Rational versus Emotional Reasoning in a Realistic Multi-Objective Environment

Seyed Mohammad Hossein Mayboudi

Submitted to the
Institute of Graduate Studies and Research
in Partial Fulfilment of the Requirements for the Degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
January 2011
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yilmaz
Director (a)

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Muhammad Salamah
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Asst. Prof. Dr. Adnan Acan
Supervisor

Examining Committee

1. Asst. Prof. Dr. Adnan Acan

2. Asst. Prof. Dr. Ahmet Unveren

3. Asst. Prof. Dr. Mehmet Bodur

ABSTRACT

Emotional intelligence and its associated with models have recently become one of new active studies in the field of artificial intelligence. Several works have been performed on modelling of emotional behaviours such as love, hate, happiness and sadness. This study presents a comparative evaluation of rational and emotional behaviours and the effects of emotions on the decision making process of agents in a realistic multi-objective environment. NetLogo simulation environment is used to model a realistic multi-objective environment, where an agent is in continuous interaction with a set of objects in its surrounding. The agent living and acting in its environment employs reasoning procedures which combine a series of inferences, evaluation, evolution processes, adaptation, learning and rule based emotions. Experimental evaluations showed that agents with the rational and emotional models in their decision to making processes achieve better performance compared to those with the rational decision model only.

Keywords: Artificial Intelligence, Agent-Based Modelling, Rational Agent, Emotional Intelligence, NetLogo.

ÖZ

Duygusal zeka ve onunla ilişkili modeller son zamanlarda yapay zeka alanında yeni aktif araştırma konularından biri haline gelmiştir. Sevgi, nefret, mutluluk ve öfke gibi duygusal davranışların modellenmesi üzerine çeşitli çalışmalar yapılmıştır. Bu çalışmada rasyonel ve duygusal davranışların karşılaştırmalı değerlendirilmeleri ve çok amaçlı gerçekçi bir ortamda ajanların karar alma süreci üzerinde duyguların etkisi sunulmuştur. NetLogo benzetim ortamı kullanılarak bir ajanın çevresindeki nesnelerle sürekli etkileşimi çok amaçlı bir ortam için modellenmiştir. Ajan yaşadığı çevrede etkili öğrenme, çıkarımlar yapma, değerlendirmelerde bulunmayı geliştirme ve uyum gibi bir dizi duygusal kural ve akıl süreci kullanır. Deneysel değerlendirmeler rasyonel ve duygusal karar alma sürelerinin birlikte kullanımının daha etkili olduğunu göstermiştir.

Anahtar Kelimeler: Yapay Zeka, Ajan Tabanlı Modelleme, Rasyonel Ajan, Duygusal Zeka, NetLogo.

ACKNOWLEDGMENTS

I would like to thank Asst. Prof. Dr. Adnan Acan for his continuous support and guidance in the preparation of this study. Without his invaluable supervision, all my efforts could have been short-sighted.

I owe extremely to my family who allowed me to travel all the way from Iran to Turkish Republic of Northern Cyprus and supported me all throughout my studies. I would like to dedicate this study to them as an indication of their significance in this study as well as in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

ABM .... Agent-Based Modelling

ACDL .... Course-Duration-List of Academy Place

ACFL .... Course-Fee-List of Academy Place

ACL .... Course-List of Academy Place

AI .... Artificial Intelligence

AKCGL .... Knowledge-Capacity-Gained-List of Academy Place

AKCGL .... Working-Capacity-Gained-List of Academy Place

BDI .... Belief-Desire-Intention

BDICTL .... Belief-Desire-Intention and Computation Tree Logic

CG .... Current-Goal of Agent

DIA .... Distributed Artificial Intelligence

EBDI .... Emotional Belief-Desire-Intention

EIA .... Emotional Intelligent Agent

EMA .... Emotional Motivated Agents

GOMASE .... Goal-Orientated, Multi-Agent Simulation Environment

GUI .... Graphic User Interface

KC .... Knowledge-Capacity of Agent

KCT .... Knowledge-Capacity-Threshold of Agent

KISS .... Keep It Simple and Stupid

LORA .... Logic of Rational Agents

ML .... Money-Level for Agent

MLSF .... Money-Level-Smooth-Factor of Agent

MLT ....Money-Level-Threshold of Agent

NEL ....Needed-Expenses-List of Agent in Places

OCC ....Ortony, Clore and Collins

OS ....Orphanage-Status of Orphanage Place

OSDF...Orphanage-Status-Decay-Factor of Environment

OST ....Orphanage-Status-Threshold of Orphanage Place

PG ....Previous-Goal of Agent

PRS ....Procedural Reasoning System

RIA ....Rational Intelligent Agent

SADL ....Activity-Duration-List of Social Club

SAFL ....Activity-Fee-List of Social Club

SAL ....Activity-List of Social Club

SC ....Social-Capacity of Agent

SCT ....Social-Capacity-Threshold of Agent

SG ....Selected-Goal of Agent

SSCGL ....Social-Capacity-Gained-List of Social Club

SWCGL..Working-Capacity-Gained-List of Social Club

TRA ....Theory of Reasoned Action

WC ....Working-Capacity of Agent

WCR ....Capacity-Required of Workplace

WCT ....Working-Capacity-Threshold of Agent

WJD ....Job-Duration of Workplace

WOJ ....Offering-Job of Workplace

WOS ....Offering-Salary of Workplace

Chapter 1

AN INTRODUCTION TO EMOTION

Humans are talented at making decisions, based on incomplete information caused by interaction between rational and emotional processes. Irrefutably, emotions influence human life in many respects [27]. According to recent studies, emotion has various interpretations in the Artificial Intelligence (AI) community: for some it is simply facial expressions [27]; for others, fundamentally internal processes [28] [29] [30]. There are some potential roles in artificial agents further than exposing and identifying emotional interactions with humans [31]:

1. action selection (for example, what to do next based on the current emotional state)
2. adaptation (for example, short or long-term changes in behaviour as a consequence to the emotional states)
3. social regulation (for example, communicating or exchanging information with others by emotional expressions)
4. sensory integration (for example, emotional filtering of data or blocking of integration)
5. alarm mechanisms (for example, fast reflexes like reactions in critical situations which interrupt other processes)
6. motivation (for example, creating motives as part of an emotional coping mechanism)

7. goal management (for example, creation of new goals or reprioritization of existing ones)
8. learning (for example, emotional evaluations as Q-values in reinforcement learning)
9. attention focus (for example, selection of data to be processed based on emotional evaluation)
10. memory control (for example, emotional bias on memory access and retrieval as well as decay rate of memory items)
11. strategic processing (for example, selection of various search plans based on overall emotional state)
12. self-model (for example, emotions as representations of "what a situation is like for the agent")

P. Baillie, M. Toleman, and D. Lukose studied the generation of emotions in artificial beings and how they filter information; as it is input, output and processed by such agents which are called Emotional Motivated Agents (EMA) with affective reasoning and decision making [32] [33]. In other words, they proposed an examination of the atomic elements and the assessment of these elements when combined to generate emotional states. Response mechanisms in a temporal situation, or events based on the Theory of Reasoned Action (TRA) by using Goal-Orientated, Multi-Agent Simulation Environment (GOMASE) [34], and the Tile-World system are used as an experimental environment.

The Ortony, Clore and Collins (OCC) cognitive appraisal model and subsets of it have been successfully implemented in a number of emotional architectures, such as Emotional Belief-Desire-Intention (EBDI) architecture. Based on the authors' justifiably claim:

> The OCC model categorizes emotions based on goals as a whole. It is difficult to address adaptive emotional behaviour stemming from attitudes about items that may interact with the individual during goal attainment and use this information to predict and extrapolate future behaviours under overlapping circumstances. [33]

H. Jiang, J. M. Vidal introduced a new emotional architecture [29] [30], which reflect practical reasoning of humans by adding the influence of emotions into decisions to make processes of traditional Belief-Desire-Intention (BDI) architecture. In addition, it handles limited resources by using primary emotions as the first filter for adjusting the priority of beliefs to speed up decision to make and refine when time permits by using secondary emotions. The Tile-world system is used as an experimental environment, like some other research [32] [33]. Based on their claim: "Most of the study into agents has focused on the development of rational utility-maximizing agents. This study assumes decisions to derive from an analysis of the future outcomes of various options and alternatives." [29] [30]

1.1 Emotion

Emotions are affected side of cognition and mostly assumed as representing a deductive logic of subjective experience, expressive behaviour and neurochemical activity. Most researchers claim that emotions are part of the human evolutionary heritage and serve adaptive ends by adding to general consciousness and the expedition of social communication. Some animals are also measured to have emotions, as first described by Charles Darwin in 1872. A significant early theory of emotion was suggested independently by William James and Carl Georg Lange between 1834 and 1900. All modern theorists grant consent to emotions, which influence what people notice, see, recognise, learn, and remember, play an important part in personality development [2].

In computer science, "Emotion" is the term for usually describing personal feelings, which command moods. In psychology, emotion is measured a reaction to stimuli and inclines in itself to inspire the individual toward further activity. Early psychological studies of emotion tried to determine whether a certain emotion appears before the action, simultaneously with it or as a response to automatic physiological processes. In the 1960s, the Schachter-Singer theory showed that cognitive process is not just physiological reactions, also playing an important role in determining emotions. In the 1980s, Robert Plutchik developed a theory which represents eight primary human emotions: joy, acceptance, fear, submission, sadness, disgust, anger and anticipation, and claimed all of them can be derived from these [3]. Gerd Ruebenstrunk carried out a very interesting survey on emotion theories [16].

In the 2000s, research in computer science, engineering, psychology and neuroscience has been intended at developing devices, which identify and model human emotions [4]. Affective computing is an interdisciplinary field of computer science, psychology and cognitive science and deals with designing and developing of AI systems and devices, which are able to recognize, interpret and process human emotions [5]. Despite the origins of the area may be traced as far back as to early philosophical enquiries into emotions [6], the more modern branch of computer science is founded upon Rosalind Picard's study [7] on affective computing [8] [9]. Identifying emotional information initiates with inactive sensors, which collect data about person's physical state or behaviour without interpreting or explaining the input. The data gathered is corresponding to the signals humans use to identify emotions in others. Another area within affective computing is the design of calculative devices proposed to display either natural emotional capabilities or is enabled persuasively simulating emotions. Emotional speech processing recognizes the person's emotional state by analysing speech patterns [10] [11]. The detection and processing of facial expression [12] [13] or body gestures [14] [15] is obtained through detectors and sensors.

1.2 Theory of Emotion from Ortony, Clore and Collins (OCC)

Ortony, Clore and Collins established their theoretical method [35] explicitly with the purpose to implement it in a computer. The theory adopts emotions, which develop as a consequence to certain perceptions and explanations. Thus it completely focuses on the cognitive elicitors of emotions. Events, agents and objects are assumed three aspects, which determine these cognitions.

Emotions, their central statement, represent reactions to these perceptions of the environment. For example, the results of an event can be pleased or not (pleased/displeased), the actions of an agent can be endorsed or rejected (approve/disapprove), or aspects of an object can be liked or disliked (like/dislike). A further difference consists of the detail which events can have concerns for others or for oneself and an acting agent can be another or oneself. The concerns of an event for another can be separated into desirable and undesirable; the results for oneself as related or unrelated expectations. Finally, related expectations can be distinguished again consistent with whether to occur or not (confirmed/disconfirmed). This discrimination leads to the following assembly of emotion types [Figure 1].

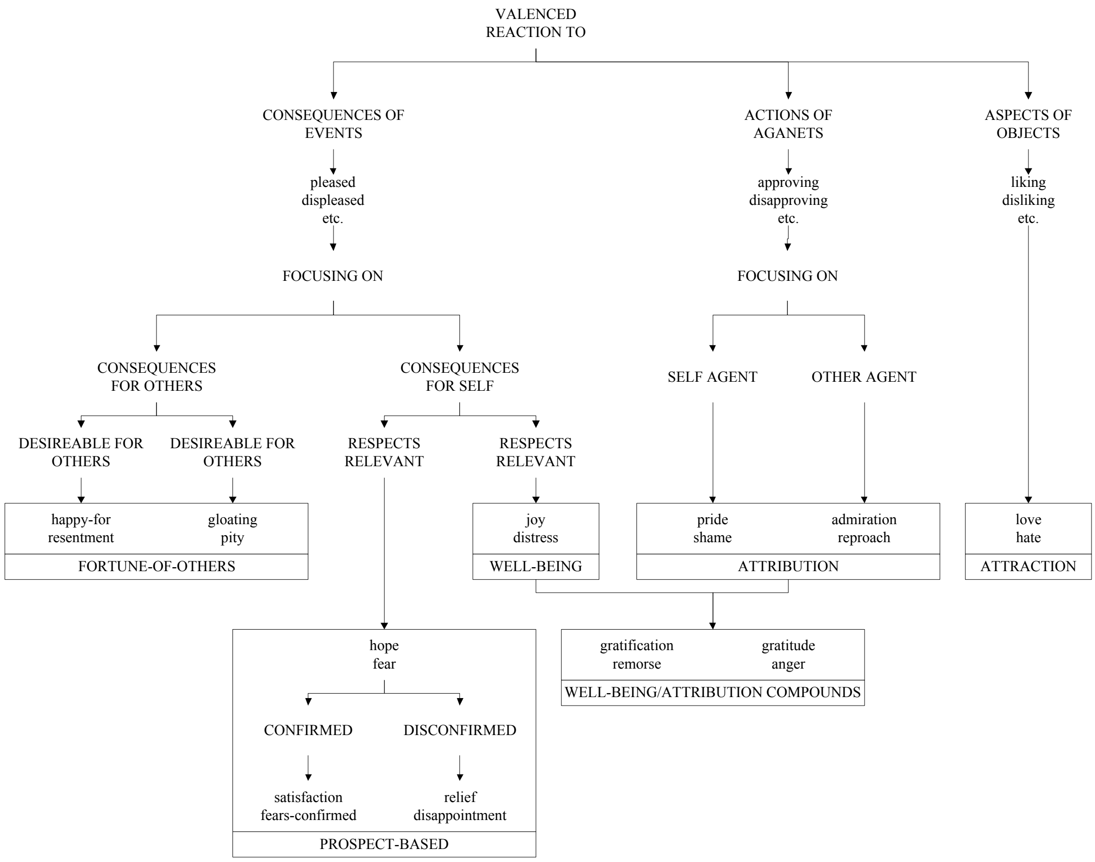

Figure 1: Schema of Emotion Types in the Theory of OCC [35]

The strength of an emotional feeling is decided mainly by three central strength variables as follows:

1. Desirability: it is linked the reaction to events and is appraised with regard to goals.
2. Praiseworthiness: it is linked with the reaction to actions of agents and it is appraised with regard to standards.
3. Attractiveness: it is linked with the reaction to objects and it is appraised with regard to attitudes.

Furthermore, a set of strength variables are explained. Sense of reality, proximity, unexpectedness, and arousal are the four global variables which operate over all three emotion categories, and the local variables are illustrated [Table 1].

Table 1: Local Variables associated with the Theory of OCC [35].

EVENTS	AGENTS	OBJECTS
Desirability	Praiseworthiness	Appealingness
Desirability for other	Strength of Cognitive Unit	Familiarity
Deservingness	Expectation Deviation	
Liking		
Likelihood		
Effort		
Realization		

In a real case, each of these variables is allocated a value and a weight. Moreover, there is a threshold value for each emotion, below which an emotion is not subjectively felt.

They propose no formalization for all of their explained emotions but give only a few examples. However, they assume which all emotions can be described by using a formal symbolization, although with many emotions. This is more complex than the presented example. With the help of such a formal system, a computer should be able to draw conclusions about emotional episodes, which are presented to it.

1.3 Belief-Desire-Intention (BDI)

The BDI [44] is a software model developed for programming intelligent agents. Superficially categorized by the implementation of an agent's beliefs, desires, and intentions are used to solve a specific problem in agent programming. In core, a method is provided for identifying the activity of selecting a plan from the performance of currently active plans. Thus BDI agents are capable of levelling the time consumed on thinking about plans and performing those plans. However, a third activity, planning, is not within the scope of the model and is left to the system designer and programmer.

In order for realizing this identification, this software model implements The Belief-Desire-Intention (BDI) model of human practical reasoning, which was developed by M. E. Bratman as a way of explaining a future-directed intention. This means that it implements the notions of belief, desire and intention in a way motivated by him. For him, intention and desire are both professional attitudes and mental attitudes concerned with action, but the intention is various as a behaviour adjusting professional attitude. He classifies an obligation as the feature factor between desire and intention, noting which it guides to temporal persistence in plans and further plans being made on the foundation of those to which it is already dedicated. This software model moderately addresses these issues. Temporal persistence is not examined in respect of explicit reference to time. The hierarchical nature of plans is more easily implemented: a plan consists of a number of steps, some of which may appeal against other plans. Since the primary plan remains, in effect, while subsidiary plans are being executed, the hierarchical explanation of plans itself implies a kind of temporal persistence.

An important aspect of this software model in terms of its significant research is the existence of logical models in which it is possible to define and reason about agents. For example, research in this area has led to the assumption of some implementations, as well as to proper logical explanations such as Anand Rao and Michael Georgeff's Belief-Desire-Intention and Computation Tree Logic (BDICTL). In addition, the Logic of Rational Agents (LORA) has been incorporated action logic into BDICTL by Michael Wooldridge, which allows reasoning not only about single-agents, but also about communication in a multi-agent system.

The BDI software model is thoroughly accompanying with intelligent agents, but does not guarantee all the features associated with such agents. For example, it allows agents to have private beliefs, but does not force them to be private. Furthermore, it has nothing to perform about agent communication. Finally, this software model is an attempt to solve a problem which has more to do with plans and planning than it has to do with the programming of intelligent agents.

1.3.1 Architecture

The ideal architectural components of this software model are as follows:

1. Beliefs represent the informational state of the agent, in other words, they represent its opinions about the environment. They can also comprise inference rules, which lead to new beliefs. They are saved in a database (belief base or belief set) although, which is an implementation decision.
2. Desires represent the motivational state of the agent, put differently, they represent objectives or situations which it would like to accomplish or make something happen.
3. Goals represent the active benefited desire of the agent. Usage of this term adds the further restriction which the set of active desires must be consistent.
4. Intentions represent the deliberative state of the agent; this means that what the agent has selected to do. They are desires to which the agent has, to some extent, committed; this means the agent has begun executing a plan in implemented systems.
5. Plans represent the process of the agent which it can perform to achieve one or more of its purposes, and may include other plans. This reflects in Bratman's

model, which they are originally only incompletely considered with details being filled in as they progress.

6. Events represent the triggers for reactive activity of the agent, which may update beliefs, trigger plans or modify goals. They may be generated externally and received by sensors or integrated systems. They may be generated internally to trigger decoupled updates or plans of activity.

1.3.2 BDI Interpreter

An ideal BDI interpreter is defined [Figure 2] and afforded the foundation of the Procedural Reasoning System (PRS) which it is a framework for constructing real-time reasoning systems, which can accomplish complex tasks in dynamic environments [17].

```
initialize-state
repeat
    options: option-generator(event-queue)
    selected-options: deliberate(options)
    update-intentions(selected-options)
    execute()
    get-new-external-events()
    drop-unsuccessful-attitudes()
    drop-impossible-attitudes()
end repeat
```

Figure 2: Pseudo Code of Main Loop of a BDI Agent [44]

It should be noted, this basic algorithm has been extended in many ways, for instance, to support planning ahead [18], automated teamwork [19], and maintenance goals [20].

1.4 Emotional Belief-Desire-Intention (EBDI)

An important aspect of this software model in terms of its significant research is by introducing primary and secondary emotion into BDI architecture, H. Jiang and J. M. Vidal present a generic architecture for an emotional agent, EBDI [29] [30], which can merge various emotion theories with the reasoning process of an agent. It implements practical reasoning techniques separately from the specific emotion mechanism. The separation allows us to plug in emotional models as needed or upgrade the agent's reasoning engine independently.

1.4.1 Architecture

The state of an EBDI agent is defined by E, B, D and I, which they are the set of all possible emotions, beliefs, desires and intentions. These components are connected by Belief Revisions, Emotion Updates, Option Generate, Filter, Plan, and Plan Execution functions. Its architecture is shown [Figure 4].

1.4.2 EBDI Interpreter

The interpreter is shown by [Figure 3] used in the architecture. The architecture is managed to integrate emotions into the standard processing loop of a BDI agent.

```
E ← E0;              E0 are initial emotions
B ← B0;              B0 are initial beliefs
I ← I0;              I0 are initial intentions
while true do
    Bp ← brf -see (Env);
    Bm ← brf -msg (C ont);
    E ← eufl (E, I , Bp ∪ Bm );
    B ← brf -in (E, I , B ∪ Bp ∪ Bm );
    D ← options (B, I );
    I ← filter (E, B, D, I );
    E' ← E
    E ← euf2 (E, I , B);
    if time permits and E = E'
            then B ← brf -in (E, I , B);
                 D ← options (B, I );
                 I ← filter (E, B, D, I );
    π ← plan (I , Ac);
    execute(π)
```

Figure 3: Pseudo Code of a Main Loop of an EBDI Agent [29]

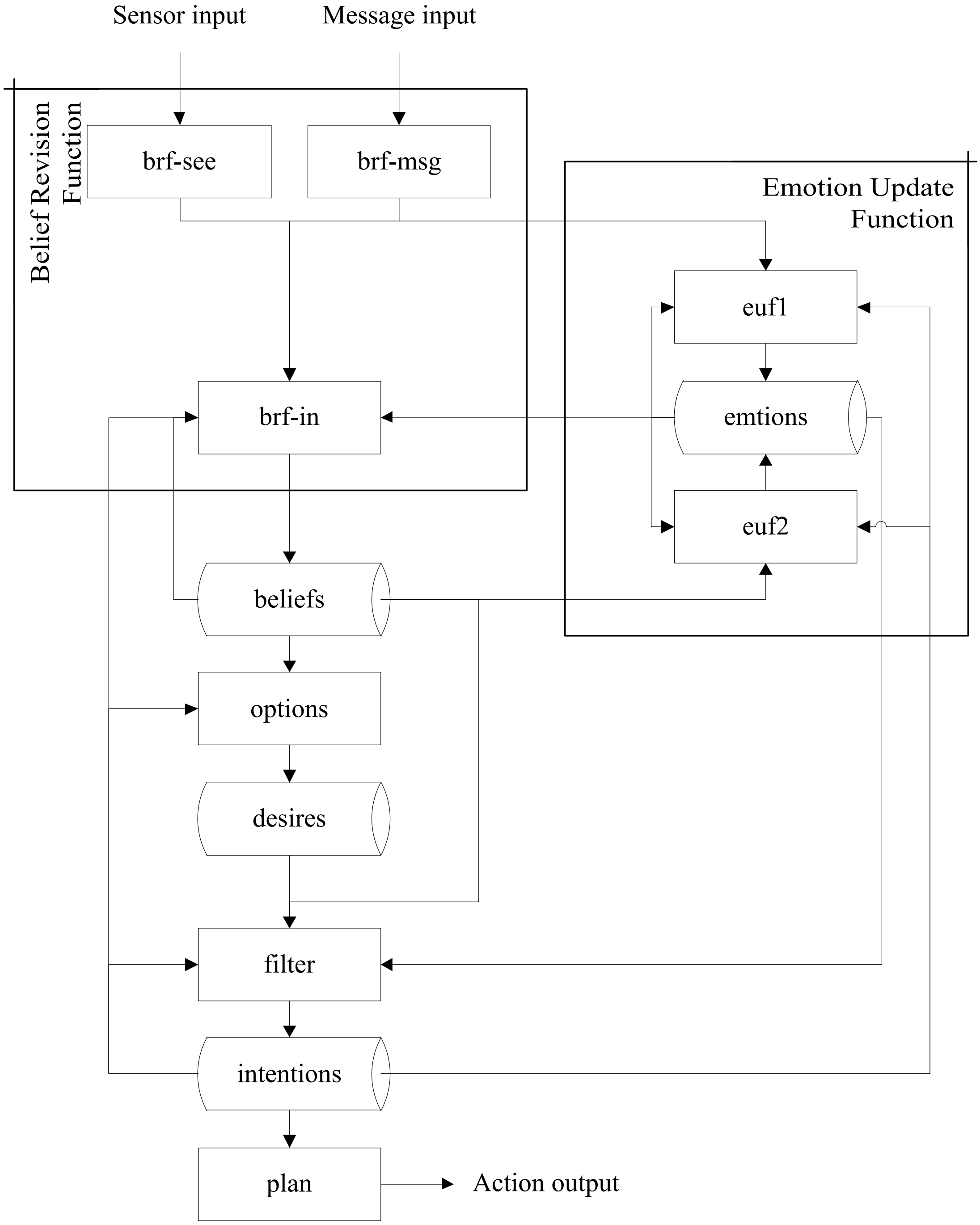

Figure 4: Schema of EBDI Architecture [29]

The execution cycle can be summarised as follows:

1. When some new information is sensed from the environment by sensor or communication messages, belief sets are generated.

2. These beliefs with current intentions trigger emotional updating, thus first feelings of the agent are acquired about the information.
3. Beliefs are evaluated again based on the new emotion status and information with current intentions as a guide.
4. Desires are generated from the beliefs and intentions.
5. The best options or intentions are chosen based on current beliefs, desires and intentions under influences of the emotions. Current working intention often has the highest priority because of intentions persist, unless they are already achieved, are found impossible to achieve or the reason for this intention is no longer existing.
6. The secondary emotions are triggered and updated based on current intentions, beliefs and previous emotions.
7. If there is no time for careful consideration, or emotion status is not changed, detail plan will be generated and will be executed; otherwise the decision to make is considered and refined precisely. It will be reconsidered if current beliefs are appropriate, as in line 14 and reconsider the desires and intentions, as in line 15 and 16. After this reconsideration, the agent then generates a plan and executes it.

Chapter 2

AN INTRODCUTION TO AGENTS

2.1 Intelligent Agent

This autonomous entity observes and performs in an environment and directs its activity to obtain goals. It may also acquire or practise knowledge to realise their goals. It may be very simple or complex: a reflex machine such as a thermostat or human being works with others to obtain a goal in a human community.

It is often defined schematically as a simple practical system akin to a computer program. For this reason, it is sometimes called abstract intelligent agent to various them from their real world implementations as computer systems, biological systems or organizations. Some definitions of it emphasize their independence and prefer the term autonomous intelligent agent. Still others, notably Russell and Norvig [36], consider goal-directed behaviour based on intelligence and prefer a term borrowed from economics, "Rational Agent".

Agents are sorted into five types based on their level of perceived intelligence and capability by Russell and Norvig: Simple Reflex Agents, Model-Based Reflex Agents, Goal-Based Agents, Utility-Based Agents and Learning Agents.

2.1.1 Simple Reflex Agents

They perform only based on the current percept. Their function is based on a condition than an action. They function to succeed when the environment is fully observable. Some of them can also include information on their current state, which permits them to ignore conditions are already triggered by actuators.

2.1.2 Model-Based Reflex Agents

They can handle partly observable environments. Its current state is saved inside the agent supporting some kind of structure, which describes the inaccessible part of the environment. This behaviour demands information on how the environment performs and operates. This additional information finalises the "World View" model. They keep following the current state of the environment using an internal model and choose an action in the same way as the reflex agent.

2.1.3 Goal-Based Agents

They are model-based agents which save information concerning circumstances, which are necessary. This aims the agent to choose the one, obtaining a goal state, among multiple possibilities.

2.1.4 Utility-Based Agents

They only difference between goal states and non-goal states. It is possible to define a measure of how necessary a particular state is. This measure can be developed by a utility function which plans a state to a measure of the utility state.

2.1.5 Learning Agents

They have an advantage which it allows the agents to operate originally in unidentified environments and to develop more competent than their basic knowledge alone might allow.

2.2 Rational Agent

In AI, it is an agent which has clear preferences, uncertain models by anticipated values and often chooses to complete the action which results in the best result for itself from among all possible actions. In other words, a rational agent can be anything, which makes decisions, typically a person, to firm machine or software with intelligence. AI borrowed the term "Rational Agents" from economics to define independent programs, which are capable of direct goalless behaviour.

2.3 Multi-Agent Systems

They are systems composed of multiple interacting intelligent agents. They can be used to explain difficulties, which are problematic or impossible for a monolithic system or individual agent to solve.

2.4 Agent-Based Model

It is a class of computational models for simulating the operations and interactions of individual and communal autonomous agents with a view to evaluating their effects on the system as a whole. The models simulate the concurrent actions and interactions of multiple agents, in an attempt to remake and predict the impression of complex phenomena. The process is one of the emergences from a micro system to a macro system. A key notion is simple behavioural rules, which generate complex behaviour. This principle, known as “Keep it simple and stupid” (KISS), is introduced by Robert Axelrod [38] before all else and is extensively accepted in the modelling community. Another central principle is which the whole is greater than the sum of the parts. Individual agents are classically described as bounded rational, supposed to be acting in what they observe as their own benefits, such as reproduction, economic benefit or social status, using heuristics or simple decision to make rules [39]. They also may experience learning, adaptation and reproduction [40].

2.5 Agent-Based Modelling

It [37] is a system made up of a group of agents which autonomously interact on networks. Each individual agent is reliable for various behaviours, which result in communal behaviours and help to define the workings of the network as a whole. It focuses on human social interactions and how people collaborate and communicate with each other without having a single grouped mind. This means that it tends to focus on the result of interactions between people or agents in a population. This type of modelling is better understood by modelling these dynamics on a smaller and more localized level. Simple individual rules or actions can result in the consistent group behaviour. Changes in these individual performances can affect the communal group in any given population.

In other words, it is an exploratory tool for hypothetical research. It enables one to deal with complex individual behaviours. In general, based on this type of modelling, the researchers propose to model the behaviour of agents and the communication between them in order for better understanding how these individual interactions influence an entire population. In essence, it is a way of modelling and understanding various global patterns.

Verification and validation of simulation models are extremely important [41] [42]. Verification includes debugging the model to guarantee it works correctly and Validation guarantees which the right model has been made.

2.6 Performance

Performance is a function which measures the quality of the actions the agent took. Such as Safe, Fast, Legal, Comfortable trip, Maximize Profit, etc.

2.7 Environment

The environment refers to the area in which an agent or a set of agents operate. If a sensory apparatus of agent gives it access to the complete state of the environment, then the environment is said to be accessible to that agent. In an accessible environment, the sensors detect all aspects that are relevant to the choice of action.

On the other hand, if the next state of the environment is completely determined by the current state and the actions selected by the agents, then environment is said to be deterministic. If only actions of other agents are non-deterministic, the environment is called strategic.

Considering the interaction of agents with their environment, if the environment can change while an agent is acting, then the environment is called dynamic for that agent, other-wise it is static. Static environments are easy to deal with be-cause the agent need not keep looking at the environment while it is deciding on an action.

2.8 NetLogo

NetLogo [1] was designed in the essence of the Logo programming language to be "low threshold and no ceiling" enabled easy entry by beginners and yet meet the needs of advanced users. This simulation environment aids investigation of developing phenomena. It includes an extensive library of models in many domains such as art, biology, chemistry, physics, computer science, earth science, mathematics, networks, social science, system dynamics and many others. It is enabled a quick and easy model development and particularly well suited for modelling complex systems developing over time. It can give commands to hundreds or thousands of autonomous agents and all operating parallels. Thus this makes it able to investigate the relation between the micro-level behaviour of individuals and the macro-level patterns which arise from the interaction of many individuals.

It should be noted, since the commands and definitions for NetLogo are standard and same as other guidance and tutorials, as some common words may be founded in this section which they are unavoidable. Significantly, they are adopted from “NetLogo Programming Guide”. Some modification has been done with respect to the studied model to help readers for greater understanding. Nevertheless, the main issue is covering all programming guide lines related to the model.

2.8.1 Understanding the Graphic User Interface (GUI)

It has three tabs at the top: the interface tab, which has tools to check and change what is occurring inside the model and shows a model run; the information tab, which typically has some information on model and mechanism written by modeller; and the procedure tab, which typically includes all the relevant code for making the model. Only one tab at a time can be visible, but switching between them can be done by clicking on the tab titles at the top of the window and below of the main menu or by using shortcut keys.

The interface tab shows the environment and includes various ways of setting variables for the model, such as buttons, sliders, choosers and switches, and viewing results, such as plots, monitors and output. By right-clicking on the environment, one can manually adjust its overall size, layout and number of patches by the "model settings" window [Figure 5].

The modeller can declare and set values of variables and issue commands by the model interface, using the "add" button shown at the top of the interface. This provides a number of options for manipulating variables, depending on their type, such as true/false, discrete and continuous. The "button" option is typically used for executing procedures. "The NetLogo User Manual" has a detailed description of how to add new and adjust new variables and run procedures by the interface.

Most programs typically have at least two buttons in the interface: a "setup" button, which runs the procedure which initializes the environment, and a "go" button, which runs the procedure which executes some set of commands run in each step of the model. It is usually wanted the "go" procedure to run indefinitely or until some criteria are met. Rather than pressing once for each model step, it is usually making this a “forever” button. This means that once pressed this command will continue to run until the modeller presses the button again or something internal to the program makes it stop.

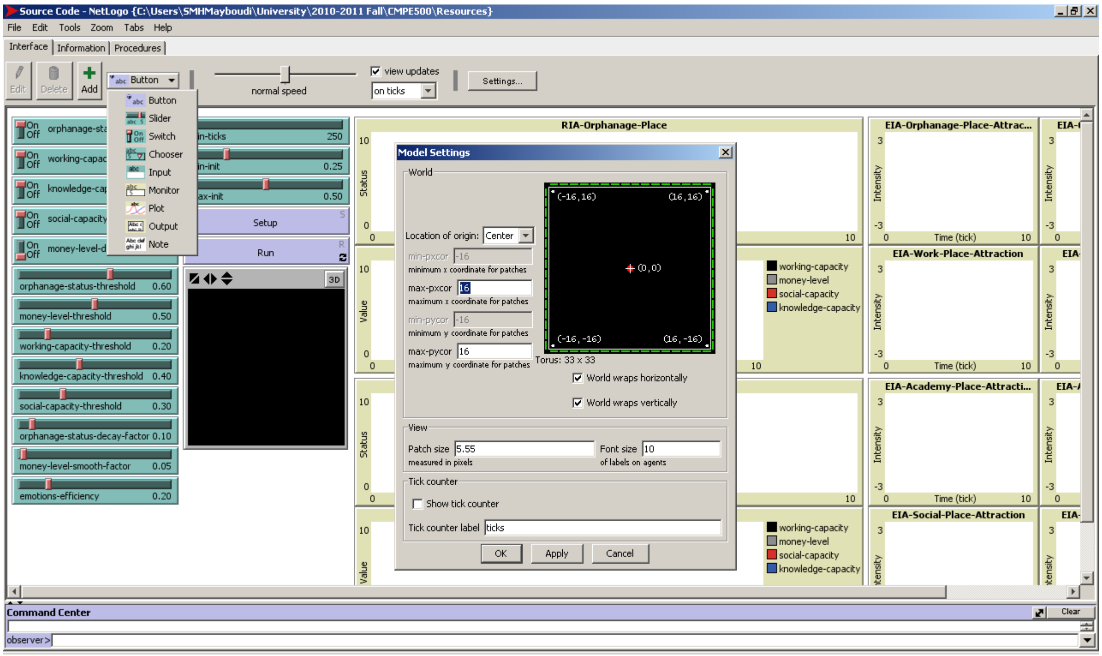

Figure 5: NetLogo GUI

2.8.2 Agents

The simulation environment is made up of agents. Agents are beings, which can follow instructions. Simultaneously, each agent can carry out its own activity. There are four types of agents: turtles, patches, links, and the observer.

Turtles are agents which move around in the environment and have two coordinates, “xcor” and “ycor”. These coordinates are the decimal, which means a turtle can be positioned at any point within its patch, in other words, it doesn't have to be in the centre of the patch. The environment is two dimensional and is divided up into a grid of patches. Each patch is a square piece of ground over which turtles can move. Patches have two coordinates, “pxcor” and “pycor”, and are always the integer.

Links are agents which connect two turtles. They have two endpoints instead of two coordinates, along the shortest path possible, even if means wrapping around the environment.

The observer doesn't have a location; this means that one can imagine it as looking out over the environment of turtles, patches, and links.

2.8.3 Primitives and Procedures

Commands and reporters tell agents what to do. A command is an action for an agent to carry out. A reporter computes a result and reports it. Most commands begin with verbs, such as “ask”, "clear", "create", “set”, and so on, whereas most reporters are nouns or noun phrases.

Commands and reporters built into NetLogo are called primitives, which are listed in “The NetLogo Dictionary”. Nonetheless, defined ones are called procedures. Each of them has a name, preceded by the “to” keyword. The “end” keyword marks the end of the commands in the procedure [Figure 6]. Once a procedure is defined, it can be used elsewhere in a program.

Both primitives and procedures can take inputs. To create a procedure which accepts inputs, include a list of input names in square brackets after the procedure name [Figure 6].

```
to Setup-all …
  RIA-Setup …
end
to Run-all [tick-counter] ; tick-counter is always loading min-ticks
  if ticks > tick-counter [ … stop ]
  RIA-Run …
end
```

Figure 6: Sample Code for Defining Procedures without/with Input Variable

Just like commands can be defined, reporters as well. However, two special things should be done: use the “to-report” keyword instead of the “to” keyword to begin the procedure, then in the body of the procedure, use the "report" command to report the value is wanted to report [Figure 7].

```
to-report Utility-Random-Float
  report min-init + random-float (max-init - min-init)
end
to-report Utility-Random [range]
  report -1 ^ random (2) * Random range
end
```

Figure 7: Sample Code for Defining Reporters without/with Input Variables

It should be noted, the "clear-all" and "crt" commands can only be run by the observer. Nevertheless, the "fd" command can only be run by turtles. In addition, some other commands and reporters, such as "set", can be run by different agent types.

2.8.4 Variables

Variables of an agent are places to store values, such as numbers, in the agent. They can be a global variable for observer or private variable for a turtle, patch, or link. If a variable is a global variable, there is only one value for it, and can be accessed at any time by every agent. It is possible to think of global variables as belonging to the observer. Turtle, patch, and link variables are different. Each turtle has its own value for every turtle variable, and each patch has its own value for every patch variable, and the same for links. Some variables are built into the simulator.

It is possible to make a global variable by adding a switch or a slider to a model, or by using the "globals" keyword at the beginning of a code [Figure 8].

```
globals [
  EIA-Academy-Place-random-selection
  EIA-Social-Place-random-selection
]
```

Figure 8: Sample Code for Defining Global Variables

Furthermore, new variables of turtle, patch, and link can be defined by using the "turtles-own", "patches-own", and "links-own" keywords [Figure 9].

```
breed [RIA-Work-Places RIA-Work-Place]
RIA-Work-Places-own [ …
 offering-salary ; (random setting)
 job-duration ; (random setting)
]
```

Figure 9: Sample Code for Defining Breeds and their Variables

It should be noted, the use of semicolons to add comments to a program. This makes the program easier to read and understand.

These variables can then be used freely in a model. Use the "set" command to set them otherwise their default value is zero [Figure 10]. Global variables can be read and set at any time as well as patch variables of the patch, which is stood on, by a turtle. In other situations where an agent is wanted to read a different agent's variable, the "of" keyword can be used.

```
to RIA-Run
 ask RIA-Work-Place 2 [
  set offering-salary Utility-Random-Float
  set job-duration 1
 ] …
end
```

Figure 10: Sample Code for Setting Variables

A local variable is defined by the "let" keyword, and used only in the context of a particular procedure or part of a procedure [Figure 11].

```
to RIA-Visit-Work-Places …
 let offering-salary-duration [ offering-salary ] of RIA-Work-Place 2 / [ job-
duration ] of RIA-Work-Place 2 …
end
```

Figure 11: Sample Code for Defining Local Variables

2.8.5 Lists

In the simplest models, each variable holds only a piece of information, usually a number or a string. The list feature allows storing multiple pieces of information in a single variable by collecting those pieces of information in a list. Each value in the list can be any type of value, such as a number, or a string, an agent, or even another list.

Lists allow for the convenient packaging of information. If agents carry out a repetitive calculation on multiple variables, it might be easier to have a list variable, instead of multiple number variables. Several primitives simplify the process of performing the same computation on each value in a list.

A list can simply be made by putting the wanted values in the list between brackets. The individual values are separated by spaces. If a list is wanted to be made which the values are determined by reporters, contrary to being a series of constants, by using the “list” reporter. The “list” reporter accepts two other reporters, runs them, and reports the results as a list [Figure 12].

```
to RIA-Setup …
  create-RIA-Academy-Places 1 [ …
    set course-list ( list "CMPE101" … ) …
    set course-duration-list ( list 1 … ) …
    set course-fees-list ( list Utility-Random-Float … ) …
    ] …
end
```

Figure 12: Sample Code for Defining Lists

It should be noted, some commands and reporters involving lists and strings may take a varying number of inputs. In these cases, in order to pass them a number of inputs other than their default, the primitive and its inputs must be surrounded by parentheses.

Lists are singly linked lists, in other words, when software needs to find an item in a list, it must start at the beginning of the list and go from item to item until it finds the one it wants.

The empty list is written by putting nothing between the brackets. To add an item, to the end of a list use the “lput” reporter; the beginning, “fput”.

The “foreach” command is used to run a command or commands on each item in a list. It takes an input list and a block of commands. The variable “?” holds the current value from the input list [Figure 13].

If an entire list is wanted to be operated, some other technique may be needed to use such as a loop using “repeat” or “while” [Figure 13], or a recursive procedure which is inadvisable.

```
to RIA-Visit-Academy-Places …
  let choices [ ]
  foreach … [ ? < [ money-level ] of RIA 0 ] … [ set choices lput … ] …
  while [ [ status ] of RIA-Orphanage-Place 1 > orphanage-status-threshold and
…] [ … ] …
end
```

Figure 13: Sample Code for Using Loops

2.8.6 Random Numbers

The random numbers are what are called "pseudo-random". Nonetheless, they are actually generated by a deterministic process. This means that the same results are gotten every time, if the same random seed is used.

If the random seed is not set, it is set to a value based on the current date and time. If a model is wanted to run to be reproducible, the random seed must be set ahead of time. Therefore, there is no way to find out what random seed it chose.

In the context of scientific modelling, pseudo-random numbers are actually desirable. Because of this, a scientific experiment can be reproducible. Since pseudo-random numbers are used, the experiments can be reproduced by others.

In addition to the uniformly distributed random integers and floating point numbers generated by the "random" and "random-float" commands, other random distributions are also offered such as "random-normal", "random-poisson", "random-exponential", and "random-gamma" [Figure 14].

```
to-report Utility-Random-Float
  report min-init + random-float (max-init - min-init)
end
to-report Utility-Random
  report -1 ^ random (2) * Random 4
end
```

Figure 14: Sample Code for Using Random Variables

2.8.7 Tick Counter

In many models, time passes in discrete steps, called "ticks". A built-in tick counter is included so track can be kept of how many ticks have passed [Figure 15]. The current value of the tick counter is shown above the view. The button of settings can be used to hide the tick counter, or change the word "ticks" to something else.

In code, to retrieve the current value of the tick counter, use the "ticks" reporter. The "tick" command advances the tick counter by 1. The "clear-all" command resets the tick counter to 0 [Figure 15]. If resetting the counter to 0 is wanted without clearing everything, use the "reset-ticks" command. If a model is set to use tick-based updates, then the tick command will usually also update the view.

```
to Setup-all
  clear-all …
end
to Run-all
  if ticks > min-ticks  […] …
  tick …
end
```

Figure 15: Sample Code for Using Ticks Counter

2.8.8 Plotting

Plotting features create plots, which help to understand what's going on in a model. Before a plot can be plotted, one or more plots are needed to create in the Interface tab. Each plot should have a unique name. Its name will be used to refer in a code of the model.

If there is only one plot in the model, it is possible to start plotting it right away; more than one plot, it should be specified which one is wanted to plot to. To do this, use the "set-current-plot" command with the name of the plot enclosed in double quotes [Figure 16].

When a new plot is made, it just has one pen in it. If the current plot only has one plot pen, then plotting to it can be started right away. However, multiple pens can also be used in a plot. Additional pens can be created by editing the plot and using the controls in the section of plot pens at the bottom of the edit dialog. Each pen should have a unique name. Its name will be used to refer to it in the code. For a plot with multiple pens, a pen has to be specified to plot. If a pen is not specified, plotting will take place with the first pen in the plot. To plot with a different pen, use the "set-current-plot-pen" command with the name of the pen enclosed in double quotes.

With "plot" the y value wanted plotted, should only be specified [Figure 16]. The x value will automatically be 0 for the first point, 1 for the second, and so on, if "interval" of the plot pen is the default value of 1. The interval can be changed.

```
to RIA-Plot …
  ask RIA-Orphanage-Place 1 [
    set-current-plot "RIA-Orphanage-Place"
    set-current-plot-pen "status"
    plot status * 100
    ] …
end
```

Figure 16: Sample Code for Plotting Quantisation

2.8.9 Strings

To input a constant string, surround it with double quotes. The empty string is written by putting nothing between the quotes, like "". A few primitives are specific to strings, such as “substring”, and “word” [Figure 17].

```
to Run-all
  if ticks > min-ticks [ ...
    let dateandtime ( word remove ":" substring date-and-time 0 5 remove "-"
remove " " substring date-and-time 13 27) ...
  ]...
end
```

Figure 17: Sample Code for Using Strings

2.8.10 Output

This part is about output to the screen. Output to the screen can also be later saved to a file using the “export-interface”, “export-output” and “export-world” command. The basic commands for generating output to the screen are the “print”, “show”, “type”, and “write” commands. These commands send their output to the Command Centre [Figure 18].

A NetLogo model may optionally have an "output area" in its Interface tab, separate from the Command Centre. To send output there instead of the Command Centre, use the “output-print”, “output-show’, “output-type”, and “output-write” commands.

```
to Run-all
  if ticks > min-ticks  [
    let dateandtime  …
    export-interface  (word dateandtime  " Interface.png")
    export-output  (word dateandtime  " Output.txt")
    export-plot "RIA-Orphanage-Place"  (word  dateandtime  " RIA-OS.csv")
    export-plot "RIA-Properties"  (word dateandtime  " RIA-Properties.csv")
    export-plot "EIA-Orphanage-Place"  (word dateandtime  " EIA-OS.csv")
    export-plot "EIA-Properties"  (word dateandtime  " EIA-Properties.csv")
    export-world  (word dateandtime  " World.csv")
    stop
  ]
  RIA-Run
  EIA-Run
  tick
  print "-------------------------------------------------------------------------------"
end …
to RIA-Track  …
  ask RIA  0 [ …
    type " RIA  OS: " type  precision  [ status  ] of RIA-Orphanage-Place  1 2 …
    ]
end
```

Figure 18: Sample Code for Exporting Output

Chapter 3

THE PROPOSED WORK

Regarding the potential roles of artificial agents [31], as mentioned in basic explanations, the role of emotions in agent-based modelling and the control of emotionally intelligent agents are studied by using a simple multi-agent system. EBDI is selected for emotional architecture and is based on the OCC cognitive appraisal model. Furthermore, the triple tower model [43] is used as a general conceptual model for the agents with some extension, which deal with the "Perceptron and Evaluation" and the "Rational and Emotional Reasoning" mechanisms due to be increasing the performance and decreasing the run-time [Figure 19]. Concerning the acting environment, it is observable, stochastic (non-deterministic), sequential (non-episodic), dynamic, discrete and single-agent.

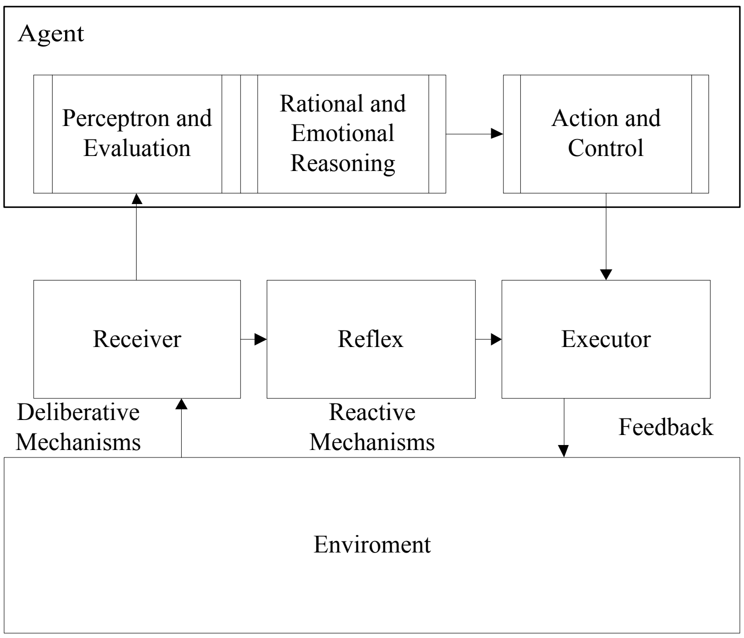

Figure 19: Generalized Conceptual Model of Agents

3.1 Orphanage Care Problem

The orphanage care problem [28] is selected as a test case and NetLogo, a computer based modelling tool, is selected as a simulation environment. Two various goal-based intelligent agents have been equipped with rational and emotional behaviour separately. These agents have been titled Rational Intelligent Agent (RIA) and Emotional Intelligent Agent (EIA). The environment is divided into two regions. Entities in both regions are identical and thus behave in the same manner. Each agent has an orphanage, a workplace, an academy place, and a social club [Figure 20], which they are objectives.

The parameters and objects of the above described as simulation environment are parts of a realistic multi-objective scenario in which the two agents, RIA and EIA mentioned previously, act to achieve their goals. All parameter values are normalized within the interval between zero and one. The main goal of each agent is to keep status of orphanage above a predefined threshold level.

To achieve the main goal, agent should spend money on the orphanage; or go to, and work in. This depends on its money level. Thus it should go to the workplace to earn money. In addition, it needs a working capacity to work in the orphanage or workplace. For this reason, it should go to the academy place to improve its level as well as knowledge capacity, and study their available courses. Additionally, it needs to increase its social capacity to be a companionable agent. Accordingly, it should go to the social club to enhance its level besides working, and learn social skills. The academy place and social club need expenses which cost money.

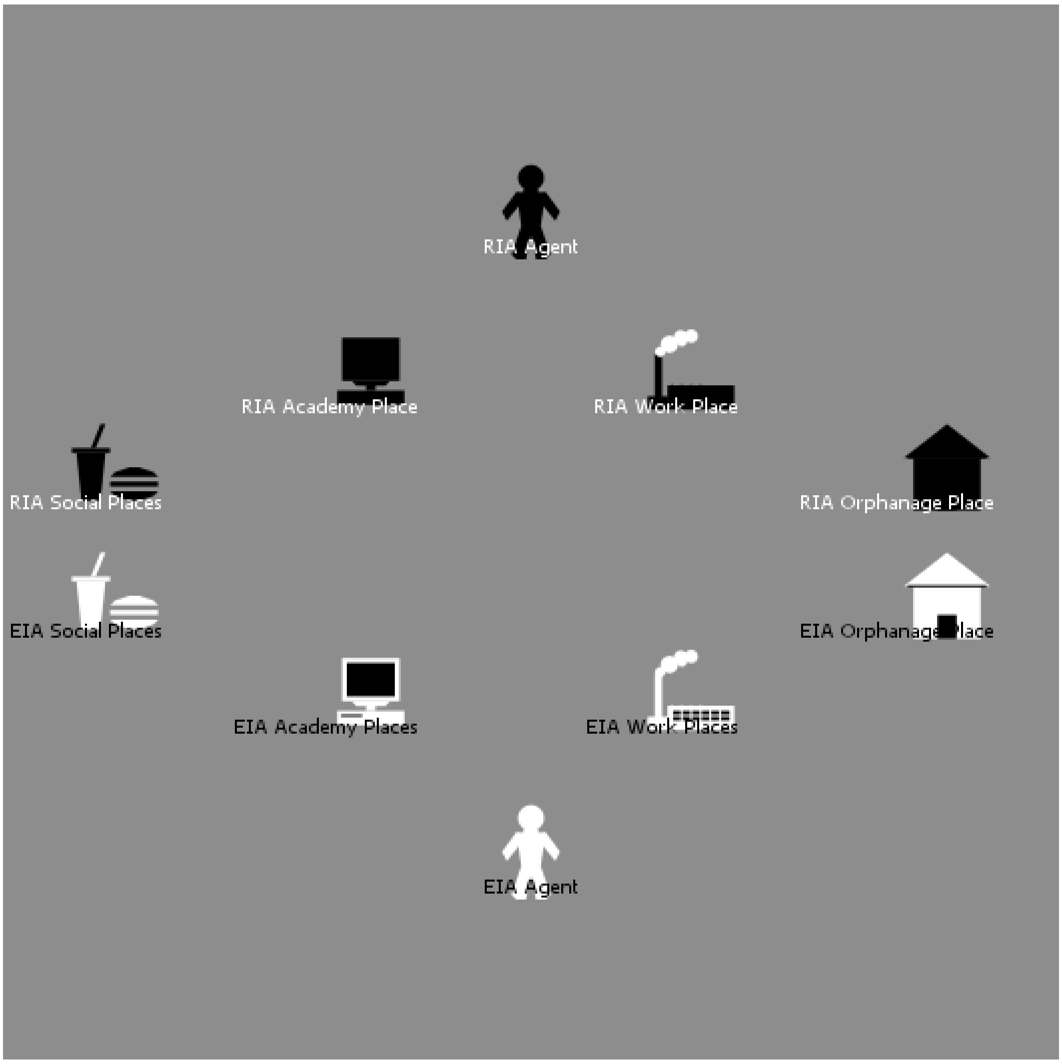

Figure 20: The Environment

3.2 Agent-Based Model

To realise the main goal, the agent should go to the orphanage place and work in it. Taking care of the orphanage depends on the working capacity of agent and its threshold. The agent should spend money on the orphanage. Supporting the orphanage financially depends on the money level of the agent and its threshold. To keep the money level from decreasing to nothing, the agent needs to find work to earn money and needs the time to finish it. The amount of salary and time for doing a job is offered by the workplace. The time should not be spent completely to gain a benefit, and it may be partial. Probably, this situation would occur when the status of the orphanage becomes less than its threshold.

Working capacity can be improved at the academy place and social club. The agent should go to the academy place to improve its knowledge capacity, thus improving its money level and studying their offered courses. Each available course requires a fee and enough time to finish its course. The expense is taken in advance and the time should be spent completely to gain a benefit; otherwise there is no improvement in working and knowledge capacity. Probably, this situation would occur when the status of the orphanage becomes less than its threshold. The "course-list" (ACL), "course-duration-list" (ACDL), "course-fee-list" (ACFL), "knowledge-capacity-gained-list" (AKCGL), and "working-capacity-gained-list" (AKCGL) values are offered by the academy and are randomly selected. Furthermore, the agent should go to the social club to improve its social capacity, thus improving its money level and joining in the offered activities. Each available activity requires a fee and time spent at the social club to complete its activity. The expense is taken in advance and the time should not be spent completely to gain a benefit, and it may be partial. Probably, this situation would occur when the status of the orphanage becomes less than its threshold. The "activity-list" (SAL), "activity-duration-list" (SADL), "activity-fee-list" (SAFL), "social-capacity-gained-list" (SSCGL), and "working-capacity-gained-list" (SWCGL) values are offered by the social club and are randomly selected.

In the agent-based model used in the simulations, agents RIA and EIA and other environmental objects have several attributes in their representations such that values of these attributes at any time are used in decision making procedures of the agents. Most of these attributes have dynamically changing values which are continuously updated at every tick. Updates are by decrements due to decaying and by increments due to participation of agents to various activities, which make the problem a realistic and challenging one. The attributes and their default values associated with RIA and EIA are illustrated [Table 2]. Similarly, the attributes and their default values associated with the four environmental objects mentioned above are illustrated [Table 3]. Values of these attributes are actually the sensory information for RIA and EIA, which are used to generate the next goal-based action to follow.

Status of orphanage is a dynamic parameter that decreases by a predefined factor, "orphanage-status-decay-factor" (OSDF), at every execution step of the simulation, which is named as the tick. Additionally, each capacity of the agents is adjustable.

Table 2: Attributes, Default Values, and Aliases associated with RIA and EIA.

Type (Agent)	**Attributes (default value), Alias**
Goals (RIA and EIA)	previous-goal ("")
	selected-goal ("")
	current-goal ("")
Agent Status (RIA and EIA)	money-level (random setting), ML
	working-capacity (random setting), WC
	knowledge-capacity (random setting), KC
	social-capacity (random setting), SC
Emotion Aspects (EIA)	event-based-emotions (0)

Table 3: Attributes associated with Environment Objects.

Object	**Attributes (default value), Alias**
Orphanage	status (its threshold), OS
Work	offering-job (random setting), WOJ
	offering-salary (random setting), WOS
	job-duration (random setting), WJD
	capacity-required (predefined), WCR
Work Emotional Relation (EIA)	attraction-emotions : Job liking, disliking (random setting), WAE1
	attribution-emotions : Agents approving, disapproving (random setting), WAE2
Academy	course-list (random setting), ACL
	course-duration-list (random setting), ACDL
	course-fees-list (random setting), ACFL
	knowledge-capacity-gained-list (random setting), AKCGL
	working-capacity-gained-list (random setting), AWCGL
Academy Emotional Relation (EIA)	attraction-emotions : Course liking, disliking (random setting), AAE1
	attribution-emotions : Agents approving, disapproving (random setting), AAE2

Object	**Attributes (default value), Alias**
Social Club	activity-list (random setting), SAL
	activity-duration-list (random setting), SADL
	activity-fees-list (random setting), SAFL
	social-capacity-gained-list (random setting), SSCGL
	working-capacity-gained-list (random setting), SWCGL
Social Club Emotional Relation (EIA)	attraction-emotions : Activity liking, disliking (random setting), SAE1
	attribution-emotions : Agents approving, disapproving (random setting) , SAE2

The performance measurements are shown by graphs in the simulation environment [Figure 29] [Figure 30] and are written as a comma-separated values (CSV) file in the same directory of simulation file. It is a simple text format for a database. Each record in the database is one line of the text file. Each field value of a record is separated from the next with a comma like “a,b,c” or “1,2,3,”. Implementations of CSV can often handle field values with embedded line breaks or separator characters by using quotation marks or escape sequences. CSV is a simple file format which is widely supported; thus it is usually used to move tabular data between various computer programs, which support the format.

3.3 Emotion Parameters

The event-based emotion [Table 2] is influenced by existing values of the status of the orphanage, the money level, the work capacity, the knowledge capacity and the social capacity. It also identifies the work in the orphanage place as an action.

The attribution emotions [Table 3] are being affected by the degree to which objects generally take actions upholding standards, particularly those the agent has to interact specifically with them.

The attraction emotions [Table 3] follow the same lines as the attribution emotion, although with the liking/disliking of objects rather than the degree to which objects take actions in order for upholding standards.

The threshold values of the objects are adjusted with the value parameter by a modeller. The attitudes of the objects are calculated randomly at all steps of the simulation with an exception for orphanage liking, as is constantly a positive value, along with the standards used by the objects within the value parameters.

3.4 Agent Processes

The main agent receives its initial states of memory and other initial parameters from the environment and goals are scanned by using attributes of objects. An appropriate goal is considered more seriously than fewer likely goals and the decision to take a relevant action is made. The reasoning and executing are essential parts of the thinking mechanism of the agent as it is mentioned.

3.4.1 Setup Procedure

The properties of each agent are initialized, for the agents, the money level, working, knowledge, and social capacities, for the orphanage place the status, for the workplace, offering job, offering salary and job, for the academy, course list, course duration list, course fees' list, knowledge capacity gained list and working capacity gained list, for the social club, activity list, activity duration list, activity fees' list, social capacity gained list and working capacity gained list are set randomly.

3.4.2 Run Procedure

A selected goal is set by reasoning procedure, and then the selected goal is visited by the agent.

3.4.3 Reasoning Procedure

a) If the status of the orphanage (OS) is less than its threshold (OST), the agent checks the money level (ML). If the money level minus its threshold (MLT) is less than the money level smooth factor (MLSF), its selected goal (SG) becomes working in the orphanage place; otherwise it needs to pay money to increase status of the orphanage. The value of the orphanage payment (OP) is calculated by equation (1), then it is decreased from the money level, and it is added to status of the orphanage. Then, it goes to reasoning again. [Figure 22]

OP = OST - OS + random-float (ML - MLT - MLSF) / (10 ^ ceiling log (ML - MLT - MLSF) 10) (1)

If the status of the orphanage is greater than its threshold, the agent checks the money level. If the money level is less than its threshold, its selected goal becomes work; otherwise it checks the knowledge capacity and social capacity. [Figure 22]

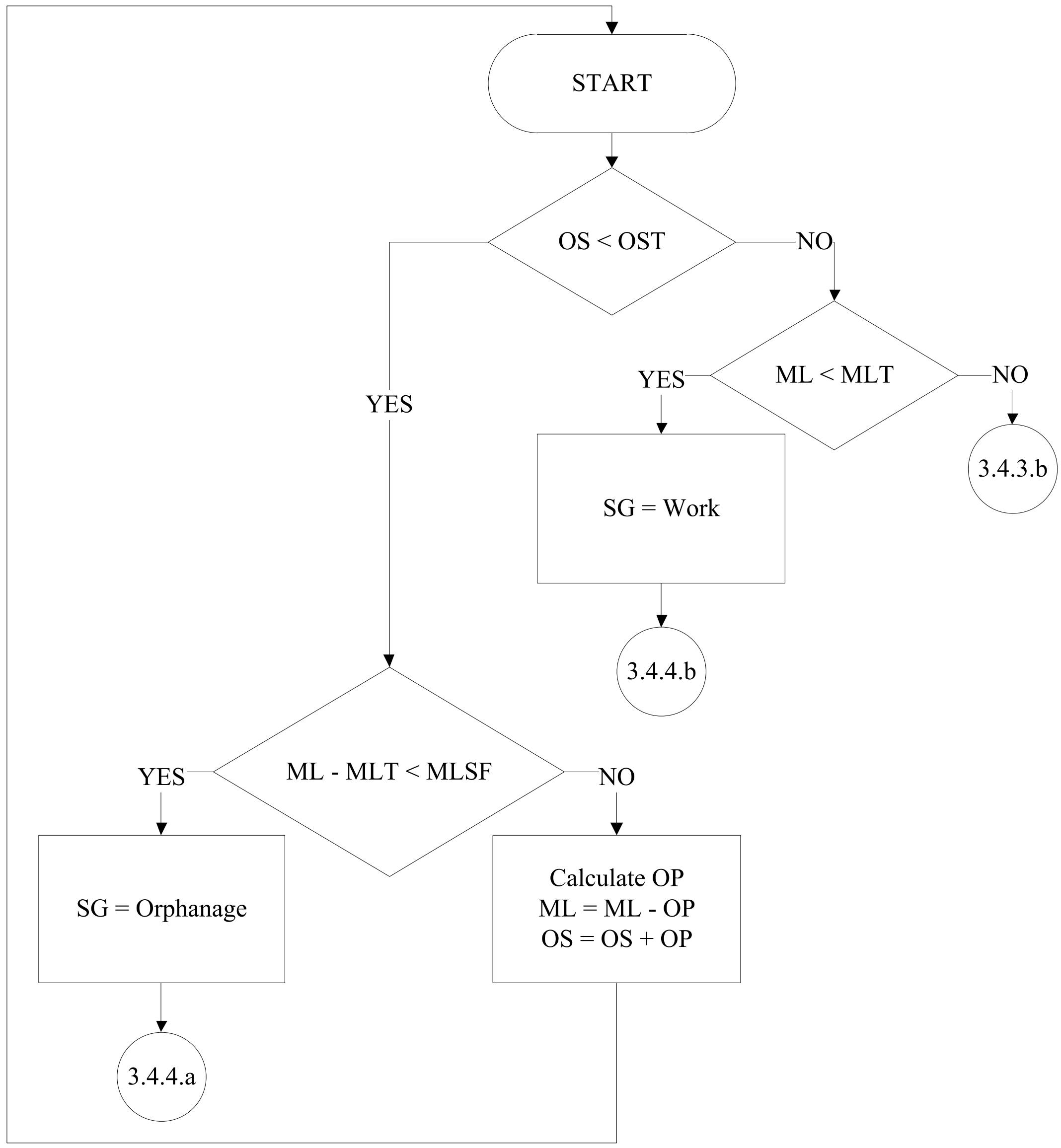

Figure 21: Reasoning associated with the OS Selection

b) If the knowledge capacity minus its threshold (KCT) is less than the social capacity minus its threshold (SCT), the agent checks the knowledge capacity and money level. If the knowledge capacity is less than its threshold and the money level is greater than the minimum needed expenses' list (NEL) for the academy, its selected goal becomes academy; otherwise it checks the social capacity and money level. If the social capacity is less than its threshold and the money level is greater than the minimum needed expenses' list for social, its selected goal becomes social; otherwise its selected goal becomes work. [Figure 22]

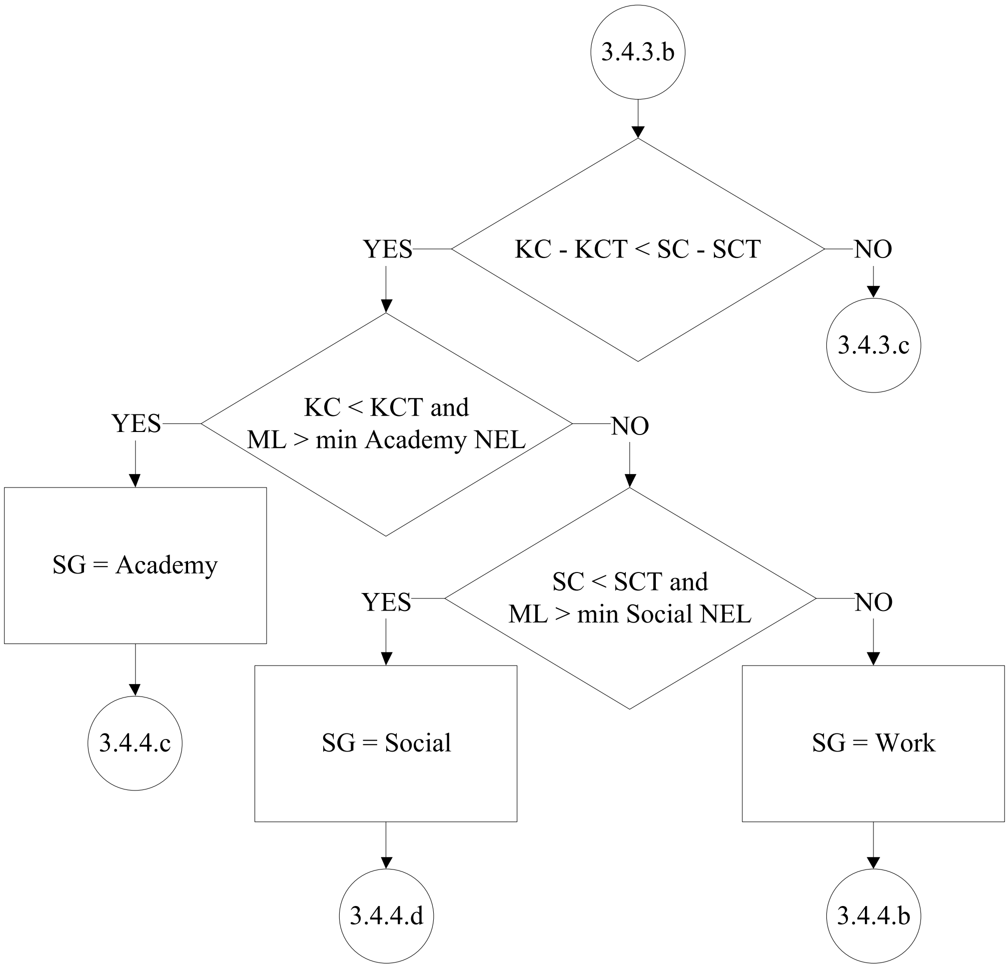

Figure 22: Reasoning associated with the KC and SC Selection

c) If the knowledge capacity minus its threshold is greater than the social capacity minus its threshold, the agent checks the social capacity and money level. If the social capacity is less than its threshold and the money level is greater than the minimum needed expenses' list for social, its selected goal becomes social; otherwise it checks the knowledge capacity and money level. If the knowledge capacity is less than its threshold and the money level is greater than the minimum needed expenses' list for academy, its selected goal becomes academy; otherwise its selected goal becomes work. [Figure 23]

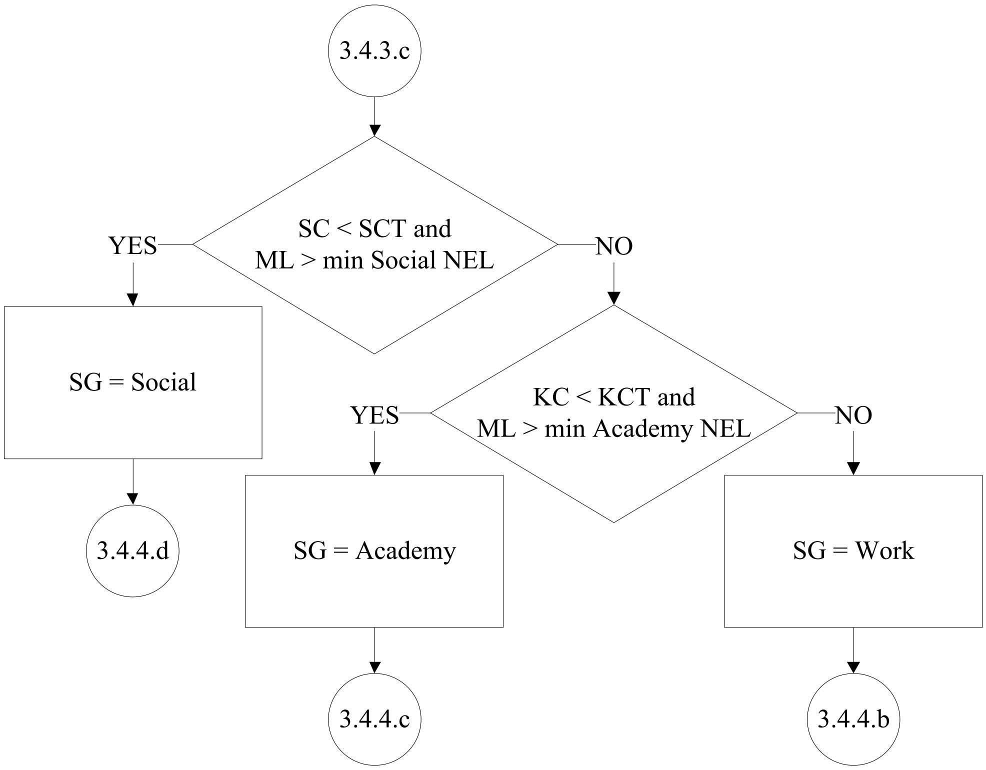

Figure 23: Reasoning associated with the SC and KC Selection

It should be noted, the fundamental difference between RIA and EIA is in the decision making related to the academy place and the social club. RIA acts based on the difference between current values and their associated with thresholds only [Appendix A: Source Code, Line 248, RIA-Reasoning-Rules]. Nevertheless, EIA selects the most affordable activities among the most attractive ones [Appendix A: Source Code, Line 764, EIA-Reasoning-Rules]. That is, EIA has preferential options among the available choices, which are unlike to RIA for which rationality is the only concern in action selection.

3.4.4 Meeting Objectives

a) If the agent visits the orphanage place, it works there while the status of the orphanage is less than its threshold. The value of the status of the orphanage is calculated by the equation (2). [Figure 24]

$$OS = OS + OSDF + Utility\text{-}Random\text{-}Float * OS \quad (2)$$

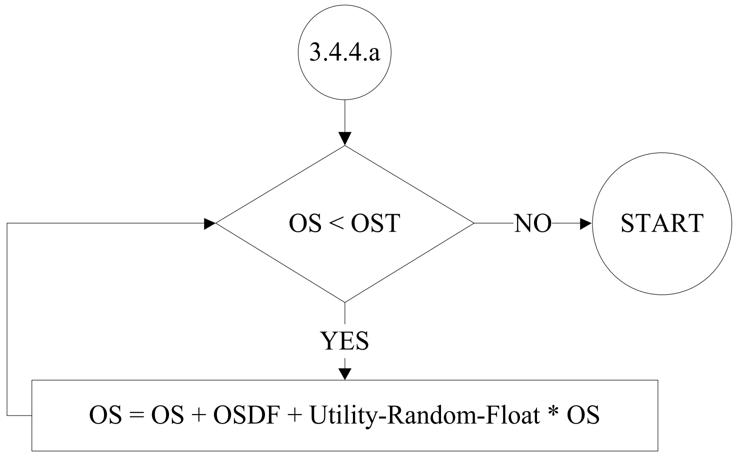

Figure 24: Orphanage Procedure associated with the OS Addition

b) If the agent visits the workplace, it works there while the status of the orphanage is greater than its threshold and the completion of work. The value of the money level is calculated by equation (3). [Figure 25]

ML = ML + WOS / WJD + Utility-Random-Float * WC + Utility-Random-Float * KC + Utility-Random-Float * SC (3)

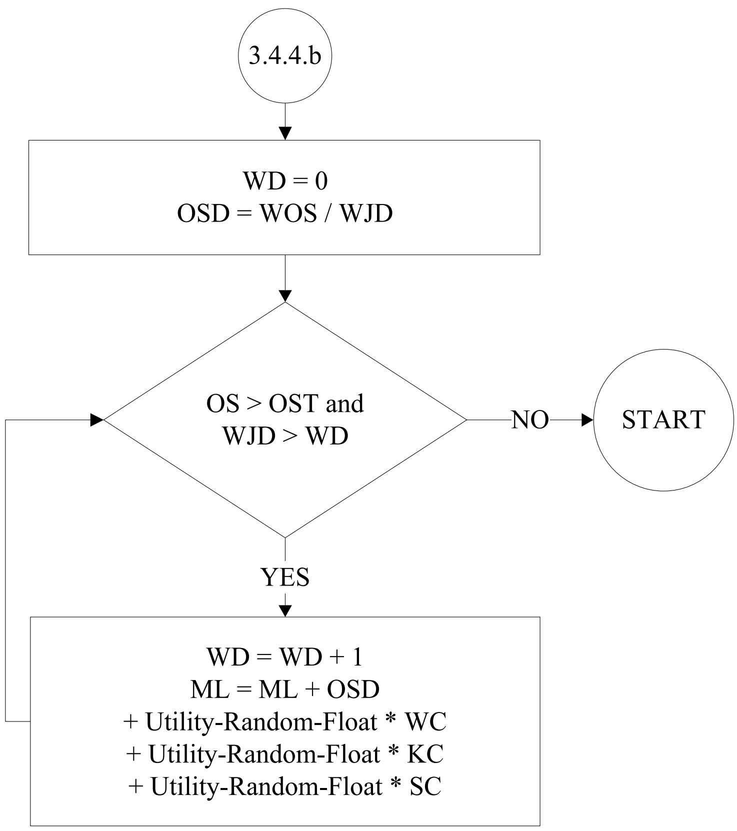

Figure 25: Work Procedure associated with the ML Addition

c) If the agent visits the academy, it selects one of the affordable courses and attends while the status of the orphanage is greater than its threshold and the completion of the course. The value of the knowledge capacity and working capacity is increased with the completion of the course. The value of the knowledge capacity is calculated by equation (4) and the value of the working capacity is calculated by equation (5). [Figure 26]

KC = KC + item random-selection AKCGL (4)

WC = WC + item random-selection AWCGL (5)

d) If the agent visits the social club, it selects one of the affordable activities and attends while the status of the orphanage is greater than its threshold and the completion of the activity. The value of the social capacity and working capacity is increased in each step. The value of the social capacity is calculated by equation (6) and the value of the working capacity is calculated by equation (7). [Figure 27]

SC = SC + item random-selection SSCGL / item random-selection SADL (6)

WC = WC + item random-selection SWCGL / item random-selection SADL (7)

It should be noted, another fundamental difference between RIA and EIA is the increase in the values of properties. A property is increased by a factor of its attribution. The new value is calculated by equation (8).

value = value * (1 + [attribution-emotions] of EIA * emotions-efficiency) (8)

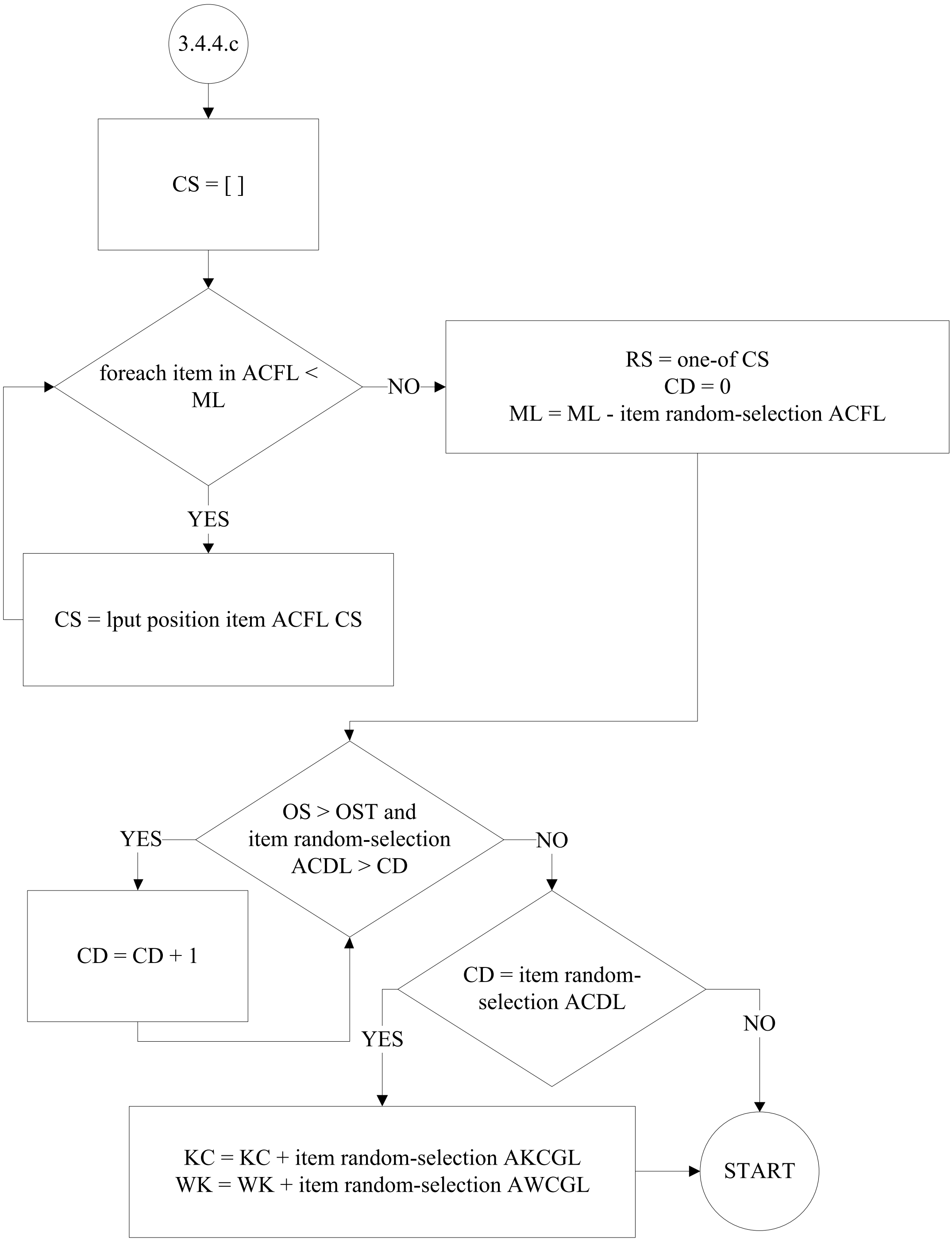

Figure 26: Academy Procedure associated with the KC and WC Addition

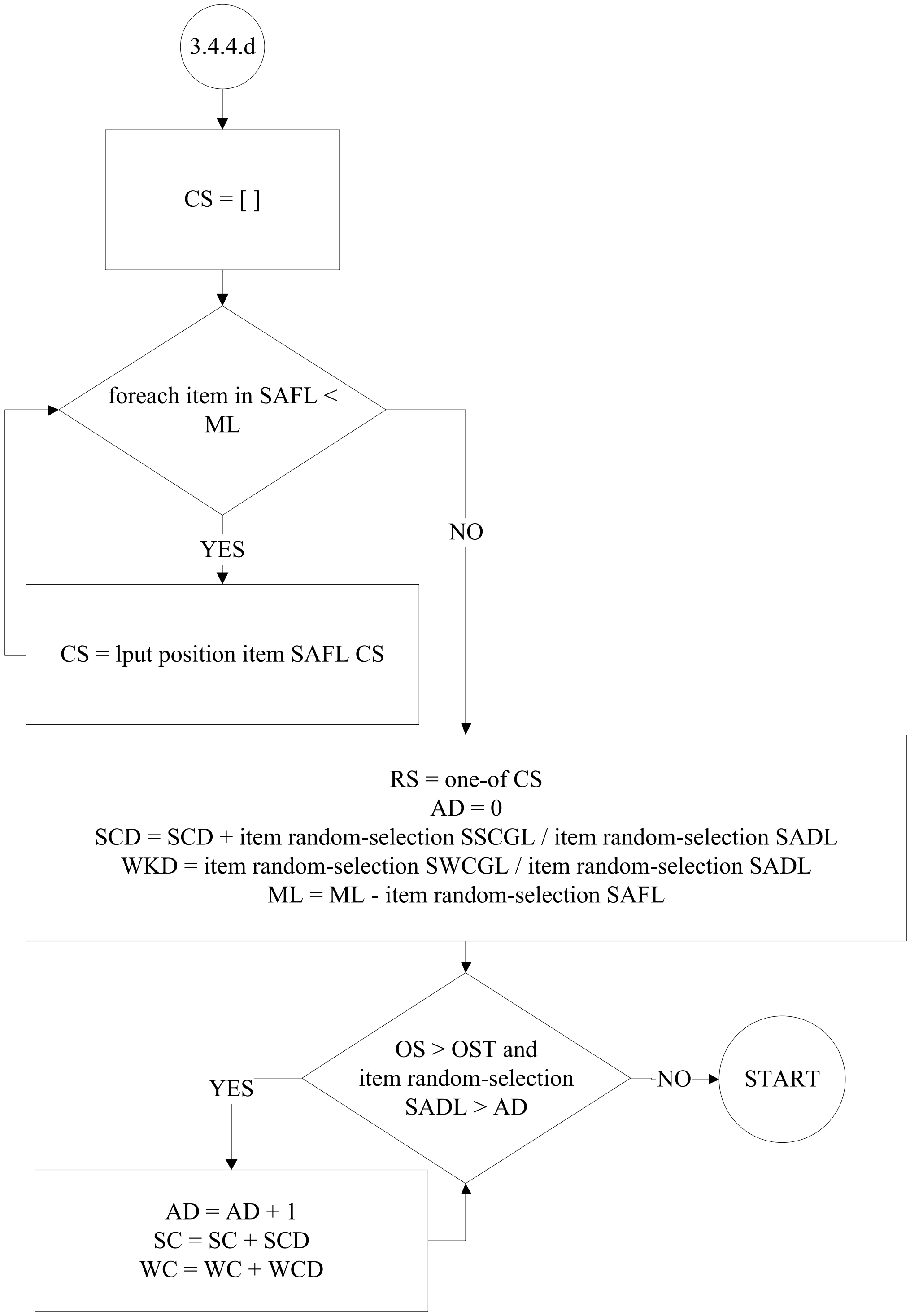

Figure 27: Social Procedure associated with the SC and WC Addition

Chapter 4

EXPERIMENTAL EVALUATIONS

The goal of experimental evaluations, through NetLogo simulations, for the proposed work, is to demonstrate the effectiveness of artificial emotions on goal-based agent behaviour when it acts within a multi-objective framework with limited resources.

The model variables describing the simulation environment are shown [Figure 28]. Experimental results presented below correspond to the average of ten runs and as a consequence to the presence of a unified module for perception and reasoning. Evaluation of each of them takes a short period of time even for a large number of ticks, for example, 1000 ticks take 11 seconds. It should be noted, the use of the modified triple tower model, one module for perceptron and reasoning modules, also is another factor of keeping the simulation time within reasonably small bounds. The user-friendly simulation interface makes it possible to enable, disable, increase or decrease specific parameters of the agents and the environment to observe their online and/or offline effects [Figure 28]. The parameter values, which used in the last experimental simulation and the presented results are obtained, are shown [Figure 28].

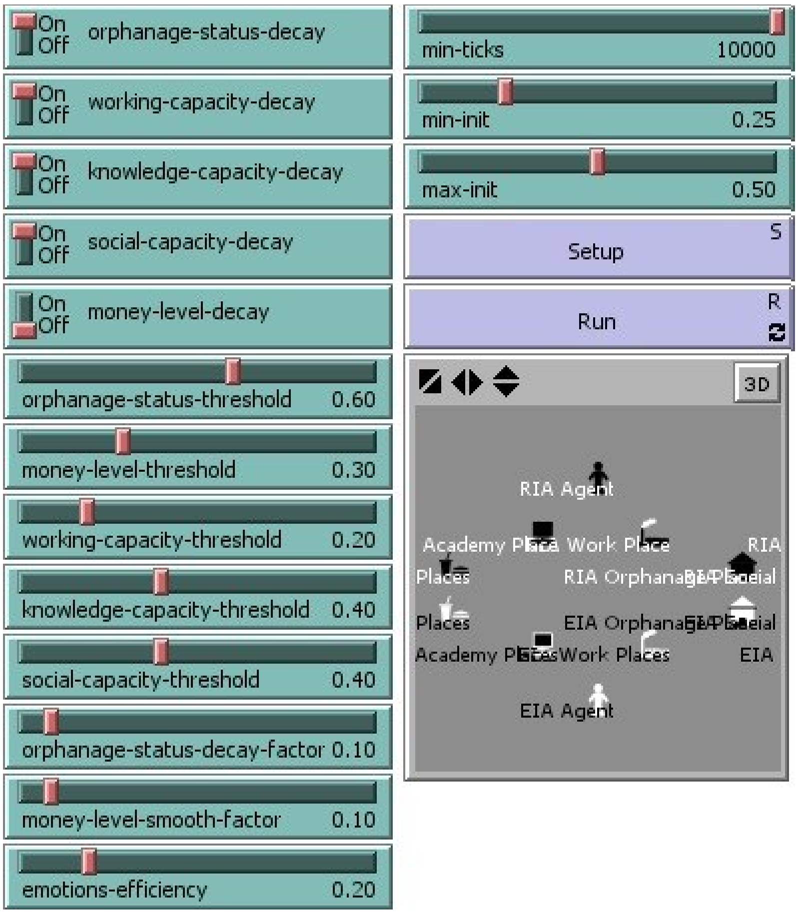

Figure 28: The Variables of the Model and the World

Successes of RIA and EIA are compared head to head over their associated with performance graphs, attributes in real time, for each objective [Figure 29] [Figure 30] under consideration, instead of comparing them after separate simulations. They try to improve their status, because of their decaying over time and threshold, by attending to a place associated with improving that status. They increase status of the orphanage by working in the orphanage place or paying money to, based on the comparison between the money level and its threshold. They increase the money level by attending to workplace with respect to status of the orphanage is more than its threshold. They increase knowledge and working capacity by attending to the academy with respect to status of the orphanage is more than its threshold and completion of the affordable course duration. They increase social and working capacity by attending to the social club with respect to status of the orphanage is more than its threshold and selecting an affordable activity. Therefore, status of the orphanage is significantly more important than any other status, thus if the status of the orphanage is becoming less than its threshold, the agent leaves the current objective and tries to improve.

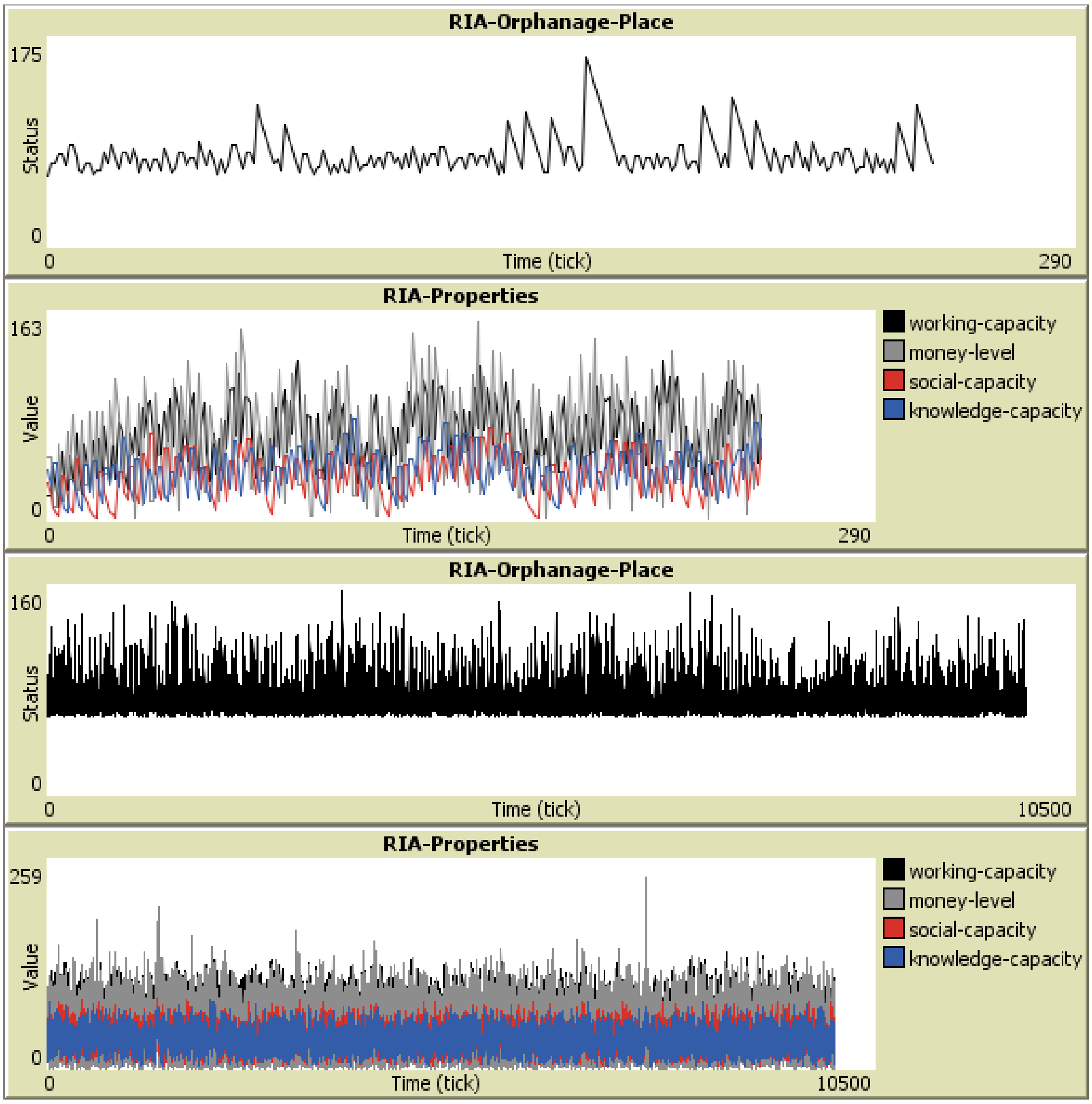

Figure 29: Line-chart Representation of the Statuses of the RIA in the Short and Long Run

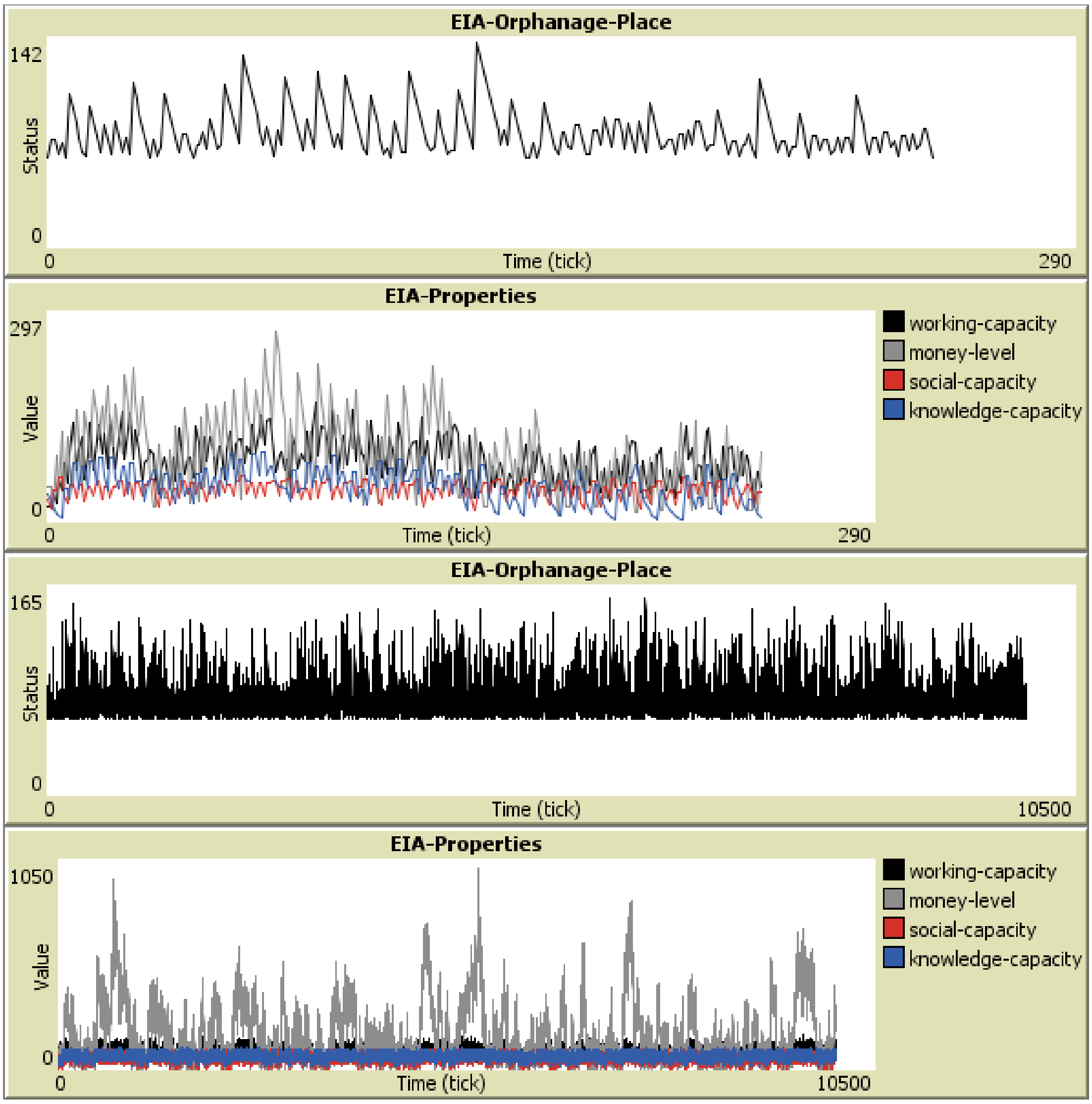

Figure 30: Line-chart Representation of the Statuses of the EIA in the Short and Long Run

The emotional behaviours of EIA towards the orphanage place, the workplace, the academy place, and the social club are represented as bar charts [Figure 31], which the emotional intensification of EIA is represented as +3 for liking (or approving), -3 for disliking (or disapproving), and 0 for neutral. The attraction emotions are used in reasoning phase to select the highest attractive activity between affordable offered courses in the academy place and affordable activities of the social club, which are then intensified by the associated with attribution emotion in meeting the objectives.

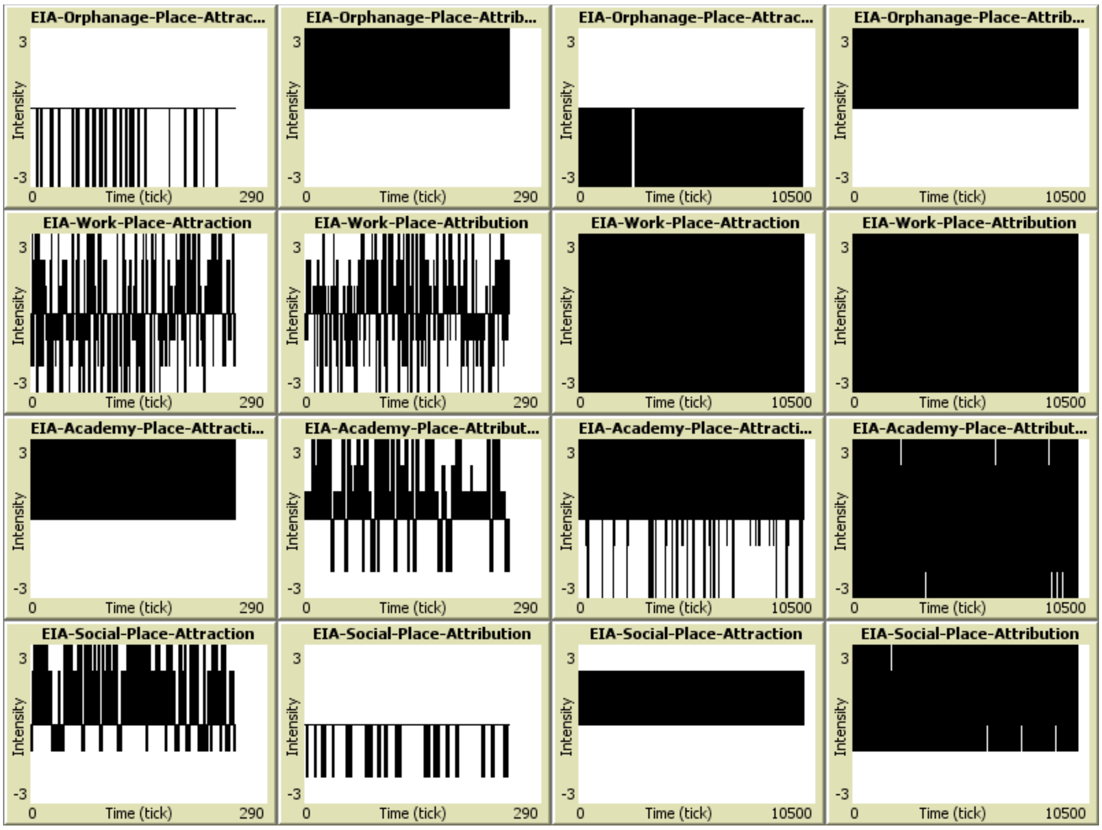

Figure 31: Bar-chart Representation for Emotional Behaviours of the EIA in the Short and Long Run

It should be noted, there are some possible selections for the attraction and attribution emotions, for example, positive attractions and negative attributions like drinking alcoholics or smoking, and negative attractions when the money level of the agent is less than the fee of other positive attractions.

Since the status of the agents has a lot of fluctuations [Figure 29] [Figure 30] during simulation, caused by simulation environment configuration; the average running of them is represented [Figure 32] [Figure 33] and the changes become clearer.

The experimental results presented in [Figure 32] to [Figure 38] clearly illustrate the effectiveness of using rational with emotional in contrast to rational reasoning. The overall performance of RIA for five objectives under considerations is shown [Figure 32]. It is clearly seen RIA keep all objectives around their thresholds and cannot improve them further. Nonetheless, as illustrated [Figure 33], EIA performs better than RIA in the sense of all objective values is not only kept above their thresholds; moreover, they are increasing steadily as the simulation proceeds. The separate tick-based achievement graphs for each individual objective are presented [Figure 34] to [Figure 38]. On each of these graphs, one can easily compare successes of RIA and EIA for a particular objective. Clearly, that EIA outperforms RIA in all objectives under considerations.

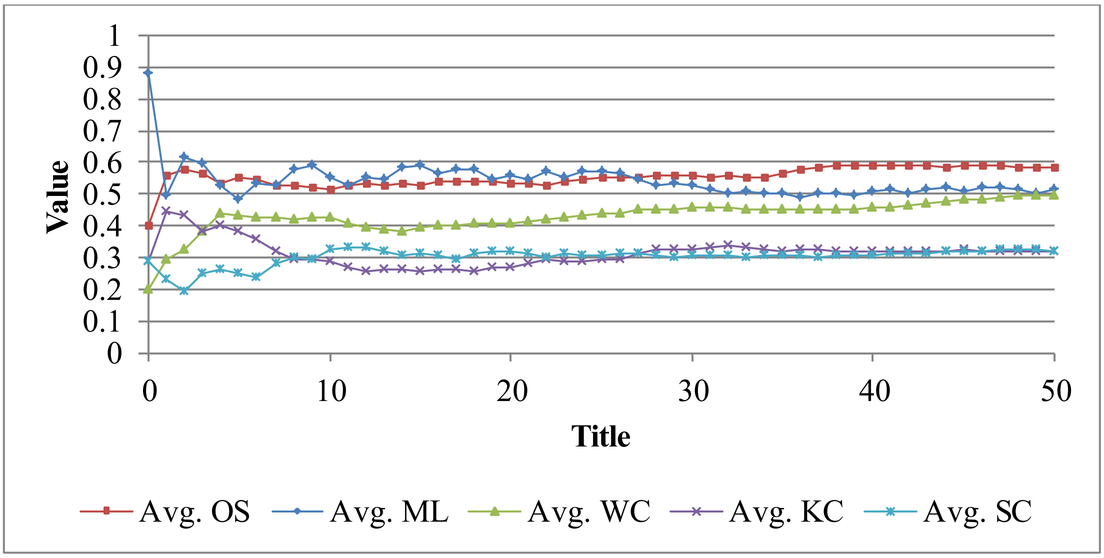

Figure 32: The Average Run of the Statuses of the RIA in the Short Run

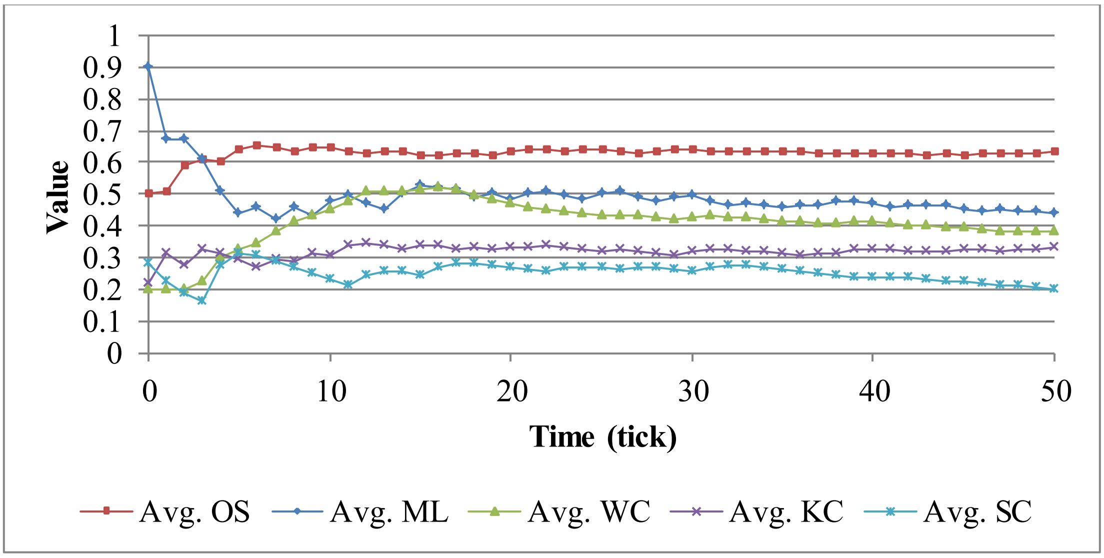

Figure 33: The Average Run of the Statuses of the EIA in the Short Run

There is a significant increase with a fluctuation in the OS of EIA from 0.50 to 0.68 between 0 and 500 ticks. Then, there is a noticeable decrease in the amount from 0.68 to 0.65 between 500 and 2000 ticks. The quantity of the OS is increasing gradually from 0.65 to 0.68 between 2000 and 10000 ticks. On the other hand, the amount of the OS being controlled by RIA is decreasing slightly from 0.60 to 0.58 all the time. [Figure 34]

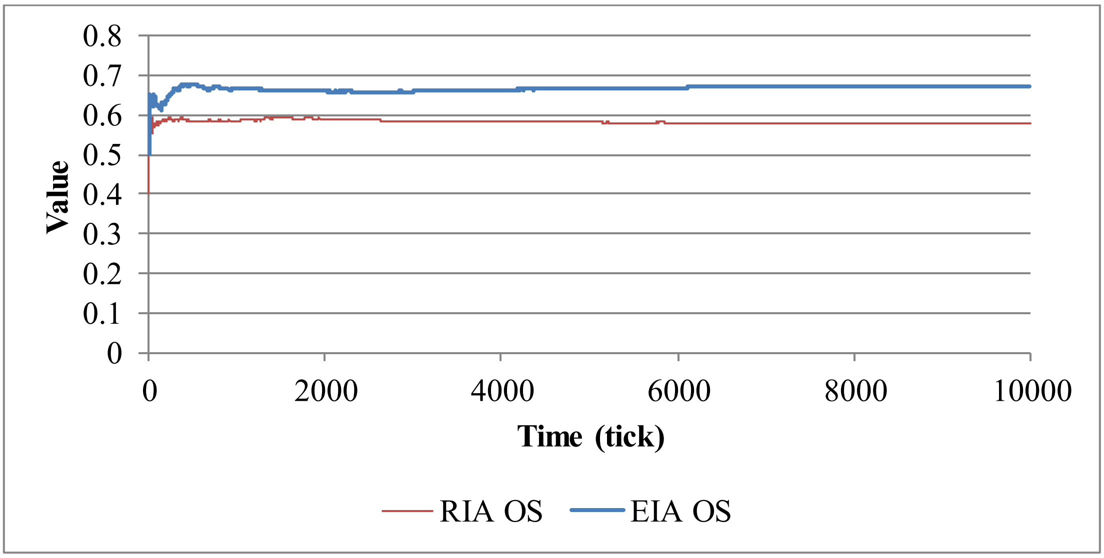

Figure 34: The Comparison of the OS of the RIA and EIA in the Long Run

Despite the fact that the ML which is made by EIA is unpredictable; it fluctuates around 1.75. In contrast, The ML of RIA is decreasing almost unnoticeable from 0.55 by 0.05 at all times. [Figure 35]

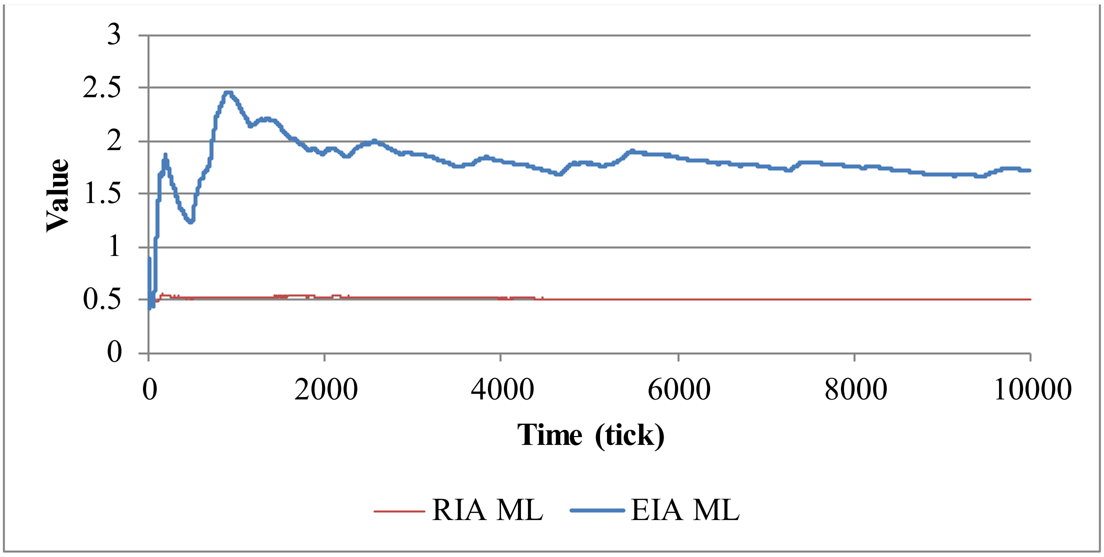

Figure 35: The Comparison of ML of the RIA and EIA in the Long Run

The WC of EIA is showing an impressive growth of 0.45 between 0 and 1000 ticks. Subsequently, it illustrates a slight decrease to 0.61 by the end of the simulation time. By contrast, the WC of RIA is climbing steeply to 0.55 before falling slightly to 0.50 in 500 ticks. [Figure 36]

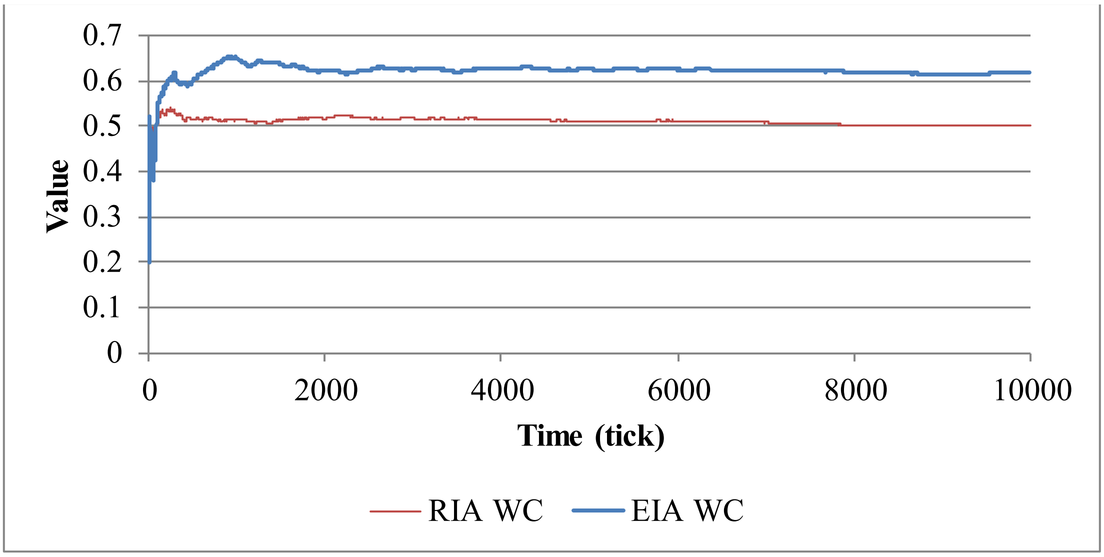

Figure 36: The Comparison of the WC of the RIA and EIA in the Long Run

There is a significant increase in the KC of EIA from 0.22 to 0.50 between 0 and 500 ticks. Afterwards, the value is declining slightly to 0.48 in 10000 ticks. Although the

KC of RIA is fluctuating at the first, it is decreasing slightly to 0.3 at the end. [Figure 37]

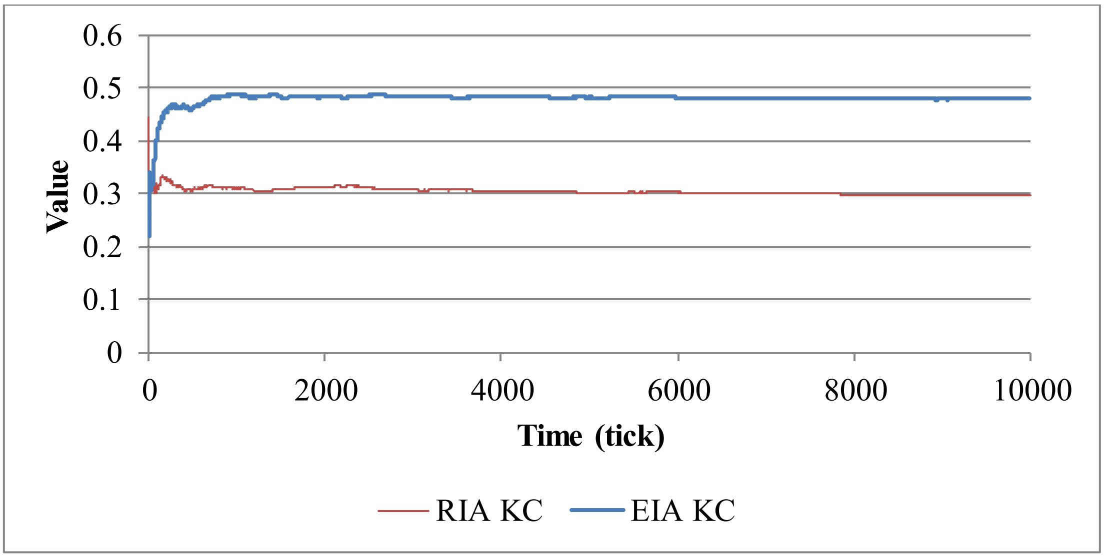

Figure 37: The Comparison of the KC of the RIA and EIA in the Long Run

The SC of EIA is climbing steeply to 0.65 before falling slightly to 0.62 in 1000 ticks. While, the SC of RIA is rising dramatically to 0.55 before decreasing slightly to 0.50 in 1000 ticks in the time of simulation. [Figure 38]

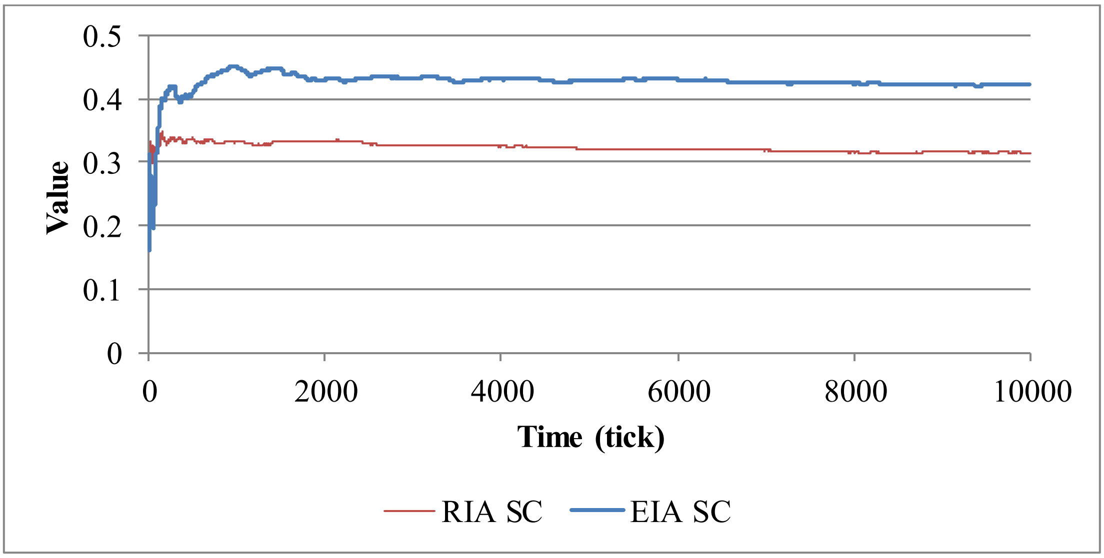

Figure 38: The Comparison of the SC of the RIA and EIA in the Long Run

Chapter 5

CONCLUSION

This study explains the modelling of artificial emotions based on agents, which emotional behaviours are modelled using emotional systems and a small subset of potential roles in artificial agents (selection, alarm mechanisms, goal management and strategic processing). It is implemented by computer based modelling NetLogo based on EBDI. The performance test environment is observable, stochastic (non-deterministic), sequential (non-episodic), dynamic, discrete and multi-agent. An orphanage care problem with more realistic multi-objective environment simulation is selected as a test environment and agents perform head to head in a real time simulation to compare their performance easily.

Moreover, this study presents a comparative analysis of emotional reasoning coupled with rationality, and rationality based decision making alone. It also uses commonly accepted representations for agents and provably efficient tools for simulations. The main distinction of the presented approach from the existing studies is its multi-objective realistic environment that is mostly compatible with real-life applications.

The simulations are conducted several times with the same experimental settings to get the comparable performances of RIA and EIA. EIA outperforms RIA for all the objectives under consideration as illustrated in the above section, which clearly shows the effectiveness of emotional behaviour when it is combined with rationality, compared to rationality applied alone.

The results of this study demonstrate which artificial emotions are applicable in agents. However, due to the very various mechanisms in implementing functions in agents, the benefits of emotions existing in humans are not easily translated to agents. It is tried to find more advanced modelling for the artificial emotions mechanisms and new scenarios and implement them in various applications. It is confided which artificial emotions can be very beneficial to the development of agents and hence to many applications in information technology.

REFERENCES

[1] Wilensky, U, (1999), "NetLogo", http://ccl.northwestern.edu/NetLogo/, Centre for Connected Learning and Computer-Based Modelling, Northwestern University, Evanston, IL.

[2] Encyclopedia Britannica, 2008, Encyclopedia Britannica Online, http://www.britannica.com/EBchecked/topic/185972/emotion.

[3] The Columbia Electronic Encyclopedia Copyright © 2004. Licensed from Columbia University Press.

[4] Emotional Circuits and Computational Neuroscience - Fellous J.M., Armony J.L., LeDoux J.E. - In 'The handbook of brain theory and neural networks' Second Edition. M.A. Arbib (editor), The MIT Press.

[5] Tao, Jianhua; Tieniu Tan (2005). "LNCS volume = 3784". Affective Computing and Intelligent Interaction. Springer. pp. 981–995. doi:10.1007/11573548

[6] James, William (1884). "What is Emotion". Mind 9: 188–205. Cited by Tao and Tan.

[7] "Affective Computing" MIT Technical Report #321 (Abstract), 1995

[8] Kleine-Cosack, Christian (October 2006). "Recognition and Simulation of Emotions" (PDF). Archived from the original on May 28, 2008. Retrieved May 13, 2008. "The introduction of emotion to computer science was done by Pickard (sic) who created the field of affective computing."

[9] Diamond, David (December 2003). "The Love Machine; Makeing computers which care.". Wired. Retrieved May 13, 2008. "Rosalind Picard, a genial MIT

professor, is the field's godmother; her 1997 book, Affective Computing, triggered an explosion of interest in the emotional side of computers and their users."

[10] Dellaert, F., Polizin, t. and Waibel, A., Recognizing Emotion in Speech", In Proc. Of ICSLP 1996, Philadelphia, PA, pp.1970-1973, 1996

[11] Lee, C.M.; Narayanan, S.; Pieraccini, R., Recognition of Negative Emotion in the Human Speech Signals, Workshop on Auto. Speech Recognition and Understanding, Dec 2001

[12] G. Caridakis, L. Malatesta, L. Kessous, N. Amir, A. Raouzaiou, K. Karpouzis, Modelling naturalistic affective states via facial and vocal expressions recognition, International Conference on Multimodal Interfaces (ICMI'06), Banff, Alberta, Canada, November 2-4, 2006

[13] T. Balomenos, A. Raouzaiou, S. Ioannou, A. Drosopoulos, K. Karpouzis, S. Kollias, Emotion Analysis in Man-Machine Interaction Systems, Samy Bengio, Herve Bourlard (Eds.), Machine Learning for Multimodal Interaction, Lecture Notes in Computer Science, Vol. 3361, 2004, pp. 318 - 328, Springer-Verlag

[14] J. K. Aggarwal, Q. Cai, Human Motion Analysis: A Review, Computer Vision and Image Understanding, Vol. 73, No. 3, 1999

[15] Vladimir I. Pavlovic, Rajeev Sharma, Thomas S. Huang, Visual Interpretation of Hand Gestures for Human-Computer Interaction; A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997

[16] Gerd Ruebenstrunk, "Emotional Computers: computer models of emotions and their meaning for emotion-psychological research", November 1998, http://www.ruebenstrunk.de/emeocomp/content.HTM.

[17] a b c Rao, M. P. Georgeff. (1995). "BDI-agents: From Theory to Practice". Pro-ceedings of the First International Conference on Multiagent Systems (ICMAS'95).

[18] de Silva, L. Sardina, S. and Padgham, L. First principles planning in BDI systems, in Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, 2009.

[19] Kaminka, G. A. and Frenkel, I. Flexible Teamwork in Behaviour -Based Robots. In Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05), 2005.

[20] Kaminka, G. A. Yakir, A. Erusalimchik, D. and Cohen-Nov, N. Towards Col-laborative Task and Team Maintenance. In Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-07), 2007.

[21] Phung, Toan; Michael Winikoff; Lin Padgham (2005). "Learning Within the BDI Framework: An Empirical Analysis". Knowledge-Based Intelligent Information and Engineering Systems.

[22] Guerra-Hernández, Alejandro; Amal El Fallah-Seghrouchni and Henry Soldano (2004). "Learning in BDI Multi-agent Systems". Computational Logic in Multi-Agent Systems.

[23] Rao, M. P. Georgeff. (1995). "Formal models and decision procedures for multi-agent systems". Technical Note, AAII.

[24] Georgeff, Michael; Barney Pell, Martha Pollack, Milind Tambe and Michael Wooldridge (1999). "The Belief-Desire-Intention Model of Agency". Intelligent Agents V: Agents Theories, Architectures and Languages.

[25] Pokahr, Alexander; Lars Braubach, Winfried Lamersdorf (2005). "Jadex: A BDI Reasoning Engine". Multi-Agent Programming.

[26] Sardina, Sebastian; Lavindra de Silva, Lin Padgham (2006). "Hierarchical planning in BDI agent programming languages: a formal approach". Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems.

[27] Aruna Chakraborty, Amit Konar, "Emotional Intelligence: A Cybernetic Approach", Studies in Computational Intelligence, Vol. 234, ISBN: 978-3-540-68606-4.

[28] Khulood Abu Maria, Raed Abu Zitar, Emotional agents: A modelling and an application, Information and Software Technology, Volume 49, Issue 7, July 2007, Pages 695-716, ISSN 0950-5849, DOI: 10.1016/j.infsof.2006.08.002.

[29] Hong Jiang, José M. Vidal, "From Rational to Emotional Agents", Computer Science and Engineering, University of South Carolina.

[30] Jiang, H., Vidal, J. M. and Huhns, M. N. 2007. EBDI: an architecture for emotional agents. In Proceedings of the 6th international Joint Conference on Autonomous Agents and Multiagent Systems (Honolulu, Hawaii, May 14 - 18, 2007). AAMAS '07. ACM, New York, NY, 1-3. DOI= http://doi.acm.org/10.1145/1329125.1329139.

[31] Scheutz, M. 2004. Useful roles of emotions in artificial agents: a case study from artificial life. In Proceedings of the 19th National Conference on Artifical intelligence (San Jose, California, July 25 - 29, 2004). A. G. Cohn, Ed. Aaai Conference On Artificial Intelligence. AAAI Press / The MIT Press, 42-47.

[32] Penny Baillie, Mark Toleman, Dickson Lukose, Could Emotions be the Key to Real Artificial Intelligence, in Proceedings of ISA'2000, Wollongong, Australia, ICSC Academic Press, 45-50

[33] Penny Baillie, Mark Toleman, Dickson Lukose, Emotional Intelligence for Intuitive Agents, in Proceedings of AISAT 2000, Hobart, Australia, 2000, University of Tasmania, Hobart, 134--139.

[34] Baillie (Ray), P. and D. Lukose. To Believe, to Desire, to Intent: Motivating Au-tonomous Intelligent Agents with a Mind of their Own. in Managing Enterprises 99. 1999. Newcastle.

[35] A. Ortony, G. L. Clore, A. Collins, editors. The Cognitive Structure of Emotions. Cambridge University Press, Cambridge, UK, 1988.

[36] Russell, Stuart J.; Norvig, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2, http://aima.cs.berkeley.edu/, chpt. 2.

[37] Macal, C. M. and North, M. J. 2008. Agent-based modelling and simulation: ABMS examples. In Proceedings of the 40th Conference on Winter Simulation (Miami, Florida, December 07 - 10, 2008). S. Mason, R. Hill, L. Mönch and O. Rose, Eds. Winter Simulation Conference. Winter Simulation Conference, 101-112.

[38] Axelrod, Robert (1997), The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration, Princeton: Princeton University Press, ISBN 978-0-691-01567-5.

[39] Agent-Based Models of Industrial Ecosystems. Rutgers University, October 6, 2003.

[40] Agent-based modeling: Methods and techniques for simulating human systems. Proceedings of the National Academy of Sciences. May 14, 2002.

[41] Balci, O. 1997. Verification validation and accreditation of simulation models. In Proceedings of the 29th Conference on Winter Simulation (Atlanta, Georgia, United States, December 07 - 10, 1997). S. Andradóttir, K. J. Healy, D. H. Withers and B. L. Nelson, Eds. Winter Simulation Conference. IEEE Computer Society, Washington, DC, 135-141. DOI= http://doi.acm.org/10.1145/268437.268462.

[42] Galán, José Manuel, Izquierdo, Luis R., Izquierdo, Segismundo S., Santos, José Ignacio, del Olmo, Ricardo, López-Paredes, Adolfo and Edmonds, Bruce (2009). 'Errors and Artefacts in Agent-Based Modelling'. Journal of Artificial Societies and Social Simulation 12(1)1 <http://jasss.soc.surrey.ac.uk/12/1/1.html>.

[43] Nils J. Nilsson: Teleo-Reactive Programs and the Triple-Tower Architecture. Electronic Transactions on Artificial Intelligence, Vol. 5 (2001), Section B, pp. 99-110. http://www.ep.liu.se/ej/etai/2001/006/.

[44] Bratman, M. E. (1999) [1987]. Intention, Plans, and Practical Reason. CSLI Publications. ISBN 1-57586-192-5.

APPENDICES

Appendix A: Source Code

```
;==============================================================;
;                                        ;
;              Common  Sections                ;
;                                        ;
;==============================================================;

globals [
 EIA-Academy-Place-random-selection
 EIA-Social-Place-random-selection
]

;=====================================;
;     Application  Setup and Run     ;
;            Section              ;
;=====================================;

to Setup-all
 clear-all
 RIA-Setup
 EIA-Setup
end

;=====================================;
;     Application  Setup and Run     ;
;            Section              ;
;=====================================;

to Run-all
 if ticks > min-ticks  [
  let dateandtime  ( word remove ":"  substring  date-and-time  0 5 remove  "-" remove
" " substring  date-and-time  13 27)
  export-interface  (word dateandtime  " Interface.png")
  export-output  (word dateandtime  " Output.txt")
  export-plot "RIA-Orphanage-Place"  (word  dateandtime  " RIA-OS.csv")
  export-plot "RIA-Properties"  (word dateandtime  " RIA-Properties.csv")
  export-plot "EIA-Orphanage-Place"  (word dateandtime  " EIA-OS.csv")
  export-plot "EIA-Properties"  (word dateandtime  " EIA-Properties.csv")
  export-world  (word dateandtime  " World.csv")
  stop
 ]
 RIA-Run
 EIA-Run
 tick
 print "------------------------------------------------------------------------"
end

;=====================================;
```

```
;        Utility  procedures          ;
;            Section              ;
;=======================================;

to-report Utility-Random-Float
  report min-init  + random-float  (max-init  - min-init)
end

to-report Utility-Random
  report -1 ^ random (2) * Random 4
end

;============================================================;
;                                        ;
;            Regular  Intelligent  Agent (RIA)          ;
;                    Sections                    ;
;                                        ;
;============================================================;


;=======================================;
;   Regular  Intelligent  Agent  (RIA)   ;
;        Definition  Section         ;
;=======================================;

breed [RIAs RIA]
RIAs-own [
  ;Goals
  previous-goal
  current-goal
  selected-goal

  ;Agent  Performance  Measures
  money-level
  working-capacity
  knowledge-capacity
  social-capacity
  ]

breed [RIA-Orphanage-Places  RIA-Orphanage-Place]
RIA-Orphanage-Places-own  [
  status ; Healthy  , Not healthy  (random setting)
  ]

breed [RIA-Work-Places RIA-Work-Place]
RIA-Work-Places-own [
  offering-job  ; (random setting)
  offering-salary  ; (random setting)
  job-duration  ; (random setting)
  ]
```

```
breed [RIA-Academy-Places  RIA-Academy-Place]
RIA-Academy-Places-own  [
  course-list  ; (random  setting)
  course-duration-list  ; (random  setting)
  course-fees-list  ; (random  setting)
  knowledge-capacity-gained-list  ; (random  setting)
  working-capacity-gained-list  ; (random  setting)
  ]

breed [RIA-Social-Places  RIA-Social-Place]  ; Related  to Emotional  relation
RIA-Social-Places-own  [
  activity-list  ; (random  setting)
  activity-duration-list  ; (random  setting)
  activity-fees-list  ; (random  setting)
  social-capacity-gained-list  ; (random  setting)
  working-capacity-gained-list  ; (random  setting)
  ]

;==========================================;
;              RIA  Setup              ;
;              Section                ;
;==========================================;

to RIA-Setup
  ask patches [
    set pcolor  gray
    ]
  create-RIAs  1 [
    set color  black
    set shape  "person"
    set size  3
    setxy  0 10
    set label  "RIA  Agent"
    set label-color  white
    set money-level  Utility-Random-Float  ;money-level-threshold
    set working-capacity  Utility-Random-Float  ;working-capacity-threshold
    set knowledge-capacity  Utility-Random-Float  ;knowledge-capacity-threshold
    set social-capacity  Utility-Random-Float  ;social-capacity-threshold
    ]
  create-RIA-Orphanage-Places  1 [
    set color  black
    set shape  "house"
    set size  3
    setxy  -20 2
    set label  "RIA  Orphanage  Place"
    set label-color  white
    set status  Utility-Random-Float  ;orphanage-status-threshold
    ]
  create-RIA-Work-Places  1 [
```

```
    set color black
    set shape "factory"
    set size 3
    setxy 5 5
    set label "RIA Work Place"
    set label-color white
    set offering-job "Job"
    set offering-salary Utility-Random-Float
    set job-duration 1
    ]
  create-RIA-Academy-Places 1 [
    set color black
    set shape "computer workstation"
    set size 3
    setxy -5 5
    set label "RIA Academy Place"
    set label-color white
    set course-list ( list "CMPE101" "CMPE201" "CMPE301" "CMPE401"
"CMPE501" "CMPE601" "CMPE701" "CMPE102" "CMPE202" "CMPE302"
"CMPE402" "CMPE502" "CMPE602" "CMPE702" "CMPE103" "CMPE203"
"CMPE303" "CMPE403" "CMPE503" "CMPE603" "CMPE703" )
    set course-duration-list ( list 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ) ; (random
setting)
    set course-fees-list ( list Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float ) ; (random setting)
    set knowledge-capacity-gained-list ( list Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float ) ; (random setting)
    set working-capacity-gained-list ( list Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float ) ; (random setting)
    ]
  create-RIA-Social-Places 1 [
    set color black
    set shape "food"
    set size 3
    setxy 20 2
```

```
    set label "RIA Social Places"
    set label-color white
    set activity-list ( list "A01" "A02" "A03" "A04" "A05" "A06" "A07" "A08" "A09"
"A03" "A10" "A11" "A12" "A13" "A14" "A15" "A16" "A17" "A18" "A19" "A20" )
    set activity-duration-list ( list 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ) ; (random
setting)
    set activity-fees-list ( list Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float ) ; (random setting)
    set social-capacity-gained-list ( list Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float ) ; (random setting)
    set working-capacity-gained-list ( list Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float ) ; (random setting)
    ]
end

;=======================================;
;           RIA Run              ;
;           Section              ;
;=======================================;

to RIA-Run
  ask RIA-Work-Place 2 [
    set offering-salary Utility-Random-Float
    set job-duration 1
  ]
  RIA-Reasoning-Rules
  RIA-Executing-Rules
  RIA-Show-Results
end

;=======================================;
;       RIA Thinking Process       ;
;      Reasoning Rules Section      ;
;=======================================;

```

```
to RIA-Reasoning-Rules
 ask RIA 0 [
  ifelse [ status ] of RIA-Orphanage-Place 1 < orphanage-status-threshold
  [
   ifelse money-level - money-level-threshold < money-level-smooth-factor
   [ set selected-goal "Orphanage" ]
   [
    let money-x money-level - money-level-threshold - money-level-smooth-factor
    let orphanage-payment orphanage-status-threshold - [ status ] of RIA-
Orphanage-Place 1 + random-float money-x / (10 ^ ceiling log money-x 10)
    set money-level money-level - orphanage-payment
    ask RIA-Orphanage-Place 1 [
     set status status + orphanage-payment
    ]
    type ticks type " RIA OP: " print orphanage-payment
    RIA-Reasoning-Rules
    ]
   ]
  [
   ifelse money-level < money-level-threshold
   [
    set selected-goal "Work"
;     ifelse working-capacity > working-capacity-threshold
;     [ set selected-goal "Work" ]
;     [
;      ifelse knowledge-capacity - knowledge-capacity-threshold < social-capacity -
social-capacity-threshold
;      [
;       ifelse money-level > min [ course-fees-list ] of RIA-Academy-Place 3; and ([
status ] of RIA-Orphanage-Place 1 - orphanage-status-threshold) / orphanage-status-
decay-factor > min [ course-duration-list ] of RIA-Academy-Place 3
;       [ set selected-goal "Academy" ]
;       [
;        ifelse money-level > min [ activity-fees-list ] of RIA-Social-Place 4
;        [ set selected-goal "Social" ]
;        [ set selected-goal "Work" ]
;        ]
;       ]
;      [
;       ifelse money-level > min [ activity-fees-list ] of RIA-Social-Place 4
;       [ set selected-goal "Social" ]
;       [
;        ifelse money-level > min [ course-fees-list ] of RIA-Academy-Place 3; and
([ status ] of RIA-Orphanage-Place 1 - orphanage-status-threshold) / orphanage-
status-decay-factor > min [ course-duration-list ] of RIA-Academy-Place 3
;        [ set selected-goal "Academy" ]
;        [ set selected-goal "Work" ]
;        ]
;       ]
;      ]
```

```
      ]
     [
      ifelse  knowledge-capacity  - knowledge-capacity-threshold  < social-capacity  -
social-capacity-threshold
      [
        ifelse  knowledge-capacity  < knowledge-capacity-threshold  and money-level  >
min  [ course-fees-list  ] of RIA-Academy-Place  3; and ([ status  ] of RIA-Orphanage-
Place  1 - orphanage-status-threshold)  / orphanage-status-decay-factor  > min  [ course-
duration-list  ] of RIA-Academy-Place  3
        [ set selected-goal  "Academy"  ]
        [
          ifelse  social-capacity  < social-capacity-threshold  and money-level  > min  [
activity-fees-list  ] of RIA-Social-Place  4
          [ set selected-goal  "Social"  ]
          [ set selected-goal  "Work"  ]
          ]
        ]
      [
        ifelse  social-capacity  < social-capacity-threshold  and money-level  > min  [
activity-fees-list  ] of RIA-Social-Place  4
        [ set selected-goal  "Social"  ]
        [
          ifelse  knowledge-capacity  < knowledge-capacity-threshold  and money-level
> min  [ course-fees-list  ] of RIA-Academy-Place  3; and ([ status  ] of RIA-
Orphanage-Place  1 - orphanage-status-threshold)  / orphanage-status-decay-factor  >
min  [ course-duration-list  ] of RIA-Academy-Place  3
          [ set selected-goal  "Academy"  ]
          [ set selected-goal  "Work"  ]
          ]
        ]
      ]
     ]
    type ticks type " RIA-Reasoning-Rules  [      ] MPG: " type previous-goal  type "
MCG: " type current-goal  type " MSG: ->" print  selected-goal
    ]
end

;=======================================;
;        RIA Thinking  Process        ;
;       Execution  Rules  Section       ;
;=======================================;

to RIA-Executing-Rules
  ask RIA  0 [
    set current-goal  selected-goal
    type ticks type " RIA-Executing-Rules  [BEFORE]  MPG: " type previous-goal
type  " MCG: ->" type current-goal  type " MSG: " print  selected-goal
    ]
  if [ selected-goal  ] of RIA  0 = "Orphanage"  [ RIA-Visit-Orphanage-Places  ]
  if [ selected-goal  ] of RIA  0 = "Work"  [ RIA-Visit-Work-Places  ]
```

```
  if [ selected-goal ] of RIA 0 = "Academy" [ RIA-Visit-Academy-Places ]
  if [ selected-goal ] of RIA 0 = "Social" [ RIA-Visit-Social-Places ]
  ask RIA 0 [
    set previous-goal current-goal
    type ticks type " RIA-Executing-Rules [AFTER ] MPG: ->" type previous-goal
type " MCG: " type current-goal type " MSG: " print selected-goal
    ]
end

;======================================;
;       RIA Thinking Process       ;
;    Go to Orphanage Place Section    ;
;======================================;

to RIA-Visit-Orphanage-Places
  type ticks print " RIA-Visit-Orphanage-Places"
  while [ [ status ] of RIA-Orphanage-Place 1 <= orphanage-status-threshold ] [
    type ticks print " RIA-Visit-Orphanage-Places LOOP"
    ask RIA-Orphanage-Place 1 [
      set status status + orphanage-status-decay-factor + Utility-Random-Float * status
      ]
    ]
end

;======================================;
;       RIA Thinking Process       ;
;      Go to Work Place Section      ;
;======================================;

to RIA-Visit-Work-Places
  type ticks print " RIA-Visit-Work-Places"
  let work-duration 0
  let offering-salary-duration [ offering-salary ] of RIA-Work-Place 2 / [ job-duration
] of RIA-Work-Place 2
  while [ [ status ] of RIA-Orphanage-Place 1 >= orphanage-status-threshold and [
job-duration ] of RIA-Work-Place 2 > work-duration ] [
    type ticks print " RIA-Visit-Work-Places LOOP"
    set work-duration work-duration + 1
    ask RIA 0 [
      set money-level money-level + offering-salary-duration + Utility-Random-Float
* working-capacity + Utility-Random-Float * knowledge-capacity + Utility-
Random-Float * social-capacity
      ]
  ]
end

;======================================;
;       RIA Thinking Process       ;
;     Go to Academy Place Section     ;
;======================================;
```

```
to RIA-Visit-Academy-Places
  type ticks print " RIA-Visit-Academy-Places"
  let choices [ ]
  foreach filter [ ? < [ money-level ] of RIA 0 ] [ course-fees-list ] of RIA-Academy-
Place 3 [set choices lput position ? [ course-fees-list ] of RIA-Academy-Place 3
choices]
  let random-selection one-of choices
  let course-duration 0
  ask RIA 0 [
    set money-level money-level - item random-selection [ course-fees-list ] of RIA-
Academy-Place 3
    ]
  while [ [ status ] of RIA-Orphanage-Place 1 >= orphanage-status-threshold and item
random-selection [ course-duration-list ] of RIA-Academy-Place 3 > course-duration
] [
    type ticks print " RIA-Visit-Academy-Places LOOP"
    set course-duration course-duration + 1
    ]
    if course-duration = item random-selection [ course-duration-list ] of RIA-
Academy-Place 3 [
    ask RIA 0 [
      set knowledge-capacity knowledge-capacity + item random-selection [
knowledge-capacity-gained-list ] of RIA-Academy-Place 3
      set working-capacity working-capacity + item random-selection [ working-
capacity-gained-list ] of RIA-Academy-Place 3
      ]
    ]
end

;======================================;
;       RIA Thinking Process        ;
;    Go to Social Place Section      ;
;======================================;

to RIA-Visit-Social-Places
  type ticks print " RIA-Visit-Social-Places"
  let choices [ ]
  foreach filter [ ? < [ money-level ] of RIA 0 ] [ activity-fees-list ] of RIA-Social-
Place 4 [set choices lput position ? [ activity-fees-list ] of RIA-Social-Place 4
choices]
  let random-selection one-of choices
  let activity-duration 0
  let social-capacity-duration item random-selection [ social-capacity-gained-list ] of
RIA-Social-Place 4 / item random-selection [ activity-duration-list ] of RIA-Social-
Place 4
  let working-capacity-duration item random-selection [ working-capacity-gained-list
] of RIA-Social-Place 4 / item random-selection [ activity-duration-list ] of RIA-
Social-Place 4
  ask RIA 0 [
```

```
    set money-level  money-level  - item  random-selection  [ activity-fees-list  ] of RIA-
Social-Place  4
    ]
  while  [ [ status  ] of RIA-Orphanage-Place  1 >= orphanage-status-threshold  and item
random-selection  [ activity-duration-list  ] of RIA-Social-Place  4 > activity-duration  ]
[
    type ticks print " RIA-Visit-Social-Places  LOOP"
    set activity-duration  activity-duration  + 1
    ask RIA  0 [
      set social-capacity  social-capacity  + social-capacity-duration
      set working-capacity  working-capacity  + working-capacity-duration
      ]
    ]
end

;========================================;
;          RIA Show Result          ;
;             Sections              ;
;========================================;

to RIA-Show-Results
  ;Plot the result
  RIA-Plot
  ;Track information  in command  center
  RIA-Track
  ;Since measurements  variables  decay over the time
  RIA-Current-Capacity-Decay
  ;Track information  in command  center
  RIA-Track
end

to RIA-Plot
  ask RIA  0 [
    set-current-plot  "RIA-Properties"
    set-current-plot-pen  "working-capacity"
    plot working-capacity  * 100
    set-current-plot-pen  "money-level"
    plot money-level  * 100
    set-current-plot-pen  "social-capacity"
    plot social-capacity  * 100
    set-current-plot-pen  "knowledge-capacity"
    plot knowledge-capacity  * 100
    ]
  ask RIA-Orphanage-Place  1 [
    set-current-plot  "RIA-Orphanage-Place"
    set-current-plot-pen  "status"
    plot status  * 100
    ]
end
```

```
to RIA-Current-Capacity-Decay
  ask RIA-Orphanage-Place 1 [
    ;Orphanage status decay over the time
    if orphanage-status-decay and [ current-goal ] of RIA 0 != "Orphanage" [ set status
status - orphanage-status-decay-factor ]
    ]
  ask RIA 0 [
    ;Working capacity decay over the time
    if working-capacity-decay and current-goal != "Work" [ set working-capacity
working-capacity - working-capacity * Utility-Random-Float ]
    ;Knowledge capacity decay over the time
    if knowledge-capacity-decay and current-goal != "Academy" [ set knowledge-
capacity knowledge-capacity - knowledge-capacity * Utility-Random-Float ]
    ;Social capacity decay over the time
    if social-capacity-decay and current-goal != "Social" [ set social-capacity social-
capacity - social-capacity * Utility-Random-Float ]
    ;money level decay over the time
    if money-level-decay and current-goal != "Work" [ set money-level money-level -
money-level * Utility-Random-Float ]
    ]
end

to RIA-Track
  ;Tracing values
  ask RIA 0 [
    type ticks
    type " RIA OS: " type precision [ status ] of RIA-Orphanage-Place 1 2
    type " ML: " type precision money-level 2
    type " WC: " type precision working-capacity 2
    type " KC: " type precision knowledge-capacity 2
    type " SC: " print precision social-capacity 2
    ]
end


;==============================================================;
;                                        ;
;          Emotional Intelligent Agent (EIA)          ;
;                    Sections                    ;
;                                        ;
;==============================================================;


;======================================;
;  Emotional Intelligent Agent (EIA)  ;
;        Definition Section        ;
;======================================;

breed [EIAs EIA]
EIAs-own [
```

```
 ;Goals
 previous-goal
 current-goal
 selected-goal

 ;Agent Performance Measures
 money-level
 working-capacity
 knowledge-capacity
 social-capacity

 ;Emotion Aspects
 ;emotional-polarity ; (positive or negative for the reaction that originated the
emotion) Causing emotional reactions
 event-based-emotions ; pleased/displeased reactions to events
 ]

breed [EIA-Orphanage-Places EIA-Orphanage-Place]
EIA-Orphanage-Places-own [
 status ; Healthy , Not healthy (random setting)
 ]

breed [EIA-Work-Places EIA-Work-Place]
EIA-Work-Places-own [
 offering-job ; (random setting)
 offering-salary ; (random setting)
 job-duration ; (random setting)
 attraction-emotions ; Job liking, disliking (random setting)
 attribution-emotions ; Agents approving, disapproving (random setting)
 ]

breed [EIA-Academy-Places EIA-Academy-Place]
EIA-Academy-Places-own [
 course-list ; (random setting)
 course-duration-list ; (random setting)
 course-fees-list ; (random setting)
 knowledge-capacity-gained-list ; (random setting)
 working-capacity-gained-list ; (random setting)
 attraction-emotions ; Course liking, disliking (random setting)
 attribution-emotions ; Agents approving, disapproving (random setting)
 ]

breed [EIA-Social-Places EIA-Social-Place] ; Related to Emotional relation
EIA-Social-Places-own [
 activity-list ; (random setting)
 activity-duration-list ; (random setting)
 activity-fees-list ; (random setting)
 social-capacity-gained-list ; (random setting)
 working-capacity-gained-list ; (random setting)
 attraction-emotions ; Activity liking, disliking (random setting)
```

```
  attribution-Emotions ; Agents approving, disapproving (random setting)
  ]

;=========================================;
;            EIA Setup           ;
;            Section             ;
;=========================================;

to EIA-Setup
  create-EIAs 1 [
    set color white
    set shape "person"
    set size 3
    setxy 0 -10
    set label "EIA Agent"
    set label-color black
    set money-level Utility-Random-Float ;money-level-threshold
    set working-capacity Utility-Random-Float ;working-capacity-threshold
    set knowledge-capacity Utility-Random-Float ;knowledge-capacity-threshold
    set social-capacity Utility-Random-Float ;social-capacity-threshold
    set event-based-emotions 0;
    ]
  create-EIA-Orphanage-Places 1 [
    set color white
    set shape "house"
    set size 3
    setxy -20 -2
    set label "EIA Orphanage Place"
    set label-color black
    set status Utility-Random-Float ;orphanage-status-threshold
    ]
  create-EIA-Work-Places 1 [
    set color white
    set shape "factory"
    set size 3
    setxy 5 -5
    set label "EIA Work Places"
    set label-color black
    set offering-job "Job"
    set offering-salary Utility-Random-Float
    set job-duration 1
    set attraction-emotions Utility-Random
    set attribution-emotions Utility-Random
    ]
  create-EIA-Academy-Places 1 [
    set color white
    set shape "computer workstation"
    set size 3
    setxy -5 -5
    set label "EIA Academy Places"
```

```
    set label-color black
    set course-list ( list "CMPE101" "CMPE201" "CMPE301" "CMPE401"
"CMPE501" "CMPE601" "CMPE701" "CMPE102" "CMPE202" "CMPE302"
"CMPE402" "CMPE502" "CMPE602" "CMPE702" "CMPE103" "CMPE203"
"CMPE303" "CMPE403" "CMPE503" "CMPE603" "CMPE703" )
    set course-duration-list ( list 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ) ; (random
setting)
    set course-fees-list ( list Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float ) ; (random setting)
    set knowledge-capacity-gained-list ( list Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float ) ; (random setting)
    set working-capacity-gained-list ( list Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float ) ; (random setting)
    set attraction-emotions ( list Utility-Random Utility-Random Utility-Random
Utility-Random Utility-Random Utility-Random Utility-Random Utility-Random
Utility-Random Utility-Random Utility-Random Utility-Random Utility-Random
Utility-Random Utility-Random Utility-Random Utility-Random Utility-Random
Utility-Random Utility-Random Utility-Random ) ; (random setting)
    set attribution-emotions ( list Utility-Random Utility-Random Utility-Random
Utility-Random Utility-Random Utility-Random Utility-Random Utility-Random
Utility-Random Utility-Random Utility-Random Utility-Random Utility-Random
Utility-Random Utility-Random Utility-Random Utility-Random Utility-Random
Utility-Random Utility-Random Utility-Random ) ; (random setting)
    ]
  create-EIA-Social-Places 1 [
    set color white
    set shape "food"
    set size 3
    setxy 20 -2
    set label "EIA Social Places"
    set label-color black
    set activity-list ( list"A01" "A02" "A03" "A04" "A05" "A06" "A07" "A08" "A09"
"A03" "A10" "A11" "A12" "A13" "A14" "A15" "A16" "A17" "A18" "A19" "A20" )
    set activity-duration-list ( list 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ) ; (random
setting)
```

```
   set activity-fees-list ( list Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float ) ; (random setting)
   set social-capacity-gained-list ( list Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float ) ; (random setting)
   set working-capacity-gained-list ( list Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-Float
Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-Random-
Float Utility-Random-Float Utility-Random-Float Utility-Random-Float Utility-
Random-Float ) ; (random setting)
   set attraction-emotions ( list Utility-Random Utility-Random Utility-Random
Utility-Random Utility-Random Utility-Random Utility-Random Utility-Random
Utility-Random Utility-Random Utility-Random Utility-Random Utility-Random
Utility-Random Utility-Random Utility-Random Utility-Random Utility-Random
Utility-Random Utility-Random Utility-Random ) ; (random setting)
   set attribution-emotions ( list Utility-Random Utility-Random Utility-Random
Utility-Random Utility-Random Utility-Random Utility-Random Utility-Random
Utility-Random Utility-Random Utility-Random Utility-Random Utility-Random
Utility-Random Utility-Random Utility-Random Utility-Random Utility-Random
Utility-Random Utility-Random Utility-Random ) ; (random setting)
   ]
end

;======================================;
;          EIA Run          ;
;          Section          ;
;======================================;

to EIA-Run
  ask EIA-Work-Place 7 [
   set offering-salary Utility-Random-Float
   set job-duration 1
   set attraction-emotions Utility-Random
   set attribution-emotions Utility-Random
  ]
  EIA-Reasoning-Rules
  ask EIA 5 [
   ifelse [ status ] of EIA-Orphanage-Place 6 < 0
   [ set event-based-emotions 3 ]
```

```
   [
     ifelse  1 < [ status  ] of EIA-Orphanage-Place  6
     [ set event-based-emotions  -3 ]
     [ set event-based-emotions  round (1 - [ status  ] of EIA-Orphanage-Place  6 /
orphanage-status-threshold)  * 3 ]
     ]
  ]
  EIA-Executing-Rules
  EIA-Show-Results
end

;=======================================;
;        EIA Thinking  Process        ;
;       Reasoning  Rules  Section       ;
;=======================================;

to EIA-Reasoning-Rules
  let is-social-attraction  false
  let is-social-expenses-need  false
  let is-social-attribution  false
  ;=======================================
  ;select  the most  attractive  activity
  let decision-attraction  max [ attraction-emotions  ] of EIA-Academy-Place  8
  if decision-attraction  <= max [ attraction-emotions  ] of EIA-Social-Place  9 [
    set decision-attraction  max [ attraction-emotions  ] of EIA-Social-Place  9
    set is-social-attraction  true
    ;=======================================
    ;select  the lowest  expenses  need activity  amoung  up
    let decision-expenses-need  1
    foreach [ course-fees-list  ] of EIA-Academy-Place  8 [
      if decision-attraction  = item ? [ attraction-emotions  ] of EIA-Academy-Place  8
and item  ? [ course-fees-list  ] of EIA-Academy-Place  8 < decision-expenses-need
      [ set decision-expenses-need  item  ? [ course-fees-list  ] of EIA-Academy-Place  8 ]
      ]
    if is-social-attraction  = true [
      foreach  [ activity-fees-list  ] of EIA-Social-Place  9 [
        if decision-attraction  = item  ? [ attraction-emotions  ] of EIA-Social-Place  9 and
item  ? [ activity-fees-list  ] of EIA-Social-Place  9 <= decision-expenses-need
        [ set decision-expenses-need  item  ? [ activity-fees-list  ] of EIA-Social-Place  9
set is-social-expenses-need  true ]
        ]
;     ;=======================================
;     ;select  the most  attribution  activity  amoung  up
;     let decision-attribution  max [ attribution-emotions  ] of EIA-Academy-Place  8
;     foreach  [ attribution-emotions  ] of EIA-Academy-Place  8 [
;       if decision-expenses-need  = item  ? [ course-fees-list  ] of EIA-Academy-Place  8
and item  ? [ attribution-emotions  ] of EIA-Academy-Place  8 < decision-attribution
;       [ set decision-attribution  item  ? [ attribution-emotions  ] of EIA-Academy-Place
8 ]
;       ]
```

```
;     if is-social-expenses-need  = true [
;       foreach  [ attribution-emotions  ] of EIA-Academy-Place  8 [
;         if decision-expenses-need  = item  ? [ course-fees-list  ] of EIA-Academy-Place
8 and item  ? [ attribution-emotions  ] of EIA-Academy-Place  8 <= decision-
attribution
;         [ set decision-attribution  item  ? [ attribution-emotions  ] of EIA-Academy-
Place  8 set is-social-attribution  true  ]
;         ]
;       ]
    ]
  ]

  ask EIA  5 [
    ifelse  [ status  ] of EIA-Orphanage-Place  6 < orphanage-status-threshold
    [
      ifelse  money-level  - money-level-threshold  < money-level-smooth-factor
      [ set selected-goal  "Orphanage"  ]
      [
        let money-x  money-level  - money-level-threshold  - money-level-smooth-factor
        let orphanage-payment  orphanage-status-threshold  - [ status  ] of EIA-
Orphanage-Place  6 + random-float  money-x  / (10 ^ ceiling  log money-x  10)
        set money-level  money-level  - orphanage-payment
        ask EIA-Orphanage-Place  6 [
          set status  status  + orphanage-payment
        ]
        type ticks type " EIA OP: " print orphanage-payment
        EIA-Reasoning-Rules
        ]
      ]
    [
      ifelse  money-level  < money-level-threshold
      [
        set selected-goal  "Work"
;       ifelse  working-capacity  > working-capacity-threshold
;       [ set selected-goal  "Work"  ]
;       [
;         ifelse  is-social-attraction  = false  and is-social-expenses-need  = false; and is-
social-attribution  = false
;         [
;           ifelse  money-level  > min  [ course-fees-list  ] of EIA-Academy-Place  8; and ([
status  ] of EIA-Orphanage-Place  6 - orphanage-status-threshold)  / orphanage-status-
decay-factor  > min  [ course-duration-list  ] of EIA-Academy-Place  8
;           [ set selected-goal  "Academy"  ]
;           [
;             ifelse  money-level  > min  [ activity-fees-list  ] of EIA-Social-Place  9
;             [ set selected-goal  "Social"  ]
;             [ set selected-goal  "Work"  ]
;             ]
;         ]
;         [
```

```
;          ifelse money-level > min [ activity-fees-list ] of EIA-Social-Place 9
;          [ set selected-goal "Social" ]
;          [
;            ifelse money-level > min [ course-fees-list ] of EIA-Academy-Place 8; and
([ status ] of EIA-Orphanage-Place 6 - orphanage-status-threshold) / orphanage-
status-decay-factor > min [ course-duration-list ] of EIA-Academy-Place 8
;            [ set selected-goal "Academy" ]
;            [ set selected-goal "Work" ]
;            ]
;          ]
;        ]
      ]
     [
      ifelse is-social-attraction = false and is-social-expenses-need = false; and is-
social-attribution = false
      [
       ifelse knowledge-capacity < knowledge-capacity-threshold and money-level >
min [ course-fees-list ] of EIA-Academy-Place 8; and ([ status ] of EIA-Orphanage-
Place 6 - orphanage-status-threshold) / orphanage-status-decay-factor > min [ course-
duration-list ] of EIA-Academy-Place 8
       [ set selected-goal "Academy" ]
       [
        ifelse social-capacity < social-capacity-threshold and money-level > min [
activity-fees-list ] of EIA-Social-Place 9
        [ set selected-goal "Social" ]
        [ set selected-goal "Work" ]
        ]
       ]
      [
       ifelse social-capacity < social-capacity-threshold and money-level > min [
activity-fees-list ] of EIA-Social-Place 9
       [ set selected-goal "Social" ]
       [
        ifelse knowledge-capacity < knowledge-capacity-threshold and money-level
> min [ course-fees-list ] of EIA-Academy-Place 8; and ([ status ] of EIA-
Orphanage-Place 6 - orphanage-status-threshold) / orphanage-status-decay-factor >
min [ course-duration-list ] of EIA-Academy-Place 8
        [ set selected-goal "Academy" ]
        [ set selected-goal "Work" ]
        ]
       ]
      ]
     ]
    type ticks type " EIA-Reasoning-Rules [     ] MPG: " type previous-goal type "
MCG: " type current-goal type " MSG: ->" print selected-goal
  ]
end

;=======================================;
;       EIA Thinking Process        ;
```

```
;     Execution  Rules  Section      ;
;========================================;

to EIA-Executing-Rules
  ask EIA 5 [
    set current-goal  selected-goal
    type ticks type " EIA-Executing-Rules  [BEFORE]  MPG: " type previous-goal  type
" MCG: ->" type current-goal  type " MSG: " print selected-goal
    ]
  if [ selected-goal  ] of EIA  5 = "Orphanage"  [ EIA-Visit-Orphanage-Places  ]
  if [ selected-goal  ] of EIA  5 = "Work"  [ EIA-Visit-Work-Places  ]
  if [ selected-goal  ] of EIA  5 = "Academy"  [ EIA-Visit-Academy-Places  ]
  if [ selected-goal  ] of EIA  5 = "Social"  [ EIA-Visit-Social-Places  ]
  ask EIA 5 [
    set previous-goal  current-goal
    type ticks type " EIA-Executing-Rules  [AFTER ] MPG: ->" type previous-goal
type " MCG: " type current-goal  type " MSG: " print selected-goal
    ]
end

;========================================;
;       EIA  Thinking  Process      ;
;    Go to Orphanage  Place Section   ;
;========================================;

to EIA-Visit-Orphanage-Places
  type ticks print " EIA-Visit-Orphanage-Places"
  while  [ [ status  ] of EIA-Orphanage-Place  6 < orphanage-status-threshold  ] [
    type ticks print " EIA-Visit-Orphanage-Places  LOOP"
    ask EIA-Orphanage-Place  6 [
      set status status  + orphanage-status-decay-factor  + Utility-Random-Float  * status
* (1 + [ event-based-emotions  ] of EIA  5 * emotions-efficiency)
      ]
    ]
end

;========================================;
;       EIA  Thinking  Process       ;
;     Go to Work  Place  Section      ;
;========================================;

to EIA-Visit-Work-Places
  type ticks print " EIA-Visit-Work-Places"
  let work-duration  0
  let offering-salary-duration  [ offering-salary  ] of EIA-Work-Place  7 / [ job-duration
] of EIA-Work-Place  7
  while  [ [ status  ] of EIA-Orphanage-Place  6 >= orphanage-status-threshold  and [
job-duration  ] of EIA-Work-Place  7 > work-duration  ] [
    type ticks print " EIA-Visit-Work-Places  LOOP"
    set work-duration  work-duration  + 1
```

```
    ask EIA 5 [
      set money-level money-level + offering-salary-duration + (Utility-Random-Float
* working-capacity + Utility-Random-Float * knowledge-capacity + Utility-
Random-Float * social-capacity) * (1 + [ attribution-emotions ] of EIA-Work-Place 7
* emotions-efficiency)
      ]
    ]
end

;=======================================;
;       EIA Thinking Process        ;
;     Go to Academy Place Section     ;
;=======================================;

to EIA-Visit-Academy-Places
  type ticks print " EIA-Visit-Academy-Places"
  let choices [ ]
  foreach filter [ ? < [ money-level ] of EIA 5 ] [ course-fees-list ] of EIA-Academy-
Place 8 [ set choices lput position ? [ course-fees-list ] of EIA-Academy-Place 8
choices ]
  let choices2 [ ]
  let decision min [ attraction-emotions ] of EIA-Academy-Place 8
  foreach choices [ if decision <= item ? [ attraction-emotions ] of EIA-Academy-
Place 8 [ set decision item ? [ attraction-emotions ] of EIA-Academy-Place 8 set
choices2 lput ? choices2 ] ]
  let random-selection one-of choices2
  set EIA-Academy-Place-random-selection random-selection
  let course-duration 0
  ask EIA 5 [
    set money-level money-level - item random-selection [ course-fees-list ] of EIA-
Academy-Place 8
    ]
  while [ [ status ] of EIA-Orphanage-Place 6 >= orphanage-status-threshold and item
random-selection [ course-duration-list ] of EIA-Academy-Place 8 > course-duration
] [
    type ticks print " EIA-Visit-Academy-Places LOOP"
    set course-duration course-duration + 1
    ]
    if course-duration = item random-selection [ course-duration-list ] of EIA-
Academy-Place 8 [
    ask EIA 5 [
      set knowledge-capacity knowledge-capacity + item random-selection [
knowledge-capacity-gained-list ] of EIA-Academy-Place 8 * (1 + item random-
selection [ attribution-emotions ] of EIA-Academy-Place 8 * emotions-efficiency)
      set working-capacity working-capacity + item random-selection [ working-
capacity-gained-list ] of EIA-Academy-Place 8 * (1 + item random-selection [
attribution-emotions ] of EIA-Academy-Place 8 * emotions-efficiency)
      ]
    ]
end
```

```
;=======================================;
;        EIA Thinking  Process        ;
;     Go to Social  Place  Section      ;
;=======================================;

to EIA-Visit-Social-Places
  type ticks print " EIA-Visit-Social-Places"
  let choices [ ]
  foreach filter [ ? < [ money-level ] of EIA 5 ] [ activity-fees-list ] of EIA-Social-
Place 9 [ set choices lput position ? [ activity-fees-list ] of EIA-Social-Place 9
choices ]
  let choices2 [ ]
  let decision min [ attraction-emotions ] of EIA-Social-Place 9
  foreach choices [ if decision <= item ? [ attraction-emotions ] of EIA-Social-Place 9
[ set decision item ? [ attraction-emotions ] of EIA-Social-Place 9 set choices2 lput ?
choices2 ] ]
  let random-selection one-of choices2
  set EIA-Social-Place-random-selection random-selection
  let activity-duration 0
  let social-capacity-duration (item random-selection [ social-capacity-gained-list ] of
EIA-Social-Place 9) / item random-selection [ activity-duration-list ] of EIA-Social-
Place 9 * (1 + item random-selection [ attribution-emotions ] of EIA-Social-Place 9 *
emotions-efficiency)
  let working-capacity-duration (item random-selection [ working-capacity-gained-
list ] of EIA-Social-Place 9) / item random-selection [ activity-duration-list ] of EIA-
Social-Place 9 * (1 + item random-selection [ attribution-emotions ] of EIA-Social-
Place 9 * emotions-efficiency)
  ask EIA 5 [
    set money-level money-level - item random-selection [ activity-fees-list ] of EIA-
Social-Place 9
    ]
  while [ [ status ] of EIA-Orphanage-Place 6 >= orphanage-status-threshold and item
random-selection [ activity-duration-list ] of EIA-Social-Place 9 > activity-duration ]
[
    type ticks print " EIA-Visit-Social-Places LOOP"
    set activity-duration activity-duration + 1
    ask EIA 5 [
      set social-capacity social-capacity + social-capacity-duration
      set working-capacity working-capacity + working-capacity-duration
      ]
    ]
end

;=======================================;
;          EIA Show Result          ;
;             Sections              ;
;=======================================;

to EIA-Show-Results
```

```
  ;Plot the result
  EIA-Plot
  ;Track information in command center
  EIA-Track
  ;Since measurements variables decay over the time
  EIA-Current-Capacity-Decay
  ;Track information in command center
  EIA-Track
end

to EIA-Plot
  ask EIA 5 [
    set-current-plot "EIA-Properties"
    set-current-plot-pen "working-capacity"
    plot working-capacity * 100
    set-current-plot-pen "money-level"
    plot money-level * 100
    set-current-plot-pen "social-capacity"
    plot social-capacity * 100
    set-current-plot-pen "knowledge-capacity"
    plot knowledge-capacity * 100
    ]
  ask EIA-Orphanage-Place 6 [
    set-current-plot "EIA-Orphanage-Place"
    set-current-plot-pen "status"
    plot status * 100
    ]
  ask EIA 5 [
    set-current-plot "EIA-Orphanage-Place-Attraction"
    set-current-plot-pen "event-based-emotions"
    plot event-based-emotions
    set-current-plot "EIA-Orphanage-Place-Attribution"
    set-current-plot-pen "event-based-emotions"
    plot 3
    ]
  ask EIA-Work-Place 7 [
    set-current-plot "EIA-Work-Place-Attraction"
    set-current-plot-pen "attraction-emotions"
    plot attraction-emotions
    set-current-plot "EIA-Work-Place-Attribution"
    set-current-plot-pen "attribution-emotions"
    plot attribution-emotions
  ]
  ask EIA-Academy-Place 8 [
    set-current-plot "EIA-Academy-Place-Attraction"
    set-current-plot-pen "attraction-emotions"
    plot item EIA-Academy-Place-random-selection attraction-emotions
    set-current-plot "EIA-Academy-Place-Attribution"
    set-current-plot-pen "attribution-emotions"
    plot item EIA-Academy-Place-random-selection attribution-emotions
```

```
  ]
  ask EIA-Social-Place 9 [
    set-current-plot "EIA-Social-Place-Attraction"
    set-current-plot-pen "attraction-emotions"
    plot item EIA-Social-Place-random-selection attraction-emotions
    set-current-plot "EIA-Social-Place-Attribution"
    set-current-plot-pen "attribution-emotions"
    plot item EIA-Social-Place-random-selection attribution-emotions
  ]
end

to EIA-Current-Capacity-Decay
  ask EIA-Orphanage-Place 6 [
    ;Orphanage status decay over the time
    if orphanage-status-decay and [ current-goal ] of EIA 5 != "Orphanage" [ set status
status - orphanage-status-decay-factor ]
    ]
  ask EIA 5 [
    ;Working capacity decay over the time
    if working-capacity-decay and current-goal != "Work" [ set working-capacity
working-capacity - working-capacity * Utility-Random-Float ]
    ;Knowledge capacity decay over the time
    if knowledge-capacity-decay and current-goal != "Academy" [ set knowledge-
capacity knowledge-capacity - knowledge-capacity * Utility-Random-Float ]
    ;Social capacity decay over the time
    if social-capacity-decay and current-goal != "Social" [ set social-capacity social-
capacity - social-capacity * Utility-Random-Float ]
    ;money level decay over the time
    if money-level-decay and current-goal != "Work" [ set money-level money-level -
money-level * Utility-Random-Float ]
    ]
end

to EIA-Track
  ;Tracing values
  ask EIA 5 [
    type ticks
    type " EIA OS: " type precision [ status ] of EIA-Orphanage-Place 6 2
    type " ML: " type precision money-level 2
    type " WC: " type precision working-capacity 2
    type " KC: " type precision knowledge-capacity 2
    type " SC: " print precision social-capacity 2
    ]
end
```