Documentation of the EmotionConfig

Group: Gamygdala-Integration

B.L.L. Kreynen

M. Spanoghe

R.A.N. Starre

Yannick Verhoog

Joost Wooning

June 2, 2015

Contents

1 Overview

In this document you can find the information to work with the EmotionConfiguration file. For this configuration you will need to create a .txt file and set certain commands you might need. Keep in mind that it is not particularly necessary to configure it. If you don not set anything the default values will be used.

1.1 Default values

When you do not specify any commands in the EmotionConfig, default values will be used. Here are the settings that will be present if you don not use a .txt file or if you have an empty .txt file.
DEFAULT GOAL UTILITY = 1;
DEFAULT DROPPED CONGRUENCE = -0.1;
DEFAULT ACHIEVED CONGRUENCE = 0.5;
DEFAULT BELIEF LIKELIHOOD = 1;
DEFAULT ISINCREMENTAL = false;

For more information on what these settings stand for, check the dedicated subsections.

1.2 Summary list

In this tabular you can find a complete list of all the commands in the form of an example. If you view this online, you can click on the tags in the first column and be directed to the regarding subsections.

NOTE: in the .txt file the column are separated ","

DEFAULT GOAL UTILITY	1			
DEFAULT ACHIEVED CONGRUENCE	0.5			
DEFAULT DROPPED CONGRUENCE	-0.1			
DEFAULT BELIEF LIKELIHOOD	1			
DEFAULT ISINCREMENTAL	false			
WHITELIST	ON			
GOAL	agent1	getBlock/1	0.6	
GOAL	agent1	getBlock/1		
SUB	getBlock/1	0.16	getAllBlocks/1	1
REL	agent1	agent2	-1	

2 Commands

Here you can find all the commands that can be used to set certain default values to your own needs. Keep in mind that the strings (the first part of each command) can also be written in small letters since they will be transformed to uppercases automatically.

2.1 Default goal utility

This sets the default goal utility to a specific value. The value is a double.

Types	String	double
Command	DEFAULT GOAL UTILITY	x

2.2 Default achieved congruence

This sets the default congruence for goals that are achieved. It specifies how "good" your default feedback is when you achieve a goal.

Types	String	double
Command	DEFAULT ACHIEVED CONGRUENCE	x

2.3 Default dropped congruence

This sets the default congruence for goals that are NOT achieved. It specifies how "good" your default feedback is when you cannot achieve a goal.The default value is still expressed as how "good" it is to drop goals. Therefore keep in mind that when you want to have a negative feedback that you have to put in a negative number.

Types	String	double
Command	DEFAULT DROPPED CONGRUENCE	x

2.4 Default belief likelihood

This rule specifies the default setting for the likelihood of beliefs. This value tells thus how "likely" belief are to be true when using this default setting.

Types	String	double
Command	DEFAULT BELIEF LIKELIHOOD	x

2.5 Default isIncremental

This specifies if the beliefs of goals are incremented or not. See documentation of Gamygdala for more information on what this feature does. You can write any upper/lower case variant of true/false. (True,False,true,false,fAlse...). (The value is being transformed to uppercases) Writing anything wrong in the syntax will throw an error instead of setting it on false by default.

Types	String	Boolean
Command	DEFAULT ISINCREMENTAL	true/false

2.6 Whitelist definition

This specifies that a white list is used for specifying goals. With this we mean that all the goals that you specify will be used with the specified values, but all other goals will not be used. This is when white list is set on. When white list is off, then the specified goals will be used with their specified values, but other goals, not specified will be using the default settings.There is no OFF setting. If you do not want white listed goals, simply don not mention is.

Types	String	String
Command	WHITELIST	ON

2.7 Goal definition

2.7.1 With utility

This definition defines the properties of a goal. Agent is the agent for which this specification is true (this is not yet implemented the idea is that eventually you could define different properties for different agents but for the same goal. At the moment it can be filled in with anything and the specification will hold for all agents.). Goal is the name of the goal, this should the same name as the goal is given in GOAL including the arity. Defining a goal like this also adds it to the whitelist if this feature is being used.

Types	String	String	String	double
Command	GOAL	agentName	goalName/arity	utility

2.7.2 without utility

This Goal definition can be used to add goals to the whitelist without setting a specific utility (between [-1, 1]), for these goals the default utility will be used. The Agent parameter again specifies which agent this goal should be whitelisted for (again this is not yet implemented).

Types	String	String	String
Command	GOAL	agentName	goalName

2.8 Belief definition

Here you can define a relation between a goal and its subgoal. The likelihood you define tell how the belief of the second goal is altered when the first goal is achieved. You can also set whether it needs to be incremental or not. For more information on what the incremental setting does you can look in the Gamygdala paper.

Types	String	String	double	String	double	Boolean
Command	SUB	goalName1/arity	x	goalName2/arity	y	true/false

2.9 Relation definition

This defines a relation between two agents. The relation is a relation that the SourceAgent has with the DestinationAgent, the value is how good or bad this relation is (between [-1,1]). The relation is one sided so DestinationAgent does not have the same relation with SourceAgent unless you add it separately. Both SourceAgent and DestinationAgents should be the names that the agents are given in GOAL.

Types	String	String	String	double
Command	REL	agentName1	agentName2	relation