

# Documentation of the EmotionConfig

**Group: Gamygdala-Integration**

B.L.L. Kreynen

M. Spanoghe

R.A.N. Starre

Yannick Verhoog

Joost Wooning

June 10, 2015

# Contents

<b>1</b>	<b>Overview</b>	<b>3</b>
1.1	Default values . . . . .	3
1.2	Summary list . . . . .	3
<b>2</b>	<b>Commands</b>	<b>4</b>
2.1	Default goal utility . . . . .	4
2.2	Default achieved congruence . . . . .	4
2.3	Default dropped congruence . . . . .	4
2.4	Default belief likelihood . . . . .	4
2.5	Default isIncremental . . . . .	5
2.6	Whitelist definition . . . . .	5
2.7	Goal definition . . . . .	5
2.7.1	Common Goal . . . . .	5
2.7.2	Individual Goal . . . . .	5
2.7.3	Optional Utility . . . . .	5
2.8	Subgoal definition . . . . .	6
2.9	Relation definition . . . . .	6

# 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 do not set anything the default values will be used. It is recommended that you have read the paper on Gamygdala[1].

## 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 do not use a .txt file or if you have an empty .txt file.

DEFAULT GOAL UTILITY	1
DEFAULT ACHIEVED CONGRUENCE	1
DEFAULT DROPPED CONGRUENCE	-1
DEFAULT BELIEF LIKELIHOOD	1
DEFAULT ISINCREMENTAL	false

For more information on what these settings stand for, you can 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. For the detailed explanation and syntax you can check the dedicated subsections.

NOTE: in the .txt file the column are SEPERATED BY ", "

DEFAULT GOAL UTILITY	1				
DEFAULT ACHIEVED CONGRUENCE	1				
DEFAULT DROPPED CONGRUENCE	-1				
DEFAULT BELIEF LIKELIHOOD	1				
DEFAULT ISINCREMENTAL	false				
WHITELIST	ON				
IGOAL	agent1	getBlock/1	0.6		
CGOAL	agent1	getBlock/1	0.6		
CGOAL	agent1	getBlock/1			
SUB	getBlock/1	0.16	getAllBlocks/1	1	True
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. This specifies how "good" or "bad" the goal is for the agent. In other words, the value the agent attributes to this goal becoming true ( $\in [-1, 1]$  where a negative value means the agent does not want this to happen).

Types	String	double
Command	DEFAULT GOAL UTILITY	x

### 2.2 Default achieved congruence

This sets the default congruence for goals that are achieved. Congruence is a number  $\in [-1, 1]$  where negative values mean this belief is blocking the goal and positive values mean this belief facilitates the goal. (Because you achieve or drop a goal in GOAL with certainty these values are set on -1 and 1 and should not be changed for a normal program setup. Only advanced users can use this setting to create more complex programs)

Types	String	double
Command	DEFAULT ACHIEVED CONGRUENCE	x

### 2.3 Default dropped congruence

This sets the default congruence for goals that are NOT achieved and therefore being dropped. It specifies how "bad" 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. (Because you achieve or drop a goal in GOAL with certainty these values are set on -1 and 1 and should not be changed for a normal program setup. Advanced users can use this setting to create more complex programs)

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. In other words, the likelihood that this information is true ( $\in [0, 1]$ ) where 0 means the belief is disconfirmed and 1 means it is confirmed.

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 values that can be set with the default commands. There is no OFF setting. If you do not want white listed goals, simply do not mention this command.

Types	String	String
Command	WHITELIST	ON

## 2.7 Goal definition

### 2.7.1 Common Goal

This goal definition defines a goal that is common for all agents. This means that when this goal is achieved, it will be appraised by all the agents with this goal. The definition of the goal itself stays the same. Only the predicate is different to address the different kinds of goals.

Types	String	String	double
Command	CGOAL	goalName	utility

### 2.7.2 Individual Goal

This goal definition defines a goal that is just for 1 agent. Therefore when this goal is achieved, only the agent itself will appraise this goal. No emotions will be changed by appraising this goal for all the other agents that have the same goal in their goal base. This is done by renaming the goal internally when you register it for an agent. The implementation extends the goal name with the agent name so that it gets a unique ID for appraising only for a specific agent.

Types	String	String	double
Command	IGOAL	goalName	utility

### 2.7.3 Optional Utility

For both the IGOAL and the CGOAL the utility parameter is optional. When not set, it will use the default goal utility that you can set to whatever you want.

Types	String	String
Command	IGOAL	goalName

## 2.8 Subgoal definition

Here you can define a relation between a goal and its subgoal. The likelihood you define tells the intensity of this goal being true. Putting a low likelihood means that it is not very likely that this happened for real although you get the percept. This alters the emotions of course. The congruence setting is about how good the main goal is for the agent. You can define a relation of a subgoal with a main goal that you do not want to happen. 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	likelihood	goalName2/arity	congruence	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 ( $\in [-1, 1]$ ). 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

## References

- [1] Gamygdala emotion engine <http://ii.tudelft.nl/~joostb/gamygdala/index.html>