DIPLOMA THESIS

Implementation of Psychoanalytic Defense Mechanisms in Artificial Intelligence

Submitted at the
Faculty of Electrical Engineering and Information Technology,
Vienna University of Technology

under supervision of

Prof. Dr. Dietmar Dietrich
Institute number: 384
Institute of Computer Technology

and

Dipl.-Ing. Friedrich Gelbard
Institute number: 384
Institute of Computer Technology

by

Lotfi Younes
Matr.Nr. 9927133

<Day><Month> <Year> ____________________________

Abstract

Increasing demands in building automation require intelligent systems, which are able to process and evaluate large amounts of sensor data**.** However, the systems must also be capable of reacting to certain situations in real time. The existing artificial intelligence models cannot yet resolve these complexities. Therefore, a new approach is required for this domain.

A technical model of the human psyche according to psychoanalytic principles can provide this new approach. This model is being developed and tested by the Artificial Recognition System (ARS) team at the Institute of Computer Technology at Vienna University of Technology.

One of the most important aspects of psychoanalysis are the defense mechanisms. They control the behavior of humans and influence their decision making through resolving conflicts which appear due to the violations of rules.

The focus of the work at hand is the further development and technical implementation of the defense mechanisms in the ARS project. These mechanisms will be implemented for drives, as well as for perceptions. Furthermore, the module of the composition of feelings, is extended.

The defense mechanisms are implemented in a simulation environment for software agents and tested with the help of use cases. The results of the test and the evaluation show that the agents are able to resolve their conflicts using different defense mechanisms.

Kurzfassung

Steigende Anforderungen in der Gebäudeautomation, fordern intelligente Systeme, die in der Lage sein müssen, große Mengen von Sensordaten zu verarbeiten und auszuwerten. Die Systeme müssen aber auch in der Lage sein Situationen in Echtzeit zu erfassen und zu bewerten. Die bestehenden Modelle der künstlichen Intelligenz können nicht in dieser Komplexität arbeiten und können Situationen nicht ganzheitlich erfassen. Daher ist ein neuer Denkansatz für diese Problemstellung erforderlich.

Ein technisches Modell der menschlichen Psyche nach psychoanalytischen Grundsätzen kann diesen neuen Ansatz liefern. Dieses Modell wird vom Artificial Recognition System (ARS) Team am Institut für Computertechnik an der Technischen Universität Wien entwickelt und getestet.

Einer der wichtigsten Aspekte der Psychoanalyse sind die Abwehrmechanismen. Sie steuern das Verhalten von Menschen und beeinflussen ihre Entscheidungsfindung. Abwehrmechanismen lösen Konflikte, welche durch widersprüchliche Triebe, Wahrnehmungen und soziale Regeln ausgelöst werden.

Der Schwerpunkt dieser Arbeit ist die Weiterentwicklung und technische Implementierung der Abwehrmechanismen im ARS Projekt. Hier werden sowohl Abwehrmechanismen für Triebe, als auch Abwehrmechanismen für Wahrnehmung implementiert. Außerdem ist das ARS-Modul „Aufbau der Gefühle" ein Teil der vorliegenden Arbeit.

Die Abwehrmechanismen werden in einer Simulationsumgebung für Software-Agenten implementiert und mit Hilfe von Use Cases getestet. Die Ergebnisse der Tests und der Evaluierung zeigen, dass Software-Agenten in der Simulationsumgebung in der Lage sind durch die Verwendung von verschiedenen Abwehrmechanismen ihre Konflikte zu lösen.

Acknowledgements

The present work would not have been possible without the help of several persons. First of all, I would like to thank Prof. Dietmar Dietrich for the opportunity to write this thesis at the Institute of Computer Technology. I would also like to thank my supervisor Friedrich Gelbard for his encouraging way and the great support. Further, I would like to thank Klaus Doblhammer, our psychoanalyst. Thanks also to my family, for the opportunity to study. Special thanks go to my wife Andrea and my sons Karim and Samir.

Abbreviations

AGI	Artificial general intelligence
AI	Artificial intelligence
ARS	Artificial Recognition System
BDI	Belief Desire and Intention
DM	Drive Mesh
GWT	Global Workspace Theory
LIDA	Learning Intelligent Distribution Agent
MASON	Multi Agent Simulator Of Neighborhood or Networks
SOAR	State, Operator Apply Result
TP	Thing Presentation
TPM	Thing presentations mesh
WM	Working memory
WP	Word Presentation
WPM	Word Presentation mesh

Table of Contents

1. Introduction

The increasing nodes in the fieldbus systems in building automation required more intelligence into the individual node to control the building [Die00, p. 347]. For example, the Frankfurt main airport consists of more than 200.000 sensor nodes [HMK+08, p. 41-43]. Classical artificial intelligence (AI) approaches are unable to deal correctly with this complexity [PP05, p. 56]. To provide a solution, a new paradigm is required in AI. This was the main reason for the genesis of the Artificial Recognition System (ARS) project, which uses the functionalities of human thinking as defined in the psychoanalysis. The present work uses psychoanalytical defense mechanisms as basis to implement them in the decision unit of embodied autonomous agents.

1.1 Motivation

At the start of the ARS project, D. Dietrich and his research team at the Institute of Computer Technology at Vienna University of Technology chose the institute's kitchen to start to create a bionical model called "Smart Kitchen (SmaKi)" [Rus03, Fue03]. SmaKi was equipped with different types of sensors and actuators in order to perceive and recognize the situation and be able to react correctly so as to provide comfort, security, safety, and better energy management [SRF00, p. 1]. To attain this goal the kitchen was controlled by the perceptive awareness modules and later by pattern recognition modules [TDR01, p.6]. Thereafter, the idea of using psychoanalysis is followed and the ARS project was built. ARS utilises concepts of the human psyche in order to evaluate sensory information and to make decisions.

As discussed in [DZ08, p.13], AI can be categorized into four generations. The first generation is described as symbolic AI, where the focus is mainly on the symbolization and manipulation of the data submitted. The second dealt with learning and neural networks and is called statistical AI. The third phase was a very crucial realization that intelligence is associated with a body and that neither intelligence nor body exists independently. This is the principle of Embodied Intelligence. The last generation was emotional AI, where emotions, language and consciousness are considered to be significant to human beings. A new approach to AI is provided by Dietrich et al. [DZ08, p.13] which leads to a fifth generation of AI with the following features:

- Top-down design
- Uniform model with no contradictions

- Cooperation between engineers and scientists, who are able to describe the functionalities of human brain

Based on these conditions and the psychoanalytical model of the human mind called second topographical model - Id, Ego, Superego - ARS is build. More details of ARS functional model will be discussed in the present work. The second topographical model of S. Freud describes the correlation of Id, Superego, and Ego. The Id consists of external perceptions and inner drive demands. Superego represents a set of rules consisting of prohibitions and restrictions which cause conflicts with the drive demands of the Id finally the Ego deals with the connection to the outer world and plays a realistic role by satisfying the desires of the Id and Superego [DDS+14, p. 43-45]. One of the most important aspects of the psychoanalysis are defense mechanisms which are the focus of the present work. They were studied by S. Freud and his daughter A. Freud who adopted the researches and further developed the details [Wer05, p. 17]. These mechanisms are used by the instance of Ego to defend against conflicts caused by the demands of the forbidden drive or by perceptions in the instance of Superego. Applying defense mechanisms leads to produce conflict free processes in the secondary process due to the altering or suppressing of perceptions and/or drives [DDS+14, p. 62].

In comparison to software agents, conflicts can occur as discussed in [GB12, p. 4208] from the following situations:

- Input data do not fulfill the desired and predefined goals
- Inference rules in the planning section of agent
- Facts in the long and short term memory
- Social rules in multi agents systems and the desire of the agent

Using defense mechanisms support the system in a way that such situations can’t occur in order to improve the stability and the reliability.

The reasons for not yet using defense mechanisms in technical systems are: the technical complexity, differing opinions on the psychoanalytical processes, and insufficient knowledge of the functionalities of human mind [GBD+11, p. 4].

In our ongoing project, not enough defense mechanisms were for disposal and the main problem was conflicts in the phase of action planning. Therefore, the aim of the work at hand is to reduce conflicts in the decision making by implementing more defense mechanisms in ARS which have been chosen and devised with a team of psychoanalysts [GBD+11, p. 4-5]. These mechanisms are: displacement, reaction formation, sublimation, turning against the self, projection, depreciation, and idealization. Furthermore, the module of the composition of feeling has been extended. The implemented program code is written by using the programming language JAVA.

1.2 Approach, Assignment, and Goals

The aim of ARS project is to cope with the existing limitation of AI by providing a new interdisciplinary approach where psychoanalytic experts and engineering working for modeling and publishing process. As discussed above, the defense mechanism methods are still not used perfectly in AI. There are not many projects that have applied defense mechanisms. Some AI architectures have tried to apply them, but not as defined in psychoanalysis, such as the model Volitron, developed by Buller and Modeling Human Mind developed by Nitta et al. On the contrary, ARS takes this aspect very seriously. To arrange the present work with different AI architectures and to demonstrate the role of defense mechanisms in ARS, different AI architectures will be presented and discussed. The work at hand discusses the way the defense mechanisms are applied by Volitron and Modeling Human Mind. Furthermore, the missing mechanisms in the other AI architectures such as Learning Intelligent Distribution Agent (LIDA), State, Operator Apply Result (SOAR), and Belief Desire and Intention (BDI) will be also discussed.

Since defense mechanisms change drives or perceptions and the composition of feeling module transforms emotions into feelings, the functional model of ARS, its data structure, and the modules of defense mechanisms and composition of feeling in ARS will be presented. Furthermore, the description of the well known defense mechanisms will be illustrated using human life examples.

The evaluation of the implemented models has to be done by using a use-case and the simulation tool Multi Agent Simulator Of Neighborhood or Networks (MASON) and their inspectors. Therefore, different multi agent system modeling tools will be briefly described and the reasons for choosing MASON for ARS are argued. Additionally, the use-case is presented, this consists of agents who carry the whole ARS model and are placed within an environment that consists of entities such as walls, nutrition, and other agents. The input data of the agents correspond to those of a human and consist of perceptions of the environment and of inner drive demands.

To be able to document and test the different defense mechanisms of autonomous agents, their timing and quantitative behaviors have to be analyzed. For that reason, chart tools are required and have to be added to the simulation platform in order to visualize the information from each defense mechanism. These tools are time charts which analyze timing behavior of mechanisms and indicate which defense mechanism at a particular moment is activated and bar charts which denote the quantitative behaviors of each mechanism and show the number of the altered drives or perceptions.

1.3 Methodology

The processing steps of the technical models of defense mechanisms and composition of feelings are shown in the Figure 1.1 which presents the method used for the work at hand. First of all, the required literatures, which contain the description of defense mechanisms, have to be analyzed. Due the fact that psychoanalytical descriptions cover many definitions and a universally accepted model is not available, the theories used in the present work are based on psychoanalysis from examples in human life and also the work done before at the institute of computer technique such as [GBD+11] which demonstrates the comparison of functions of defense and technical filter mechanisms. Further, [GB12] explains how defense mechanisms influence the decision making process in autonomous agents by reducing conflicting data. The work done in [GBK+11] and [Rie09] describes different mechanisms. To be able to start working on the ARS project and to carry out the tasks of the presented work, the descriptions of modules with their interfaces and data structures have to be studied.

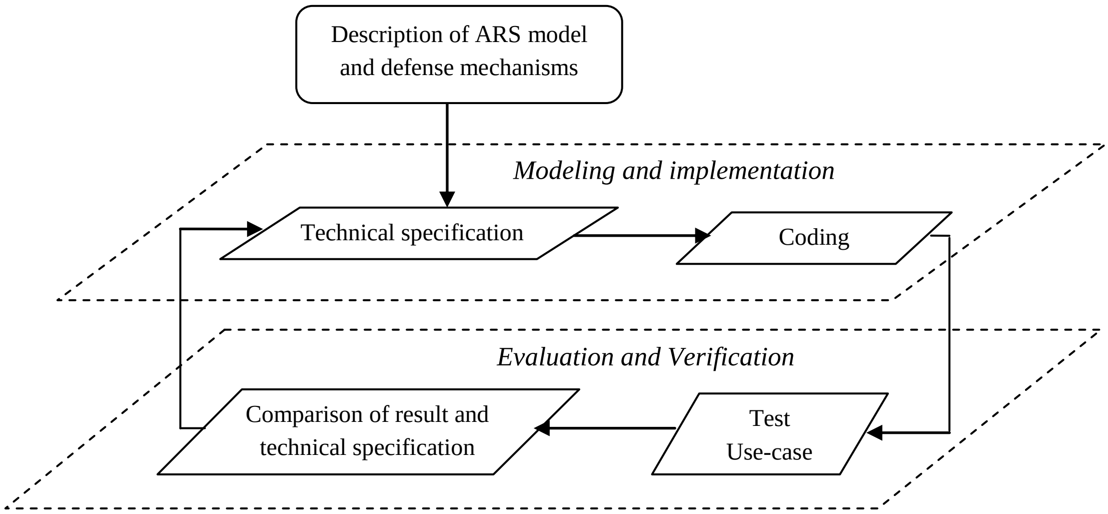


Figure 1.1: Processing steps

As seen in Figure 1.1, the processing steps consist of two levels: Modeling and implementation level and evaluation and verification level.

- Modeling and implementation: The modeling process deals with transforming the descriptions of defense mechanisms and composition of feelings into technical specifications based on the available information in the ARS project and the utilized literature. After the modeling phase, implementation begins. It maps the resultant technical model into the programming language which is used in ARS.
- Evaluation and verification: To ensure that the implemented models satisfy the goals of the present work, the evaluation and verification level is done by using a use-case and simulation tool. The use-case defines the conditions, the start, and desired outcomes of situations of agents. Its goals are to demonstrate the conflicts between Superego claims and drive demands and/or perceptions and to resolve them by the application of different defense mechanisms. To visualize the result of the implemented models, a simulator debug tool, called MASON, is used and additionally, inspectors will be implemented in order to prove their correctness.

  Figure 1.1shows that the modeling phase has been reviewed and that the steps are repeated in case of gaps or inconsistencies.

2. State of the Art

Some AI researchers adopted the term of artificial general intelligence (AGI) to refer to their research field and the reason was to attain the general nature of the research goal and scope. The AGI projects have to consist of the following criteria: A theory of "intelligence", an engineering plan for the implementation of the applied theory of intelligence must exist, and some results of the project must be published to be evaluated by the research community [WG06, p. 2-3]. Comparing these criteria with ARS, let us summarize that ARS uses instead of theory of intelligence psychoanalysis which is considered as science to investigate concepts and especially models of unconscious mental processes and mental functions [Schu97, p. 12].

The main task of the present work is applying psychoanalytical defense mechanisms in ARS and it is necessary to arrange the present work in relation to previous AI projects. As stated in Section 1.1, not many projects have used defense mechanisms. Therefore, the following Section will present some AI projects based on cognitive architecture and discusses the missing mechanisms in their processing cycle. These projects are Learning Intelligent Distribution Agent (LIDA), State, Operator Apply Result (SOAR), and Belief Desire and Intention (BDI). A further part of the Chapter "State of the Art" will present projects that have used defense mechanisms such as Volitron and Modelling Human Mind and discussed how they incorporated it. The outcome of these discussions is that the tasks and the application of defense mechanisms in ARS, where the concepts of these mechanisms are primarily taken from psychoanalysis and are transformed into technically feasible terms. Finally, different multi agent system modeling tools are briefly presented and the reasons for choosing MASON for ARS are introduced.

2.1 AI Architectures

Various approaches to human mind are used for AI such human reasoning, human cognition, and psychoanalysis. One of the most important aspects in the AI is the decision making and the mechanisms used before an action plan is selected where in the most of situations there are many actions available and only one has to be chosen. Based on this view different AI models are presented, which are based on cognitive architectures and psychoanalysis. Further, a discussion of the used and the missing mechanisms to reduced irrelevant data is discussed. Finally, ARS model and the role of defense mechanisms are presented.

2.1.1 Cognitive Architectures

The aim of cognitive architectures is to try to maintain a human cognition process by specifying an infrastructure that includes mechanisms to represent the knowledge, to store domain content, and to process this knowledge [LC06, p. 1469].

Projects such as Learning Intelligent Distribution Agent (LIDA), State, Operator Apply Result (SOAR), and Belief Desire and Intention (BDI) are presented and discussed by focussing on the approach used and the missing mechanisms. Out of these discussions, the tasks for the present work are derived.

LIDA

LIDA is an extension of the model Intelligent Distribution Agent (IDA), which is an intelligent, autonomous software agent and uses software technology developed at the University of Memphis. IDA achieves two goals: A science and an engineering goal, where the science goal is to generate the model of human cognition and the engineering goal which deals with the implementing and creating of the applications of the human work [FP06, p. 2].

The learning IDA implements the Global Workspace Theory (GWT) and extends the IDA by adding three modes of learning: perceptual learning, episodic learning, and procedural learning.

GWT treats conscious and unconscious processes and was developed by Baars. The main constructs of this theory are a global workspace, a set of specialized conscious processes and a set of unconscious context that define, choose and evoke conscious content [FP06, p. 2]. GWT makes the following assumptions [FB09, p. 2]:

- The brain is a collection of distributed processes or networks
- Consciousness is combined with a global workspace, which is a fleeting memory capacity. The contents of this memory are distributed to many unconscious processors
- Unconscious processes (contexts) may form conscious content
- To join the constrain conscious events, some unconscious networks, called contexts, can work together
- Motives and Emotions can be as parts of goal contexts
- The executive functions serve as hierarchies of goal contexts

Figure 2.1 shows the LIDA cognitive cycle. The model analyses the human cognition cycle of processes. It begins with perception and ends with an action. This Model can be described in nine steps [FRD+07, p. 63]:

1. Perception: The received internal or external stimuli are processed by perception where the beginnings of meaning are added and it produces percept.
2. Percept to preconscious buffer: Some of the data, the meaning, and relational structures are added to the produced percept. The current percept is stored in the preconscious LIDA’s working memory (workspace), and temporary structures are built.

3. Local associations: To produce the local associations, the incoming percept and the content from the working memory (WM) including emotional content are used as cues for transient episodic and declarative memory. These local associations are stored in long-term WM.
4. Competition for consciousness: Attention codelets are the content form coalitions of the long-term WM. Thus providing novel, most relevant, urgent or insistent events to consciousness.
5. Conscious broadcast: A coalition of codelets gets access to global workspace and has its contents broadcast. This process allows many form of learning and the recruitment of internal resources. Conscious broadcast is based on global workspace theory.
6. Recruitment of resources: The relevant schemes from procedural memory respond to conscious broadcast. Those schemes have context that are relevant to the information in the conscious broadcast.
7. Setting goal context hierarchy: the recruit schemes use conscious contents and emotions/feelings to set new goal context hierarchies into the behavior net.
8. Action chosen: The behavior net chooses one behaviour (scheme, goal context)
9. Action taken: Finally, LIDA act in the environment.

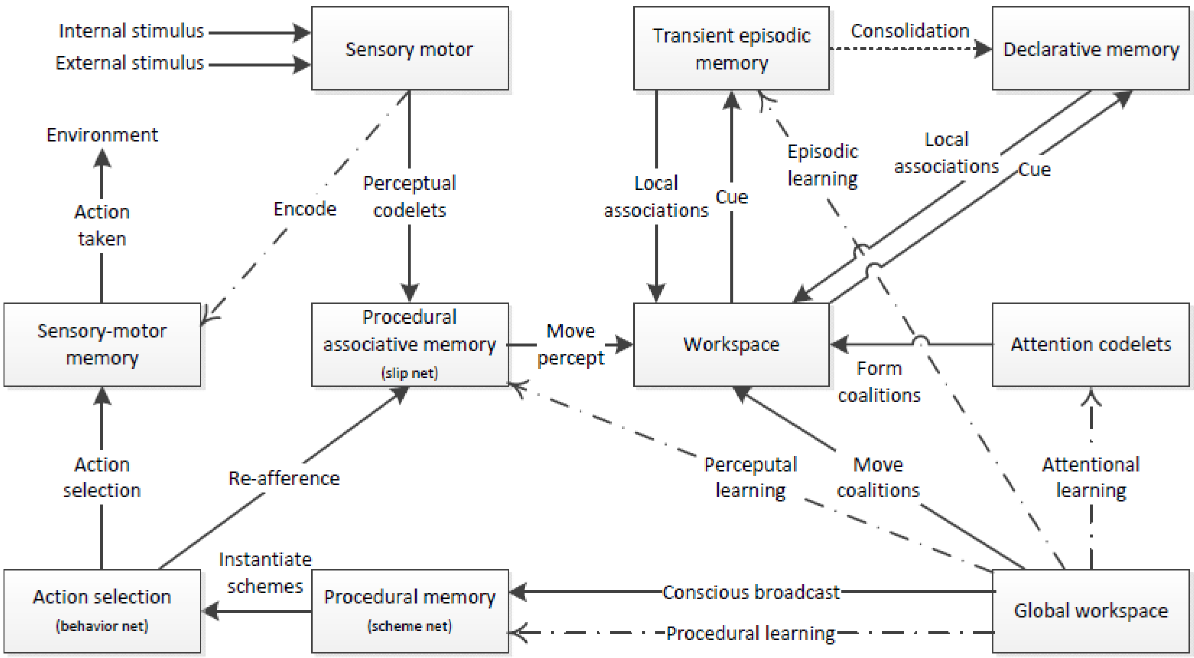


Figure 2.1: The LIDA cognitive cycle [FB09, p.4]

The LIDA was used by the U.S. NAVY and did the personal work. It communicates with the sailors, who move from one location to another. It uses natural language and email to satisfy their needs and answers the question “What do I do next?” taking care of the sailor’s preferences, time, location, and cost. Other possible applications of LIDA were discussed in [FP06, p.6] which mentioned examples like autonomous space Exploration, autonomous undersea exploration, autonomous product inspection, complex planning scheduling, and human information agent [FB06, p.7].

LIDA is based on GWT and covers some psychological and neuropsychological theories such as situated cognation, perceptual symbol systems, WM, memory by affordances, long-term WM, and transient episodic memory [FB09, p. 3]. As described in the cognitive cycle, the passage to the consciousness phase in LIDA is done by attention codelets which ensure the most urgent, important, and relevant coalition contents that become conscious. This means that the attention codelets in LIDA are responsible for passage to the conscious phase. Analysing cognitive cycle of LIDA, let a question arise concerning the irrelevant coalition contents. Applying defense mechanisms to LIDA resolve this situation in such a way: First rules have to be declared and after the irrelevant coalition contents which do not agree with these rules have to be modified by using different defense mechanisms in order to be useful for decision making.

SOAR

State, Operator Apply Result (SOAR) is based on cognitive architecture and was proposed by Laird Newel and Rosenbloom in 1987. SOAR has progressed, where eighth versions were developed between 1982 and 2007 [Lai08, p.1].

As shown in Figure 2.2, the SOAR 9 architecture consists of two kinds of memory: Symbolic Long-Term memory and Symbolic Short-Term memory. The first memory contains three types: procedural memory presents knowledge as production rules, semantic memory presents facts, and episodic memory presents experiences. The Symbolic Short-Term memory contain a graph structure, which presents objects with properties and relations.

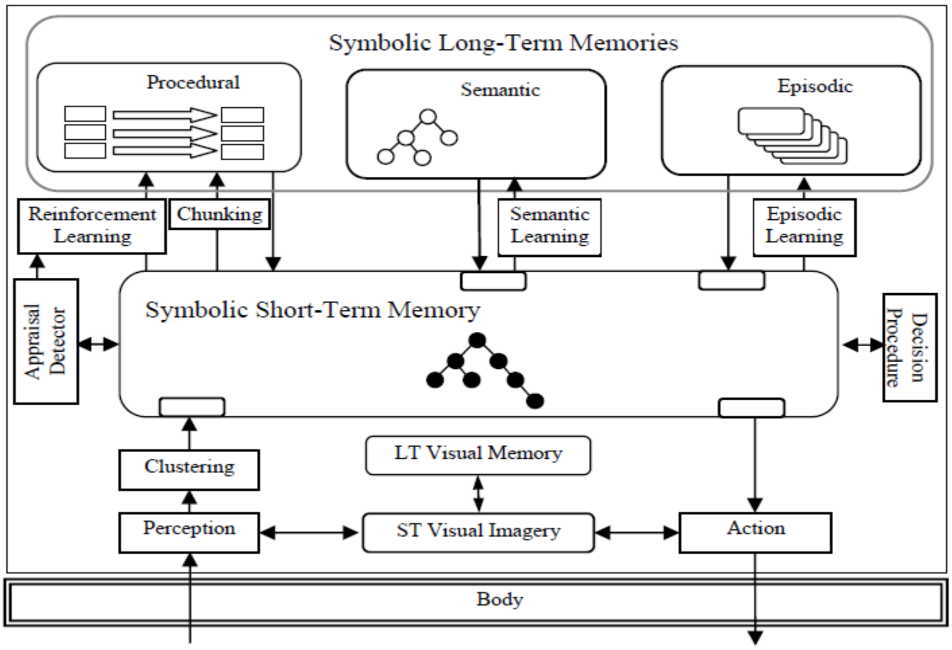


Figure 2.2: Structure of SOAR 9 [Lai08, p. 3]

In addition, SOAR has a reinforcement learning, chunking, an appraisal detector and a decision procedure, where reinforcement learning regulates the knowledge of operator selection, Chunking is a mechanism to learn new procedural knowledge, the appraisal detector generates emotions, feeling, and an internal reward signal to reinforcement learning, and the decision procedure chooses the operator and detects impasses [Lai08, p.2].

The execution of SOAR occurs over by number of cycles and each cycle consists of five steps as shown in Figure 2.3 [Lai08, p.4]:

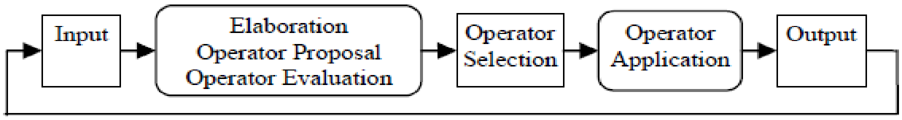


Figure 2.3: Soar processing cycle [Lai08, p. 4]

- Input: Short term memory receive new sensory data
- Elaboration: The current state is elaborated. For example, if the goal is to grasp an object, a structure is made that indicates whether the object within reach.
- Operator Proposal: Rules propose operators for the current situation,
- Operator Evaluation: The evaluation of an operator is made by creating the preferences for them based on the current situation and goal.
- Operator selection: A new operator can be selected from the generated preferences or, if the preferences are insufficient to make a decision an impasse appears and a new substate is created to resolve it. In the substate, SOAR uses the same processing cycle to select and apply operator.
- Operator Application: Rules perform action of operator that agree with the current situation and the structure of operator.
- Output: Output commands are sent to the motor system.

SOAR architecture was used in TacAir-Soar by U.S. military, which contains 5200 production rules, 450 operator, and 130 abstract operators. TacAir-SOAR agent is able to provide a flight operation that allows interaction between two agents via simulated radio systems. Other applications are the controlling of unmanned aircraft and vehicles [HJL09, p.71].

SOAR is a rule based system describing the world in terms of problem space with states and goals, where performed actions are moved in the problem states by operators. Before an action is taken, SOAR reduces data by selecting operators. When more than one operator can appear in a single cycle just one will be chosen to transform the current state to desired goal state. As discussed above, an evaluation of an operator is done by adding the preferences. This is how SOAR reduces data in order to make a decision. In the step of operator selecting in the producing cycle, if the preferences are not able to make a decision, SOAR creates a new substate and repeats the same processing cycle. Defense mechanisms release such situation so that all data are processed in one producing cycle. This don by defining rules where their violation cause conflicts which is resolved by applying different mechanisms.

BDI

Belief Desire and Intention (BDI) is based on the philosophical theory of by Bratman (1987) and the theory and practice of by Rao and Georgeff (1995). The main goal on BDI agent is to select an action to satisfy its desire [AG95, p. 312].

The BDI agent is defined by its beliefs, its desires, and its intentions. Beliefs provide information about itself, the environment, and may be information about other agents. Desires or goals are factors that cause an action and intentions are engagements aimed at achieving particular goals. Usually the agent holds different plans for different situations, those plans describe how the intentions can be achieved and the steps that have to be taken.

BDI agent has to determine goals (desires) that are to follow next and to achieve these goals the correspondent plans have to be selected. To attain this process, the agent analyses the goal base and derives those that are not yet satisfied. Next it obtains all plans from plane base which can be use to achieve these goals. Finally it verifies the current belief base whether it agrees with the context of the plan which was design for it [BACR08, p. 38].

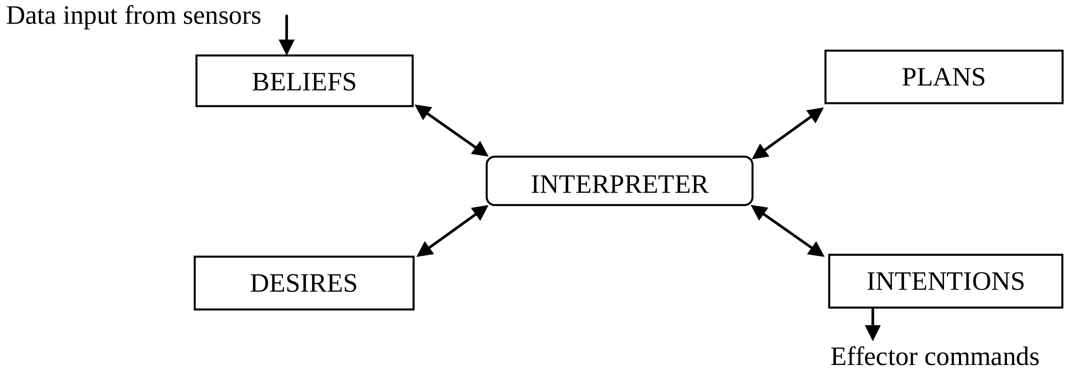


Figure 2.4: BDI agent architecture [Woo96, p. 2]

BDI was used in an air-traffic management system, OASIS. This is an agent-oriented and consists of one aircraft agent and a number of global agents. The aircraft agent is responsible for flying along a flight path which is described by coordinates and a sequence of waypoints. The global agents can be up to seventy or eighty agents such as sequencers, wind modelers, coordinators, and trajectory checker agent. Beliefs in OASIS can include all possible wind rapidity and trajectories, the desires are the correct expected times of arrival and the intention of the aircraft agent is the achieving of the most efficient trajectories [AG95, p. 317]. The Procedural Reasoning System (PRS) and the distributed multi-agent reasoning system (dMARS) used the BDI model in other large applications such as telecommunication network management, air combat modeling, business process management and space shuttle diagnosis [AG95, p.318].

BDI is based on folk-psychological view of reasoning [N04, p. 202] which provides a robust method of human reasoning. Decisions in BDI are made by selecting one plan from many available plans after it has been verified by the belief base. Checking and verifying all plans increases computational time and makes the BDI agent dependent on the number of plans. Similar to the previous models, unused data are not considered in BDI. This in contrast to defens mechanisms, where the unused or undesired data are modified in order to become usefull in choosing an action. These mechanisms reduce the computational times by processing all data in one processing cycle to react in real time.

2.1.2 Models Based on Psychoanalysis

Psychoanalysis and defense mechanisms are new aspects in the AI [GBD+11, p.6]. There are two models, which are until now trying to implement defense mechanisms. In this section, the models Volitron and Modeling Human Mind are presented and discussed the application of defense mechanisms. Finally, based on the missing mechanisms in the AI architectures discussed in this section and the section before, the ARS model and the principles of defense mechanisms are presented.

Volitron

Buller developed a controller called Volitron, which increases the competence in activities such as self-initiated exploration of an environment, new goal acquisition, and planning /execution of action, by its host robot [Bul02, p. 1].

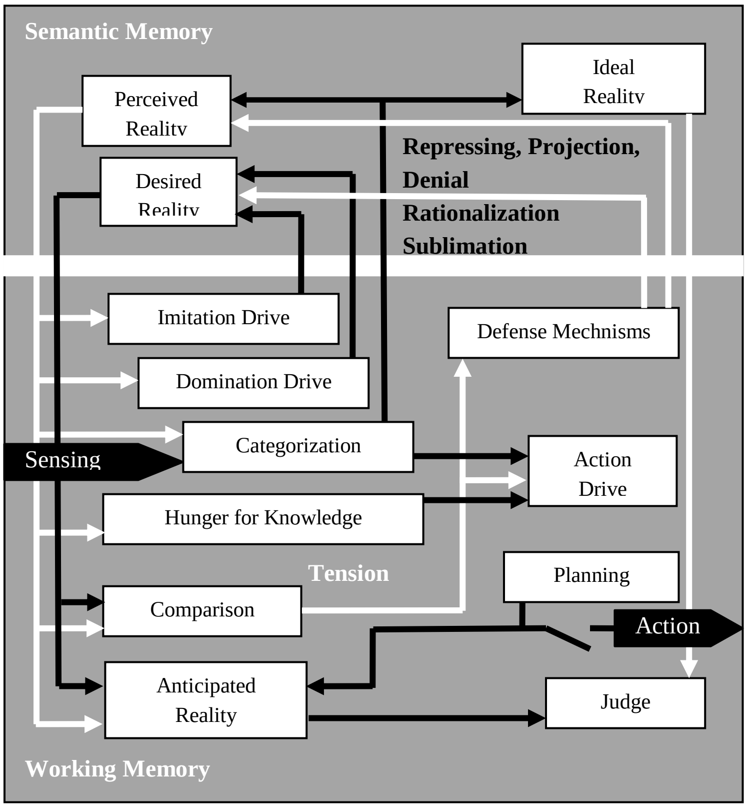


Figure 2.5: Structure of Volitron [Bul02, p. 2]

In relation to psychoanalysis, Buller introduces tension and defense mechanisms into the structure of Volitron. As in the human memory system, Volitron uses a structure created by Tulving and consist of five types of memories: perceptual representation system (PRS), procedural memory, semantic memory, episodic memory are used to store information for long periods of time, and the working memory (WM) which is used to process pieces of information (memes) and to change the content of these memories.

Figure 2.5 presents the architecture of Volitron, which is based on four models: model for perceived reality, model of desired reality, model of ideal reality, and model of anticipated reality. The memes (pieces of information) appear into WM, for example, memes of satisfaction and memes of dissatisfaction. The tension is calculated based on a numerical balance of these memes. The manner in which memes become satisfied or dissatisfied, as Buller explains it, is as in the theory of Freud, "mental life is a kind of a continuous battle between conflicting psychological forces such as wishes, fears, and intentions".

In Volitron, memes are injected from semantic memory into WM. The essentials memes processing in WM is explained in the following [Bul02, p. 2]:

- Categorization appears in the production of memes representing candidate recognitions
- Hunger for knowledge production results if the memes have the knowledge of missing elements in the model of perceived reality.
- Imitation-drive production executes on memes, which carry perceived action achieved by observed people or other robots, they changed to memes that carry the same action that is performed by the robot itself.
- Model comparison processes the memes coming from the model of perceived reality and the model of desired reality. The result of this process is the production of memes of dissatisfaction. The aim of this process is to change the environment so that perceived reality resembles the desired reality.
- Candidate-plan generation generates and tests the plans and the intention of this process is to produce anticipated reality in the WM.
- Judgment of candidate plans and anticipated results appear in the device judge, which compares the anticipated reality with relevant part of the model of ideal reality. Depending on the result of this comparison, an acceptable plan appears in this process, which can be physically executed.
- Defense mechanisms in Volitron are repression, projection, denial, rationalization and sublimation. This process occurs when no acceptable plan exists for long time or the execution of a plan hasn't changed the environment. The memes produced in this process can create change in the models of the reality and reduce the tension.

Volitron was tested in three robots *Neko*, *Miao* and *Miao-v*, where the task of *Neko* was obstacle avoidance and has tensions like fear, excitation, boredom, and anxiety. *Miao* is a simulated robot and has the same tensions as *Neko* but includes hunger. *Miao-V* is a simulated robot like *Miao*, its brain is such as neural network which can develop and grow in a literal sense: Each pleasure-related experience adds new cells and connection to its neural network. The tension of *Mia-V* is expressed as a desire for red, green, and yellow objects and the learning process of *Miao-V* was achieved with a caregiver. *Miao-V* and its caregiver developed a common, mutually understandable proto-language [PAL08, p. 61].

Volitron uses some psychoanalysis terms such as defense mechanism and tension. As discussed in [Deu11, p. 45], tension in Volitron is similar to drive in psychoanalysis. However, the drives consist of sexual and aggression drives and they were missing in Volitron. Furthermore, how tensions are composed, is also missing. Buller uses defense mechanisms if no acceptable plan is produced for a long time or the produced plan did not change the environment. This causes a delay in the processing time. According to psychoanalyse, defense mechanisms are used to solve conflicts. Therefore rules have to be declared in Volitron and their violation cause conflicts which lead to the activation of defense mechanisms. Volitron uses five defense mechanisms: repression, projection, denial, rationalization, and sublimation. However, which mechanism is activated at certain moment and their dependence are not declared.

Modeling Human Mind

Modeling Human Mind was developed by Nitta et al. [NTMI99]. According to this work, the human mind is defined as non-intellectual functions such as emotions and consciousness and intellectual activities, such as learning, reasoning, judging, estimating, memorizing, and association. Based on these non-intellectual functions this project is modeled using psychoanalysis with the assumption that the defense mechanisms play an important function in human personality and use inductive probability to model them [NTMI99, p. 2].

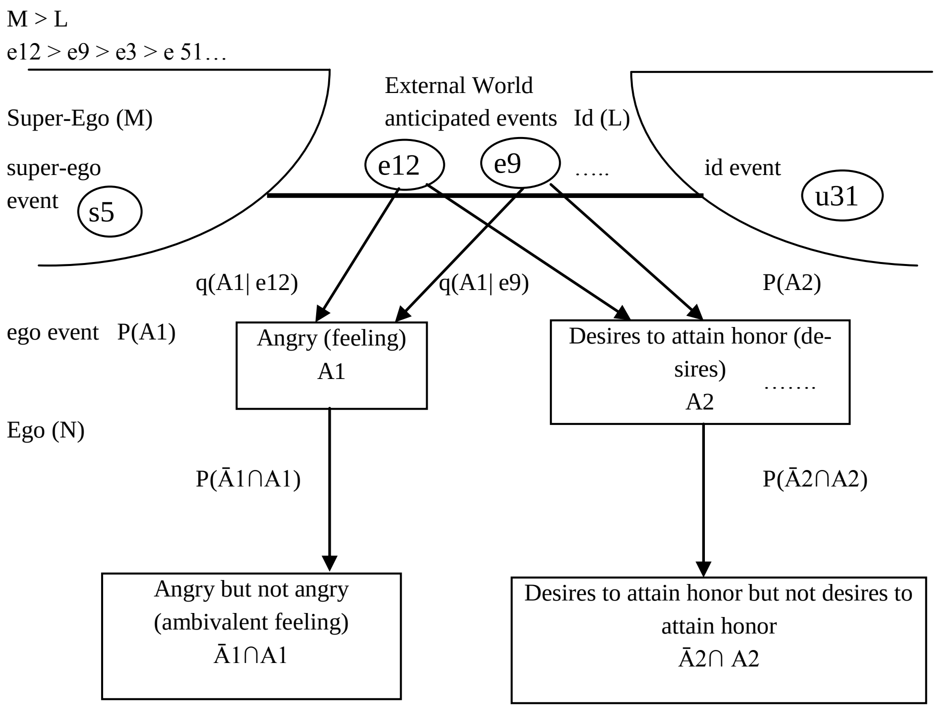


Figure 2.6: Framework of the personality model [NTNI99, p.4]

A framework is modeled in this project using the following mechanisms:

Trigger ⇨ Conflict ⇨ Anxiety ⇔ Defense

As shown on the Figure 2.6, the framework is composed of three modules: Ego, Id, and Superego. For each module certain events occur. The actualization of the events of the Superego “u” and the events of Id “s” are dependent on the person. Ego events contain feeling, memories, ideas, and desires and are defined by the inductive probability which is calculates by anticipated events. Each module holds energy, where the sum of the energy of Ego (N), Id (M), and SuperEgo (L) are equivalent to 1 and remains constant ($M + N + L = 1$) [NTMI99, p. 3-4].

Defense mechanisms are dependent on the energy of the Ego N, where, if the magnitude of N is too low, it implies that the Ego is weak and that defense mechanisms have to be activated. On the other hand when N is large, the Ego is strong and the person has to face the reality to solve the conflict without using defense mechanisms. The conflicts are based on the equation of the inductive probability $0 \leq P(A \cap \bar{A}) \leq 1/4$, and they occur if this equation is not true. Anxiety is defined as a real number and called anxiety parameter which activates defense mechanism repression if it is very strong [NTNI99, p. 3-6].

Modeling Human Mind uses psychoanalytical terms such as Ego, Superego, Id, defense mechanisms, anxiety, psychic energy, and conflict, but they are not perfectly defined. In psychoanalysis defense mechanisms are used to resolve conflicts that are defined in the Super-ego as rules and not as inductive probability. The term “drive” is missing too in this model. Before an action is taken, modeling human mind uses defense mechanisms like repression if the Ego energy is too low otherwise Ego solves the conflict by facing the reality without any defense mechanism. But defense mechanisms are not dependent on the state of the Ego’s energy.

ARS-Project

The aim of the ARS project is to develop an automation system, which can cope with an immense amount of data, as in building automation where many of computer nodes with sensors and actuators could be used to control a large building. Such a system must be more intelligent. There are many methods and principles in the nature, one example is how humans reach a decision.

According to the conditions of fifth generation of AI provided by Dietrich et al. [DZ08, p.13] which are discussed in the Chapter 1, the ARS model is built by project team members who are from different scientific fields and are working together to apply the Freud’s definitions and explanations using a top-down approach of a second topographical model of Freud to design and develop design unit for an autonomous agent. The Figure 2.7 shows the mind of the agent that consists of a sensor interface, an actuator interface, and the psychic apparatus. Information from the environment is collected by the sensor system and the resulting data are transmitted to psychic apparatus which is the central part of the model and creates the control decision. The actuator interface couples the mechanical body elements with the Psychic apparatus.

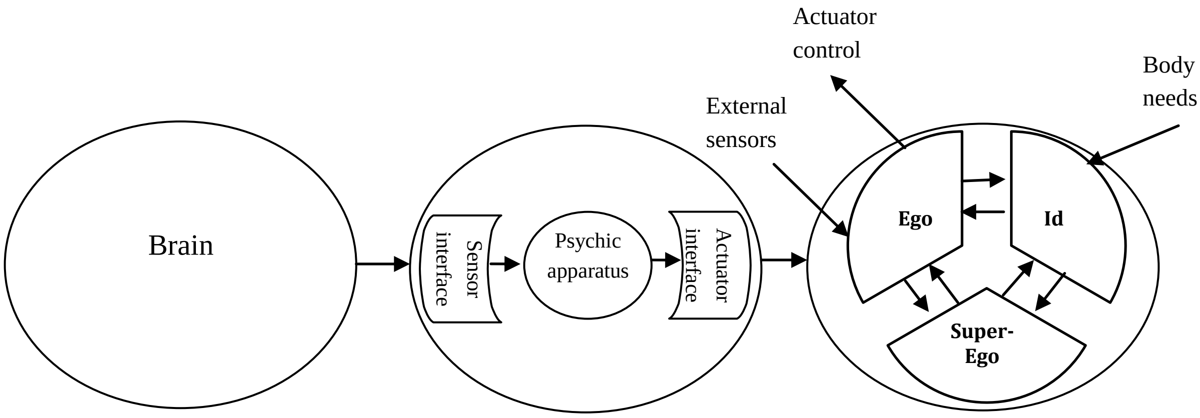


Figure 2.7: ARS model [DFZB08, p. 56]

Psychic apparatus is a distributed and purely functional system. Further it represents the second topographical model of Freud and consists of sub modules such as Ego, Id, and Super-Ego. Figure 2.7 also shows the connection between the sub modules where the Ego receives data from the environment and sends control decision to an actuator control. All processes within the psychic apparatus modules are centred on Ego. Id represents data from body needs and external perceptions which are received from the Ego. The body needs can be understood in the software agents or robotic systems as, for example, a desire to maintain temperature levels or a desire to recharge batteries. Super-Ego represents social rules which cause conflicts with the drive demands of the Id. Conversely, Ego tries to avoid these conflicts using different defense mechanisms.

In this section, the description of ARS summarized the tasks and principles of defense mechanisms that have to perform during the action planning. The description of entire functional model of ARS will be presenting in the Chapter 3.

The steps that are used in ARS before an action is selected are shown in the Figure 2.8. The first, the construction of data have to be built which consist of the perceptions of the environment and body needs and some of this data cause conflict. As discussed in Chapter 1, conflicts arise from different situations such as the input data do not agree with the desired goals, rules in multi agents system, and facts in the long and short memory. In ARS project, these conflicts have to be detected, thereafter different psychoanalytical defense mechanisms are used to resolve them.

The Figure 2.8 shows also a backward loop to perceptions before action is selected. This loop checks the altered expectation in different context and situation to prove the reality check for perceptions in the environment. Defense mechanisms are placed before action planning module in the primary process in order to choose an action plan without conflicting rules or plans. ARS uses two arts of defense mechanisms: Mechanisms for drives and mechanisms for perceptions.

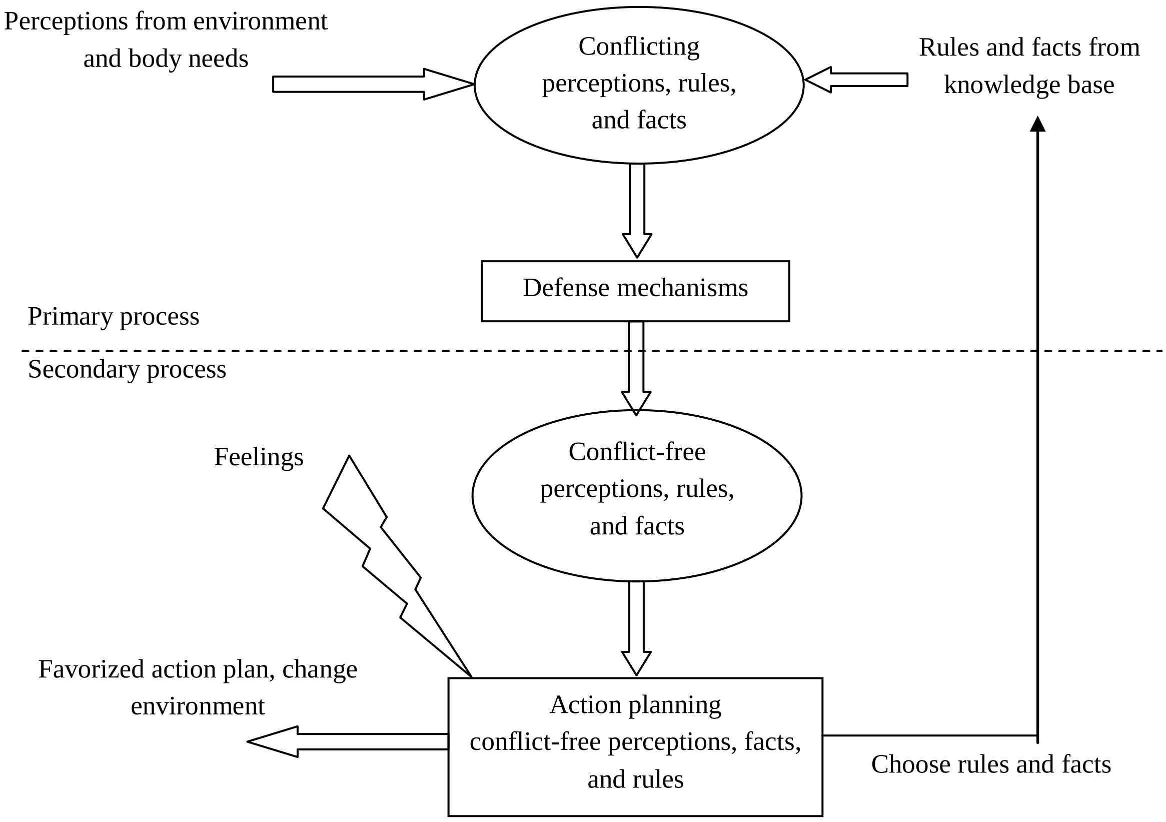


Figure 2.8: Defense mechanisms and action planning modules in ARS

As sketched in the Figure 2.8, another aspect influences decision in ARS, which are the feelings. They are defined in the primary process as emotions and converted to the feeling in the secondary process.

Summary

As discussed above, Volitron and modelling human mind do not implement defense mechanisms as they are defined in psychoanalysis. In our ongoing ARS project, defense mechanisms are implemented as defined in the psychoanalysis, which resolves conflicts and discharges decision making unit from conflicting actions and improve its convergence. Furthermore, in ARS data which are not corresponding to the rules are transformed or repressed by the use of different mechanisms in order to choose a conflicting free action. This process will take place in one processing cycle. This leads to a reduction of any delay and computation time in the producing cycle.

2.2 Multi Agent Simulation

The most widely used definition of an agent is presented by Wooldridge. It states that an agent is a computer system, which has to behave and be able to execute autonomous action in an environment to achieve their design objectives. Multi agent systems (MAS) are systems composed of multiple interacting agents [W02, p. 1]. Further Wooldridge and Jennings were suggested that the intelligent agent has to have the following properties:

- Reactivity: The capability to perceive the environment and timely acting, in case of changes, to achieve the design objectives.
- Proactiveness: The capability to initiate goal-directed behaviour to achieve the design objectivities.
- Social ability: The capability to interact with other agents or humans to achieve design objectivities.

The comparison of these properties with ARS model summarize that our model is suitable for multi agent systems. The reactivity in ARS is done by sensors to collect information about the environment in real time and procativeness is done in the secondary process, where a decision is selected and executed by actuator. The most significant property for the present work is the social ability that is defined as rules in ARS, which have to be considered during decision making. The violation of these rules causes conflicts and different defense mechanisms can be selected to resolve them.

It can be predicated that multi agent system is a basic for testing defense mechanisms. Other implicit mean of multi agent communication is the environment, which can be considered as place of the interactions of agents with objects and resources. As discussed in Chapter 1, an environment is implemented by ARS team project, which consist of wall, nutrition, agents etc. This will be described later in the Chapter 5.

As illustrated in Chapter 1, that the validation of the implementation of defense mechanisms required visualization tools such as time charts and bar charts and they have to be added to the simulation platform in order to document and test defense mechanisms. Furthermore, the mental processes, which trigger the movement of agents, have to be analyzed and especially modules of defense mechanisms and composition of feelings with their interfaces. Therefore, the simulator has to provide the following features:

- Supporting the programming language “Java”, which is used by ARS project
- Simulation of many agents carrying complex decision units, which required higher execution speed
- Feasibility of simulation of a social system with 2D visualization to achieve the interaction with other agents and objects
- Inspectors to view internal state of agents and analyze different modules of ARS. Further, support of Graphic User Interface (GUI) to be able to implement tools, such time charts and bar charts

Previous possible simulation platforms are available to test agents in an environment. The Team of ARS project chose MASON to evaluate the model. In this Chapter the following platform are briefly described in order to present the reason for choosing MASON as simulator for our project: Swarm, NetLogo, RePast, and AnyLogic. Their description includes the programming language, supported operating systems, and the provided tools. Next, general architectural of MASON is presented and finally the reason for choosing MASON is described.

Swarm

Swarm is a framework and library platforms for agent-based models [RLJ06, p. 609] and was written in Objectiv-C on a Unix-based operating system. Objectiv-C is an object-oriented extension of the C programming language. The Current version, Swarm 2.2, makes Objective-C libraries available to Java code and can be used on Windows using Cygwin, Linux, or Macintoch. The Swarm platform provides a framework for designing, describing, controlling of an agent-based model, and software implementation. A community of users and developers share ideas, software, and experience [3].

The disadvantages of Swarm are the Objective C programming language, error handling, and the lack of developer tools [RLJ06, p. 609].

NetLogo

NetLogo was developed in 1999 by Uri Wilensky and is an open source project. It was designed for educational use and based on a Logo programming language. NetLogo is cross-platform running on Windows, Linux, and Mac Os. NetLogo is a programmable platform for simulating natural and social phenomena [3].

The main disadvantage of NetLogo is the used programming language (Logo), which is a non standard programming language like Java or C++.

RePast

RePast developed at the University of Chicago, used many concepts from Swarm, is implemented in Java and is free open source. A variety of agent templates and examples can be used, RePast supports concurrent discrete event schedulers and provides graphing tools and two dimensional agent environments. Development of RePast models can use many programming languages like Java, C#, Visual Basic.Net, Managed Lisp, Managed Prolog, and Python scripting. RePast can be used on Windows, Linux, and Mac Os [5].

The problem of RePast is the execution speed, where MASON 1-35% faster than Repast [RLJ06, p. 610].

AnyLogic

AnyLogic is developed by XJtek in Java and support modelling of agent-based, system dynamic, discret-event, continous, and dynamic system models. It provides graphing tools and an easy two dimensional agent environments [6].

The previous versions of ARS used Bubble Family Game (BFG) as a simulator, this was implemented in AnyLogic, and the aim of this simulator was to generate an artificial world, which consisted of foods, agents, points of interest, and different landscape. During development of BFG, there were many problems with features such as execution time, debugging abilities, and the state chart. However, these do not affect the basic requirements for ARS development [Deu11, p. 48].

MASON

MASON is a multi agent system, a single process discrete event simulation core and a visualization toolkit. The main advantage of MASON is its flexibility and can be used for a wide range of simulations and to support a large number of agents (up to millions). The system is written in Java and is open-source [L11].

Figure 2.9 shows MASON architecture, it is divided into two layers: Model used for simulation and Visualisation which allows 2D or 3D projection. The layers can be run separately, this means that the model can run with or without visualisation [L11, p 9-10].

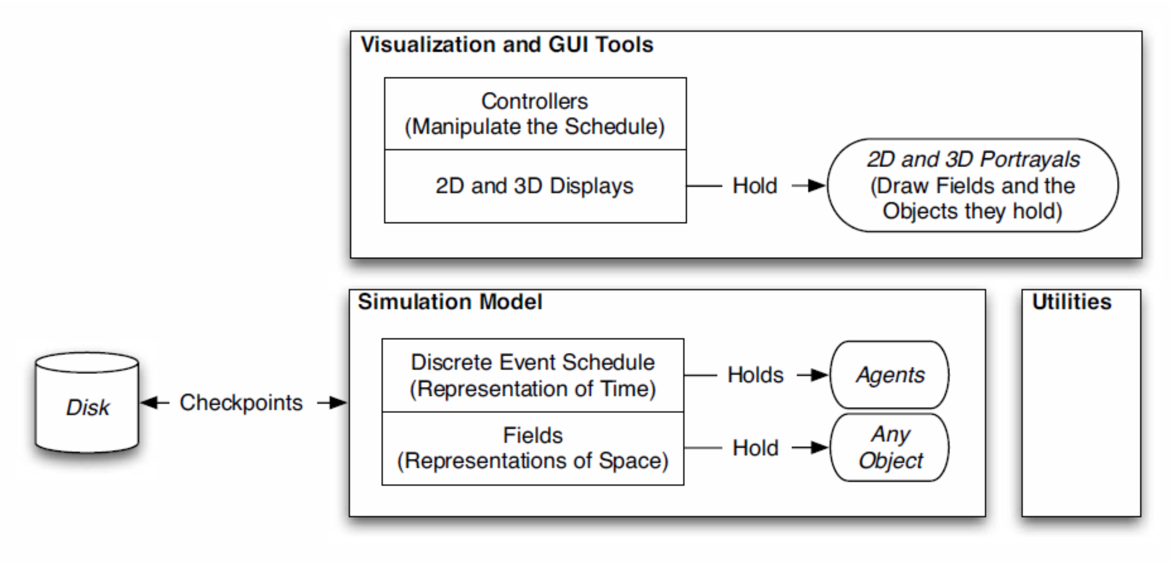


Figure 2.9: MASON utilities, model, and visualization layers [BR+03, p. 5]

- ***Model:*** It consists of schedule, agents, field, objects or values, and a high-quality random number generator. The schedule is a discrete event, which schedule different agents by calling them at the sometime in the future. Schedule and agents represent time in the MASON model and allows agents to perform an action in the future. To represent space, MASON provides one or more fields to associate object or values together and they are an arbitrary data structure. In additional, MASON contains a high-quality random number generator.
- ***Visualization:*** MASON allows 2D and 3D visualization tools. The visualization layer contains a controller which is responsible for starting, stopping, and manipulating the schedule. Further the controller manages the displays, which project the fields as a 2D or 3D visualization using field portrayals. To view objects or values a simple portrayal is designed. The simple portrayal also provides inspection of model details by selecting objects. The GUI contains its own auxiliary schedule which is managed tied to the underlying schedule in order to queue visualization that update the GUI displays. The both schedules are stepped through a controller.

- ***Utilities:*** The Utilities layers consist of java classes, which include bags, immutable 2D and 3D vectors, and a high-quality implementation of random number generator [BR+03].

The main advantage of MASON is the execution speed and it was designed with the focus on models with many agents over much iteration. The facility of stopping the simulation, detach and reattach graphical interfaces are considered as priority for long simulation in MASON [RLJ06, p. 613].

3. Model and Concept

Based on the research results of the different AI architectures introduced in Section 2.1, this chapter presents a technical model of defense mechanisms according to the concept discussed in the Section 2.1.2. This is done by first presenting the functional model of ARS, which consists of 41 modules. The data structures of ARS are presented and the modules of ARS are described in detail. The main task of the present work is the implementation of defense mechanisms reaction formation, sublimation, displacement, turning against the self, projection, idealization, and depreciation. Further, the module of composition of feelings in ARS is extended. Due to the task of the work at hand, theories of psychoanalytical defense mechanisms are illustrated with examples of human life. Finally, to be able to implement the modules of composition of feelings and the defense mechanisms, technical models are presented.

3.1 ARS Model

To be able to model the defense mechanisms and the extension of the module of composition of feeling, the data structures, the functional model of ARS, and the modules of defense mechanisms and composition of feelings are presented in this section.

3.1.1 Data Structures in ARS

The information representation in ARS is shown in the Figure 3.1 where the modules of the reasoning unit are receiving data from the neuro-symbolic layer and accessing the database to retrieve or store information. This neuro-symbolic layer is responsible for turning sensory signals into symbols where the resulting symbols are similar to thing presentation in psychoanalysis [Deu11, p. 74].

There are two modes in ARS which manipulate the data in the reasoning unit: the primary and the secondary process. The thing presentations (TPs) are manipulated in the first process and word presentation in the second process. Word presentations (WPs) are a set of signs and combine a TP to one sign, similar to human language where the alphabets and vocabularies are represented by set of signs [Zei10, p. 58].

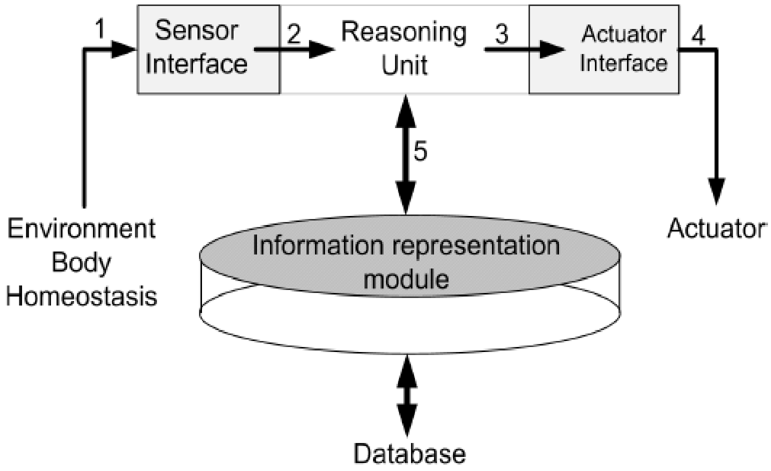


Figure 3.1: Information flow through the reasoning unit [ZPK10, p.710]

TPs are defined by acoustic, visual, haptic, olfactory and taste modalities. There are two types of TPs: First represents the sensor information of a physical object and the second one defines scenarios like movement, handling, and basic ability. Several TPs are mapped in thing presentation mesh (TPM) and they are temporally or locally associated. The temporally associations are dependent on the time such as distance or position and the locally associations describe the properties of data structure such as form, color, and size [DDS+14, p. 43].

TPMs are linked to tuples, which define the homeostatic impact and are labeled as drive mesh (DM) in ARS. One TPM can be associated with diverse DMs. For example, an object can be satisfied in different ways [Zei10, p.57].

DM consists of psychic categories, which are used as indicator for a particular drive. These categories are the name of the drive, the drive type, and the quota of affect. Furthermore, they are divided into four partial drives: "anal", "oral", "phallic" and "genital". There are two drive types AGGRESSIVE and LIBIDINOUS. The quota of affect defines the quantitative need for a drive which has to be achieved. In the case of avoidance of entities, the quota of affect can be a PLEASURE (positive) or UNPLEASURE (negative). For example, a certain DM is associated with something edible and could have the following values: categories type “LIBIDINOUS”, drive name “nourish” and the quota of affect 1.0 (high pleasure). For the agent, this means that this entity is edible and it would be even preferred [DDS+14, p. 44].

Conversely the TPMs depend on the sensors, WPs are depending on the environment and introduce scenarios and plans. As sketched in the Figure 3.2, the structure of the information is divided in the primary and secondary processes. The primary process is represented by TPMs and they are associated. Further, as seen in the Figure 3.2, affects with their TPs are attached to a mesh. In the second process, the structure of information is represented as WPs which are associated to TP and between each other. TPs of the associated word presentations are not connected. Relations between WPs contain various components like temporal attributes and actions which have to be applied to the object [ZPK10, p. 710].

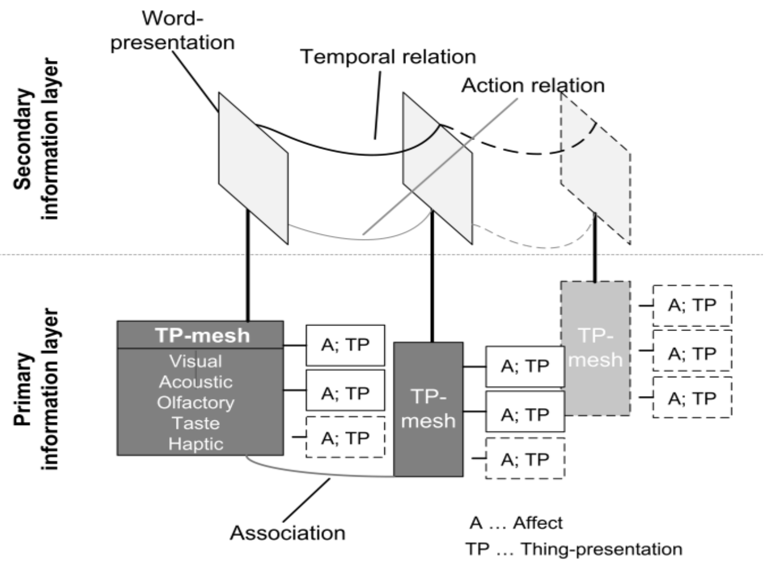


Figure 3.2: Information representation structure [ZPK10, p. 711]

The associations between thing presentation (temporal and attribute) contain a weight, which defines the importance of the link. This weight varies in the future implementation. For example, if a TP is retrieved from memory, its activation level increases. This leads to increase all the activation level of the connected TP and affects [Deu11, p. 74].

3.1.2 ARS Functional Model

The Figure 3.3 shows the functional model of ARS, which covers the processes of the perception, decision making and acting. The model is divided into functional tracks, and the labelled modules are important for the present work. They are presented with their interfaces like the module of composition of feelings, super-ego reactive and proactive, and the modules of defense mechanisms for drives and perceptions. The description of the functional model of ARS is cited from [DDS+14, p. 60-96].

Self-preservation drive track

The self-preservation drive track consists of parts of all layers (1, 2 and 3) and includes three functions: Sensor Metabolism F1, Neurosymbolization of needs F2, and Partial self-preservation drives F65.

The sensors of the module F1 provide data about state values of body organs such as information about metabolism, stomach tension, blood pressure, energy, stomach etc. This is converted into homeostatic neurosymbols in F2 so that it can be processed by F65 which generates from each bodily need an aggressive and libidinous drives (Drive Mesh), that contain drive object, drive aim, drive source, and quota of affect.

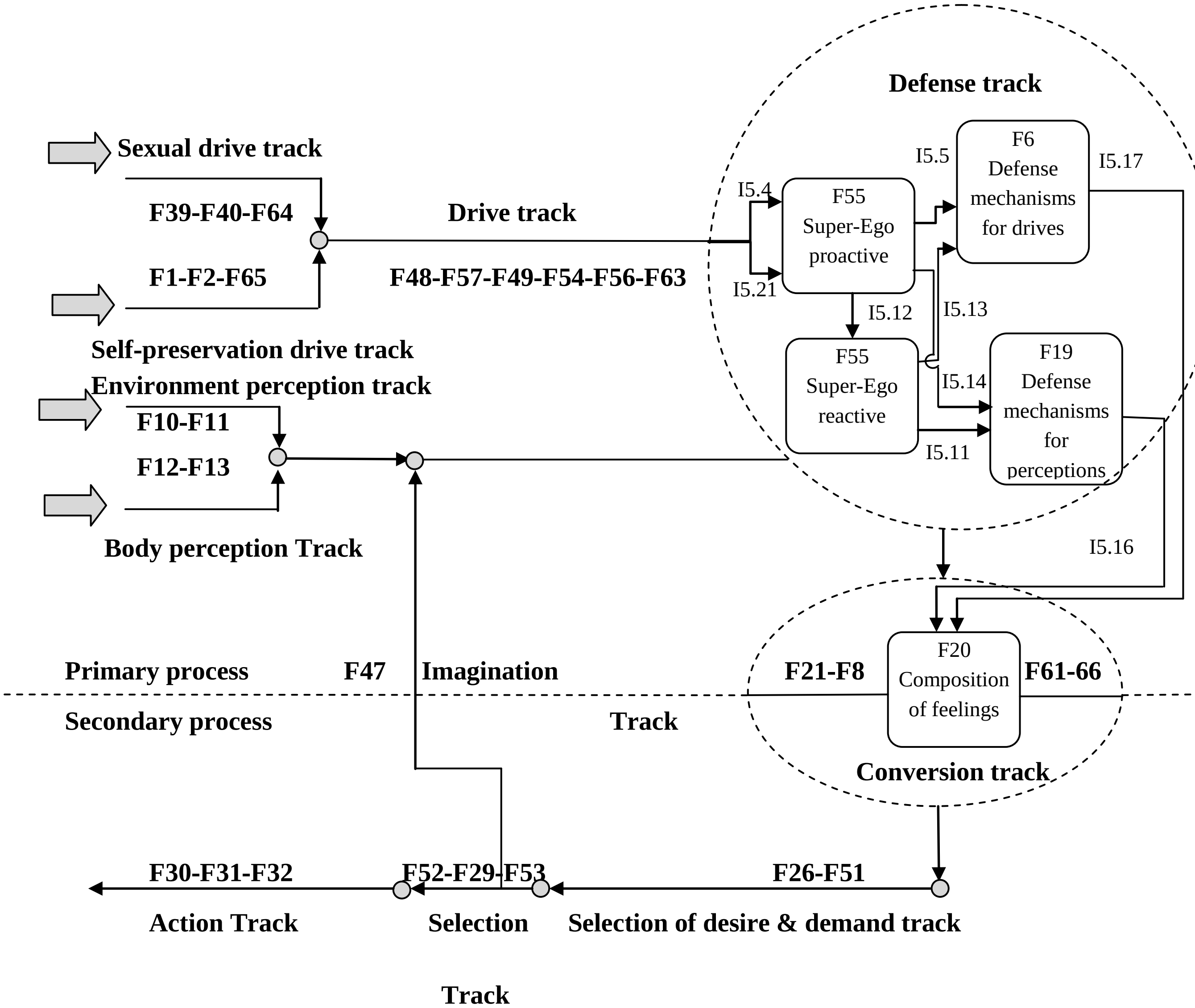


Figure 3.3: ARS Functional model

Sexual drive track

This track consists also of parts from all layers (1, 2 and 3) and includes three modules: F39 Seeking system, F40 Neurosymbolization of libido, and F64 partial sexual drives.

The seeking system assembles all the information based on the libido creation by various somatic sources and by erogenous zones and forwards them to F40, which convert them into neurosymbols. Similarly to F65 in the self preservation track, F64 also generates drives with the same components, and are divided into partial drives (DM), which are composed of oral, anal, phallic and genital.

Drive track

Drive track is composed of partial sexual drives F64, partial self preservation drives F65, accumulation of quota of affects for drive F48, memory traces for drives F57, primal repression for drives F49, and Emersion of blocked drive content F54.

The sexual drives and self-preservation drives are combined into a single list in F48 and the size of the reduction is calculated and stored as pleasure. In F57, drive object and drive aim for the generated drives are remembered as satisfaction options. The result of this process is the arrangement of the data structures involved as a tree, so that the drives are linked with the remembered objects and actions. The module F49 attaches to the content of drives, which are repressed by defense mechanism repression. They are forwarded to the module F54 which manages their contents over a different context to pass the defense mechanism.

The function of the module F56 is the generation of neutralized energy from quota of affect of the drive which is then provided to other modules.

Emotions are composed in module F63, which are one of the most important in the present work, where they have to be converted to word presentation and emotion anxiety triggered defense mechanisms. This is therefore, described in detail here.

According to Damasio emotions are a pre-stage to the “Feelings” and are built a level below conscious feelings, above life-regulation [report2]. In the ARS psychoanalytic model, there are two art of emotion: basic emotions such as anxiety, anger, sadness, love, elation, and saturation. The second type are complex emotions which are formed through defense mechanisms such as feeling guilt, sadness (depression), shame, pity, disgust, hatred (with object), love (with object), and envy.

An emotion is represented by a 4-dimensional vector. Those factors are unpleasure (Sum of all quota of affect), pleasure (decrease of amount of affect), the sum of all amounts of aggressive quota of affect, and the sum of all libidinous quota of affect. The following basic emotions are formed:

- Anger : Unpleasure + Aggression
- Sadness: Unpleasure + Libidinous
- Saturation: Pleasure + libidinous
- Elation: Pleasure + Aggression
- Anxiety: Unpleasure
- Joy: Pleasure

Feeling are created with the support of core consciousness from the emotions according to Damasio, they cause a specific somatic reaction.

Environment perception track

Sensor environment F10 and neurosymbolization environment F11 are arranged in the environment track, where the module F10 changes the physical or chemical variables from the environment into measurable values. The available data are multimodal such as sight, hearing, smell, touch, and taste.

The collected sensor data from F10 is forwarded to F11 and subsequently converted into neurosymbols in order to make it available to the mental apparatus.

Body perception track

Body perception track is covered by layers 1 and 2 and consists of sensors body F12 and neurosymbolization body F13. The module F12 is responsible for the recognition of the self image, for example, detecting the angular position of the arms, and the module F13 deals with converting the resulting data from F12 into neurosymbols.

Perception track

The Perception track consist of the following modules: external perception F14, memory traces for perception F46, primal repression for perception F37, emersion of blocked content F35, libido discharge F45, and composition of quota of affect for perception F18.

The module F14 generates a thing presentation from the neurosymbolic contents so that the body and environment sensors can be processed by the mental apparatus. F46 generates associations for the TPs produced in F14 with previously experienced and stored memory traces.

The modules F37 and F49 have access to the memory of repressed content, where F37 compares each drive from the primal repression with drive presented in perception and returns those which have sufficient matching.

The function of the module F35 is similar to F54 from the drive track and deals with changing the repressed content in the way that can be passed by the defense mechanism.

Module F45 is responsible for discharging libido by comparing the incoming perceptions with the memory, where libido discharge is a pleasure gain which is forwarded to F18. This module sums pleasure and unpleasure for all drives having the same type.

Defense track

The key aspect of the present work is the defense mechanism, the modules and their interfaces, which are involved, are presented in Figure 3.3.

The modules of the defense mechanisms in the ARS model operate in two ways, one processes the drives (F06) and the second one processes the perception (F19) and both of them are part of Ego.

Modules:

Defense mechanisms for drives F06: This module is responsible for the execution and the selection of defense mechanisms relating to the information and the conflict that F7 provide. F06 decides which drive representation can be preconscious and conscious. There are various options to process the drives in F06: forbidden drives contents are left in their original condition and will be passed or the forbidden drives contents with their quota of affect are changed or blocked by using defense mechanisms. The repressed drives are sent back to the module Emersion of blocked drive content (F54) [2]. The existing mechanisms in the current version of ARS are: intellectualization, reversal of affect, repression. The defense mechanisms for this module that have to be implemented in the present work are displacement, reaction formation, sublimation, turning against the self and projection.

Defense mechanisms for perception F19: Similar to F06 the main difference being the available data, where F19 defends against the forbidden perception and the forbidden emotions in the state of the drives like in F06. The difference is that the mechanisms used in F06 are not identical to the ones used in F19. Examples for these mechanisms are idealization and the depreciation which has to be implemented. The repressed perceptions are sent back to the module Emersion of blocked content (F35) [2].

Superego proactive F55: F55 represents the internalized rules like "Be always friendly to other People". The rule of F55 in the current version of ARS is, that the amount from quota of affect from the aggressive drives will shift to the libidinous drives.

Superego reactive F7: F7 control the incoming drives, perceptions, and emotions with regard to the internalized rules. If one of these rules makes a conflict, the drive, the perception, or the emotion is added to the forbidden drive list, the forbidden perception or the forbidden emotion list, where the forbidden drives are sent to F6 and the forbidden perceptions and the forbidden Emotion to F19.

Interfaces:

I5.21: The emotions list which is produced by F63 are sent to the module F55.

I5.4: The second incoming connection of F55 is a list of drives with data type drive mesh.

I5.12: This interface transports emotions and drives from F55 to F7.

I5.5 and 5.14: Not yet implemented.

I5.13: Forbidden drives and emotions generated in F07, with drives and emotions are transported to the module F6.

I5.11: Perceptions, emotions, forbidden perceptions and emotions are sent to F19.

I5.16: Emotions are forwarded by F19 to composition of feelings.

I5.17: This interface distributes quota of affects from F6 to composition of feelings.

I5.18: Drives and altered drives which are modified by F6 are forwarded to transformation to secondary process (drive-wishes) F8.

I5.15: Perceptions and altered perceptions done by F19 are sent to transformation secondary process (perception) F21.

Conversion Track

The conversion track is the passage to the secondary process where data structures are converted and extended with the data structures of the secondary process. This track consist of the following modules: Transformation to secondary process (Perception) F21, composition of feelings F20, Transformation to secondary process (drive-wishes) F8, semantic and syntactic speech ability F66, and localization F61.

The module F20 the composition of the feeling has to be extended in the present work. It does not to be understood at this stage what this feeling means what it has triggered, etc. But it is important to convert the emotion that it can be processed in the secondary process, which affects, later the decision-making. Here in this module, the quota of affect amounts that are (possibly in modified form) transmitted by the defense mechanisms has to be perceived as feelings after a transformation has taken place in the secondary process. They are connected to word-presentations [2].

Interfaces:

I5.17: Quota of affect is sent from F6 to F20.

I5.16: Emotions are transported from F19 to F20.

I6.2: the composed Feelings are transported to the modules semantic and syntactic speech ability F66, decision making F26, and evaluation of imaginary F29.

I6.5 and I6.4: Not yet implemented.

The transformation of perceptions is done by searching for each perceived image in the memory of the secondary process and associations are made. The incoming drives are converted into form of drive wishes which can be processed in the secondary process.

Selection of desire and demand

The selection of desire and demand is done by the module decision making F26 and Reality check wish fulfillment F51. In F26 the reality, drives, and superego together build the consequence of a decision which can be used as motive for an action. The basic knowledge about the external reality is processed in F51 to select a desire.

Selection track

Different alternatives of implementation and achievement of action has to be prepared before it is carrying out. This set of imaginary action is made in the module generation of imaginary action F52 and they are evaluated in the module evaluation of imaginary action F29

Action track

In the action track the abstract action plans will be filtered out and the composite actions such as flee, search etc. will be resolved.

3.2 Model of Defense Mechanisms

As illustrated in Chapter 1 that the method used in the present work starts with the required literatures, which contain the description of defense mechanisms. In this subsection different defense mechanisms are explained and analyzed using human life examples. Furthermore, the models of different defense mechanisms and composition of feelings are presented.

3.2.1 Description of Defense Mechanisms

Psychoanalysts define human behavior as a compromise of a conflict between the Ego and the Id. To avoid undesired drive demands, the ego uses defense mechanisms [Wer05, p. 17]. The Term Defense mechanisms were used the first time by Freud and after his death they are investigated further by his daughter Anna Freud (1963) [Fre48].

According to Freud, the various defense mechanisms are to be deployed against the drive, while Anna Freud believed that the defense mechanisms were a defense against the drive, emotion, affect and the superego-Anxiety [Stu09, p.5].

The defense mechanisms, which Anna named in her book “The Ego and mechanisms of defense”, such as: repression, regression, reaction formation, isolation, denial, projection, introjection, turning against the self, reversal, displacement, intellectualization, and sublimation are discussed here.

Repression: The drive wishes are blocked and pushed into the unconscious. For example, someone forgets the name of a man, by he has just annoyed [Kön96, p.19].

Regression: A revertion to a previous level of psychological development. This is usually in response to stressful situations, for example, because of a threat to life or health [Kön96, p.87].

Isolation: The idea or imagination is isolated from the associated affect. For example, I’m angry with a friend and I want to hurt him, but because I like my friend, I isolate the affect “anger” from the idea “friend”.

Denial: The denial of reality means that the extent of this reality is not recognized. For example, a hiker wants to reach his goal, although there is a sign of a storm. The hiker thinks "The storm will only be there when I've reached my goal" [Kön96, p.40].

Projection: All the individual unacceptable affects, moods, and impulses are projected outwards. For example, someone projects his aggressive moods on others because he does not want to experience them [Kön96, p.47].

Introjection: Introjection refers to the construction of an internal representation of a person [Kön96, p 23]. Freud explains this term, as a projection toward. "Good" is taken in the interior, "bad" is projected outwards [Stu09, p.331].

Turning against the self: Aggressive impulses are set against the ones own Ego and don’t satisfy the original drive object. The depression serves as example [Kön96, p.32].

Reversal: reversal of a drive mechanism transforms the goal of drive form active to passive and/or may be to opposite.

Displacement: Here the drives are moved to another object or a person. For example, children, who do not dare to release their anger on the parents, move this anger onto their stuffed animal [Mye04, p.594].

Reaction Formation: In the reaction formation unacceptable unconscious drives impulses are changed into their opposites, e.g. an unacceptable unconscious "I hate him" becomes a conscious "I love him" [Mye04, p.594].

Sublimation: Anna Freud called sublimation as the most mature defense mechanism, because the unconscious impulses are allowed in a socially acceptable manner. The drive’s aim changed to socially desirable behavior [Malt11, p.89].

Intellectualization: This defense mechanism makes the drives accessible and manageable through close association with the imagination content [Ana, p.126].

Idealization: This defense mechanism separates the good from the bad properties of an object or person. For example the good mother of a baby is different to the bad mother [GBD+11, p. 5].

Depreciation: It does the opposite of idealization [GBD+11, p. 5].

3.2.2 Modeling Defense Mechanisms and Composition of Feeling

According to discussed information in the previous sections in this chapter, the following defense mechanisms such as displacement, reaction formation, sublimation, turning against the self, projection, depreciation, and idealization have to be modeled and implemented. In addition the module of the composition of feeling for ARS has to be extended.

To be able to realize these tasks, a technical model has to be developed. In this session, models and the possible realizations of the five defense mechanisms and the module of composition of feeling are presented. The modules are bordered by a block with a double line.

In contrast to other defense mechanisms displacement, reaction formation, and sublimation used an association memory to change the original drive object or original drive aim. The reason for choosing association memory is that they are considered such as personality parameters. Looking back to the human life, each person acts differently due to the defense mechanisms.

As discussed in the Section 3.1.2, there are two different modules of defense mechanisms: one for drives and the second for perceptions, as shown in the Figures 3.4 and 3.5. The inputs of the first module are a list of forbidden drives and emotions using interface I5.13, where the output is a list with altered drive and quota of affect using interfaces I5.18 and I5.17. The second module contains inputs such as list of forbidden perceptions and forbidden emotions (I5.14), and as output a list with altered emotion (I5.16) and perception (I5.15).

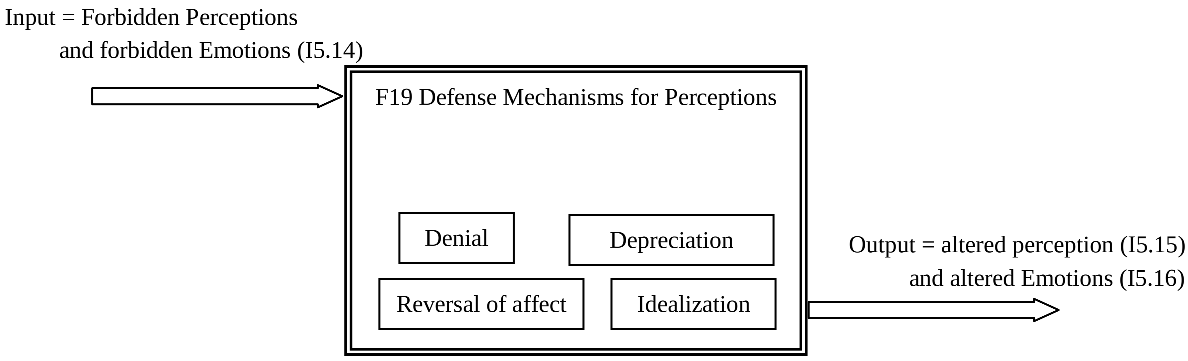


Figure 3.4: Module F19 defense mechanisms for perception

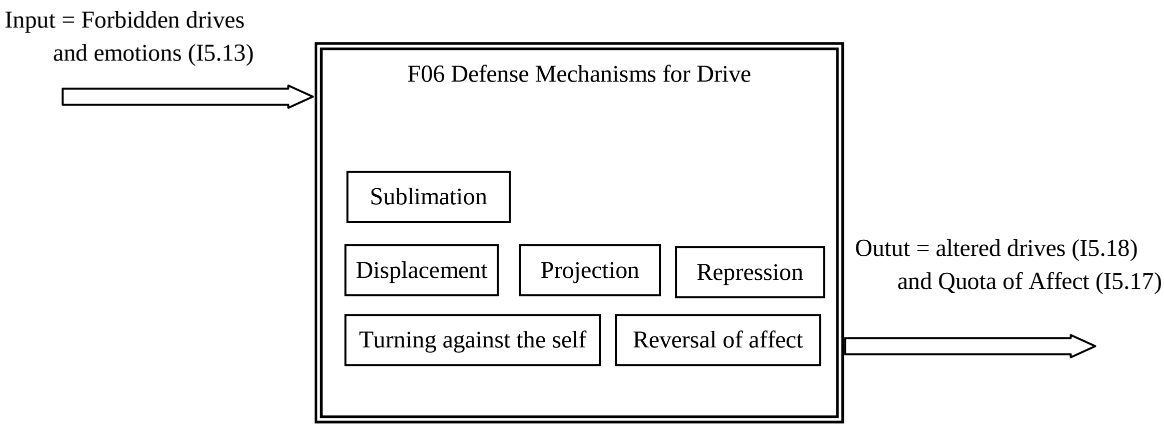


Figure 3.5: Module F06 defense mechanisms for drives

Displacement

As discussed in Section 3.2.1, displacement uses another drive object to satisfy the drive aim. For example, this defense mechanism can be used in multi agent system to protect social rules or to replace drive object which does not exist in the environment with the perceived object.

Sublimation

Applying sublimation in multi agents systems, allows agents to function culturally acceptable ways. Sublimation helps agents to react without violating the social rules. Therefore, the inacceptable drive aim has to be changed into acceptable one.

Reaction formation

Reaction formation changes the original drive aim to its opposite. This defense mechanism can be used to protect agents from danger or acting incorrectly.

Turning against the self and projection

In Section 3.2.1, Turning against the self is explained so that the original drive object will be not satisfied and the drive aim is set to the own Ego. Applying this defense mechanism to the agents can cause trouble. Therefore, the drive aim has to be checked and if the situation causes danger to the agent defense mechanism projection will be used, which projects all unacceptable drive aim outward.

Idealization and depreciation

Idealization and depreciation can be used to the agents to evaluate the properties of perceived object, where the idealization separates the good from the bad properties of an object and depreciation does the opposite.

Summary

After all the information discussed before, which explain how defense mechanisms can be applied to multi agent systems, it is possible to summarize the effect of these mechanisms on data used in ARS project. The table 3.1 shows how drives or perceptions should be altered.

Table 3.1: Effect of defense mechanisms on data used in ARS

Defense mechanisms	Effect
Displacement	Drive object
Reaction formation	Drive aim
Sublimation	Drive aim
Turning against the self	Change drive object to self
Projection	Drive object self
Idealization	Evaluate the positive association attributes
Depreciation	Evaluate the negative association attributes

Overview of defense mechanisms for drives

An overview of defense mechanisms by applying an example can be introduced where the Figure 3.6 shows how defense mechanisms can be worked for the available simulation and agents. It is assumed that the input forbidden drive consists of the following components: drive aim is to "eat" and drive object is "carrot". As discussed in Section 3.1.3, displacement replace drive object to another one, this is done by changing the "carrot" to "schnitzel". Reaction formation modifies the original drive aim to its opposite, in this example the aim "eat" is changed to "sleep" where no energy is consumed. By defense mechanism sublimation, the drive aim is altered to a socially acceptable aim. As seen in the Figure 3.6, the drive aim "eat" is replaced by "share the food".

The defense mechanism turning against the self changes the drive object to the object self and projection as discussed in [GBD+11, p. 5]is operated in such way so that, if the original drive source is the self, this is replaced by the original drive object and the new drive object gets the object self for example:

- Drive before projection

  Drive source = "SELF", drive aim = "BEAT", and drive object = "OTHER AGENT"

- Drive after projection

  Drive source = "OTHER PERSON", drive aim = "BEAT", and drive object = "SELF"

Since the source of drive in the ARS project can be an organ and/or a somatic process as discussed in Section 3.4, it is not possible to realize the projection in this approach, with the self as source of drive.

In the work at hand, projection is defined such that the unacceptable drives are ascribed to the outside. For example if drive contains the aim eat and object self defense mechanism projection changes the drive object self to another drive object. In the example of the Figure 3.6, projection is used to protect the agent from defense mechanism turning against the self, because it is unacceptable that the agent hurts himself.

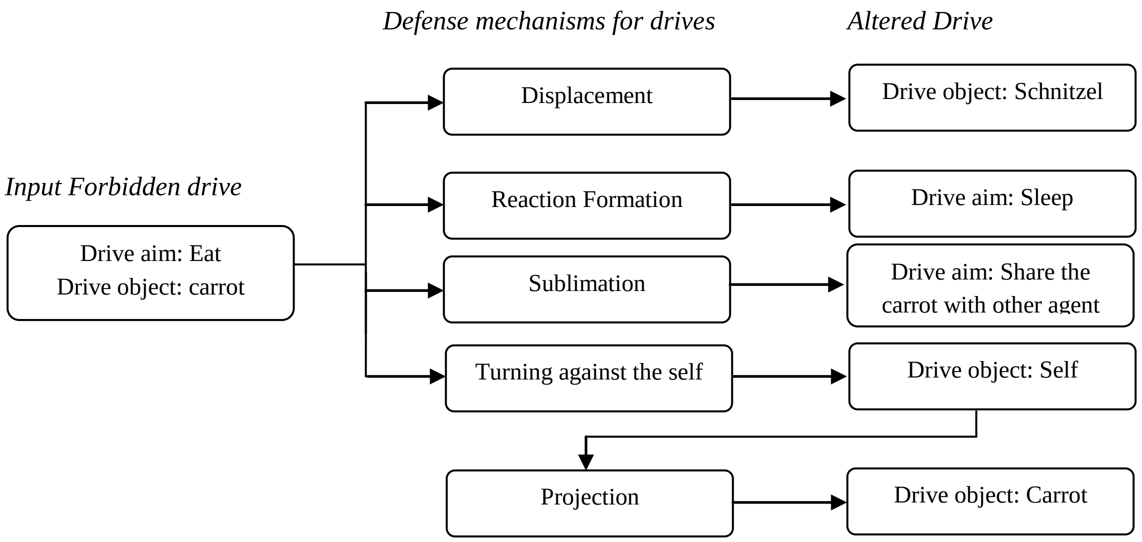


Figure 3.6: Overview of defense mechanisms for drives

The Figure 3.7 shows an example for defense mechanisms depreciation and idealization, where the perceive object is a “cake”. As discussed in session 3.1.3, idealization separates the good from the bad properties and depreciation does the opposite. In the Figure 3.7 present how the both defense mechanisms affect the perceived object, where idealization create just the good attribute from the cake and depreciation create just the bad attributes.

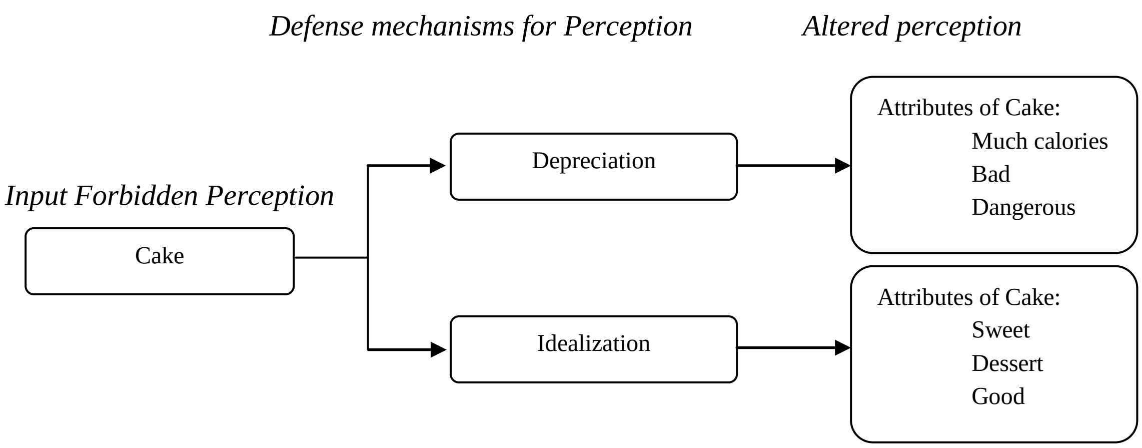


Figure 3.7: Overview of defense mechanisms for perception

Module F20 Composition of Feelings

At this stage, what this feeling means and what it has triggered does not need to be understood. But it is important that the emotions on the primary process are transformed to secondary process by converting them into data type word presentation mesh. Additional quota of affect of drive that are (possibly in modified form) transmitted by the defense mechanism reversal of affect in the module F06 are now perceived as feelings. Those feelings also affect the later decision making.

The module F20 Composition of Feelings is the passage to the secondary process. As shown in the Figure 3.8 a general model is presented where the inputs consist of lists of emotions using interface I5.16 and the quota of affect for drives using interface I5.17. The transformation to WPM is composed of two functions one to connect the quota of affect to emotion and the second to convert all the emotions to WPM.

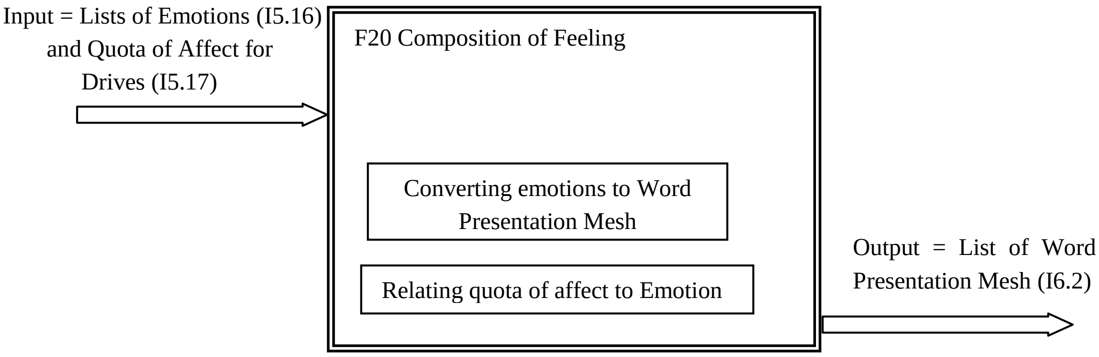


Figure 3.8: Model of composition of feeling

3.2.3 Test Process

To test the defense mechanisms, the following functions are implemented: one to indicate the time duration of the activation of each defense mechanism, the second one is to calculate the number of altered drives, and finally function to view the drives.

The function of time duration is calculated by simulation steps and the function of calculation of altered drives is done after each manipulation of the forbidden drive. The aim of this test is to control the delay or the lost drive without any change during each defense mechanism, where the number of the simulation steps has to be equivalent to the number of altered drives as long as defense mechanism is active. Further, the forbidden drive and the altered drive have to be viewed in the simulation platform in order to visualize the manipulation of drive.

The Figure 3.9 shows, how the test process of each defense mechanism operates. This is a process on running system with intention to find errors. The activation of defense mechanism followed by activation of its time chart and the forbidden drive has to be viewed. Further, each call of changing the drive succeeds by increasing the number of altered drive and the component of the new drive is viewed.

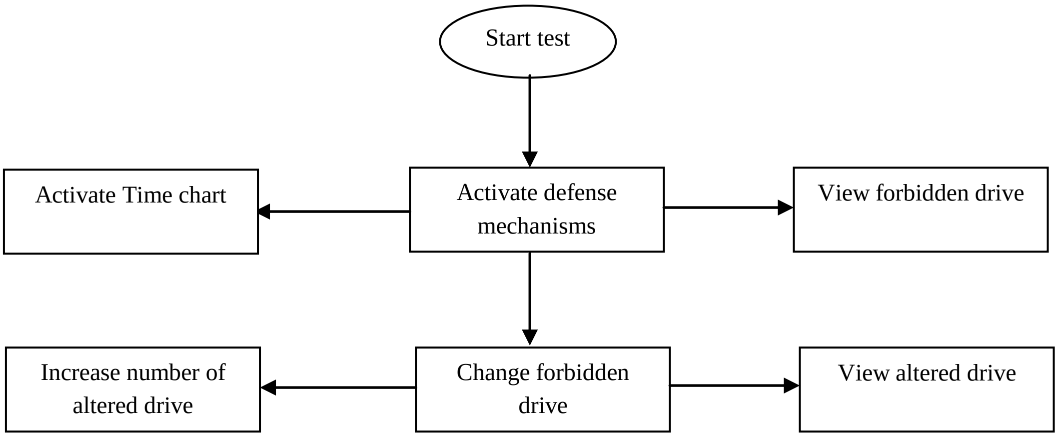


Figure 3.9: Test process of defense mechanisms

4. Technical Realization

Based on the presented general models and modules that are discussed in the Chapter 3, in this chapter a technical realizations of them are demonstrated, where Java is used as programming language. The Modules F06 defense mechanisms for drives, F19 defense mechanisms for perceptions, and F20 composition of feelings have their own Java classes. At the start, the structure of the project and the used models with their interfaces are presented. Further, the triggering of defense mechanisms and their classifications are demonstrated, as well as their justification. The used steps for each defense mechanism for the manipulations of drives or perceived objects are described and illustrated. The realization of the composition of feelings is presented where the methods and steps for the conversion from emotion into word presentation mesh and the relation between the quota of affect for drive and emotion are confirmed. Finally, the additional inspectors for the modules of defense mechanisms are presented which consist of bar chart and time chart. They are used to view and to test the correctness of each defense mechanism. The bar chart shows the number of the altered drives and the time chart presents time duration.

4.1 Project Structure and Models

As monitored in the Chapter 1, ARS project was implemented and not enough defense mechanisms modules were implemented. Before the technical realization is demonstrated, an overview about the implemented software architecture of ARS is introduced where the structure of project and the used module with their interfaces are presented.

ARS software architecture consists of seven sub-projects in order to be independent from each other. As seen in the Figure 4.1, the control architectures in the decision unit project are separate from multi agent framework MASON. The sub-project simulator is responsible for world creation. The definitions of each entity such as sensors, bodies, and interaction possibilities is implemented in sub-project world, which has two interfaces to physics engine provided by MasonPhysics2D. The classes which are needed for decision unit are implemented in sub-project DUInterface. MASON provides several inspectors to view internal state of the agents. In the project world, body and entity inspectors are stored. The necessary inspectors for the work at hand are stored in DUInspectors, which view the internal state of defense mechanisms and the implemented bar chart and time chart.

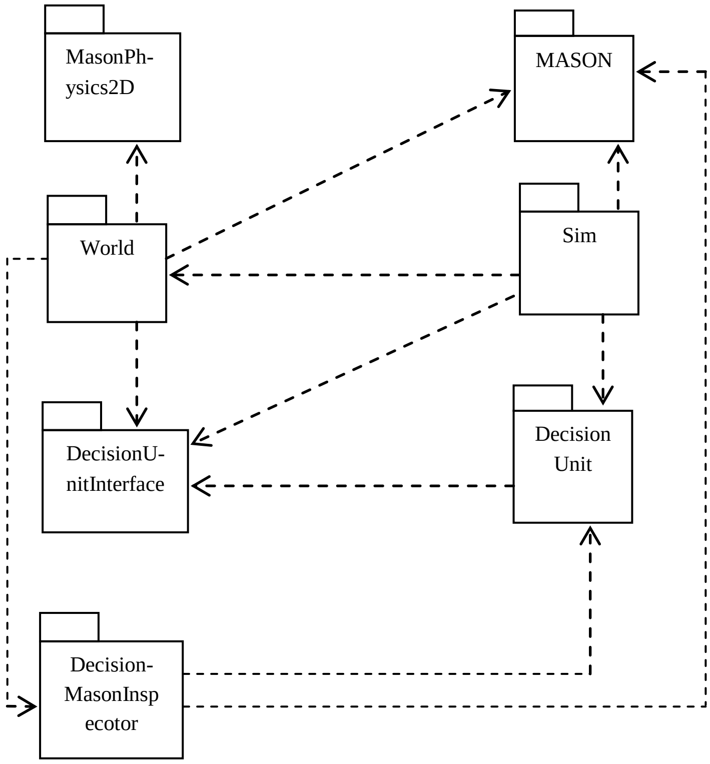


Figure 4.1: Project structure

Based on the introduced information in Section 3.1.2, the modules of defense mechanisms, superego, and composition of feelings are shown in the Figure 4.2 with their interfaces in UML. They are placed in the project DesignUnit.

The module F06 (defense mechanisms for drives) gets its data from F07 (Superego reactive) via I5.13 which contains Forbidden drives and emotions and sends the processed data via interface I5.18 to secondary process to the module F08 (Conversion to secondary process for drive wishes). Furthermore, the quota of affect is sent via interface I5.17 to the module F20 (composition of feelings). The module F19 (defense mechanisms for perceptions) gets its data via interface I5.11 from module F07 and sends resulting data to module F21 (Conversion to secondary process for perception). Additional to the discussed interfaces, there are also interfaces that are implemented in the present work and are used to transport data to the inspectors of MASON. As seen in the Figure 4.1, the interfaces for F06 are itfInspectorBarChartforF06 and itfInspectorCombinedTimeChart. they contain data for Bar Chart and Time Chart. The module F19 includes also the two interfaces itfInspectorBarChartforF19 and itfInspectorCombinedTimeChart. This is described later in this Chapter.

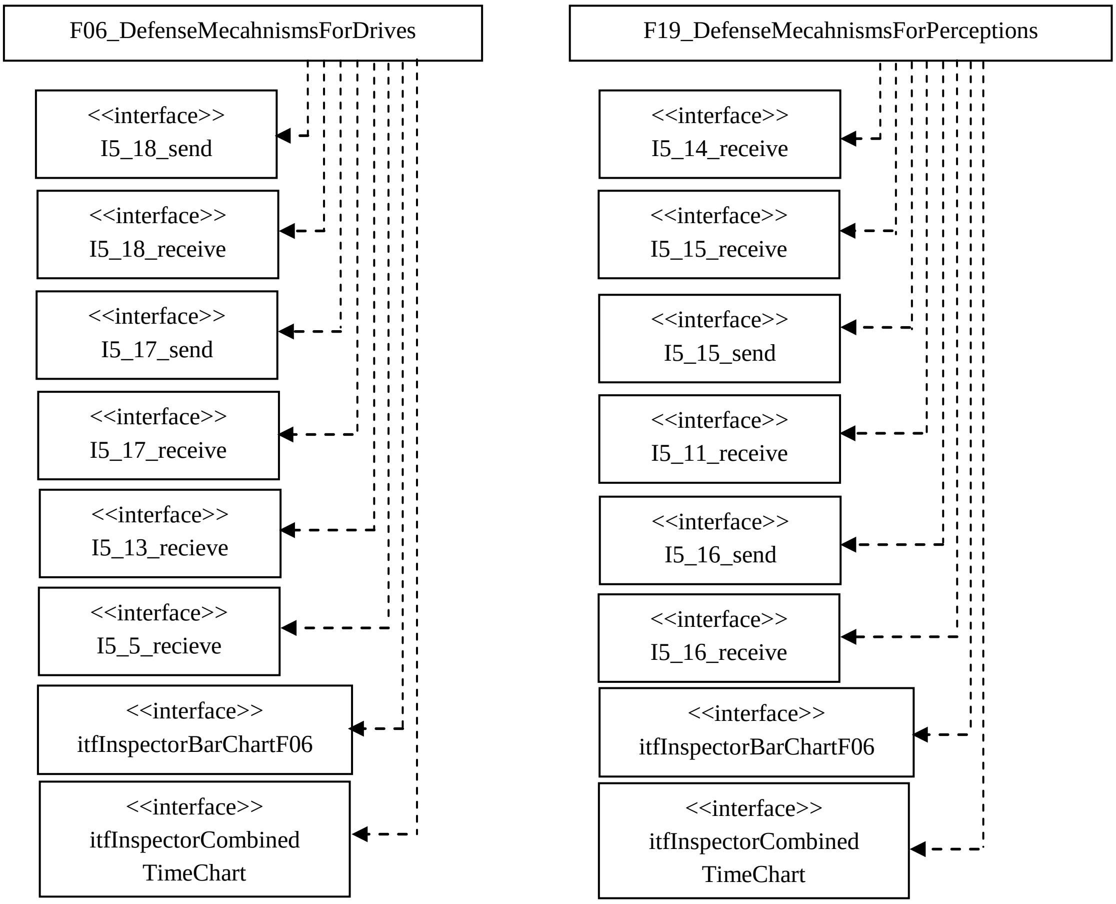


Figure 4.2: Models of defense mechanisms and their interfaces

As seen in the Figure 4.3, Module F20 receives its data, which consist of emotions, from F19 via interface I5.17 and the module F06 send the quota of affect via I5.16. The Modules F 8 (Conversion to secondary process for drive wishes) and Module F21 (Conversion to secondary process for perception) are sending data to F20 via interfaces I6.4 and I6.5. The processed data is forwarded to modules F26 (Decision making) and F29 (Evaluation of imaginary Actions)

The rectangles (F06, F19, and F20) in the Figure 4.2 and in the Figure 4.3 present abstract classes. The <<interface>> rectangle is an interface class. The arrow dashed defines that Class F06 implements interfaces.

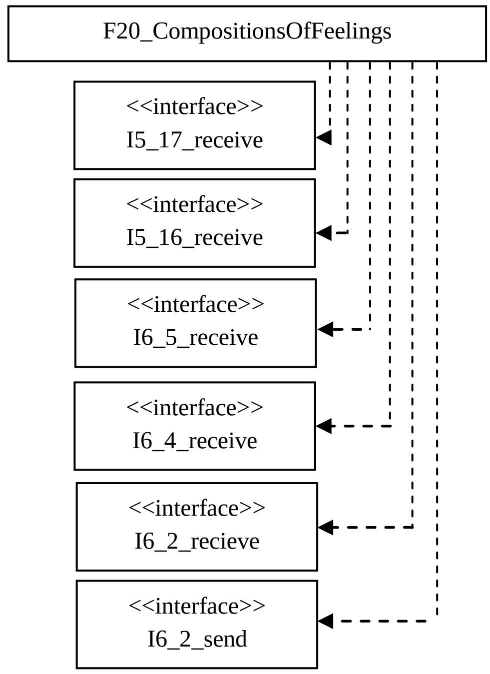


Figure 4.3: Model of composition of feelings and its interfaces

4.2 Triggering of Defense Mechanisms

The triggering of defense mechanisms is dependent on the intensity of anxiety and the quota of affect of drives in module F06 (defense mechanisms for drives). The amounts of the two factors are divided into intervals, where the selection of the intervals is arbitrary. The important aspect is choosing the defense mechanism for each interval as it is expected that with increase in the magnitude of those factors increases the strength of defense mechanisms. It is assumed that the protection grows with increasing the strength of defense mechanisms.

Table 4.1 describes the dependencies of defense mechanisms on drives, where the intervals of the intensity of anxiety and the quota of affect with their assigned defense mechanism are presented.

As seen in the Table 4.1 the classification of the implemented defense mechanisms is done in this sequence with the consideration of their strength:

Turning against the self / Projection → Reaction formation → Sublimation → Displacement

The reason for choosing turning against the self / projection as the weakest defense mechanisms because of the drive object is displaced to the self. This mechanism can cause a trouble to the agent in case of drives aim such as EAT or BITE. To control this defense mechanism projection is used. Displacement manipulates the drive object into a new one. This summarized that the meaning of the drive by displacement is changed more than sublimation. By reaction formation, the drive aim is taking its opposite, which referred to the primitive defense mechanism because the meaning of the drive is totally changed.

Table 4.1: Dependency of defense mechanisms on drives

Quota of affect	Intensity of anxiety	Defense mechanisms
0 < Quota of affect <= 0.5	0 < Anxiety <= 1.0	Turning against the self and projection
0 < Quota of affect <= 0.5	1.0 < Anxiety <= 1.3	Reaction formation
0.5 <= Quota of affect < ∞	1.3 < Anxiety <=1.4	Sublimation
0.5 <= Quota of affect < ∞	1.4 <Anxiety <= 1.6	Displacement

Similar to the module F06 the module F19 defense mechanisms for perceptions are depending on only intensity of anxiety. As shown in the Table 4.2, the classification of defense mechanisms is presented in this sequence:

Idealization → Depreciation

Table 4.2: Dependency of defense mechanisms for perceptions

Intensity of anxiety	Defense mechanisms
0.7 < Anxiety <= 0.9	Idealization
0.9< Anxiety <=1.2	Depreciation

Depreciation and idealization split the perceived object into two representations bad and good. The defense mechanism idealization is used when the intensity of anxiety is lower, in comparison, to depreciation in order to protect the agent from the negative properties of the objects.

4.3 Defense Mechanisms

For defense mechanisms displacement, sublimation and seaction Formation an association memory is used which is called in java “HashMap”. The original drive objects and aims are stored as keys and their values are the desired change after applying the defense mechanism. HashMap capacity should not be overstepped because the time for the iteration over collection is proportional to the capacity.

Model of Displacement

As mentioned above, that due to the defense mechanism of displacement the drive objects move to other objects. The original objects are stored as keys in associative memory and their corresponding associated values are the shift objects. Table 4.3 shows example of original drive objects and their shift objects in case of drive aim “EAT”.

Table 4.3: Association memory for displacement

Original drive object	New drive object
Bodo	Schnitzel
Stone	Schnitzel
Wall	Schnitzel

To apply displacement, original drive objects have to be retrieved from the forbidden drive which contains several pieces of information. After that, the original object has to be replaced by its desired object. The Figure 4.4 shows the steps that are taking for displacement. After the defense mechanism has been activated, the forbidden drive is received and the following functions are followed:

- Retrieve drive object: Return the original drive object from the forbidden drive which has to be replaced.
- Search for drive object: return true or false, if the original drive object exists in the HashMap. After, the value of the original drive object will be returned from the HashMap which is the new drive object.
- Create TPM for new drive object: Creates data structure TPM for the new drive object.
- Set TPM into DM and to output: Set the new drive object into the forbidden drive and create new DM to be set to the output.

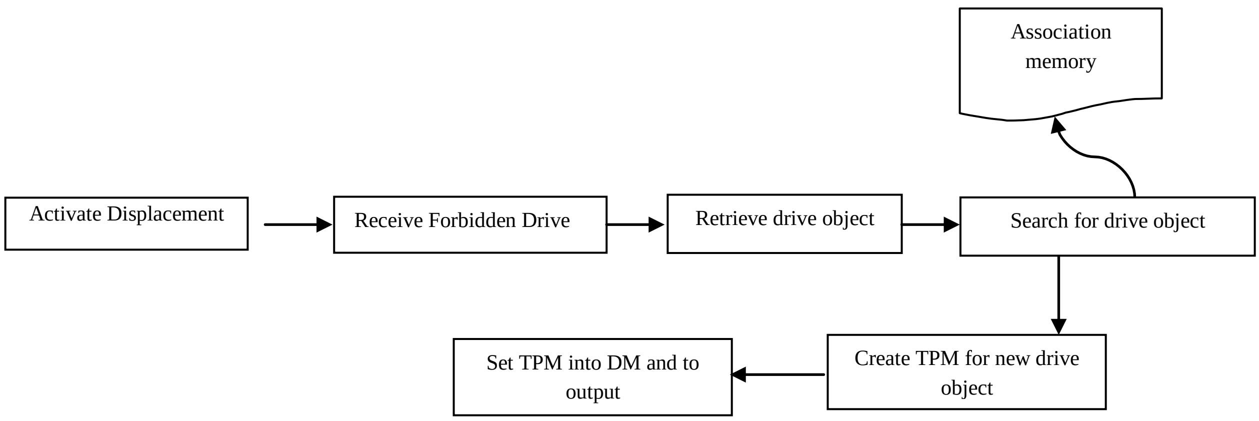


Figure 4.4: Process of the model of displacement

Models of Reaction Formation and Sublimation

On the contrary of displacement, the defense mechanisms reaction formation and sublimation the change drive aim. As discussed in the session 3.1.3, the reaction formation changes the drive aim to its opposite and sublimation modifies it to a social one.

The Figure 4.5 shows the model of reaction formation and sublimation, since the both of them have to change the drive aim, they share the same function. After the activation of reaction formation or sublimation, the association memory will be created and the function to change drive will be called. The input of the model is the forbidden drive, it is forwarded with the created association memory to the function to change the drive aim.

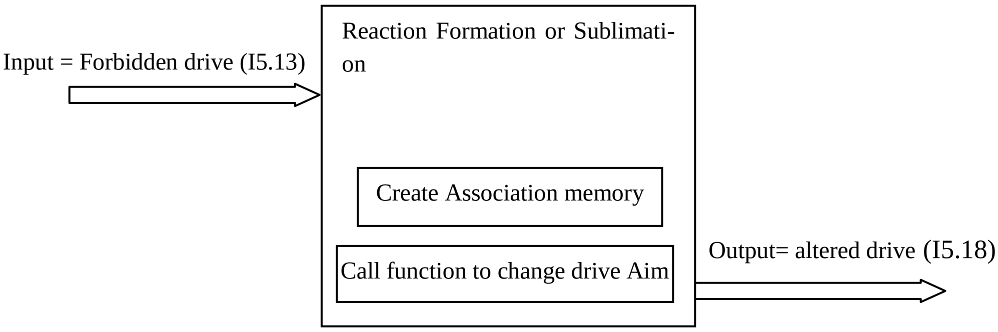


Figure 4.5: Model of sublimation and reaction formation

Similar to defense mechanism displacement, just here drive aim has to be stored in the association memory. Table 4.4 presents examples of original drive aims and their desired values for reaction formation and sublimation.

Table 4.4: Association memory for reaction formation and sublimation

Original drive aims	Desired drive aim for reaction formation	Desired drive aim for Sublimation
Eat	Sleep	Share the food
Bite	Nourish	Hug
Nourish	Bite	Taste the food for others

The Figure 4.6 presents steps that are taking to change drive aim. This steps is used for all defense mechanisms which should change the drive aim, such as intellectualization, sublimation, and reaction Formation. After the call of the defense mechanisms the association memory is created and the forbidden drive is received. The following processes are used during defense mechanisms reaction formation and sublimation.

- Retrieve drive aim: Return the original drive aim from the forbidden drive which has to be replaced.

- Search for drive aim: Return true or false, if the original drive aim exists in the HashMap of the current defense mechanism. The result is the value of the original drive aim from the HashMap of the current defense mechanism.
- Create TPM for new drive aim: Creates data structure TPM for the new drive aim.
- Set TPM into DM and to the output: Set the new drive aim into the forbidden drive and to the output.

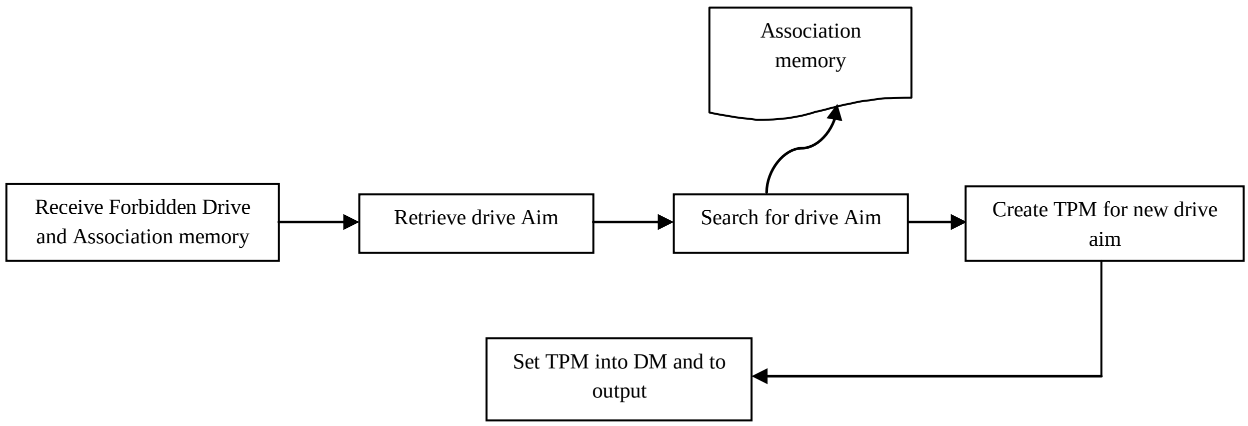


Figure 4.6: Process of changing the drive aim

Idealization and Depreciation

The defense mechanisms depreciation and idealization are used to evaluate the quality of perceived object on the basis of its attributes, where by idealization deletes all the negative attributes and depreciation does the opposite.

Since defense mechanism modules are placed in the primary process, the existing data structure is thing presentation. As discussed in the session 3.2.1, there two types of associations in the primary process: association temporary and locally. The temporally associations are dependent on the time and the locally associations define the properties of data. The both defense mechanisms deal with the bad or the good properties, therefore the locally associations have to be applied.

The Figure 4.7 shows the general model of the both the defense mechanisms. After the activation of idealization or depreciation, a list of negative or positive association attributes of the perceived object will be created. It is depending on which mechanism is called. For Depreciation the positive parameters are created and for the Idealization the negatives. Next function removes all the specified parameters.

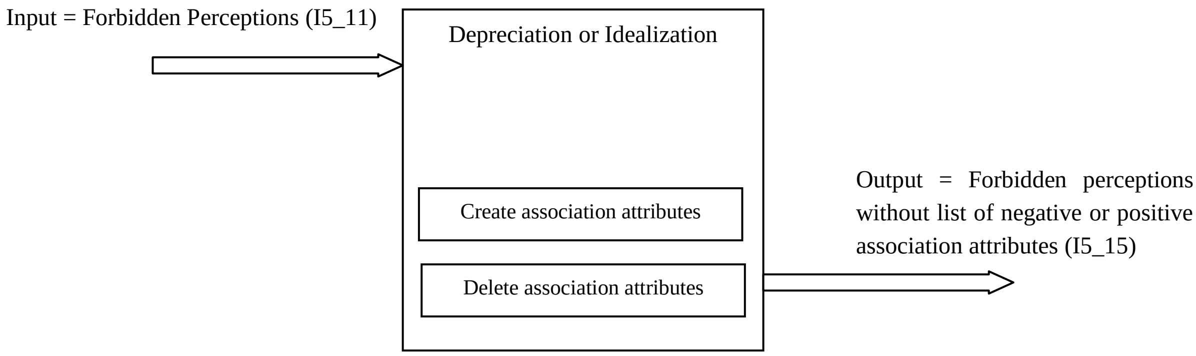


Figure 4.7: Model of depreciation and idealization

As shown in Figure 4.8, to create the association attributes of the perceived object which has data type TPM, a list with thing presentations (TPs) with their negative or positive association attributes has to be generated. Finally, the TPs are related to the TPM over their associations.

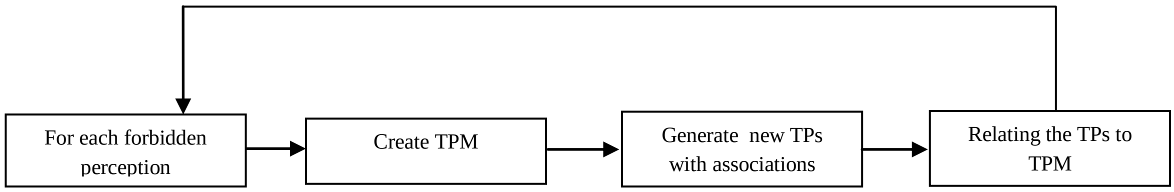


Figure 4.8: Model of creating association attributes to the perceived object

The function to delete the association parameters of the object is presented as a flowchart in Figure 4.9. After the reception of the forbidden perception object and the association attributes, the root of the associations will be checked whether they are equivalent to the object. If they are, all the associations will be deleted. It depends on which defense mechanism is activated. The depreciation removes the positive association parameters and idealization removes the negative association parameters.

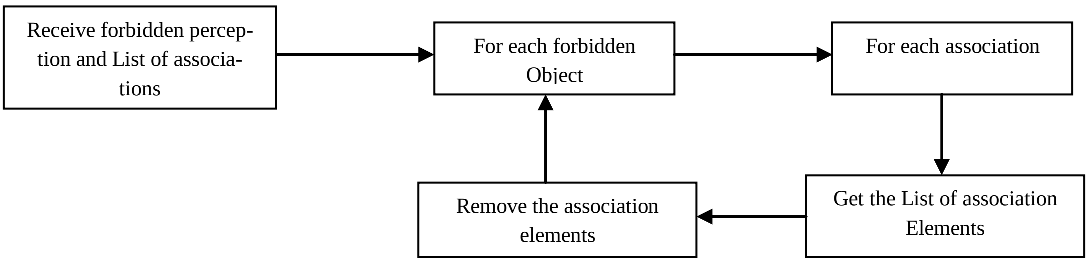


Figure 4.9: Process of deleting the association attributes

Turning against the Self and Projection

As explained in session 3.1.3, turning against the self causes such a situation like depression or hurting the self. Therefore, projection is used to control the result of turning against the self so that the agent does not destroy itself, for example, in case of drive aim such EAT or BITE. The Figure 4.10

represents the flowchart of both defense mechanisms. The following steps are used during Turning against the self and projection:

- Retrieve and store drive object: After the activation of Turning against the self, the drive object are retrieved and stored in order to be used for defense mechanism projection in the situation where the drive aim is "EAT" or "BEAT".
- Generate TPM for "SELF": To be able to add new drive object self to DM, data type TPM has to be created.
- Check drive aim: Check drive aim. For example, drive aim such as "EAT" or "BEAT". It is not the case, the drive object self with data type TPM will be set to DM and to the output.
- Function of projection: defense mechanism projection is called to project the agent from hearting itself. The stored drive object is set to the TPM and to the output.

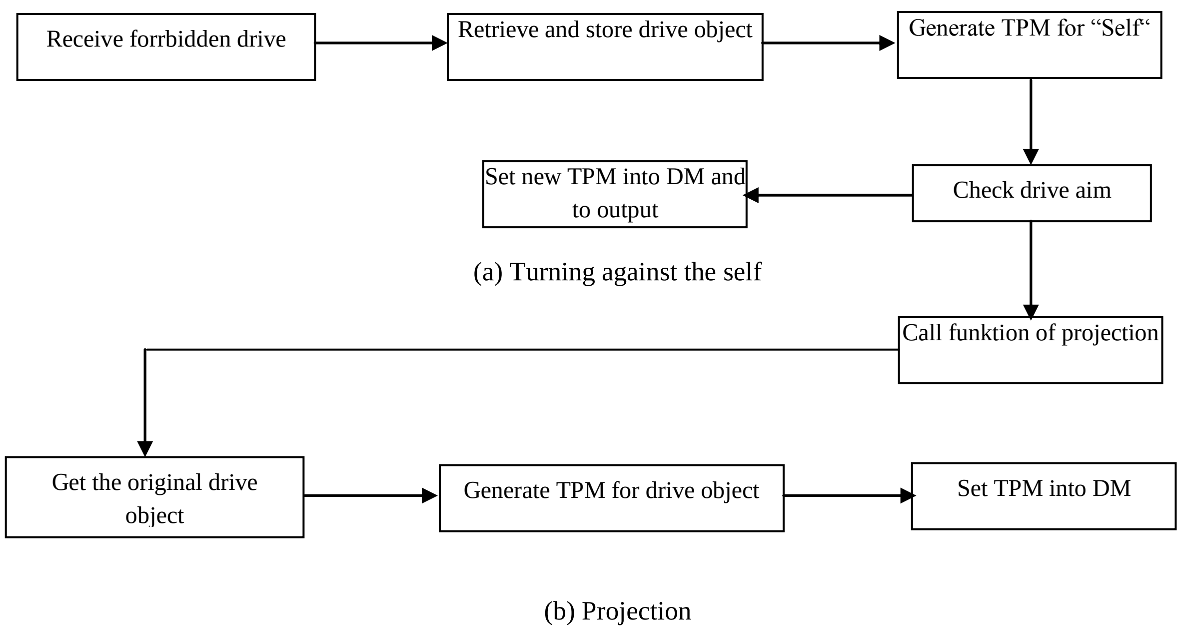


Figure 4.10: Models of turning against the self and projection

4.4 F20 Composition of Feelings

To compose the feeling, each emotion gets one WPM. As discussed in session 3.5, emotions in ARS are determined by four factors: Pleasure, unpleasure, aggression, and libidinous. All of these factors are converted to WPs and related to WPMs over association secondary.

The quota of affect of drive from the module F06 (defense mechanisms for drives) effects the anxiety by increasing its intensity.

Table 4.5 presents the dependency of quota of affect of drive and the intensity of anxiety. The intensity of anxiety increases by increasing the quota of affect. This is done by dividing the amount of Affect into intervals and for each interval, the intensity of anxiety have specific value.

Table 4.5: The dependency of quota of affect of drive and anxiety

Amount of affect	Intensity of anxiety
0< Amount of affect <= 0	Intensity of Anxiety + 0.1
0.3<= Amount of affect < 0.7	Intensity of Anxiety + 0.2
0.7 <= Amount of affect < ∞	Intensity of Anxiety + 0.3

Relating quota of affect to emotion is done by increasing the intensity of anxiety in relation to its amount, after that the call of the function to convert the emotion executes in order to get the new value of the intensity of anxiety. The emotions are converted so that each emotion gets its own WPM. This process, as shown in Figure 4.11, involves the following steps:

- Creation WPM for each emotion: For each emotion, data type of word presentation Mesh os created in order to have for each emotion, one feeling.
- Converting of emotion factors to WP: Creates for each factor of emotion one word presentation
- Relating the WP to WPM: Relating each word presentation which presents the factor of emotion to Word Presentation Mesh Feeling over association secondary.

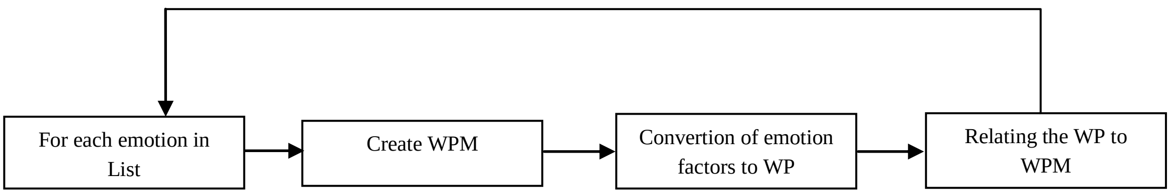


Figure 4.11: Flow chart of converting emotion to WPM

4.5 Additional Inspectors for Defense Mechanisms

An inspector is a graphical user interface (GUI), it provides to user facilities such as to inspect, track, and modify certain model object, value or the property of the object.

MASON provides a collection of inspectors to look into the internals states. The state presents relevant information about the agent during the simulation, such as the information about the agent entity, body, brain, ARSIN overview, and debug actions. The most used inspectors in the present work during the test and evaluation of tasks is the information about the brain, where various modules of ARS project with their input and output (drives, emotion, commands etc.) are viewed.

As discussed in the section before, the time chart and bar chart have to prove the result of defense mechanisms. Further, the forbidden and altered drives or perceptions have to be viewed in order to show the manipulation that caused by defense mechanisms.

To prove the correct processing of defence mechanisms a debugging tool in the form of inspectors is proposed which are classified into two groups. The first summarizes data and their textual content with Hypertext Markup Language for every module input and output. The second group uses charts to visualize the time duration of defence mechanisms and number of altered drives or perceptions while the defence mechanism arts, this is done by utilizing the open source library JFreeChart.

The following steps are implemented to set data to inspectors in the two modules of defense mechanisms and they are included in their classes:

- View the forbidden drives: This function is called before any activation of defense mechanisms to demonstrate the forbidden drives and their components.
- View the altered drives: After each defense mechanisms, drive manipulation is presented by calling this function
- View association parameters before defense mechanisms depreciation or idealisation: The attributes of the perceived object are viewed before application of depreciation or idealization.
- View association parameters after defense mechanisms depreciation or idealisation: The attributes of the perceived object are viewed before application of depreciation or idealization.

4.5.1 Bar Chart Inspector

The Reason to use the Bar chart is to view the number of altered drive or perceptions during simulation steps and to compare it with the number of simulation steps.

The Figure 4.12 shows how the implemented classes “cls_BarChartInspectorF06” and “cls_BarChartInspectorF06” are associated with interfaces “itfInspectorBarChartF19” and “itfInspectorBarChartF19”. Further, the classes of the modules F06 and F19 import these interfaces. The classes of “cls_BarChartInspectorF06” and “cls_BarChartInspectorF06” are arranged in the project “DecisionUnitMasonInspectors\src\inspectors\mind\pa_v38\autocreated”.

The interfaces “itfInspectorBarChartF19” and “itfInspectorBarChartF19” are in the directory of “DecisionUnits\src\pa_v38\interfaces” which contain the following two methods: one to return name of bar chart and is called by classes of the modules defense mechanisms for drives and perceptions. The second method returns the current set of data as association memory, which contains the name of the current defense mechanism and the value of numbers of altered drives

As seen in the Figure 4.12, the “DecisionUnitMasonInterface” includes the both classes “cls_BarChartInspectorF06” and “cls_BarChartInspectorF06” where the implementation of the following steps is done:

- Initialization of data of bar chart which contains of the following information titel, label for category axis, label for the value axis, dataset, and the orientation.
- Default value of the numbers of altered drives and perceptions.
- Updating data in bar chart in every step of the simulation.

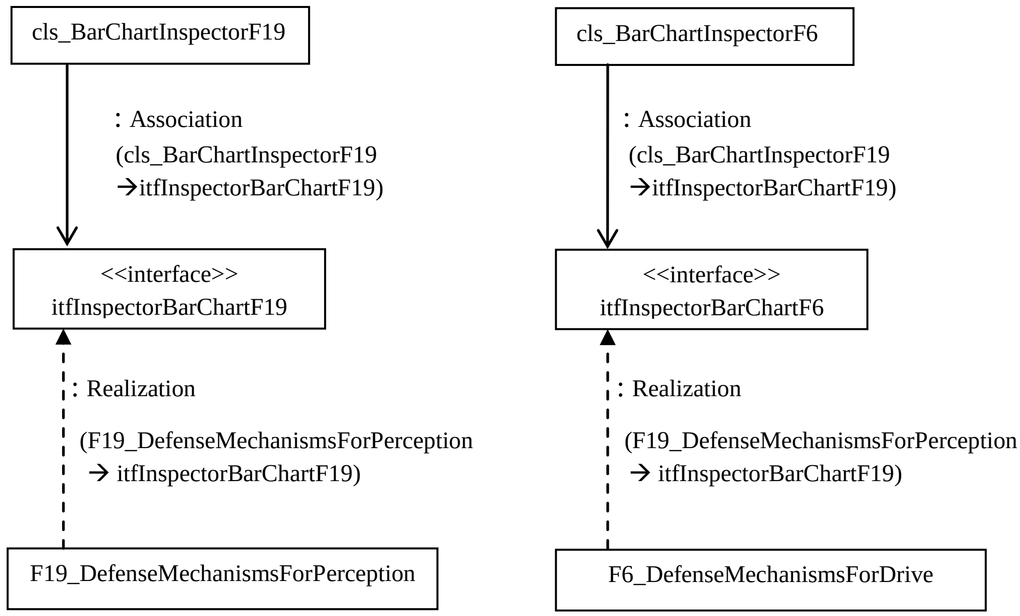


Figure 4.12: Inspectors for the Bar chart for modules F06 and F19

4.5.2 Time Chart Inspector

The Reason to use time chart inspector is to view which defense mechanism is activated and also to calculate time duration to compare them with the numbers of altered drives which is resulted by bar chart.
The inspectors of time chart are used for both classes of defense mechanisms. The Figure 4.13 shows how the class of time chart, its interface, and their association are composed. The class cls_CombinedTimeChart is arranged in the project:

"DecisionUnitMasonInspectors\src\inspectors\mind\pa_v38\autocreated" and the interface "itfInspectorCombinedTimeChart" in the project "DecisionUnits\src\pa_v38\interfaces".

The used steps for time chart are as the following:

- Initialization of data of time chart which contains of the following information titel, label for category axis, label for the value axis and dataset.
- Evaluating the label for Y-axis, in the case of the present work is 0 or 1, which indicate active or inactive.
- Evaluating the current set of data which contains the value of the number of simulation and the current defense mechanism.
- Evaluating the name of chart title which contains the name of defense mechanism.

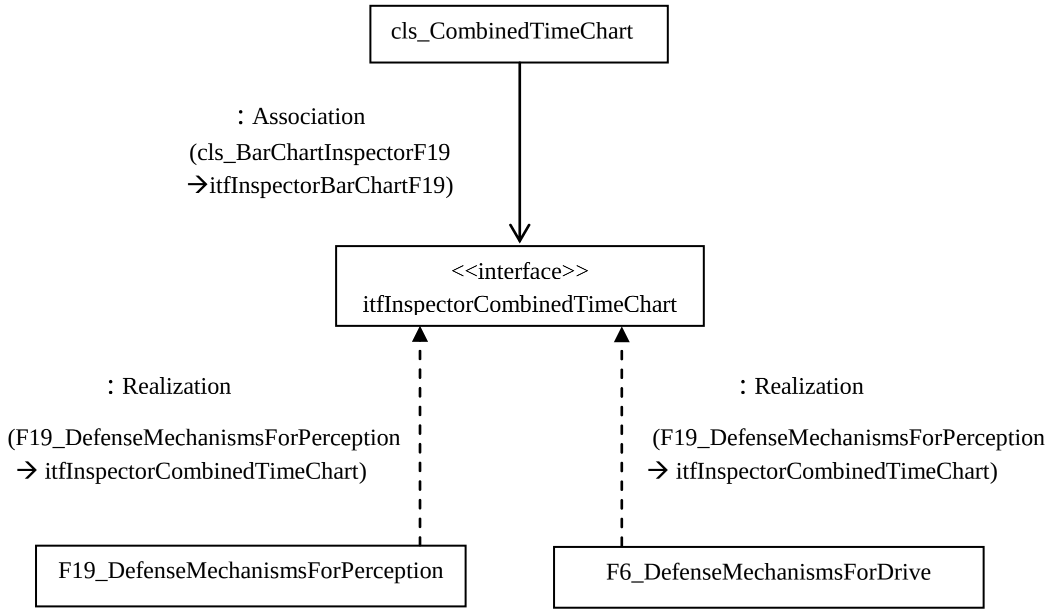


Figure 4.13: Inspectors for Time Chart

5. Simulation and Results

The result of each defense mechanism and composition of feeling are visualized using simulation platform. Based on inspectors provided by MASON, The mental process of the agents can viewed. In Chapter of simulation and result, a use case is introduced, which explains the situation of agents, will be described. The scenario consists of agents that are placed in an environment which appoints their tasks and nutrition source. These tasks allow the agents to enter to situations that lead to conflict. The aim of this use case is to prove the functionality and the application of defense mechanisms.

First the platform for the evaluation is presented, where the Agents and their environment are introduced. This is followed by description of use case and the rules of Superego. Finally the result of each defense mechanism is presented using different inspectors. This is done by viewing the forbidden drives and the altered drives in order to view the manipulations that happen after the application of defense mechanisms. To prove the correctness of the implementation, time chart and bar chart are used. For the time duration, a time chart is used, where the y-axis presents “0” (for inactive) and “1” (for active) and x-axis indicates the simulation steps. The calculation of the altered drives is presented by a bar chart where the y-axis presents the number of altered drives and the x-axis indicates the current defense mechanism that alters the drive.

5.1 Platform for Evaluation

According to psychoanalysis, defense mechanisms are functions of Ego of the human being and deal with drive information coming from Id which is developed from the body of the human being and represents bodily demands in the psyche. Furthermore, defense mechanisms manipulate drive information such as drive object or drive aim. For example, displacement replaces the original drive object by a new one and reaction formation replaces the drive aim by the opposite. Based on this motivation, to test and evaluate the implementation of defense mechanisms, a body is needed to act within artificial world and has to provide actuators, sensors, and carries the ARS model as control system. The artificial world has to consist of autonomous agents, which can interact with different objects and apply actions to change the environment. To ensure that the implemented world satisfies the needs to process tests, a use-case is needed to define the conditions, start situations, and desire outcomes. The goals of this use case are to demonstrate the conflicts in the agent between drive demands and Superego claims and to resolve them by the application of different defense mechanisms which have to be implemented in the present work.

Considering the above say, the ARS project team has implemented two agents which carry the whole ARS model. The agents are placed in an environment which consists of walls and nutrition. The input data of the agents corresponds to that of a human. They consist of perceptions of the environment and of inner drive demands like demand to eat. At the start of the situation, Adam felt hungry and the schnitzel is placed outside of its perceptual reach. The drive tension increases and the demands of drive satisfaction (Es-demand) come into conflict with control demand of the Superego. The Ego mediates between Superego, Id, and real situation and finally Ego satisfied the Superego by applying different defense mechanisms. The internalized rules in Superego in our model, for example in case of hunger, are to not eat the other agents to improve the reliability and to not eat walls to improve safety. For example, if the first rule is violated, the drive demand, which caused the conflict, becomes forbidden and defense mechanisms are activated. The observation of this use-case in the simulation takes only a few seconds. It is not important to observe the movement of Adam. But the mental processes, which trigger this movement, are analyzed

5.1.1 ARS World Platform

The idea of this simulator is to provide an artificial environment that can test ARS during the implementation. The simulator is based on the development toolkit MASON version 13 discussed above in Section 2.5.1.

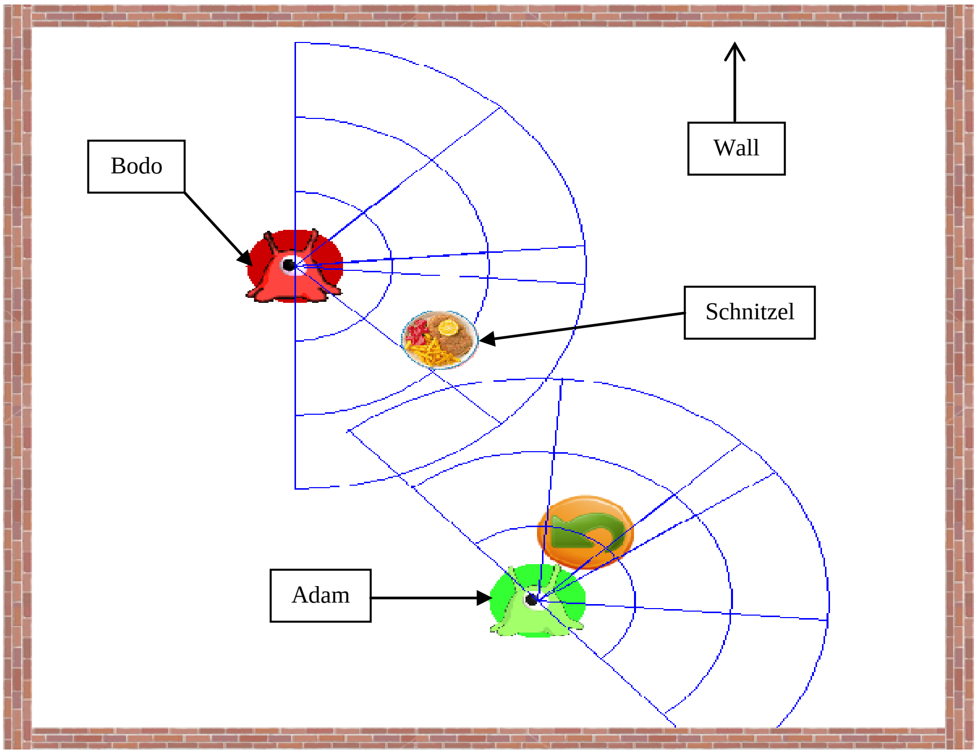


Figure 5.1: ARSIN world

As seen in the Figure 5.1, which consists of two agents colored with red and green and object which presents their source energy. The agents are called in ARS: Bodo (red agent) and adam (green agent). The environment is called "ARSIN world" and has a surface of a torus and is a two-dimensional, where the x-axis and y-axis end points are connected. The objects and agents within the world are defined in two-dimensional.

5.1.2 ARSIN Agent

The Figure 5.2 presents ARSIN agent in form of alien. This form has been chosen to avoid emulating the human body. It consists of a torso equipped with six tentacles which used to move and to manipulate objects. Furthermore, the agent contains two antennas and a single eye. Other aspect like ears or internal systems is not seen from outside [Deu11, p. 123].

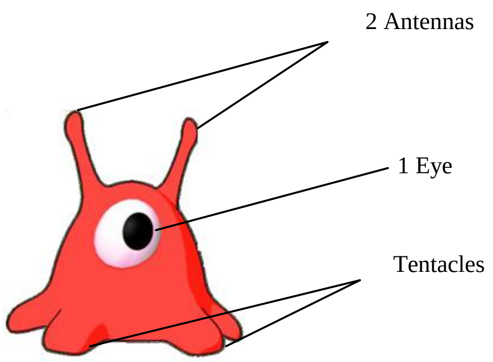


Figure 5.2: ARSIN agent

5.1.3 Use Case: Adam Searches Schnitzel

The initial situation of the case study "Adam searches Schnitzel" has a scene with the following objects: Agents named Adam and Bodo, contain the ARS decision unit as controller and an energy source like in the form of a "Wiener Schnitzel".

The Figure 5.1 shows a situation of this use case where agent Adam is hungry and, because of the empty stomach, homeostatic drives are produced in his body. This increases further because he eats nothing. At the start of the situation, the Schnitzel is placed outside of its perceptual reach and Adam does not yet perceive this and he also has very strict Superego rules which are activated now because of rising hunger and drives impulses. This situation results in a conflict between the aggressive claim of hunger and the Superego. The drives or the perceptions which do not meet the Superego rules are called forbidden drives or forbidden perceptions. Mediation by the Ego now activates defense mechanisms to convert undesired to desired drives or perceptions.

From the outside, the proffered scene offers a short course of action. Adam is ever the hungry actor. He moves towards it, but before he reached this, he takes Bodo true.

The observation of this use cases in the simulator only takes a few seconds. But it is not important the movement of Adam, but instead the mental processes that trigger this movement are analyzed. For visualizing various debug tools are used in simulator, as shown below.

Rules of Superego:

The Figure 5.3 shows the internalized rules of the module F07 Superego. They consist of rules for drives, emotions, and perceptions. The module checks data according to initialized rules. If these data don't agree with those rules they become forbidden as seen in the figure.

The selected parts in Figure 5.3 present relevant data on the module F07. The label (a) demonstrates the strength of Superego which is a double value. The parameters of drives are presented by label (b) and they consist of the case of the rule (Hunger), component of the drive (Aggression or Libidinous), organ (Stomach), and the quota of affect. Label (d) presents the perceived object. Finally, According to these rules, drives or perceptions which don't agree with these rules become forbidden as seen by label (e).

The forbidden drives have to be changed by applying different defense mechanisms. In this Chapter, the application of different mechanisms is demonstrated.

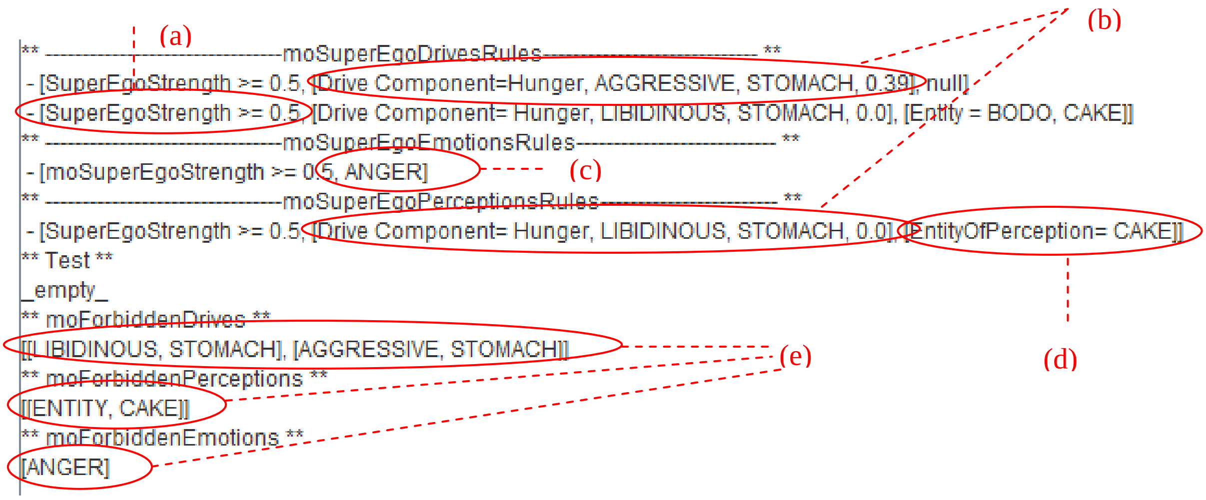


Figure 5.3: Internal data "Rules in Superego"

5.1.4 Inspectors

As mentioned in the Section 4.4, inspectors are implemented for the modules F06 and F19 to evaluate and test the implemented defense mechanisms. The Figure 5.4 shows the menu bar of the inspectors for the both modules. Based on information presented in the Section 4.4.2, the implemented inspectors for time chart and bar chart are labelled in the Figure 5.4. Furthermore, information about altered data has to be viewed. For this, the state is used which presents the input and output data for each module

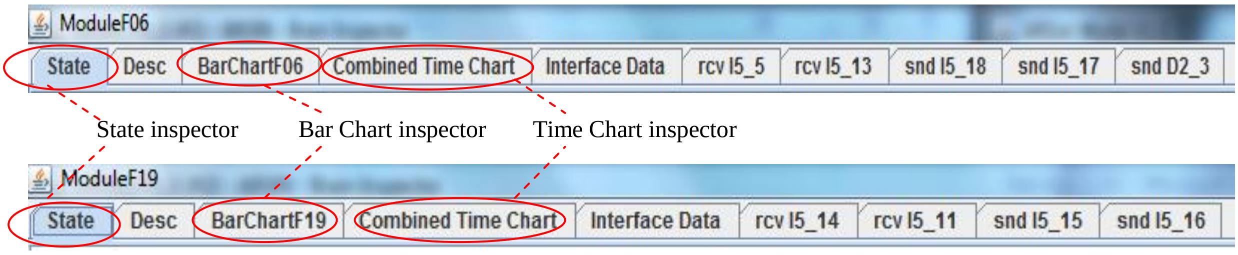


Figure 5.4: Menu bar of the additional inspectors

5.2 Defense Mechanisms

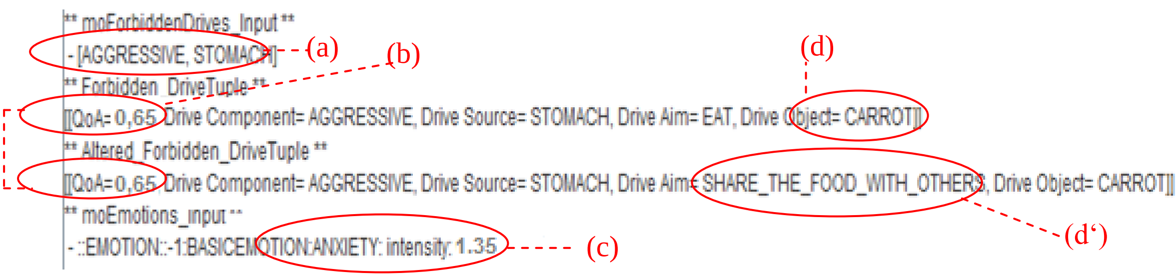
To view the result of defense mechanisms with the manipulation of data, the following information is demonstrated: input of the forbidden drives/perceptions, altered drives as tuple with quota of affect, component of drive, drive source, drive aim, and drive object. The time duration of each defense mechanism and the number of altered drives or perceptions are also shown by using charts to prove that no data are lost.

Furthermore, it has been proved that it is possible to control the result of the defense mechanism, whether it is desirable, by the activation of two defense mechanisms. This is done by using; turning against the self and projection at the same time if the drive aim is EAT or BITE in order to protect ADAM from hurting itself.

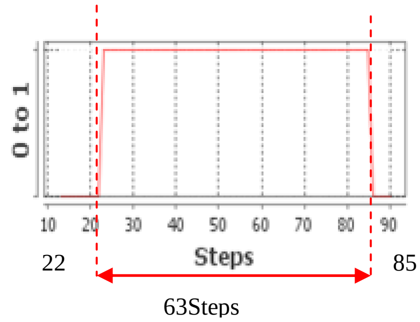
Sublimation

In Session 3.1.3 is pointed that the drive aim by sublimation should be transformed into socially acceptable actions. As seen in Figure 5.5 the result of Sublimation is demonstrated.

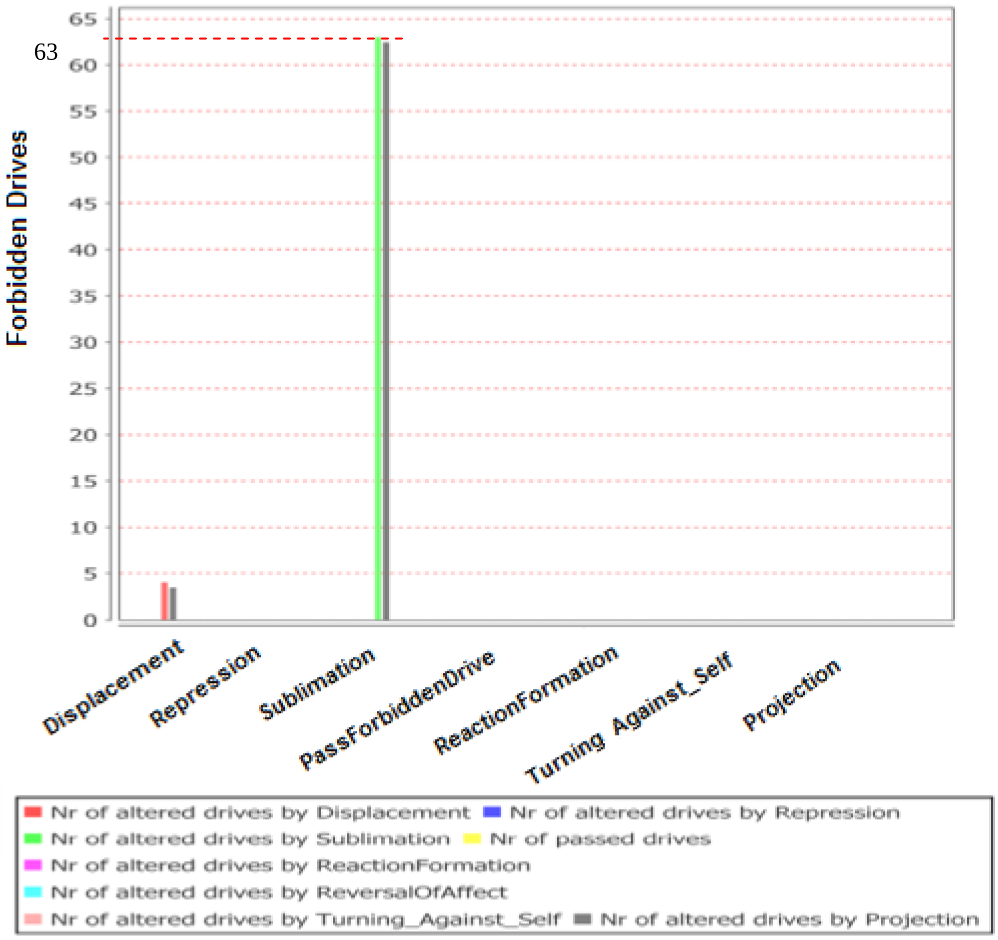
The Figure 5.5(a) presents the internal data where the label (a) indicates the component of forbidden drive such as AGRESSIVE and drive source STOMACH. Depending on the implementation, sublimation is active when the intensity of anxiety between 1.3 and 1.4 and the quota of affect is greater than “0.5” (Fig. 5.5(a) label (b) for quota of affect and (c) for intensity of anxiety). In Figure 5.5(a) the transformation of the drive aim “EAT” into drive aim “SHARE THE FOOD WITH OTHERS” is demonstrated by label (d) and (d’) where (d) is for original drive aim and (d’) is for altered drive.



(a)Internal data (screenshot)



(b)Time duration of Sublimation



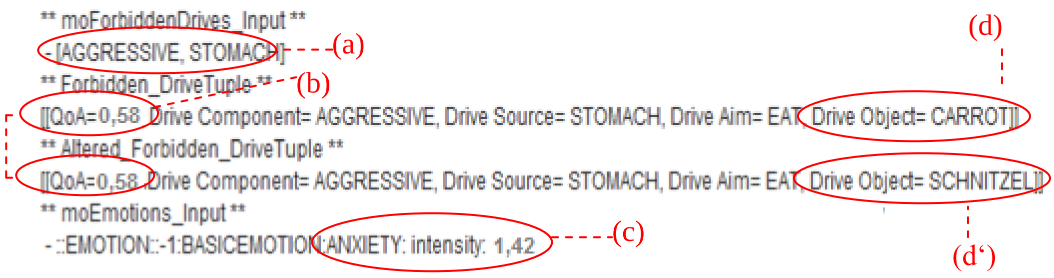
(c)Number of altered drives by Sublimation

Figure 5.5: Results of Sublimation (screenshots)

The Figure 5.5(b) shows the time duration of sublimation, where y-axis represents the activity of sublimation and the x-axis represents the simulation steps which are 63. In Figure 5.5(b) the numbers of altered drives are shown and amount to 63 drives. This summary showing that with each simulation step a drive is processed; therefore no delay or data loss has happened. In the Figure 5.5, the number of drives altered by defense mechanism displacement is starting to rise, this will be discussed next.

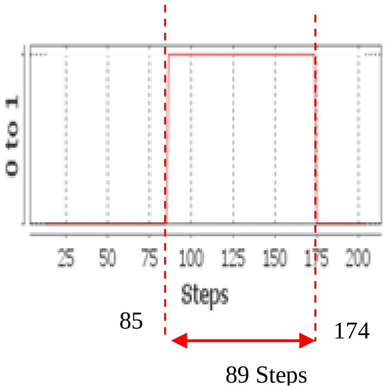
Displacement

In the case of the current simulation, after sublimation, displacement is activated. Defense mechanism displacement transforms the drive object in to another one. As shown in Fig. 5.6(a), the forbidden drive contains the following properties: drive component AGGRESSIVE, drive source STOMACH (Label (a)), drive object CARROT which is changed by the use of displacement to "SHNITZEL" (Label (d) and (d')). The activation of displacement is implemented when the value of the intensity of anxiety is within interval]1.4,1.6] and also the value of quota of affect of drive is greater than "0.5" as shown (Label (b) and (c)).

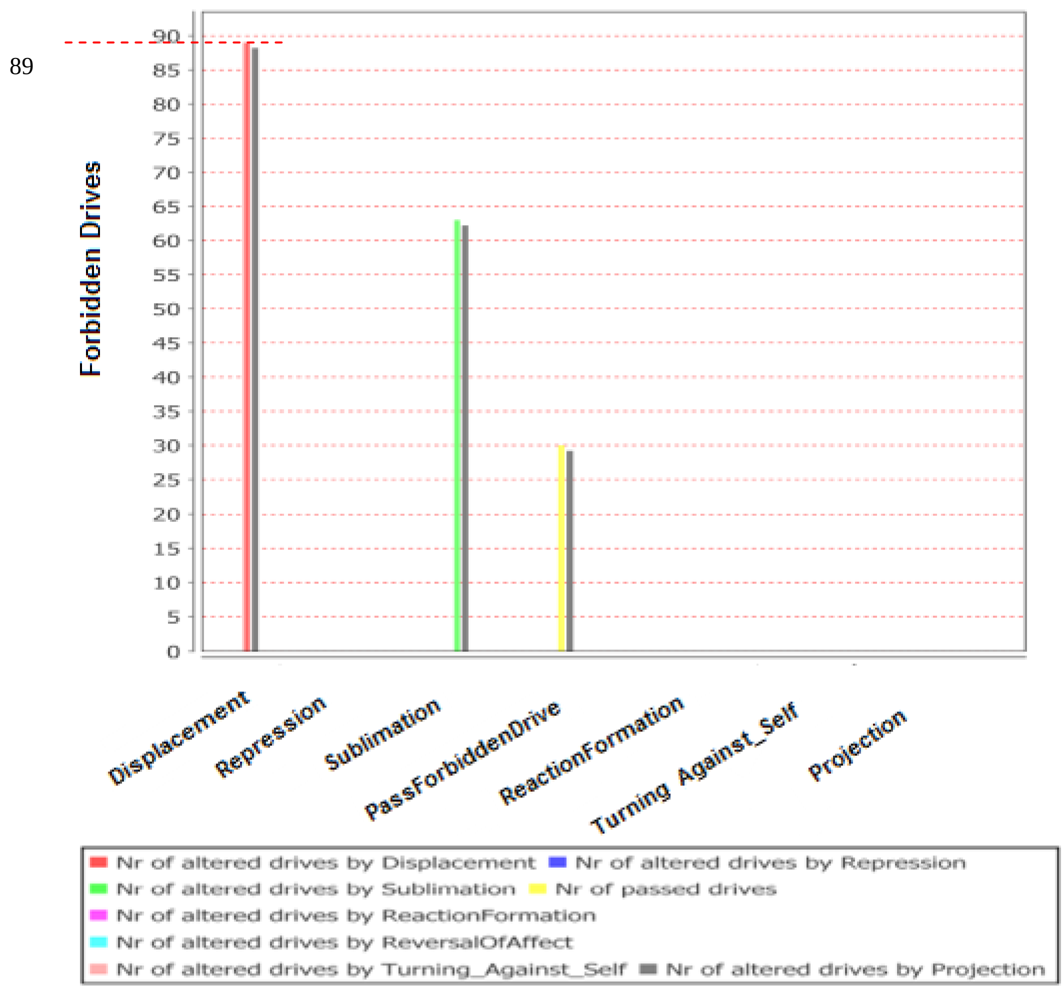


(a)Internal data

The time duration of displacement is 89 simulation steps, during this period 89 drives are altered by displacement (Fig. 5.6 (b) and (c)). This result also demonstrates that for every simulation step a drive is processed. The Figure 5.6(c) also shows that the number of passed forbidden drives is starting to rise; this means that no defense mechanism has been activated, whether this happens is dependent on the condition of the quota of affect or the value of anxiety



(b)Time duration of Displacement



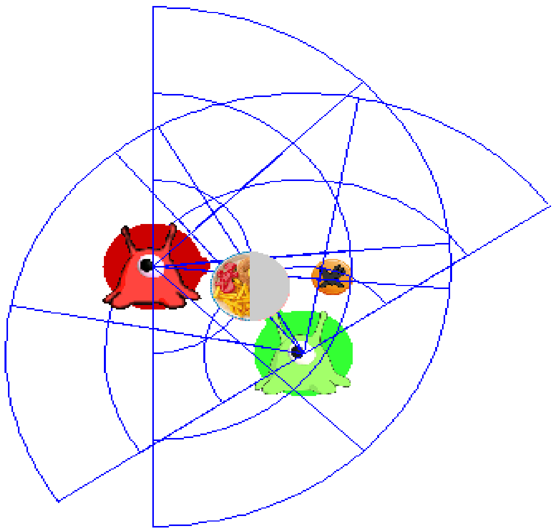
(c) Number of altered drive by Displacement

Figure 5.6: Results of Displacement (screenshots)

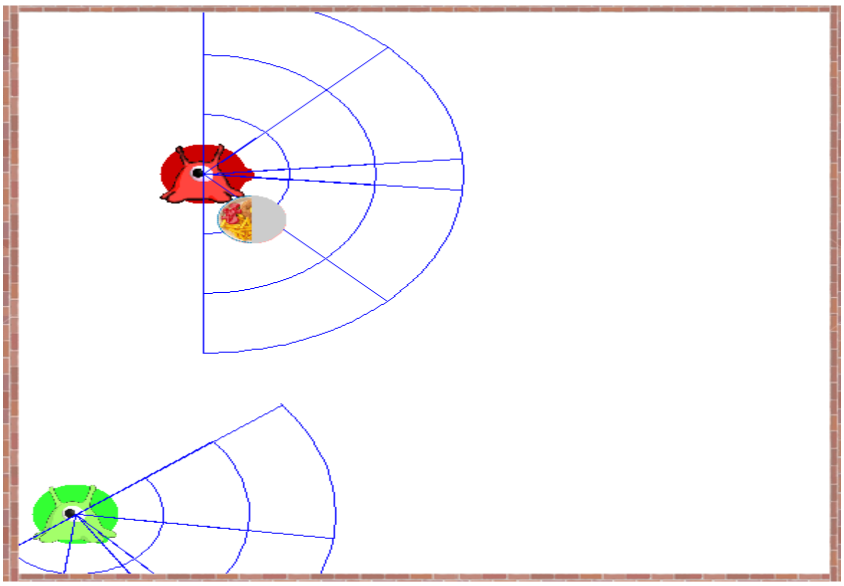
Situation of Adam

After activation of defense mechanisms sublimation, displacement, and the passed forbidden drives, Adam starts to eat, as shown in Figure 5.7(a). At this time no forbidden drives are generated because none of the input drives correspond to Superego rules. Quota of affect from all input drives and the intensity of anxiety are decreased.

After the satisfaction of Adam's need, Adam goes in the direction of the wall (Fig.5.7(b)). Now the quota of affect and the value of the Anxiety get higher, this leads to the occurrence of the forbidden drive which causes the activation of defense mechanisms.



(a) Adam eats



(b) Adam goes away

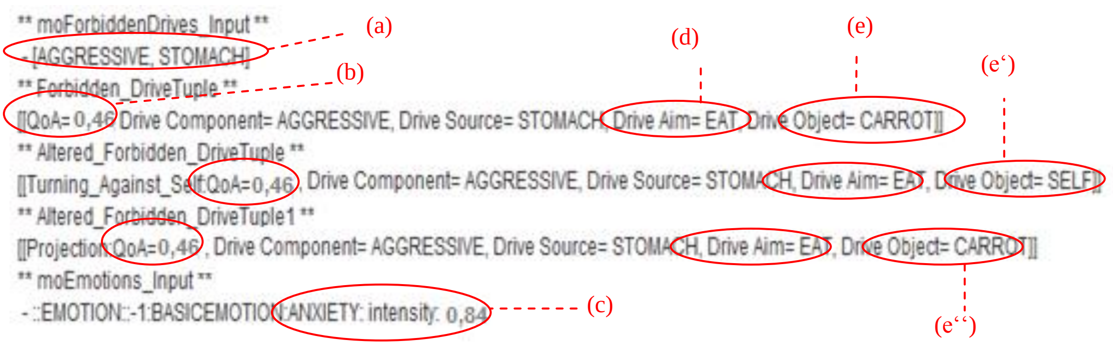
Figure 5.7: Situation of Adam

Turning against the self and projection

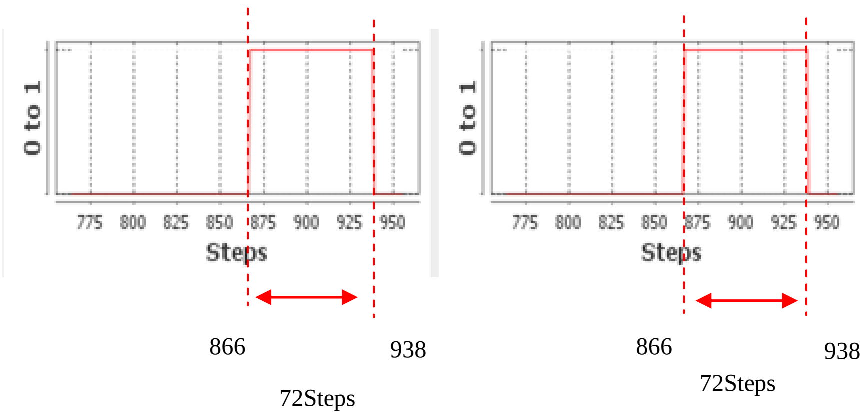
As mentioned above, that two defense mechanisms can trigger at the same time. Turning against the self and the projection are both activated if the aim of the drive is for example EAT or BITE, to protect Adam from hurting himself. Figure 5.8(a) presents the internal data similar to previews defense mechanisms, the difference is the quota of affect and the intensity of anxiety (0 < Quota of affect <=

0.5 and Intensity of anxiety within]0-1.0]) which indicate the activation of turning against the self. The Fig. 5.8 (a) label (e) presents the original drive object "CARROT" and label (e') presents the drive object after turning against the self "SELF". Because of the drive aim is "EAT" (Fig. 5.8 label (d)) the defense mechanism projection is used to get the original drive object "CARROT" (Fig. 5.8 label (e'')) back.

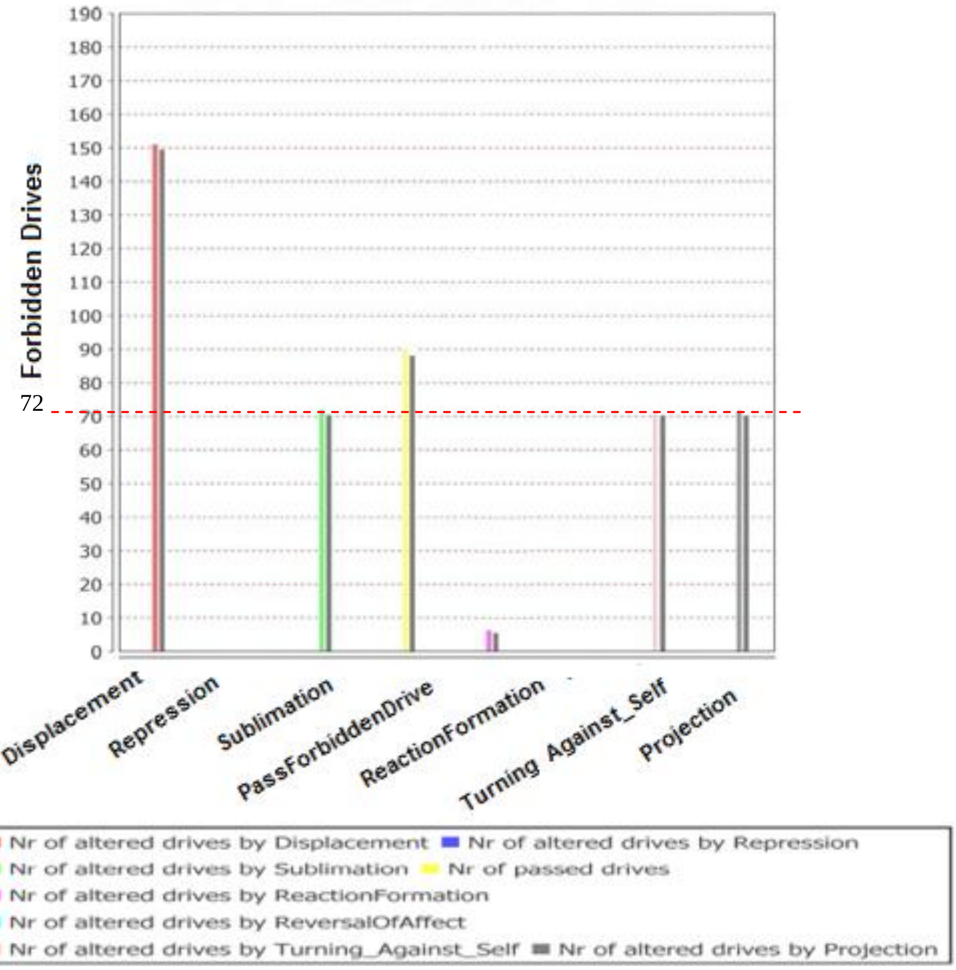
As shown in the Figure 5.8 (b) the both the defense mechanisms are activated and their time duration amounts to 72 steps. Figure 5.8 (c) presents the number of altered drives (72) which is equal to the simulation steps.



(a)Internal data



(b)Time duration of Projection and Turning against the self

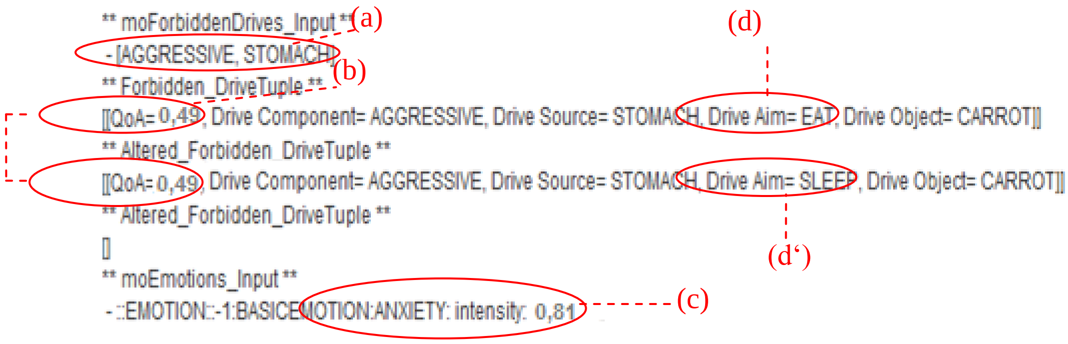


(c) Number of altered forbidden drives by turning against the self and projection

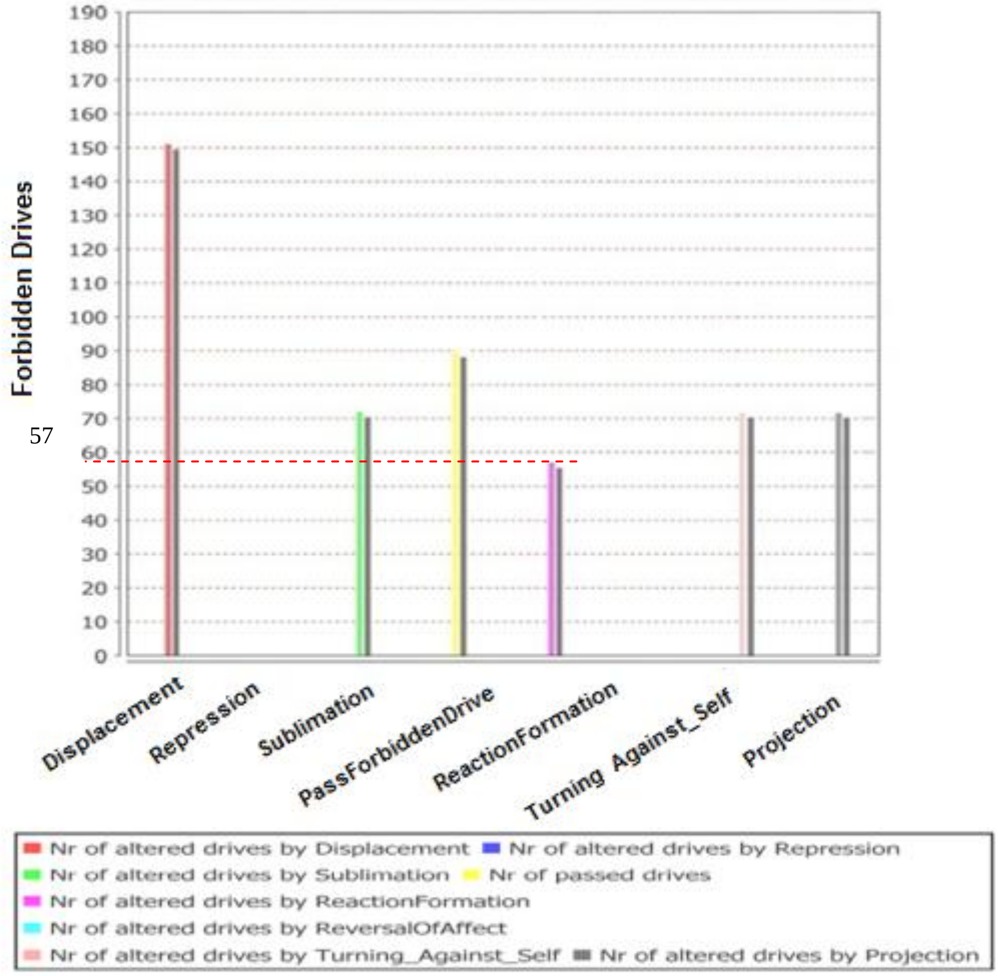
Figure 5.8: Result of Turning against the self and Projection

Reaction formation

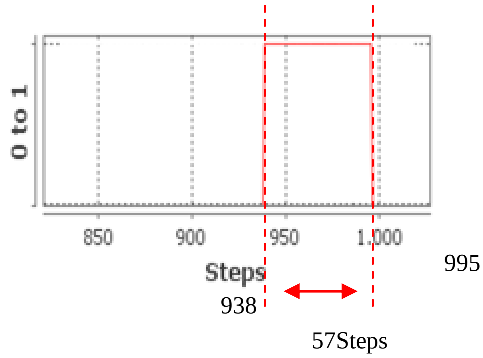
By defense mechanism reaction formation the drive aim is changed into its opposite. As seen in Figure 5.9 (a), the forbidden drive has drive component AGGRESSIVE, drive source STOMACH (Label (a)), and drive aim EAT (Label (d)) which changes to its opposite SLEEP where no action is assigned (Label (d')). Reaction formation is activated, depending on the manner of the implementation, when the value of the intensity of anxiety is within interval]1.0,1.3] and the value of quota of affect of drive is within 0< quota of affect <= 0,5(Label (c)).



(a)Internal data



(b)Number of altered forbidden drives by Reaction Formation

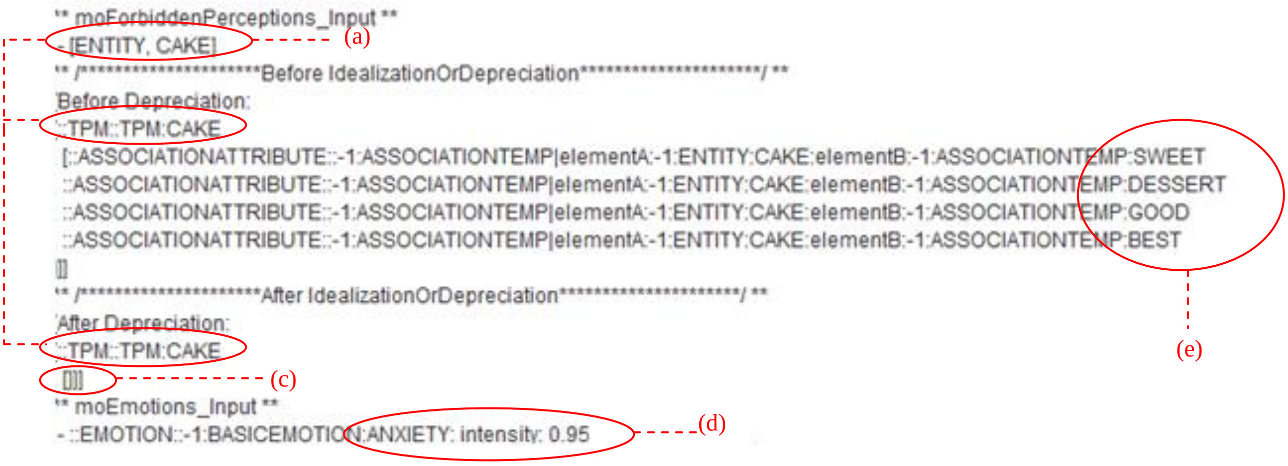


(b) Time duration of reaction formation

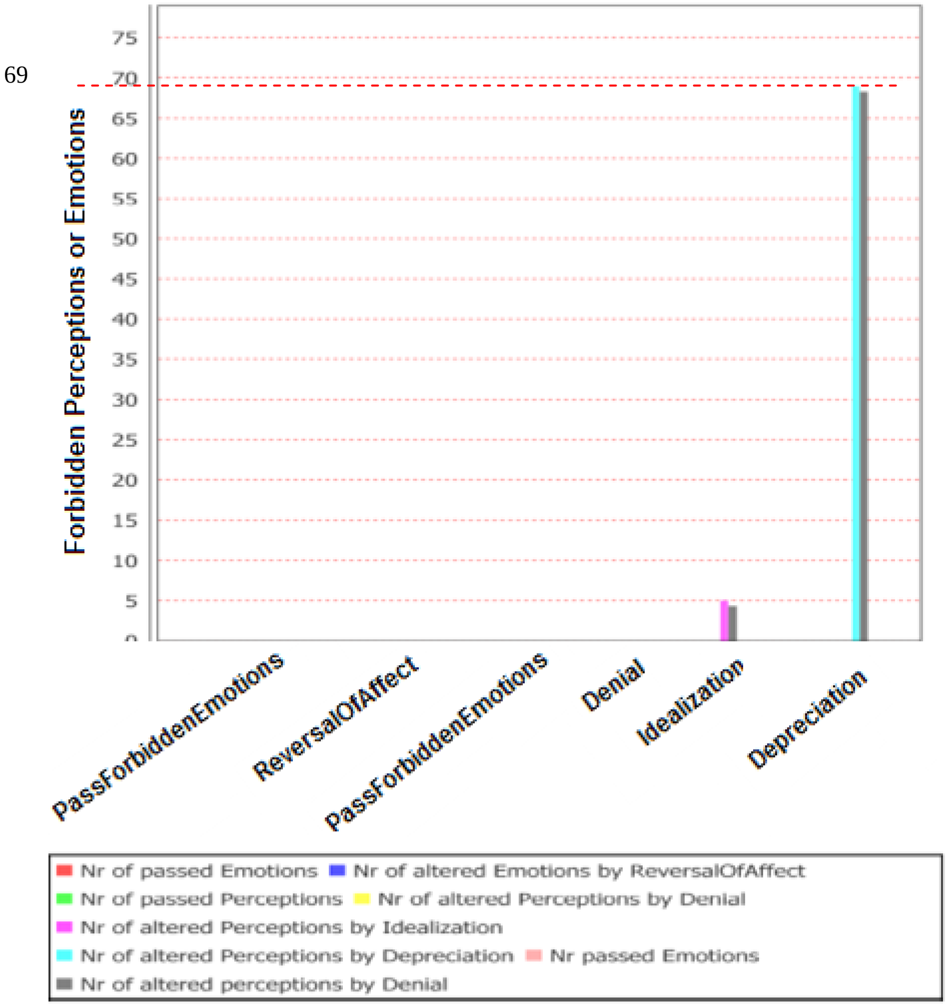
Figure 5.9: Result of Reaction formation

Depreciation

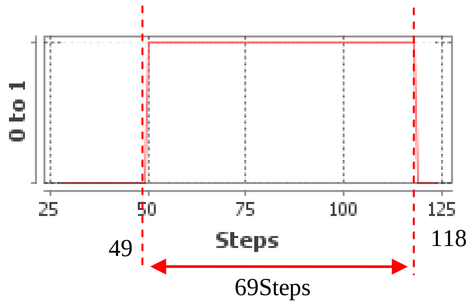
Defense Mechanism depreciation deletes all the positive properties of the perceived object. On implementation, depreciation is activated if the value of the anxiety is between interval [0.9, 1.0[, which is shown in Figure 5.10 (a) label (d) equal to ca. 0.95 . In the same Figure, the forbidden perception object is CAKE (Label (a)), where; TPM stands for thing presentation mesh and ASSOCIATIONTEMP for the properties of the perceived object. During the activation of depreciation, a list of positive associations are created such as SWEET, DESSERT, GOOD, and BEST (Label (e)) and later all of these associations are deleted (Label (c)). Figure 5.10 (b) and (c) indicate that the number of simulation steps and the number of altered perception are identical and is 69. As shown in Figure 5.10 (b), after the depreciation defense mechanism idealization is activated, which is discussed next.



(a)Internal data (Screenshot)



(b) Number of altered perception by Depreciation



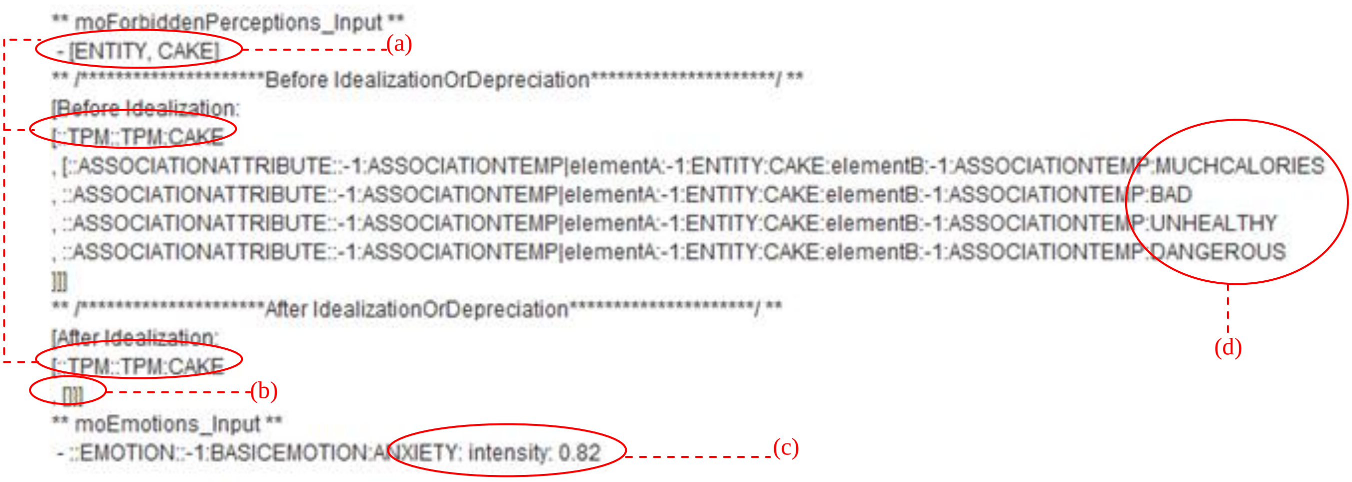
(c)Time duration of Depreciation

Figure 5.10: Result of depreciation

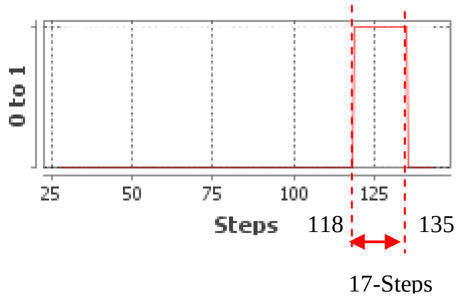
Idealization

As discussed above, idealization deletes the negative properties of the perceived object. Label (a) the Figure 5.8 (a) shows the forbidden object CAKE and its bad properties: MUCHCALORIES, BAD, UNHEALTHY, and DANGEROUS which are created (Label (e)). Defense mechanism idealization is activated when the intensity of anxiety, within the interval of [0.7, 0.9[which on fig.5.8 (a) is label (c) with amount ca. 0.82, at the end all the negative associations are removed (Label (b)).

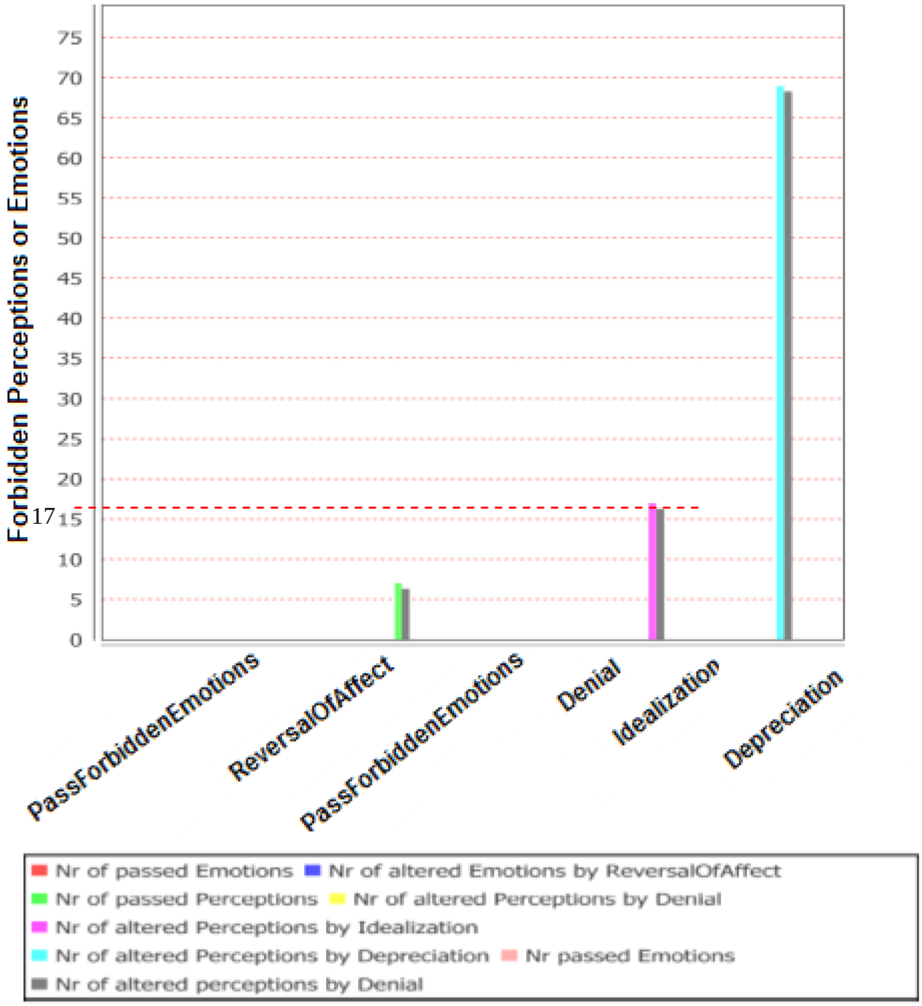
The Figure 5.8 (b) and (c) presents the equivalence of time duration and number of altered perceptions by idealization which indicates that in each simulation step one perception is processed. As shown in fig. 5.11(c) after idealization is passed, forbidden perception is activated. This means that no defense mechanism has been applied.



(a) Internal data screenshot



(b) Time duration of Idealization



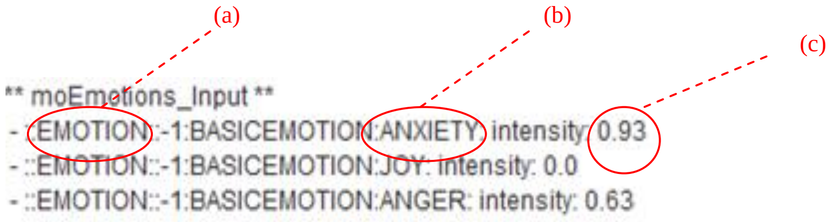
(c) Number of altered perception by idealization

Figure 5.11: Result of Idealization

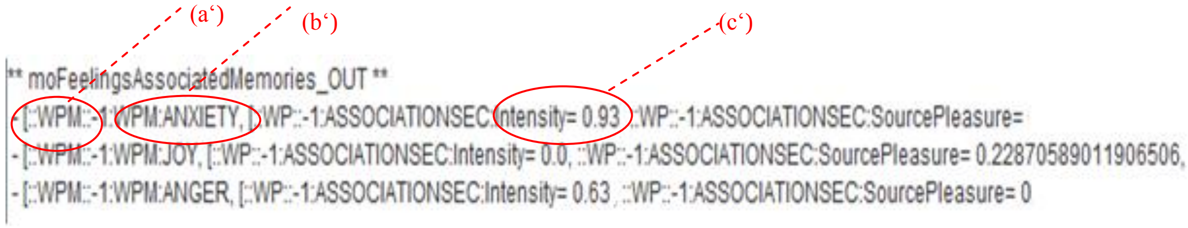
5.3 F20 Composition of Feelings

The Transformation of emotion to feeling in module F20 is done by converting data type of emotion from “clsEmotion” to Word Presentation Mesh (WPM). A list is created which contains all the emotions in order to get from each emotion one feeling.

Figure 5.12 presents the transformation of emotion Anxiety where its parameters and their values are shown in Figure 5.12 (a). Those parameters are converted to data type word presentation mesh and are related to WPM over association secondary, as demonstrated in the Figure 5.12 (b). The labelled data in the Figure 5.12 (a) presents relevant data that are converted into feelings. Label (a) indicates data structure of input data of the module 20 which is emotion. The type of the emotion is presented by the label (b) and the intensity of the emotion is labelled by (c). The output of the F20 is presented by Figure 5.12 (b), where the data structure is WPM and for each emotion is one feeling is made. Furthermore, the figure shows data structure is labelled by (a'), the type of feelings (b'), and the intensity of Anxiety which is related to WPM over association secondary.



(a)Internal data



(b)Data structure of feeling

Figure 5.12: Result of Composition of Feeling

6. Conclusion and Outlook

The increasing amount of sensory data requires a new approach to deal with this data. A possible solution is the application of psychoanalytical defense mechanisms where the realisation of these mechanisms is done in two stages. In the first, rules have to be declared, these consist of forbiddances and restrictions and in the second stage, selection a specific defense mechanism. Data, which does not agree with these rules, causes conflicts that have to be altered or repressed by these mechanisms. The goal of extending the ARS model by applying defense mechanisms is to reduce conflicts in system and to facilitate the selecting of a decision.

In the Chapter of Conclusion and Outlook, a discussion of the present work is introduced and the possible application of defense mechanisms in technical systems is presented.

6.1 Discussions

As discussed in Chapter 1, the complexity in the expressing of defense mechanisms and differing opinions on the psychoanalytical processes complicate their application in AI models. Therefore, there are not many AI projects which used these mechanisms. The projects presented in Chapter 2, based on cognitive architecture like BDI, SOAR, and LIDA, use various approaches to the human mind. Some of AI architectures tried to use defense mechanisms but not in the manner way as defined in psychoanalysis. Buller developed a model called Volitron and he applies these mechanisms in the case where no acceptable plan exists or the execution plan hasn't changed the environment. As discussed in the Section 2.1.2, the way defense mechanisms are implemented in Volitron may cause a delay in the processing time. Another AI model called Modelling Human Mind activates defense mechanisms in dependence on the energy of the Ego and defined the conflicts as inductive probability. In our ongoing ARS-Project, defense mechanisms are placed in the primary process before action planning in order to choose a conflict-free action. Conflicts are defined in ARS such as rules which have to be considered. Examples of rules used in our model are those which protect the social life in the multi agent system from the desires of agent and if input data don't fulfil the desired goals of agent. The application of defense mechanisms in the ARS-project is similar how they are defined in psychoanalysis.

The implementation of the different defense mechanisms in the present work is achieved due to results from previous work done in (e.g [GB12], [GBD+11], [Rie09]) and it's an alternative implementation of these mechanisms. It is done by taking care of the conflicts that are defined in the current version of ARS using the use case "Adam searches schnitzel". The implemented mechanisms were tested and they function correctly, as demonstrated in Chapter 4 and Chapter5.

The classifications of defense mechanisms are made according to their strength of data manipulation and it is dependent on the quota of affect and the intensity of anxiety. The quota of affect in our ARS project describes how strong a certain drive demand is and anxiety is defined as unpleasure, which increases along with rising drive tension [DDS+14, p. 57]. Based on the amount of the value of the quota of affect and anxiety, specific defense mechanisms can be selected by classifying them according to their strength of manipulation of drives or perceptions. For example, if the values of the quota of affect and the anxiety are high, the corresponding defense mechanism has to be strong. This means that the drives or perceptions which cause conflicts have to be changed completely or repressed. This is done in order to increase the protection of agent.

The Module of Composition of Feelings is also extended in the present work, where emotions are the pre-stage to feeling. Due to the placement of this module in the second process, emotions have to be transformed into Word presentation. This is done by setting for each emotion one WPM. As discussed in session 3.5, emotions in ARS-project are determined by four factors: Pleasure, unpleasure, aggression, and libidinous. All of these factors are converted to word presentations and related to WPM over association secondary.

6.2 Outlook

Defense mechanisms can be applied in the technical systems so that certain incoming data or actions are not allowed on the basis of safety or security which are defined as rules. Furthermore, technical systems may have to cooperate with other systems to exchange data. Therefore rules have to be initialized and this should be taken into account. For example a technical video surveillance system, within an automated building, may have to detect people, define their actions and interactions, and analyze their movement. In addition to this features, the technical video surveillance has to work with others videos that are placed in other environments. These systems have to deal with extremely large amounts of incoming data, which have to be classified and qualified. In other words, human actions or movements have to be analyzed and classified into authorized or forbidden behaviours, and result in a highly accurate description. This can be done through the application of rules which cannot be violated otherwise a conflict appear.

Applying the implemented defense mechanisms in a technical video surveillance system can be useful where the defense mechanism replacement can be used to replace the "not interesting" situation with one which could be "interesting" in order to focus surveillance and analyse the situation. Idealisation and depreciation filter the properties of the observed objects or situation in order to analyse them and to categorize them according to how dangerous they are. Sublimation can be applied if an observed object moves to another location where another video camera is operating; the information about the moved object would be shared. As defined above, reaction formation changes the drive aim to its opposite of it, this can be applied in such a system so as to change the system action to its opposite in case of the environment changing. Turning against the self and projection can work together. For example when finding a dangerous situation the system has to alert security staff. This is done first by turning against the self which sends a signal to the system to indicate that the observed situation is dangerous and the projection is used to alert the staff that has to deal with it.

As illustrated in the Section 3.2.1, there are 12 well known defense mechanisms. Implementing the rest of analytical defense mechanisms might be work for the future for ARS. Furthermore, the test application used in the present work is different to the original goals of the project ARS, but the aim of this test platform is to provide a test tool and to evaluate the implemented models of defense mechanisms. In the future, the simulation of the environment may have to be more complex because the current environment and the input data of the agent are limited.

Bibliography

[AG95] Anand S. Rao, Michael P. Georgeff. BDI Agents: From Theory to Practice. Proceeding of the first International Conference on Multiagent Systems, AAAI, 1995.

[BACR08] Birgit Burmeister, M. Arnold, Felicia Copaciu, Giovanni Rimassa. BDI-Agents for Agile Goal-Oriented Business Processes. Proc. of 7th Int. Conf.on Autonomous Agents and Multiagent Systems (AAMAS 2008) – Industry and Applications Track, Berger, Burg, Nishiyama (eds.), May, 12-16., 2008, Estoril, Portugal.

[BR+03] G.C.BALAN, C, Cioffi-Revilla, S. Luke, L. Panait, S. Paus. MASON: A Java Multi- Agent Simulation Library George Mason university. Available at: http://cs.gmu.edu/~eclab/projects/mason/publications/Agent2003.pdf [accessed 10.2013].

[Bul02] Andrzej Buller. Volitron: On a psychodynamic robot and its four realities. Proceedings Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, 94, 2002. Available at: http://www.lucs.lu.se/LUCS/094/Buller.pdf [accessed 04.2013].

[DDS+14] Dietmar Dietrich, Klaus Doblhammer, Samer Schaat, Alexander Wendt, Roman Widholm. Naturwissenschaftliches, psychoanalytisches Modell der Psyche für Simulation und Emulation. ARS Team, Institute of Computer Technology, Vienna University of Technology 2014.

[Deu11] Deutsch tobias. Human Bionically Inspired Autonomous Agents: The Framework Implementaion ARSi11 of the Psychoanalytical Entity Id Applied to Embodied Agents. PhD thesis, Vienna Univesity of Technology, 2011

[DFZB08] Dietmar Dietrich, Georg Fodor, Gerhard Zuker, Dietrich Bruckner, „Simulating the mind “. Wien: Springer Verlag, 2008.

[Die00] Dietmar Dietrich. Evolution potentials for _eldbus systems. In Factory Communication Systems, 2000. Proceedings. 2000 IEEE International Workshop on, volume 1, 2000. Invited Talk.

[DZ08] Dietrich, D., Zucker G. New Approach for Controlling Complex Processes. An Introduction to the 5th Generation of AI, Institute of Computer Technology, Vienna University of Technology, Vienna, Austria 2008.

[FB09] Stan Franklin, Bernard J. baars. CONSCIOUSNESS IS COMPUTATIONAL: THE LIDA MODEL OF GLOBAL WORKSPACE THEORY. In International Journal of Machine Consciousness, World Scientific Publishing Company 2009. Available at: http://ccrg.cs.memphis.edu/assets/papers/2009/GWT-IJMC-2009.pdf. [Accessed 03.2013]

[FP06] Stan Franklin, F. G. Patterson. The lida architecture: Adding new modes of learning to an intelligent, autonomous, software agent. Proceedings of Integrated Design and Process Technology, IDPT-2006, 2006. Available at: http://www.theassc.org/files/assc/zo-1010-lida-060403.pdf [Accessed 03.2013].

[FRD+07] Stan Franklin, Uma Ramamurthy, Sidney K. D'Mello, Lee McCauley, Aregahegn Negatu, Rodrigo Silva L., Vivek Datla. Lida: A computational model of global workspace theory and developmental learning. In AAAI Fall Symposium on AI and Consciousness: Theoretical Foundations and Current Approaches. Arlington, VA: AAAI, 2007. Available at: http://www.aaai.org/Papers/Symposia/Fall/2007/FS-07-01/FS07-01-011.pdf [accessed 03.2013]

[Fre48] Freud, A.: Das Ich und die Abwehrmechanismen. Frankfurt am Main : Fisher Taschenbuchverlag, 1984

[Fue03] Clara Tamarit Fuertes. Automation System Perception - First Step towards Perceptive Awareness. PhD thesis, Faculty of Electrical Engineering and Information Technology, Vienna University of Technology, 2003.

[GB12] Friedrich Gelbard, Dietmar Bruckner. Influence of Psychoanalytic Defense Mechanisms on the Decision Making Process in Autonomous Agents. Institut of Computer Technology, Vienna University of Technology 2012.

[GBD+11] Friedrich Gelbard, Christian Brandstätter, Klaus Doblhammer, Isabella Hinterleitner, Stefan Kohlhauser, Zsofia Kovacs, Heimo Zeilinger. Comparison of Technical Filter Mechanisms and Defense Mechanisms of the Human Mind - Why do We Need Psychoanalysis in Artificial Intelligence? Institute of Computer Technology, Technology Vienna University of Technology 2011

[GBK+11] Friedrich Gelbard, Dietmar Bruckner, Zsofia Kovacs, Klaus Doblhammer. Transformation of Perceptions as Filter Mechanisms in Artificial Intelligence. An Implementation of Psychoanalytic Defense Mechanisms in Artificial Intelligence, Institute of Computer Technology, published in AFRICON 2011.

[HJL09] Hanford Scott D., Janrathitikarn Oranuj, Long Lyle N.. Control of Mobile Robots Using the Soar Cognitive Architecture. Journal of Aerospace Computing, Information, and Communication. 6, February 2009.

[HMK+08] Bernd Hanke, Günter Meyer, Thomas Kohlhoff, Marius Hartel, Volker Pregizer, and Frank Schubert. Airport tests bacnet devices - competition at high stage. *BACnet Europe*, 8:41–43, 2008.

[Kön96] König K.: Abwehrmechanismen, 1996 Vandenhoeck & Ruprecht in Göttinger, Deutschland, ISBN 978-3-525-45607-1.

[L11] Sean Luke, Multiagent Simulation and the MASON Library. Available at: http://cs.gmu.edu/~eclab/projects/mason/manual.pdf [accessed 04.2013].

[Lai08] John E. Laird. Extending the Soar Cognitive Architecture. Division of computer science and engineering, University of Michigan. Proceeding of the 2008 conference on Artificial General Intelligence 2008. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.77.2473&rep=rep1&type=pdf [Accessed 04.2013]

[LC06] Pat Langley, Dongkyu Choi. A Unified Cognitive Architecture for Physical Agents. Computational Learning Laboratory Center for the Study of Language and Information Stanford University, Stanford, CA 94305, USA, 2006. Available at: http://pdf.aminer.org/000/013/692/a_unified_cognitive_architecture_for_physical_agents.pdf [Accessed 04.2013]

[N04] Norling E.. Flok Psychology for Human Modelling: Extending the BDI Paradigm. AAMAS'04, July 19-23, 2004, New York, USA. 2004 ACM 1-58113-864. Available at: http://pdf.aminer.org/000/056/398/folk_psychology_for_human_modelling_extending_the_bdi_paradigm.pdf [accessed 05.2013]

[NTMI99] Tohru Nitta, Toshio Tanaka, Kenji Mishida, Hiroaki Inayoshi. Modeling human mind. In Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on, volume 2, pages 342 - 347,1999.

[PAL08] Palensky Brigitte. From Neuro-Psychoanalysis to Cognitive and Affective Automation Systems. PhD thesis, Vienna Univesity of Technology, 2008.

[PP05] Gerhard Pratl and Peter Palensky. Project ARS - the next step towards an intelligent environment. Proceedings of the IEE International Workshop on Intelligent Environments, 2005.

[Rie09] Christiane Riediger. Psychoanalytical Defense Mechanisms Applied to Autonomous Agents. Master thesis, Faculty of Electrical Engineering and Information Technology, Vienna University of Technology 2009.

[RLJ06] Steven F. Railsback, Steven L. Lytinen, Stephen K. Jackson. Agent-based Simulation Platforms: Review and Development Recommendations. *SIMULATION*, Vol. 82, Issue 9, September 2006 609-623. 2006 The Society for Modeling and Simulation International.

[Rus03] Gerhard Russ. Situation-dependent Behavior in Building Automation. PhD thesis, Vienna University of Technology, Institute of Computer Technology, 2003.

[SRF00] Stefan. Soucek, Gerhard Russ, and Clara Tamarit Fuertes. The smart kitchen project – an application on feldbus technology to domotics. Proceedings of the 2nd International Workshop on Networked Appliances (IWNA2000), 2000.

[TDR01] C. Tamarit, D. Dietrich, K. Dimond, G. Russ. A Definition and a Model of a Perceptive Awareness System (PAS). Proceedings of the 4 IFAC Inter- national Conference on Fieldbus Systems and their Applications, 2001.

[W02] Micahel Wouldridge. An Introduction to Multiagent Systems. Department of Computer Science,University of Liverpool, UK 2002.

[Wer05] Werner, C., Langenmayr: Das Unbewusst und die Abwehrmechanismen, Psychoanalyse und Empire, 2005, Vandenhoeck & Ruprecht Gmbh & Co.KG, Göttingen, Deutschland, ISBN 3-525-45005-2.

[WG06] Pei Wang, Ben Goertzel. Introduction: Aspects of artificial general intelligence. Proceed1ing of the 2007 conference on Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms: Proceedings of the AGI Workshop 2006.

[Woo96] Michael Wooldbridge. Practical Reasoning with Procedural knowledge: A logic of BDI agents with know-how. International Conference on Formal and Applied Practical Reasoning, FAPR'96 Bonn, Germany, June 3–7, 1996 Proceedings. P 663-678. 1996.

[Zei10] Heimo Zeilinger. Bionically Inspired Information Representation for Embodied Software Agents. PhD thesis, Vienna Univesity of Technology, 2010.

[ZPK10] Heimo Zeilinger, Andreas Perner, Stefan Kohlhauser: Bionically Inspired Information Reprensentation Module. 2010 Conference on human system interaction May 13-15.2010, Rzeszow (poland).

Internet References

[1] Homepage of ARS project. http://ars.ict.tuwien.ac.at/ accessed January 2013

[2] SWARM Development Group. *Homepage*, 2013. http://www.swarm.org/index.php/Swarm_main_page, accessed February 2013

[3] Netlog. *Homepage*, 2013. http://ccl.northwestern.edu/netlogo/, accessed February 2013

[4] Repast. *Homepage*, 2013. http://repast.sourceforge.net/repast_3/index.html, accessed February 2013

[5] Anylogic. *Homepage*, 2013. http://www.anylogic.com/overview, accessed February 2013

[6] Mason. *Homepage*, 2013. http://cs.gmu.edu/~eclab/projects/mason/, accessed November 2013

A. Appendix

The important part of the present work is Chapter 4 where steps and functions are describes and they are coded to the programming language Java. In this part of the present work, the used methods and functions in the implementation are explained.

A.1 Defense Mechanisms for Drives

Displacement

defenseMechanism_Displacement (): Activation of displacement

MoDriveList_Output.remove(): Removing the forbidden drive from the List of output drive

MoDriveList_Output.add(displacement()): Inserting the result of displacement back into the output of the Module F06 where the function “displacement()” executes the process of defense mechanism displacement.

oDisplaceDriveObjectList.put(“Original drive object”,”new drive object”): Creating a HashMap for displacement

getActualDriveObject.getMoContent(): Return the original drive object from the forbidden drive which has to be replaced.

oDisplaceDriveObjectList.containsKey(): return true or false, if the original drive object exists in the HashMap.

oDisplaceDriveObjectList.get(): Return the value of the original drive object from the HashMap which is the new drive object.

clsDataStructureGenerator.generateDataStructure(): Creates data structure Thing Presentation Mesh for the new drive object.

setActualDriveObject(): Set the new drive object into the forbidden drive.

Sublimation, Reaction Formation

defenseMechanism_ReactionFormation() and defenseMechanism_Sublimation(): Activation of sublimation and reaction formation and create the HashMap.

oOppositeTP.put("Original drive aim","new drive aim"): initializing HashMap.

defenseMechanism_ReactionFormation_Sublimation_Intellectualization(): This is the shared function between sublimation and reaction formation it is also used by the defense mechanism intellectualization.

replaceDriveAim(): Receives the forbidden drive and replaces the drive aim.

getActualDriveAim().getMoContent(): Return the original drive aim from the forbidden drive which has to be replaced.

oOppositeTP.containsKey(): Return true or false, if the original drive aim exists in the HashMap of the current defense mechanism.

oOppositeTP.get(): Return the value of the original drive aim from the HashMap of the current defense mechanism.
clsDataStructureGenerator.generateDataStructure(): Creates data structure Thing Presentation Mesh for the new drive aim.
setActualDriveAim(): Set the new drive aim into the forbidden drive.

Turning against the self and Projection

*clsDataStructureGenerator.generateDataStructure():*Creates data structure Thing Presentation Mesh for the “Self”.

setActualDriveObject():Set the new drive object into the forbidden drive which self.

*getActualDriveAim().getMoContent():*Get dirve aim which is used to check the drive aim.

A.2 Defense Mechanisms for Perceptions

defenseMechanism_Depreciation() and defenseMechanism_Idealization (): perceive the forbidden perception and call the two following functions

CreateListWithPositiveObjects() and CreateListWithNegativeObjects(): The two functions create the parameters of the perceive objects. First a new data type as thing presentation mesh is created by the method clsDataStructureGenerator.generateDataStructure() with data type “TPM” and where content is the forbidden perception. After the creation of “TPM” a thing presentation is generated by the method clsDataStructureGenerator.generateTP().

deleteAssociationsFromPerception(): Deleting the associations of forbidden perception is done by calling this function which removes all the positive or negative parameters.

A.3 F20 Composition of Feelings

CreateWPMForEmotions (): Emotions in ARS-project are determined by four factors: Pleasure, unpleasure, aggression, and libidinous. All of these factors are converted to word presentations and related to WPM over association secondary. These steps by using the following methods *new clsWordPresentationMeshFeeling():*creates, for each emotion, one data type of word presentation Mesh in order to have for each emotion, one feeling.

new clsWordPresentation(): creates for each factor of emotion one word presentation.

clsMeshTools.createAssociationSecondary(): relates each word presentation which presentes the factor of emotion to Word Presentation Mesh Feeling over association secondary.

calculateQuotaOfAffect(): Calculate the average of quota of affect and organize it in intervals.

GetEmotionIntensity(): Get a value of the intensity of anxiety.

searchInEmotions(): Search for emotion anxiety in the emotion list.

emotion.getMoContent(): Return the type of emotion.

emotion.setMrEmotionIntensity(): Set the new value of the intensity of emotion.

A.4 Additional Inspectors for Defense Mechanisms

The following functions are implemented to set data to inspectors in the two modules of defense mechanisms and they are included in their classes:

ViewForbiddenDriveTupel(): This function is called before any activation of defense mechanisms to demonstrate the forbidden drives and their components.

ViewAltered_DriveTuple(): After each defense mechanisms, drive manipulation is presented by calling this function

ViewAssocBeforeDepreciation() and ViewAssocBeforeIdealization(): The attributes of the perceived object are viewed before application of depreciation or idealization.

ViewAssocAfterDepreciation() and ViewAssocAfterIdealization (): The attributes of the perceived object are viewed before application of depreciation or idealization.

Bar Chart Inspector

The interfaces itfInspectorBarChartF19 and itfInspectorBarChartF19 are in the directory:

“DecisionUnits\src\pa_v38\interfaces” which contain the following two methods:

String getBarChartTitle(): Returns name of bar chart and is called by the classes of defense mechanisms.

HashMap<String , Double> getBarChartData(): Returns the current set of data as HashMap which contains of data types String for the current defense mechanism and Double is the value of numbers of the altered drives or perceptions. This function is called in the classes of the modules F06 and F19 and returns the HashMap with the numbers of altered drives or perceptions and the corresponding defense mechanism.

The DecisionUnitMasonInterface includes the both classes cls_BarChartInspectorF06 and cls_BarChartInspectorF06 which contain the following methods:
initChart(): Initialization of data of bar chart is done by calling the method JFreeChart.CreateBarChart(), which contains of the following information titel, label for category axis, label for the value axis, dataset, and the orientation.
DefaultCategoryDataset(): Default value of the numbers of altered drives and perceptions. This is done by calling the function “getBarChartData()” from the interfaces itfInspectorBarChartF19 or itfInspectorBarChartF06.
updateDataSet(): Deal with updating data in bar chart by calling in every step of the simulation the method getBarChartData().

Time Chart Inspectors

The used methods in the interface itfInspectorCombinedTimeChart are as the following:
getCombinedTimeChartAxis(): Returns label for Y-axis. This is in case of the present work 0 or 1 (inactive or active)
getCombinedTimeChartData(): Returns an array that contains the current set of data with data types as Double. These values are the number of simulation steps.
getChartTitles(): Returns the labels for all sub charts, which indicates the names of defense mechanisms.
getValueCaptions(): Returns an array that contains the labels of the values of all sub charts.
moTimeInputChartData(): Returns HashMap that call “moTimeChartData” with data of time chart and bar chart where the keys are defense mechanisms and their association values are number of altered drives or whether specific defense mechanism is active or not.

GetCombinedTimeDefenseYaxisData(): This function deal with Y-axis data of time chart, which is ‘1’ if defense mechanism is active or’0’ if it is inactive. If a specific defense mechanism is active the others have to be reset.

getCombinedTimeChartData(): This function indicates for time chart which defense mechanism is active and with this method “moTimeChartData.get("TimeReactionFormation"))” the value is returned which is ‘0’ for inactive or ‘1’ for active.

getChartTitles(): Returns the names of different defense mechanisms that are used to view them

getValueCaptions(): This function is used to get the data from moTimeInputChartData(), which return HashMap that have value for each defense mechanism and returns ArrayList.

B. Curriculum Vitae

PERSONAL INFORMATION

Full Name	Younes LOTFI
Citizenship	Austria
Date of Birth	11.10.1973
Place of Birth	Marrakech

EDUCATION

02/08-	Master studies at the Vienna University of Technology - Faculty of Electrical Engineering and Information Technology
09/01-03/07	Bachelor of Technical Computer Science at the Vienna University of Technology - Faculty of Informatics
03/98-02/94	Study of Chemistry at University of Cadi ayyad, Marrakech Morocco
06/93	School graduation at Lycee Hassan 2, Marrakech Morocco

PROFESSIONAL EXPERIENCE

Since 2007	Privacy commissioner and working for IT department of JAW company
03/01-04/06	Web design and development, and IT consultant at “Verrein zur Förderung der Spiel kultur”