfandiari, *Software agent course*. [Online]. Available: https://www.sce.carleton.ca/cgi-babak/agentcourse.cgi.