Architecture Description

Julian Mauricio Angel Fernandez

28 de febrero de 2014

Índice general

Version	Date	Author	Comments
0.1	27/02/2014	Julian Angel	First draft of the system's design

Capítulo 1

General View

The architecture presented in this chapter was created following the Belief, Desires and Intentions (BDI) approach, as could be seen in the Figure 1.1. Additionally it was added the Description and Action modules. The first one (Description) has two goals: enable personalisation of the robot (e.g. platform and character), and describe all the necessary information to act. Each of the modules are decomposed in sub-system that working as a whole could be used to accomplish the task. The final result could be seen in the Figure 1.2, where the module Action was divided in two: Action decision and Action modulation. Also was added the sub-module feature. Each of the modules in the architecture will be described in more detail though this chapter.

1.1. Description

The action description defines all the necessary information that are required to generate emotive action and personalize the system. This system is divided into parts: general, which described the information to generate the emotive actions. And Theatre, which describes the script and the current character that should be portrayed by the robot.

1.1.1. Simple Actions

The simple actions are actions that have specialized modules that execute them. At the moment the current actions are:

- $Move(finalposition, velocity, angularPosition)$
- $OscillateMove(velocity, maximumAngle)$ the oscillation movement just implies movement in the angular velocity, thus is just change in the θ component, and $maximumAngel \in [\pi, -\pi)$
- $RotateTorso(angle, velocity)$ which moves the upper part of the waist, if there is one.
- $OscillateTorso(velocity, maximumAngle)$ which enable the oscillation of the upper part of the waist.

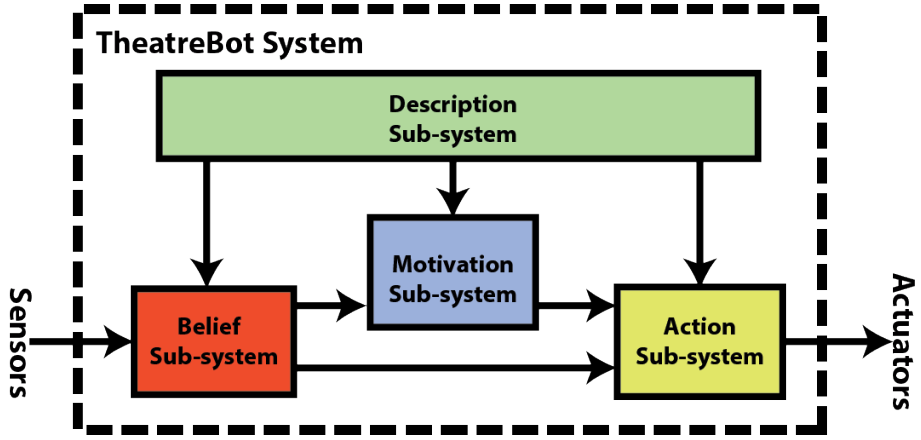

Figura 1.1: General Architecture. The dash line shows the components that are in the system. The concepts of Desires and Intentions are included in the box Motivation.

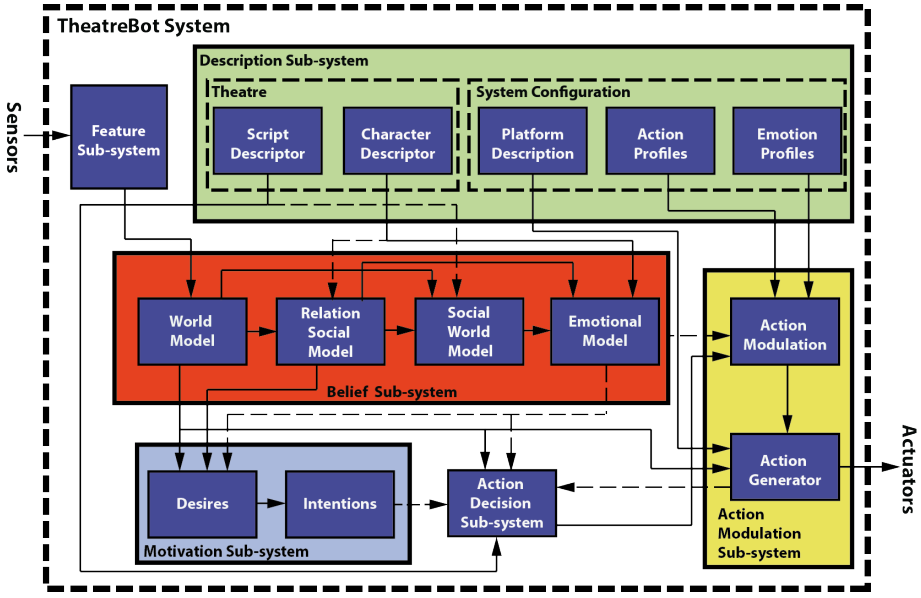

Figura 1.2: Architecture with the subsystems.

- $Grasp(object)$
- $RotateHead(angle\theta, angle\beta, angle\omega, velocity\theta, velocity\beta, velocity\omega)$
- $OscillateHead(angelID, velocity, angle)$ where $angleID \in \{angle\theta, angle\beta, angle\omega\}$
- $RorateShoulder(IDShoulder, angle\theta, angle\beta, velocity\theta, velocity\beta)$
- $OscillateShoulder(IDShoulder, angleID, velocity, maximumAngle)$ where $IDShoulder \in (\forall x | x = shoulder \bigwedge x \in robot)$

1.1.2. Abstract Action

The abstract actions are actions that could be composed by other abstract actions and simple actions. The composition of these action could use any kind of time or other kind of constrains for their synchronization, or even composition of them. However the first implementation will be simple as possible, which means that the composition of the actions will in parallel and started at the same time.
The abstract action $BalanceArms(velocity, maximumAngle)$ could be decompose as: $OscillateShoulder(IDShoulder_1, angleID_1, velocity, maximumAngle)$ and $OscillateShoulder(IDShoulder_2, angleID_1, -velocity, maximumAngle)$

1.1.3. Action Profile

The action profiles describes the simple abstract actions that people have created based on the simple actions that are available in the system.

1.1.4. Emotion Profile

The emotions profiles are based on the research done during the first year, where we come up with some basic features that enable the emotion projection. Example:

```
<emotion id="happiness">
    <simpleAction id="Move">
<parameter id="velocity">fast</parameter>
    </simpleAction>
    <simpleAction id="OscillateMove">
        <parameter id="velocity">fast</parameter>
        <parameter id="maximumAngel">small</parameter>
    </simpleAction>
    <simpleAction id="BalanceArms">
        <parameter id="maximumAngle">medium</parameter>
    </simpleAction>
    <simpleAction id="OscillateTorso">
        <parameter id="velocity">-fast</parameter>
        <parameter id="maximumAngel">small</parameter>
    </simpleAction>
</emotion>
```

1.1.5. Script descriptor

1.1.6. Character descriptor

1.2. Action Modulation

The principal role of this sub-system is to convert abstract actions (e.g. walk, gaze, etc.) and the emotional state to actions that could be performed by the robotic platform that it has been using. These emotions actions are just commands to the platform, but the emotional part is added during the process due to adding new simple actions, changing the parameters of the simple actions, or both. This could be done thanks that each abstract action should be describe in terms of simple actions, which the system knows how to modify and perform. The main inputs of these module are:

- Emotion profiles, which is the information in how to show an emotion in terms of simple actions.
- Action profiles, which describes the basic actions and the composition of the abstract actions in terms of the simple actions.
- Emotional state is described by the tuple emotion and intensity. The first says the emotions that is going to convey the robot, and the second says how much is going to be evident the emotion.
- Abstract action, which could be a simple action to a any action described in terms of the simple actions supported by the system.
- Platform descriptions gives the information about which simple actions could be performed by the current platform.
- World model.

The outputs of this module are:

- *Electronic signals:* to control each driver that has the robot.
- *Failure signal:* to inform the action decision sub-system that one of the actions could not be performed.

This sub-system is sub-divided into two blocks: actions modulation and action generation.

1.2.1. Action Modulation

This module gets the emotional state and the action(s) to generate emotive actions, which includes adding new actions and modifying parameters of the current. To do this module add new actions regarding the platform. The inputs to this module are:

- Emotion profiles
- Action profiles
- Emotional state

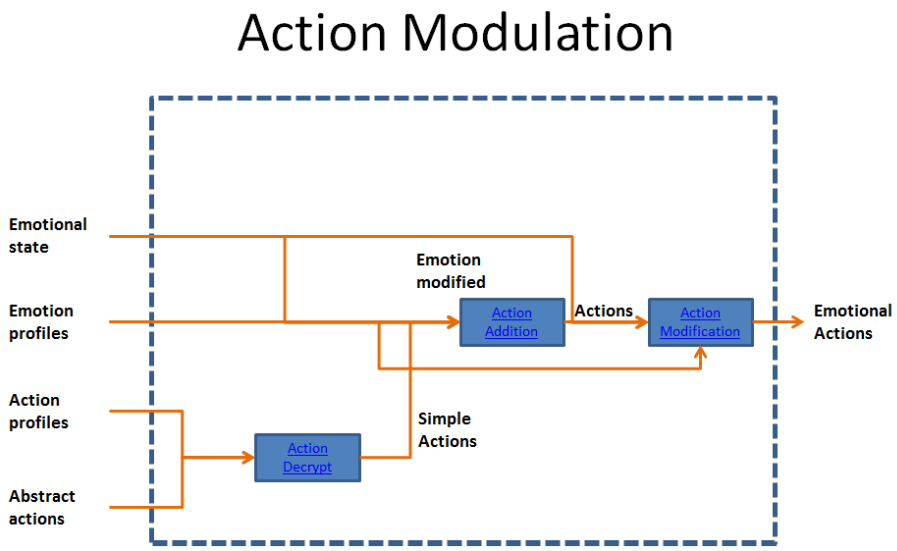

Figura 1.3: Modules of the action modulation sub-system

- Abstract action

The outputs:

- *Emotional actions:* are the set of simple actions with modified parameters to show emotions.

As well this module is divided in modules with specific objectives, the division could be seen in Figure 1.3. The goal of each module is described below:

- *Action decrypt:* take the abstract actions and converted to simple actions.
- *Action addition:* add the new actions needed to show the emotion.
- *Action modification:* changes the parameters of all the actions.

1.2.2. Action Generation

This module is in charge to decide which actions could be performed by the current platform, and execute and control each action that should be performed. The inputs of this module are:

- Platform description
- Emotional actions
- World model

The outputs are:

- Failure signal

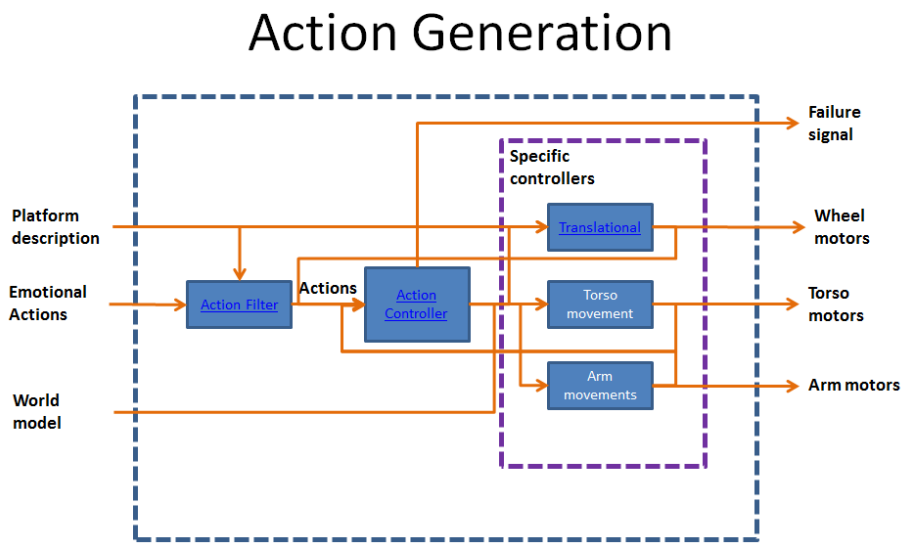

Figura 1.4: Modules of the action generation

- Signal controls

The task decomposition could be seen in Figure 1.4. The goal of each sub-module are described below:

- *Action filter:* filters the possible actions that could be perform due the constrains of the platform, and it makes any necessary change to the action to be correct performed by the platform (e.g. Constrains of movement)
- *Action controller:* decides which actions should be executed given the current actions that have performed by the robot. This module communicates with a specific module, which knows how to perform the action. This module knows which actions are executing and solve the problems of precedence and importance.
- *Specific controllers:* are modules developed with the idea to execute an specific simple action.

1.3. Belief

1.3.1. Relational Social Model

This module upgrades the relations with other people and roles, and the emotions that these produce to the character. The inputs of this module are:

- World model
- Character descriptor, which gives the concrete information of the relation during a play. This allow focus in the other modules and test the system.

The output of this module is:

- A list of people and roles that are in the current situation, and the emotions that these produce in the character.

1.3.2. Social World Model

This module is in charge to recognize the current social situation and determine other's emotional state. The inputs of this module are:

- World model
- A list of people and roles that are in the current situation, which is used as a filter.
- Script descriptor, which gives information of the current situation and the emotional state of the other people present in the current scene.

The outputs are:

- Social situation
- List of the people present in the environment and their respective emotional state

1.3.3. Emotional Model

This module calculates the current emotional state based on the people present, their emotional state, and the current situation.

1.3.4. World Model

Capítulo 2

Requirements

- How to change the emotional state without changing the current action?
- How to change the action without the emotional state?
- How to handle when the emotional state changes and the action is the same?
- How to describe the script?
  - How to handle the questions written above?
  - Is it necessary to introduce time?
  - How to add the coordination points?
    - Levels of these points?
- Which is the best way to describe the abstract actions, emotions profiles, etc.?
  - JSON
  - XML
  - YAML
  - JSON vs XML
  - JSON vs XML
  - YAML
- How to implement the specific controllers in ROS?
  - the controllers depend on the platform?

Capítulo 3

Description Design

Capítulo 4

Action Modulation Design