Robot Programming - Robocode Competition

André Pimenta Ribeiro, *Student Member, IEEE*,
and Cedric Fernandes Pimenta, *Student Member, IEEE*,
and Rafael Fernando Pereira Abreu, *Student Member, IEEE*

***Abstract*—**
This article aims to explain the study in the construction process of simulating individual behaviors and social groups in the programming environment Robocode. All behaviours coded in Java will be exposed as well as the justification and context of each, either the individuals and the social ones. In the end, it will be demonstrated all the results achieved and if it were the ones expected or not.

***Index Terms*—Robocode, Autonomous Systemas, Artificial Intelligence, MEI.**

I. Introduction

NOWADAYS, we find the use of autonomous systems more and more as a reality and a necessity from the real world, for their capacity to understand the environment in which they are found at and their ability to react and adapt to that same environment, by the use of sensors. In the case study of this pratical work, our goals are, through the use of the programming environment Robocode and the simulation capacity that this system offers, the implementation and utilization of different strategies from artificial intelligence to the development of individual robots as well as robot teams, where we will be able to study and apply cooperation, execution of tasks in group and decision-making techniques. It will be presented, in the first part, robots who are caracterized and manifest themselves by characteristics that represent, in the most possible way, emotions, feelings and typically human behaviours. Subsequently, it will be showed social behaviours, ie the behaviour that should be adopted and is expected in an environment where communication and colaboration to achieve a common goal (a team objective) is a necessity. Also will be discussed, in this same study, both at individual and team levels, the OCC model [1] which will serve to represent emotions of the respective robots as well as their reactions to certain stimulations and events provoked by the environment, in this particular case, the Robocode's battlefield. This will all be implemented and represented through the programming language JAVA.

A. Motivations

THE principal motivation, for a project like this, has to be the learning and utilization of artificial intelligence strategies that permit the development of autonomous systems. More specifically, the implementation of certain characteristics in the robots who will need to react and interpret the enviornment in their surrounds, in an automated and efficient way. These characteristics have, as the principal focus, the representation of comportamental behaviours as well as their manifestation connected to stimulus and changes caused by events that happen in the interaction environment. The use and creation of tactis and communication strategies for the robots is, also, another grand goal for this work since their intersection, communication and communication bring a much more complexity to the problem, even though it possesses the advantage of introducing an added value to the project (the socialization and collaboration of different agents (robots) towards a common goal). And, as its natural and in the case that it is well defined, these characteristics bring numerous advantages because it becomes possible to abandon the limitation of being only one entity to realize a task to the use of a whole team working towards a common objective. Furthermore, another motivational component is the integration of the OCC model (mentioned above) in the Robocode environment and the respective results and real applications that we will be able no achieve in their conjunction.

II. Specifications

A. Robocode

AS it was mentioned before, we will be working in the programming environemnt named Robocode, gifted with different classes that permit the implementation of distinct robots who present a wide range of characteristics and capacities. The different classes mentioned are the following:

Robot: base class of robots.

Droid: interface that implements robots without radar but with more energy.

Advanced Robot: advanced classes of robots that allows a bigger control and implementation freedom.

Team Robot: class that allows the implementation of team robots as well as their communication.

B. Control Architectures

Different robots rule themselves through different control architectures that define, in a certain way, their behaviour throughout the lifecycle. For a better understanding of this architectures, we are going to present a little description of their functionality:

Deliberative: This architecture is based in the detection of sensorial data, resulting from environment, which will then be used to plan future actions: Understand-¿Plan-¿Act.

A. P. Ribeiro is with the Department of Informatics, Minho University, Braga, Portugal. E-mail: pg20189@alunos.uminho.pt

C. F. Pimenta is with the Department of Informatics, Minho University, Braga, Portugal. E-mail: pg19824@alunos.uminho.pt

R. F. P. Abreu is with the Department of Informatics, Minho University, Braga, Portugal. E-mail: pg20978@alunos.uminho.pt

[1] Ortony, Clore, Collins, The Cognitive Structure of Emotions

Reative: Architecture that have, as a principal highlight, the response's quickness since the planning is omitted, the actions are limited to a defined set of condition-¿action rules. The representation of the environment is as minimum as possible.

Hibrid: Architecture that is a cross between the past two, applying reactive systems in the low level control and deliberative systems at the level of decision-making.

Behavioural: Inspired by biology, this architecture tries to imitate animal behaviour in the problem solving. It is based in a set of behaviours whose finality is to achieve or maintain certain objectives but that are influenced by emotions triggered through events happening in the environment.

C. Control Strategies

The control system, in this case study applied to Robocode, is responsible for using the information retrieved from the environment and compute the way how robots are going to interact with that same environment. The control strategies applied are the following:

Open Loop: Strategy that bases itself in the non use of sensors, being its use an advantage in static or predictable environments.

Feedforward: This strategy is going to use sensors to retrieve information. This information will then be used to refresh the variables that model the environment. It presents a weak utility when the environment is dynamic.

Feedbackward: Sensors are used by this strategy to monitor the environment continuously, adjusting the robot according to the interpretation of this environment. This strategy allows a constant update of the external world.

III. Individual Robots

In this section, we will give specifications about how the individuals robots were created. They were developed with the objective to represent a set of characteristics, behaviours and emotions that can be trigged during a battle.

A. OCC model

One of our primordial decisions were the application of the OCC model introduced to us during this semester. The applied model is not exactly the model OCC but a derivation of this, that introduces concepts a little different from the original model but that made much logic, this factor was sufficient enough to persuade us from the benefits that this model would bring to our project. Explaining in detail how this model was applied during the development, initially, we make the caracterization of the robot under the qualities/defects that we pretend to simulate (his personality). So, this is decided through the attribuiton of values to the next five variables:

```
private double openness = 0.87;
private double conscientiousness = 0.63;
private double extraversion = 0.22;
private double agreeableness = 0.48;
private double neuroticism = 0.42;
```

Fig. 1. OCEAN variables

These characteristics are translated to a universe of three variables, designated PAD (Pleasure, Arousal and Dominance) and these variables take values between -1 and 1 by the application of defined formulas that use the five values mentioned above. Through this universe, it was possible to distinguish twelve distinct emotions (following as a reference the work done by the authors' article mentioned in the bibliography). These twelve emotions were chosen taking the following values:

```
public static Point3D joy = new Point3D
(0.4, 0.2, 0.1);
public static Point3D hope = new Point3D
(0.2, 0.2, 0.1);
public static Point3D relief = new Point3D
(0.2 , 0.3, 0.4);
public static Point3D pride = new Point3D
(0.4, 0.3, 0.3);
public static Point3D gratitude = new Point3D
(0.4, 0.2, 0.3);
public static Point3D love = new Point3D
(0.3, 0.1, 0.2);
public static Point3D distress = new Point3D
(0.4, 0.2, 0.5);
public static Point3D fear = new Point3D
(0.64, 0.6 , 0.43);
public static Point3D disappointment = new Point3D
(0.3, 0.1 , 0.4);
public static Point3D remorse = new Point3D
(0.3, 0.1 , 0.6);
public static Point3D anger = new Point3D
(0.51, 0.59 , 0.25);
public static Point3D hate = new Point3D
(0.6, 0.6 , 0.3);
```

Fig. 2. Emotion values in PAD universe

Last but not least, it is import to refer that for a new emotion to act on our robot, under this model, the intensity of this emotion needs to be higher than the absolute difference between extraversion and neuroticism. Besides this, after an emotion influences a robot and along time, the emotion tends to disappear and the robot returns to his initial state, following the next formula:

$$I(t) = I * e^{(-t*n)}$$

where n is the neuroticism, t is the time that held since the emotion has triggered and I is the intensity of the new emotion. With all these factors acting upon the robot, we were able to express with precision the different emotions that are triggered on the robot when occur certain events on the environment (ie, when a robot fires a projectile, the new emotion will be hope, to hit the enemy; if it fails, the new emotion will be disappointment but if it hits, it will be joy). Through this, we can have a better simulation of the actions from the system when certains set of factors act upon it (ie, if, each time it is hit, the rage will increase, after achieving certain levels, the robots can abandon their strategy and being carried away by this emotion).

B. The Developed Robots

DURING the project, there were developed many robots that simulated a wide range of characteristics. We will take two robots as a example to show how we were able to simulate the mentioned characteristics:

B.1 Aim

From the definition of the word Aim, the following characteristics were taken in consideration for the implementation of the respective robot: *precision, cunning, concentration.*

Control Strategy	**Feedbackward**
Control Architecture	**Deliberative**
Features	• Acquires a sniper behaviour • At the beginning of a battle, moves himself to a corner and fires under a predictive algorithm (linear or circular) depending on the number of missed bullets • At the end of a battle, he checks if when he died, there were still many robots alive, if affirmative, changes the corner on the next battle • With these type of robot, emotions need to have a high intensity to act upon them since they are in a position where they need to have a tough personality

B.2 Scare

Scare is a emotion that derives a little from fear mixed with some shyness, for what we decided to characterize this robot with states of *fright, sudden terror, trepidation.*

Control Strategy	**Feedfoward**
Control Architecture	**Reactive**
Features	• Adjusts the shot power according to the distance of the nearest adversary • When the values of the PAD model are near (-0.50,30,-0.45), if becomes scared: – Becomes pale (white) – Turns around himself and fires with the maximum power randomly

IV. Team Robots

IN this section, we will present the robots developed to work as a team, as well as the tactics and strategies, both of communication and collaboration, to achieve an end, in this specific study, the victory on the battlefield.

A. Communication between Robots

A communication between the elements of a team is a fulcral component for the success of said team and so it is the used communication. Through a clear and efficient communication, it is possible to make a distribuition of tasks in order to reach a common goal and, furthermore, having a bigger source of information about the external world; this factor allows communication between elements of a team oriented to the share of knowledge to be made in a much more complete way, being achieved from different places simultaneously. However, the team communication has a danger side and can easily become harmful if there are communication failures, communication of wrong or incomplete information.

B. Ontology

To do a efficient communication between the different elements of a team, it was created an ontology based on military concepts, adapting them to our environment. The defined ontology was the following:

Attack: This command orders an attack to a position in the battlefield or to an enemy.

Help: Orders the defence of a certain team element. It is defined by a priority which have as objective defining the necessity to trample other commands. Contains information about the robot causing the damage and the location where the help is needed.

Move: This command defines a movement from a specified position to another and its principal goal is to define strategic movements regarding any element of the team.

Target: Defines a target (its name, position, energy and type).

Position: Defines a 2D position on the battlefield.

Shot: Defines the information for the projectile's shot. Its objective is to fire lethal shots.

Command: Command that is used for text communication between elements of the same team.

C. Robot Team Formation

One of the objectives of this study is the definition and creation of robots' teams who work together to achieve a common goal. In our environment (Robocode), this objective in common is the victory in the battlefield against other robots. It was decided to implement a military strategy based in a military hierarchy. This hierarchy is formed through military experience and, above all, in the capacities and knowledge of their elements that allows to assume that a higher element in the chain of command must be more capable and gifted with more knowledge and strategy than an element who is in an inferior position. There is a wide range of military positions as well as military hierarchies, so we opted for the portuguese military hierarchy, choosing some positions in order to represent our team of robots.

General: The team leader and will be the one gifted with most responsabilities in the battlefield among the elements of his team. He will be giving instructions to the other elements constantly and he is the one who decides which tactic the team should adopt.

Sargento: This element assumes some responsabilities and he is given some tactic freedom, being able to make some decisions, not only at individual but also at team level.

Atirador: The element with more specific characteristics in the team. Can be considered a special element whose objective is "shoot to kill".

Cabo: The one who is in a more vulnerable position during a battle since he is supposed to be in the line of fire.

Soldado: This element limits is actions to obeying the General and his objective is to do the tasks that the General orders. These tasks are prioritized over everything else.

D. Combat Strategies

Since the objective of the environment we use in this study is the victory in the battlefield by destroying the other robots, the use os strategies, specially in team combats, becomes fundamental in order to gain advantage over the enemies. It was our choice to choose not to build different teams for different strategies, instead we prepared a team to adapt itself with different strategies during the battle, taking always in consideration the possible different battle scenarios and events that may occur. Before a scenario where the team is in numberic inferiority (for example, team energy), the team leader adopt a movement strategy to a location in the battlefield where there are fewer enemies. This strategy, often, precede a defensive strategy; when this latter strategy is adopted, the team retreat to a position where it tries to form a more compact formation that grants protection to the leader and Atirador (mostly because of the special roles, mentioned above, they assume).

D.1 Decision Making

The decision making of the robots is made, mostly, by the team leader. The General make this decisions taking always in consideration the number of elements on his current team and his enemy teams and, also, the teams' energy levels. In case of General's death, the leadership passes on to his direct inferior, in this case, the Sargento, being more limited in the process of making decisions and leadership. From the level of Sargento to the other inferior levels in the hierarchy, there is no type of leadership, making the team desoriented and each element for himself (decision making is only made at individual levels).

D.2 Priority System

The existence of multiple events and tasks that occur during the battle can make the decision making a whole lot difficult, even disturbing the robots' actions. Trying to optimize, in a better way, the operation of this process and weighing the hierarchy system, it was implemented a priority system. This is made with the objective to highlight the type of orders and hirarchy without forgetting the simples tasks that need to be done for the greater good of the team. The priorities are defined in a scale from 1 to 5 and each robot need to interpret and scale, in the best way possible, these same priorities.

V. Conclusion

The development of different robots mentioned during this article, either at a individual as a team level, allow for concrete analysis about the use of autonomous systems as well as the necessities that these suffer. From now, we can affirm that the perception of the external world, such as the sensing of this external world, is fundamental for the function of any minimal autonomous system because, so it will be posssible to make the tasks as correctly as possible, it is needed a constant knowledge of the environment where the systems are enveloped, unless the exception where the environment is static/predictable (which in this study, isn't). We also know for a fact that different tasks, and for different personalities, fit better with different control strategies since the constant process of information isn't always necessary. At the robots' team level, it is important to give emphasis to the communication that these teams require so it is possible to obtain a high degree of confidence that leads to a correct execution of tasks, resulting in the achievement of the defined goals. However, this is a complex problem which requires a lot of careful considerations because how much more diversified is a team and how much bigger are the factors to take in consideration, the higher is the necessary care demanded to ensure that the information is always correct so everyone can always keep their information updated in a fast and secure way. Regarding the use of the OCC model, it is possible to highlight the impressive way in which it can simulate human emotions with greate precision, and in this particular case, it makes a approximation between the components in which we worked and the stimulus they receive through the exterior environment practicable and their behaviour can be modeled according to the information possessed. As future work, stays the idea of the robots' team evolution since, even though it is working at a acceptable level, it can always be subjected to improvements. It is important to mention the work done regarding the development of the communication premises that can serve as a base for future development.

References

[1] Santos R., Marreiros G., Ramos C., Neves J., Bulas-Cruz J., *Personality, Emotion, and Mood in Agent-Based Group Decision Making*, IEEE Computer Society, 2011.

[2] Maria Goreti, *Agentes de Apoio Argumentao e Deciso em Grupo*, Universidade do Minho, 2007.

[3] Jos Maia Neves, *A Logic Interpreter to Handle Time and Negation in Logic Data Bases*, Universidade do Minho, 1984.

[4] Susana Rocha Rodrigues, *Eventos Adversos e No Conformidades nos Cuidados de Sade*, Universidade do Minho, 2009.

[5] Rodrigues S., Brando P., Nelas N., Neves J., Alves V., *A logic programming approach to medical errors in imaging*, International journal of medicalinformatics, 2009.