

# MODELLING THE COGNITIVE IMMUNE SYSTEM THEORY WITH A LEARNING CLASSIFIER SYSTEM

Daniel Voigt, Henry Wirth, and Werner Dilger  
Chemnitz University of Technology  
Straße der Nationen 62, D-09107 Chemnitz, Germany  
Email: {davo, henw, dilger}@informatik.tu-chemnitz.de

**ABSTRACT:** Learning Classifier Systems have been used by several authors as a tool for modelling the immune system. In this paper we present a modification of the LCS that can be used for modelling the Cognitive Immune Theory introduced by I. Cohen. We describe how the characteristic features of the theory, namely degeneracy of recognition and context of immune reactions, can be realized in this modified LCS. Furthermore, we introduce the representations of the immune agents, the phases of activity that take place and the applied evolutionary mechanisms. The computational immune model has been implemented and the various parameters of its operational cycle are presented.

**KEYWORDS:** Cognitive Immunology, Learning Classifier Systems, Immune Cells, Cytokines, Affinity

## 1 INTRODUCTION

Biological systems have always been a rich source of inspiration for the field of Computational Intelligence (CI). For instance, the adaptive algorithms and connectionist architectures derived from the central nervous system of higher mammals have been successfully applied to problems of machine learning and pattern recognition. Evolutionary approaches have proved to be well-suited for the solution of optimization problems.

The body of vertebrates contains another interesting control system that plays an important role for the homeostasis of an individual, namely the immune system. Since the fundamental findings of Pasteur and Koch and in particular during the last 50 years biologists and physicians have made great progress in detecting the mechanisms that contribute to the overall functioning of the immune system. Biologists already began to formalize their descriptive approaches, cf. for instance the work of Perelson/Oster [1]. This was the starting point for developing Artificial Immune Systems (AIS) as a new field of research where the principles and mechanisms of the immune system were applied as problem solving methods to different kind of problems.

The computational systems developed in AIS were based on one of the leading theories in the field of immune systems, namely Burnet's Clonal Selection Theory [2] and Jerne's Network Theory [3]. In particular the Network Theory attracted many researchers, maybe because computer scientists are familiar with networks which they use for quite different purposes, maybe because they offer a link to the Neural Networks which are widely used in CI. In recent years, I. Cohen has suggested a new approach which is based on the Network Theory but goes far beyond it (see [4]). He considers the immune system as a cognitive system which can sense certain environmental aspects, build up an internal representation of it, and make decisions about actions that are required to keep the homeostasis of the individual. Special features of his theory are the degeneracy of recognition events, which is in sharp contrast to the assumption of monospecificity, and the embeddedness of immune activities in a context that is created by immune agents.

Some authors (see [5] and [6]) have shown how the Learning Classifier Systems (LCS) can be used to implement a model of the immune system. In this paper we present an modification of the LCS and show how the Cognitive Immune Theory can be modelled and implemented in this system. In chapter 2 the classic form of the LCS as suggested by J. Holland is introduced. Chapter 3 gives a short introduction to the basics of the Cognitive Immune Theory. Chapter 4 describes how we model the theory in the modified LCS. The implementation of the system and some parameter settings are presented in chapter 5. Finally, chapter 6 contains some remarks about future developments.

## 2 LEARNING CLASSIFIER SYSTEMS

In this chapter we briefly describe the basic components of the machine learning paradigm that was introduced by John Holland in [7] and is summarized under the term "Learning Classifier Systems" (LCS). In time there have been several

revisions and variations of the underlying architectures and algorithms and so it is difficult to pick out one basic LCS standard form. In our modelling approach presented here we concentrate on a form which is almost identical to the one outlined by Holland in [8]. We took this “classic” LCS as a starting point and modified its structure and its internal mechanisms according to the principles of Cohen's Cognitive Immune Theory. In the first section, we present the computational system as a whole and introduce its internal representations. After that, we look at the subsystems of the LCS and describe their functioning. The operational cycle of the system is presented in the last section.

## 2.1 OVERALL VIEW

Basically, LCS are a rule-based evolutionary approach to machine learning and can be understood as an internal control mechanism of an autonomous agent that is situated in an unknown environment. By means of sets of *detectors* and *effectors* the agent is able to interact with the objects in its surrounding. The specific perceptions that are received by the set of detectors (e.g. cameras and measuring instruments) are internally coded as binary *messages* and subsequently matched with the set of rule strings that in their entirety make up the agent's world model. In addition to these newly acquired observations the agent's previous decisions are also considered in form of an internal memory state. The rules that fit the current situation best are selected and thus are potentially in the position to determine the next action of the effectors (e.g. positional change or move of a grip arm). Depending on the appropriateness of the performed action, the environment occasionally gives a *reward* to the LCS. Through the incorporation of this environmental feedback into the structure of the world model the agent is able to learn from experience. So in future situations the agent is likely to behave in the desired manner. If required, the LCS inserts new rules into the existing world model by the use of heuristic methods. These new rules have the opportunity to prove their usefulness in controlling the agent's behaviour and eventually lead to a reward. In case these rules turn out to be unproductive or obstructive, they will be replaced by others. On account of the fact that its internal learning process never comes to a final halt, the autonomous agent has the ability to adapt itself to a perpetually changing environment.

The world model of the LCS consists of a population of *classifiers*. Each classifier is represented by a simple production rule and has a numerical parameter – called *strength* – assigned to it. The strength value is automatically altered during the learning phase (see section 2.3) and gives an estimate about the respective classifier's average efficacy in controlling the reactions of the agent in the past – so a high value stands for an efficient rule that achieves rewards quite often and a low value corresponds to an essentially unsuccessful rule. A single classifier is a compound of two rule parts: the *condition* and the *action*. The classifier condition is a string with fixed length that is made up of symbols of the alphabet  $\{0, 1, \#\}$  and is compared to all available messages during the matching procedure (see section 2.2). As it represents either a “0” or a “1” in the condition string, the “#”-symbol is commonly called *don't care*. The accompanying action part of a classifier consists of a fixed length string of symbols of the alphabet  $\{0, 1\}$  and thus is a message itself. In order to interact with the environment and to adapt its own behaviour according to the received feedback, the agent's internal classifier population is processed by means of three LCS-subsystems: the *performance component*, the *credit allocation component* and the *rule discovery component*.

## 2.2 PERFORMANCE COMPONENT

The performance subsystem contains all LCS mechanisms that ensure the agent's capacity to act in its environment. In particular this includes the acquisition of sensor inputs, their internal processing and the selection of an appropriate reaction. As already mentioned above, the LCS is equipped with a set of detectors that constantly take samples of the environment and encode these signals into fixed length message strings of the alphabet  $\{0, 1\}$ . So for example, an arbitrary detector scans the surrounding for the presence of green objects and indicates their appearance by putting a “1” at the first bit of the binary message, while another detector is coding red objects to the second bit. In the specific situation where all objects in the environment are red and no green objects are observable, the detectors would deliver the message “01”.

All the messages describing external observations are subsequently placed in an internal *message list*. So to say, this list holds a simplified description of the current environment that will be processed in the following steps. First of all, each message of the message list is compared to the condition part of each classifier in the current rule population. The condition of a single classifier is matched by a message if all bits of both strings are identical – respectively, a “#”-symbol stands for either a “0” or a “1”. So for example, the condition string “1#0” is exclusively matched by the two messages “100” and “110”. All the classifiers with a satisfied condition string are added to the so called *match set* and take part in the following competition round.

The purpose of this competition is to decide which classifiers will be allowed to publish their action string and as a result will influence the next performed action of the agent. To this end an *auction* between all elements of the match set is held which starts with a *bidding process*. For each satisfied classifier a *bid* is calculated that takes into account the current strength of the classifier and the *specificity* of its rule string. The specificity value of a classifier is the ratio

between the amount of non-“#”-symbols in the condition part and the overall length of the condition string. The bid of a classifier is defined as follows:

$$bid = c \cdot strength \cdot specificity \quad (1)$$

where  $c$  is a constant for all classifiers. As mentioned above, the strength of a classifier is an average estimate of its past performance and so an useful rule with a high value can potentially bid more in the auction than its weaker competitors. In order to prevent the dominance of over-general classifiers (rules with condition parts that almost entirely consist of don't cares) the bid is multiplied by the specificity. In case of two equally strong classifiers the one with the higher specificity can place the higher bid and as a result is more likely to win the auction – the LCS has the tendency to prefer specialized rules if there are several candidates to choose from.

After all classifiers in the match set have placed their bids, a stochastic *roulette wheel selection* takes place. Each classifier is assigned a certain section of the wheel in accordance to its bid. The winners of the auction are selected by repeatedly turning the virtual roulette wheel and adding them to the *winner set*. In the next step these winning classifiers get the chance to publish their action strings in the message list. The old messages in the list are swapped for the ones that are produced by the currently active classifiers. These internal messages may lead to the activation of other classifiers in the next computational cycle – even without any external messages from the detectors. Therefore, the LCS is able to show a kind of eigen-behaviour by processing its own messages. This internal activation of classifiers is called *rule chaining* and enables the agent to solve complex multi-step problems.

Besides the fact that classifiers may activate each other by means of publishing messages, they are also able to instruct the agent's effectors to perform actions. To this end, all messages in the message list are passed to the effectors as well. So for example, a certain bit of a message may activate the effector that is responsible for grabbing red objects while another one may activate the effector for green objects. In case there are contradictory actions to perform, a *conflict resolution mechanism* is applied that decides which action to give priority in execution.

### 2.3 CREDIT ALLOCATION COMPONENT

After the effectors have performed the instructions of the active classifiers, the environment delivers a feedback about the agent's behaviour. This feedback is expressed in form of a real valued scalar that represents the reward (or the penalty) that is handed over to the agent – according to the actions' efficacy in regard to the current environmental state. Thus, useful actions are rewarded with positive and unsuitable (or even harmful) actions with negative feedback values respectively. Based on this reinforcement information the strength properties of the elements of the classifier population are adapted in the next step. Because of the altered bidding situation, certain actions become more likely to appear in the future than others and as a result, the agent is able to achieve rewards more often. Due to the fact that the feedback of the environment may be received in an irregular and delayed manner, the efficient distribution of reward to the responsible classifiers is a difficult task for a learning algorithm. Even if the environmental feedback would be available to the agent without delay, it would be disadvantageous to give reward only to the classifiers that triggered the last effector actions. Because of the ability of classifiers to activate each other by means of passing internal messages, the pioneering effects of the rule chain must be taken into account when allocating the received external feedback. As with all constituent parts of a LCS, there have been suggested several mechanisms for solving this task. In the following we outline an approach for the adaptation of classifier strength that is commonly called – according to its underlying idea – the *bucket brigade algorithm* (for details see [8]).

The value of the environmental feedback that is received in time step  $t$  is distributed among all active classifiers of that step by adding it to their respective strength. In addition to that, certain proportions of strength are passed between the classifiers that have won consecutive auction rounds. The current auction winners transfer their bids to the winners of the previous round that actually made it possible for them to become active by publishing apt messages. So on the one hand, a classifier loses strength by being activated, but on the other hand, it regains strength by the transfers of the subsequent auction winners. Thus, a classifier can grow stronger if it is part of a rule chain that directly leads to a reward. The strength value of a classifier that only publishes inappropriate messages and does not cause environmental reward at all will come down to a point where winning an auction is almost impossible. In general, the strength of an arbitrary classifier at time step  $t + 1$  is defined as:

$$strength(t+1) = strength(t) + reward(t) + payment(t) - bid(t) \quad (2)$$

where  $strength(t)$  is the strength of the classifier at time step  $t$ ;  $reward(t)$  is the feedback value that is received from the environment at time step  $t$ ;  $payment(t)$  is the amount of all bids paid by the classifiers of the next round that are satisfied by the message of the respective classifier;  $bid(t)$  is the bid of the classifier (see (1)). So for the efficient adaptation of the classifier population the bucket brigade algorithm only requires local information about the auction winners of the previous time step.

## 2.4 RULE DISCOVERY COMPONENT

Through the use of the credit allocation component, the agent is able to dynamically map the regularities of the environment by self-adapting the strength values of certain classifiers. Yet, the agent cannot extend its internal world model by including new building blocks – its abilities are limited so far to optimizing the existing knowledge structure. In order to generate these necessary new classifiers the LCS makes use of a standard *Genetic Algorithm* (GA). The elements of the classifier population are interpreted as chromosomes whose genes are determined by the binary (and ternary) alleles of the accompanying rule strings. The classifier's strength parameter is taken as the *fitness* value of the individuals that are subject to a simulated evolutionary selection process consisting of three consecutive steps:

- 1) **REPRODUCTION:** The GA selects classifiers from the population with probability proportional to their strengths (by means of roulette wheel selection). The selected rule strings serve as parents for new classifiers.
- 2) **RECOMBINATION:** The parent strings are crossed and mutated by means of genetic operators. First, the bit-strings are cut in a random position and recombined in an opposite manner (so called 1-point-crossover) and second, the values of the bits in the resulting strings are changed with a certain probability (so called mutation). Finally, the parents' average strength value is applied to the offspring classifiers.
- 3) **EXCHANGE:** The newly generated classifiers are inserted into the initial population at the expense of existing weak classifiers. The classifiers that are being replaced are chosen by means of roulette wheel selection based on the reciprocal of the current strength value.

Because the proper fitness value of a classifier is not easily available and has to be evaluated through the continuous working of the performance and the credit allocation subsystem, the GA should not be applied in every round. The strength values of the classifier population need a certain time to stabilize and function as a representative estimate of the classifiers' actual fitness. Because of that, the rule discovery mechanism is only applied after the periodic completion of a certain number of LCS rounds. In a collaborative effort between the bucket brigade algorithm and the GA the agent is enabled to learn. While the former algorithm realizes an exploitation step, the latter is responsible for the exploration step.

## 2.5 SUMMARY

The overall LCS and its computational cycle is shown in figure 1.

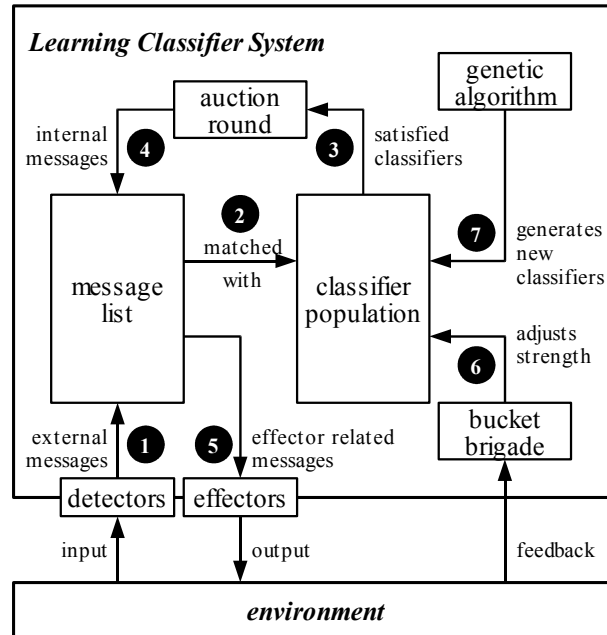


Figure 1: LCS overview (according to [9])

At the beginning of the LCS cycle, the rule strings of the classifier population are randomly initialized and a predefined strength value is assigned to all classifiers. Then the message list is cleared and the system enters the following computational cycle: (1) The detectors add messages to the message list that describe certain observations on the environment. (2) Each classifier is compared to each message located in the message list. The classifiers that are completely satisfied

are added to the match set. (3) An auction round is held between all classifiers of the match set. The set of winners is stochastically determined based on the bids of the classifiers. (4) The message list is cleared and all classifiers of the winner set publish their respective action string as a message in the message list. (5) The effectors read the relevant messages and put into effect the accompanying actions in the environment. (6) If there is feedback from the environment, the bucket brigade adapts the strength values of certain classifiers. (7) If necessary, weak classifiers are replaced by new classifiers by means of the GA. (8) The internal cycle of the LCS is restarted from step (1).

### 3 COGNITIVE IMMUNE SYSTEM THEORY

In this chapter we give an overview of Cohen's ideas regarding the cognitive immune system. At first, we look at the general functioning and the role of the system and deal with its cognitive capacities. Then the basic elements and anatomic spaces that constitute the immune system are briefly described. Subsequently, we summarize the interactions of the immune elements and the activity phases that they are subject to. On account of the fact that we are particularly interested in the ideas that can be utilized algorithmically, the biological details are left aside as far as possible. Rather than giving a detailed introduction to Cohen's theory, we concentrate on the relevant concepts for understanding our modelling approach described in chapter 4. For in-depth information on Cohen's immune theory see [4].

#### 3.1 THE COGNITIVE IMMUNE SYSTEM

In general, the immune system is an embedded system that maintains the body and protects it from malign influences that come from the outside or the inside of the organism. These factors are summarized under the term *antigens*. While doing so, the immune system incorporates the states of the body's tissues and thus provides an immune response that is based on *context*. This differs from the classical point of view that the main purpose of the immune system is to defend the body by discriminating between self and non-self and triggering a monospecific response (see Burnet's Clonal Selection Theory in [2]). Cohen characterizes the immune system as a cognitive system on the basis of three essential properties: it is able to *make decisions*, it creates *internal images* of its world, and it is able to *learn* in a *self-organized* manner.

#### 3.2 ELEMENTS AND SPACES

The immune system consists of a multitude of specialized *cells* and *organs*. The immune cells include the lymphocytes, monocytes and granulocytes. The organs divide the body into distinct compartments that the immune cells flow through and sense antigens. Certain compartments of the organism are responsible for producing the immune cells, others for transporting them, and again others contain the manifold interactions between the cells. As a result, an immune response is elicited that fits the respective antigenic situation and protects and maintains the body as a whole. From our perspective, the relevant anatomic spaces are the *bone marrow*, the *thymus*, the *lymph nodes* and the *body tissue*. While the bone marrow produces fresh immune cells, the thymus selects those cells that are immunocompetent and are allowed to enter the body. By means of the blood and lymph vessels the mature immune cells circulate between the body tissues and the lymph nodes. On their way through the tissues the cells gather information on the tissues' state, the presence of antigens and the activity of other immune cells. This information is available through abstract molecular shapes that are presented by all antigens and tissue cells. After completing this *recognition phase*, the immune cells move to the lymph nodes and exchange the gathered information by producing certain communication molecules that are summarized under the term *cytokines*. Then the immune cells jointly make a decision on an appropriate response considering the detected state of the body. Following that *decision-making phase*, the immune cells move back to the tissues to put the decided actions into effect. After finishing this *execution phase*, the immune cells re-enter the recognition phase and continue to gather updated information in the body.

#### 3.3 PHASES OF ACTIVITY

The three phases of immune cell activity continuously follow each other and thus can be described as the *functional loop* of the immune system. In each phase *spatial proximity* is an important prerequisite for the interactions between immune cells and antigens and for the interactions between immune cells alone.

## Recognition Phase

In this phase the immune cells filter out the relevant molecular signals from the total set of signals that are available in the body. The ability to discriminate between relevant and irrelevant signals is called the *specificity* of recognition. All immune cells bear *receptor* molecules on their surfaces that enable them to recognize other molecules. These recognized molecules are summarized as *ligands* and can be antigens, immune cells and other body molecules. The receptor of an immune cell consists of a *binding site* and a *reaction site*. The former is the portion of the receptor where a ligand can bind and the latter is the portion where a successful recognition event is signalled to the cell. The act of recognizing is explained best with the help of a metaphor: the binding site of a receptor is interpreted as a key and the ligand as a lock. Here, a specific interaction between the key and the lock has to take place – a key that actually does not have the proper shape will obviously not open the door. Like the metaphor pair, the binding site and the ligand both have certain (molecular) shapes that have to match in a complementary manner. Only if this complementarity is satisfied, the door opens and the receptor's reaction site indicates a recognition event.

The characteristic property of Cohen's theory is the fact that a receptor is able to bind more than one single ligand. So to say, an immune cell receptor corresponds to a general key that also opens locks with similar shapes and not just the one with the exact complementary shape. This flexible mapping between receptors and ligands is called *degeneracy*. The term *affinity* describes the specific binding energy between receptor and ligand that arises from their degree of molecular complementarity – the higher the affinity the higher the probability of a successful binding event. But besides that, the receptor's recognition is also influenced by the concentration of the respective ligand. So a relatively low ligand concentration with high affinity already causes the receptor to bind, but the low affinity of a ligand can be compensated by a high concentration as well. The shapes of the receptors' binding sites can be seen as the *internal images* of the immune system. The immune cells reflect the negative shapes of the antigens through their receptors, and thus are provided with a description of the objects that can be interpreted as their environment.

Furthermore, the receptors of the lymphocytes are created somatically. This means, that these immune cells can manufacture their receptors epigenetically from genetic raw material. While a receptor's reaction site is genetically determined by the germ-line of the organism and thus is limited to a few possible shapes, the genes of a receptor's binding site are composed individually by rearranging short pieces of DNA by chance. After that, a random sequence of mutation, deletion and addition operators is applied to the segments of the receptor's genetic "building plan". On account of the immune cell's somatic influence the possible shapes of a binding site are enormous, and therefore the receptors are potentially able to bind every single molecule that might occur in the body. In contrast to the lymphocytes, the receptors of the monocytes are manufactured without somatic modification – the possible molecular shapes of their binding and reaction sites are exclusively determined by the inherited genes of the germ-line. As a result, the monocytes can only sense antigens coarsely. Apart from the antigen receptors, the immune cells also bear receptors for cytokines. This special class of molecules is used mainly for intercellular communication and regulation. The surrounding tissue cells are also able to produce cytokines and therefore take part in the communication process. The set of available cytokines is genetically determined and the same applies to the set of cytokine receptors that is expressed by the immune cells.

Because the total amount of available immune cells in the body is (in terms of figures) smaller than the potentially dangerous molecules (pathogens), the cells' perceptions are focussed. So for example, certain lymphocytes (B-cells) sense the molecular shapes of antigens directly while other lymphocytes (T-cells) need the assistance of monocytes that present preprocessed antigen molecules to them. Since the immune cells focus on different aspect of the antigenic world in the recognition phase, a distributed multi-level perception is achieved, that yields a detailed image of the whole body's state. This molecular image is the starting point for the cellular interactions of the next phase where an appropriate immune response is selected.

## Decision-Making Phase

After the immune cells have gathered the molecular signals in the tissues of the body, they move to the lymph nodes and mutually exchange their observations. The cells react in accordance to their recent perceptions and as a result influence the reactions of other cells and thus the immune response as a whole. Consequently, an immune response is the product of a second order process: on the one hand, the reactions of cells in regard to certain target molecules are integrated, and on the other hand, the reactions of cells in regard to the reactions of other cells are integrated as well. Cohen terms this special interaction between immune cells their *co-response*. While co-responding, the immune cells produce various interaction molecules, but cytokines in particular. These cytokines can have stimulating and inhibiting effects on the immune cells. Furthermore, their exerted effects can even change in the course of an immune response and so they are an important factor in setting off and influencing the co-response. By means of cytokines an intercellular feedback mechanism is available to the immune system that helps to control the response.

The lymph nodes are distributed all over the body and are the anatomic space where most of the co-response takes place. Because of the blood circulation, in a lymph node local and global information is brought together. The

former is delivered by the immune cells that have passed their cycle through a local tissue sector and the latter by the immune cells that have returned from a non-neighbouring tissue. So the overall state of the body is always considered locally and the observations of the immune cells prepare the context for appropriate immune decisions. These decisions consist of associations between special cases of perceptions and general classes of reactions – the immune system has to *choose from several options*. Considering this, certain lymphocytes function as *memory cells* that are a result of a successful maintenance event that took place in the past. These cells proved to be efficient in causing an apt immune response, and therefore changed to this cellular memory state. So in case of repeatedly occurring antigens, the memory cells can speed up the necessary immune decision and reproduce an efficient response to the antigen more quickly. In general, the decisions are a joint product of the mutual interactions between populations of immune cells.

### *Execution Phase*

In the last phase of immune activity, the immune cells spread across the body tissues once more and execute the decision that they have mutually come to. Despite the degenerate perceptions of the cells, the immune system shows the ability to respond specifically. Certain molecular inputs that are sensed fuzzily by the immune cells, are subsequently processed through co-responsiveness and as a result lead to specific decisions that are realized by specific molecular outputs. These outputs are brought about by *effector cells* that are able to influence the other cells of the body in various kinds: they can have effects on the growth and the reproduction rate of tissue cells; they can induce the death of cells and stimulate their ability to move; they also can influence the cells' differentiation and regulate the functioning of the tissue's supply and support systems. Beyond that, certain immune cells can attack pathogens directly. Most of this influencing is realized by the cells by producing cytokines and interaction molecules that specifically counteract the detected threat to the organism's health.

### 3.4 SELECTION STEPS

In order to structure the more or less randomly generated population of immune cells and to ensure the functioning of the immune system, the repertoire undergoes *four evolutionary mechanisms* that sort out inefficient elements. Actually, not all selection steps are applied to all types of immune cells, so for example, a certain subset of the lymphocytes is only subject to two of the four steps. But for the sake of brevity, in this section we will generalize over the whole immune repertoire and will not go into the details of the evolutionary mechanisms too much. The first selection step takes place during the maturation process in the thymus where the new immune cells move after being produced in the bone marrow. In general, only those cells survive this evolutionary step that are able to bind molecules that are derived from the self of the body. Thus, the cell's ability to recognize the organism's molecules is tested. But as a restriction, the affinity between receptor and tested molecule is not allowed to be too strong nor too weak – only moderately binding cells survive and enter the circulation. All other cells are deleted. Subsequently, in the tissues of the body the mature immune cells get in contact with the various molecules. In case of high affinity, the cells are selected and become active. So the second selection step gives preference to immune cells that are able to recognize the molecules of the altered self. The cells that do not have this ability will disappear in time and make way for new cells in the repertoire.

A cell that becomes activated by a successful recognition event reproduces itself and inherits its receptor genes to its offspring. However, the daughter cells somatically modify the receptor genes once again – this process is called *hypermutation*. As a result, the daughter cells bear receptors that are similar to the shape of their mother's receptors, but show individual deviations. All the mutated cell clones that have a high affinity in regard to the antigen that caused their mother's reproduction are selected by the third selection step and are allowed to reproduce themselves. By means of mutation and subsequent selection, the immune cells' affinity is raised. This process of improving the antigenic perception of immune cells is called *affinity maturation*. In contrast to the three previous selection steps that all form the immune cells' perceptions, the fourth selection step determines the effects that result from the perceptions. Those cells are selected from the immune repertoire that show an adequate response in regard to certain molecular inputs. The cells that undermine the immune system's body maintaining function and endanger the organism's survival are deleted. As a result of the *self-organized* workings of the four complementing selection steps an immunocompetent cell repertoire emerges and the immune system *learns* to handle the influences from the antigenic environment efficiently.

## 4 OUR MODELLING APPROACH

In the following chapter, we bring together those two systems whose basic elements and mechanisms we have summarized before: we take Holland's "classic" LCS as a starting point for outlining a computational model of Cohen's immune system theory. In doing so, we establish points of contact between the two approaches, and if necessary, sug-

gest suitable modifications for the classifiers of the LCS and the algorithms that they are applied to them. We term this version of a LCS that is modified with regard to the characteristic features of Cohen's theory a "Cognitive Immune System" (CIS). At first, we briefly introduce the concepts and representations that are used in our CIS model. After that, we discuss the interactions between the elements of the CIS and the resulting dynamics of the computational system. Finally, we describe the functioning of the evolutionary components that are used for adapting the system's behaviour and summarize the whole CIS by presenting its full computational cycle.

#### 4.1 REPRESENTATIONS

In our model we assume that the classifier population is the counterpart to the immune cell repertoire. Therefore, the anatomic spaces of Cohen's immune theory can be found on part of a LCS as well as the general phases of activity. The *tissue* in its function as the place where the immune cells receive their molecular input corresponds to the *environment* of a classic LCS. This anatomic space contains the antigens and the cytokines that are produced by the tissue cells. Like the objects of the LCS environment that are encoded into messages through a set of detectors, it is the case (in a figurative sense) with the available immune signals that are subsequently processed by the immune cells. Therefore, the detector set can be seen as a kind of algorithmic auxiliary device that has no explicit counterpart in the immune system. Despite the *recognition phase* of a LCS cannot be directly localized in its environment, from an abstract point of view it still takes place there: since the detectors do not alter the received signals and just pass them through to the inner system, the shift of the input space to the LCS message list seems negligible. The LCS equivalents to the *lymph nodes* in their function as a processing space for received signals are the *message list* and the *auction round* parts of the performance component. In accordance to the co-responding interactions of immune cells, in these LCS parts the classifier population reacts to the encoded environmental observations and its own internal reactions. Thus, the computational sequence of the LCS that consists of the matching procedure, the auction round and the publishing of internal messages can be seen as an analogy to the *decision-making phase* of the immune system. This phase results in immune actions that are subsequently executed by certain cells in the tissues of the body. The same applies to the effectors of the LCS that perform the internally decided actions in the environment, and so their activity corresponds to the immune system's *execution phase*. This effector performance leads to new observations in the environment – like in the biological example, a cyclic activity of the overall system is the result.

Before we can actually have a look at the immune cells' and the molecules' representation in the CIS, we introduce two necessary symbol sets. The first one is the set  $C$  that contains the *cytokines*  $\{c_1, \dots, c_l\}$  where  $l \in \mathbb{N}^+$ , and the second one is the set  $A$  that contains the *actions*  $\{a_1, \dots, a_m\}$  where  $m \in \mathbb{N}^+$ . The former describes the cytokines that potentially can be produced in the CIS, and the latter describes the actions that can be performed by the immune cells. Both sets are determined by the user of the system before starting and remain unchanged during the whole computational cycle. This corresponds to the fact that in the immune system the total set of possible cytokines and actions is genetically limited and cannot be extended. It is for the user to decide which internal coding mechanism is applied to the elements of the sets – so for example, symbolic or numerical representations can be used here. In the end, the actual values and the sets' cardinality essentially depend on the field of application.

In order to model the signal processing chain of the CIS according to the principles of Cohen's theory, we modify the architecture of a classic LCS: we divide the message list into three distinct parts and as a result obtain an *antigen message list*, a *cytokine message list* and an *action message list*. The main inputs of the immune system consist of *antigens* and *cytokines* that are produced by tissue cells. These special molecules can be found in the CIS in form of *messages*. According to this, an antigen is encoded by a string from the alphabet  $\{0, 1\}$  that has the length  $k \in \mathbb{N}^+$ . This binary antigen string is placed by the detectors in the antigen message list and can be interpreted as an abstract description of a molecular shape. They correspond to the external messages of a LCS. A tissue-produced cytokine is represented by an element of the set  $C$ , that was already defined above. The detectors publish the internal representation of a cytokine into the cytokine message list. The cytokine messages that stem from the tissues correspond to the external messages of a classic LCS, but as we soon will see, the cytokine message list also holds cytokines that are counterparts to the internal LCS messages. The outputs of the CIS consist of elements of the set  $A$ , that was also defined above. These descriptions of actions are published in the action message list and are implemented by means of the system's effectors. The action message list has no direct equivalent in the classic LCS.

In our approach the different classes of immune cells (lymphocytes, monocytes and granulocytes) are integrated into one hybrid immune cell type. The structure of such a CIS immune cell consists of the following five parts: an *antigen receptor* string  $R_{Antigen} \in \{0, 1\}^k$ , where  $k \in \mathbb{N}^+$ , that describes the cell's possible perceptions; a *positive cytokine receptor*  $R_{Cytokine+} \in C$ , that describes the cytokine that stimulates the cell; a *negative cytokine receptor*  $R_{Cytokine-} \in C$ , that describes the cytokine that inhibits the cell; a *cytokine response message*  $M_{Cytokine} \in C$ , that is published by the cell in case of activation; and an *action message*  $M_{Action} \in A$ , that is suggested by the cell for execution. While the antigen receptor and the two cytokine receptors correspond to the condition part of a LCS rule string, the cytokine response message and the action message can be compared to the action part of a classifier. Although the syntax of a classic rule

string is modified in detail, the general functional semantics can still be identified. In addition to the receptor and the response parts, each CIS immune cell is assigned a scalar *lifetime*  $L \in \mathbb{R}$ , that gives an estimate about the respective cell's remaining lifetime in the repertoire. This value is analogous to the strength of a classifier in the classic LCS.

## 4.2 INTERACTIONS

After the detectors of the CIS have placed the descriptions of the antigens and the tissue-produced cytokines in the corresponding message lists, a matching procedure between the elements of the immune repertoire and the available immune signals takes place. This mechanism can be seen as a partial equivalent to the recognition phase of the natural immune system. Similar to the matching procedure of the LCS, the elements' mutual interaction potential is identified by comparing each element of the CIS population to each element of the antigen message list and to the elements of the cytokine message list. But the algorithm that is used in the matching procedure of the CIS differs in some regards from the one of the LCS which only considers completely satisfied rule strings. Since this matching criterion is much too strict to meet with the degenerate perception of antigen receptors, we suggest a matching mechanism that is derived from the field of AIS. In order to calculate the antigen affinity  $A_{Antigen}$  between an available antigen message  $M_{Antigen}$  and the antigen receptor  $R_{Antigen}$  of an immune cell, we apply the concept of a *binary Hamming shape-space* (see [10]):

$$A_{Antigen} = \frac{1}{k} \sum_{i=1}^k \alpha^i, \quad \text{where } \alpha^i = \begin{cases} 1 & \text{if } M_{Antigen}^i \neq R_{Antigen}^i \\ 0 & \text{else} \end{cases} \quad (3)$$

( $M_{Antigen}^i$  and  $R_{Antigen}^i$  refer to the  $i^{\text{th}}$  bit of the respective string)

where  $k$  is the total amount of bits in an antigen message or antigen receptor string. The sum yields the amount of bits where the antigen message and the corresponding antigen receptor differ, and thus describes the strings' ability to match complementarily. The actual antigen affinity is obtained by normalizing the value to the interval  $[0, 1]$ . Since only a pair of identical binary strings has no affinity at all, there is a certain degenerate interaction potential between most of the bit strings in the immune repertoire and the antigen message list. Furthermore, the matching mechanism yields the cytokine affinity  $A_{Cytokine}$  between all available cytokine messages and a single cytokine receptor  $R_{Cytokine}$  as follows:

$$A_{Cytokine} = \frac{1}{n} \sum_{j=1}^n \beta^j, \quad \text{where } \beta^j = \begin{cases} 1 & \text{if } M_{Cytokine}^j = R_{Cytokine} \\ 0 & \text{else} \end{cases} \quad (4)$$

( $M_{Cytokine}^j$  refers to the  $j^{\text{th}}$  cytokine message of the list)

where  $n$  is the total amount of available cytokines in the cytokine message list. The matching procedure only takes into account whether a successful recognition event on part of the receptor takes place or not. Instead of testing the degenerate matching of complementary molecular shapes, the generalized cytokine matching procedure focusses on the recognition event itself. The overall affinity of an immune cell in regard to an actual immune situation results from the integration of the partial affinities: first, the antigen affinity is calculated for the cell's antigen receptor (using (3)), and second, the cytokine affinities are calculated for the cell's positive and negative cytokine receptors respectively (using (4)). In case of an occurring empty message list, the corresponding partial affinity value is set to zero. On account of the degenerate matching procedure between antigens and antigen receptors, a choice between several possible antigen-receptor-pairs has to be made. Since the antigen concentration also influences the interaction potential of the pair, we suggest the use of an *affinity-based roulette wheel selection* like in the auction round of the classic LCS. By means of this stochastic step a single antigen from the antigen message list is selected, that subsequently is taken as a basis for calculating the overall affinity  $A_{Overall}$  according to:

$$A_{Overall} = f(A_{Antigen} + A_{Cytokine+} - A_{Cytokine-}) \quad (5)$$

$$\text{where } f(x) = \begin{cases} \frac{x}{2} & \text{for } x > 0 \\ 0 & \text{else} \end{cases}$$

Thus, the term  $A_{Antigen}$  describes a kind of "basic activity" with regard to an antigen, that is stimulated or inhibited by the following terms  $A_{Cytokine+}$  and  $A_{Cytokine-}$  respectively. Since the function  $f$  maps  $A_{Overall}$  to the interval  $[0, 1]$ , the result of the affinity calculation cannot be a negative value. The overall affinity is calculated for each cell of the immune repertoire. These values are the *potential activation* of the immune cells in regard to the current immune situation that is represented by the elements of the message lists. In order to determine from the set of all available immune cells the ones that actually become active, in accordance to the LCS an *immune auction round* is held. This restriction of potential cell

activity can be seen as a computational equivalent to the spatial distribution of the immune cells in the body. According to this, an immune cell only gets activated if it has the necessary proximity to the target molecule (under the assumption that there is sufficient affinity). Therefore, the auction round can be interpreted as the CIS counterpart to this anatomical restraint. In this selection process the immune cells place their respective overall affinity  $A_{Overall}$  as their “bids”. Subsequently, the set of auction winners – and thus the set of active immune cells – is obtained by turning the wheel. Furthermore, the lifetime of the cells that remain inactive in this round is reduced by a certain predefined amount. Accordingly, an immune cell is removed from the repertoire, if the value of its remaining lifetime falls below a certain threshold – the purpose of this automatic reduction of lifetime is to increase the selection pressure in the immune cell population as a whole (see section 4.3).

As a result of their activation the immune cells produce co-response signals. In the CIS these intercellular signals are modelled as internal cytokine messages that are published by the active immune cells after completing the auction round. Before entering the equivalent to the decision-making phase of the immune system, the cytokine message list is cleared. Then the activated cells publish their accompanying cytokine response messages and jointly create a new cytokine pattern in the message list. In the course of the next CIS cycle, this pattern is supplemented by the tissue-produced cytokines that are sensed through the system's detectors. Therefore, an updated cytokine context is created that influences the future processing cycles. By means of the cytokine message list the immune cells exchange information and react to this with new information. Besides that, the co-response process serves as an internal feedback mechanism. Due to their two cytokine receptors, the immune cells are able to receive positive and negative feedback at the same time. Actually, a high affinity alone will possibly not lead to a cell's activation. In case of a simultaneously occurring inhibiting cytokine message, the overall affinity of the immune cell would be decreased by a certain amount (depending on the cytokine's concentration) and the activation of the cell would become unlikely. In the same way, certain cytokines can compensate a cell's lack of antigen affinity and increase its activation probability. Even the occurrence of antigens is not a requirement for an immune cell to become activated – this already can be achieved by the mere influence of a sufficient amount of stimulating cytokines. This kind of mutual cell activation through internal cytokine messages corresponds to the rule chaining of the LCS. Generally, the cells' antigenic perceptions arise from the total context of the currently available cytokine messages.

The co-response process is a joint effort between all active immune cells, and as a result yields an actual immune decision that influences the future actions of the effector cells. In the CIS the result of the decision-making phase is modelled by the fact that the active immune cells suggest actions by publishing their respective action message in the corresponding message list. So the *support* for an arbitrary action is defined as the amount of messages in the action message list that actually suggest this action as the next one to be executed by the effectors. In order to determine the next performed action, a *support-based roulette wheel selection* takes place. The action message that is selected by this mechanism is passed to the effectors of the CIS and subsequently executed in the tissues. It might be promising to allow the system to pass a certain amount of computational cycles and to consolidate its support values before the contents of the action message list are utilized for immune decision-making.

### 4.3 EVOLUTIONARY MECHANISMS

In addition to the interaction procedures, the CIS is equipped with an algorithmic component that models the bone marrow's function as the immune organ where immature immune cells are produced. The constituent parts of a new immune cell are produced as follows: as an analogy to the cells' ability to manufacture antigen receptors somatically, the CIS antigen receptor string is created by chance from the symbols of the binary alphabet; due to the genetic restrictions of the cytokine repertoire, the symbols that are assigned to the cell's two cytokine receptors and the respective cytokine response message are derived from the predefined set  $C$ ; accordingly, the action message is derived from the set  $A$ ; the cell's lifetime parameter is set to a predefined constant. In correspondence to the first selection step that takes place in the thymus, all immature immune cells are subject to a testing mechanism that decides whether these cells are added to the immune repertoire or not. For this test we introduce the user-defined set  $S$  of *self messages*  $\{s_1, \dots, s_o\}$  where  $s \in \{0, 1\}^k$ , and  $k, o \in \mathbb{N}^+$ . The self messages describe a subset of the message space that is regarded to be part of the body's self and can be interpreted as a filter that prevents certain new immune cells from being inserted into the repertoire. The immunocompetence of a cell is determined by comparing the antigen receptor of the cell to each self message and calculating their mutual affinity (according to (3)). A cell is rejected if its antigen affinity is too high or too low – as a result of this first selection step, only those cells survive that show a moderate affinity to the self messages.

The second selection step of the immune system is realized by means of the CIS auction round that determines the cells that actually become activated. If the inhibiting effects of the cytokines are left aside for a moment, this selection mechanism particularly favours the immune cells that have a high antigen affinity. Accordingly, the activated cells can prove their usefulness in regard to an actual immune situation and can eventually increase their respective lifetimes by suggesting appropriate actions (see below). The inactive cells do not have this chance – because of their low antigen affinity they cannot compete with the high affinity cells and as a result are displaced from the repertoire. This process is

sped up by the decreasing of lifetime that is automatically applied to the immune cells that remain inactive in a round. In order to implement a CIS mechanism that corresponds to the third selection step of affinity maturation, a subset of the activated immune cells is reproduced. This is realized by copying the constituent strings of an immune cell and thus obtaining a set of identical cell clones. While the the daughter cells' respective antigen receptor string is mutated by inverting random position bits, the other parts of the cells remain unchanged. So from this modification process results a set of immune cells that only differ in their antigen receptor string. Subsequently, the respective antigen affinities between the mutated daughter cells and the antigen that caused their mother's activation are calculated (according to (3)). By means of an affinity-based roulette wheel selection one cell is selected from the set of the mother and the daughter cells. This cell replaces the mother cell in the immune repertoire; all other cells are rejected. As a result of the third selection step the immune cells' antigenic perceptions are improved.

The fourth selection step is realized in the CIS through the workings of the LCS bucket brigade algorithm. By means of the feedback that is received from the tissues as a result of performed immune actions, the bucket brigade adjusts the lifetime parameters of the immune cells in the repertoire. As described in section 2.3, the activated immune cells pass around certain proportions of their lifetime and thus lead to the establishing of effective activation chains that receive positive feedback from the tissues. An immune cell that is able to gather sufficient amounts of lifetime can be seen as a CIS equivalent to a memory cell.

#### 4.4 SUMMARY

The functional components of a CIS and its computational cycle is summarized in figure 2.

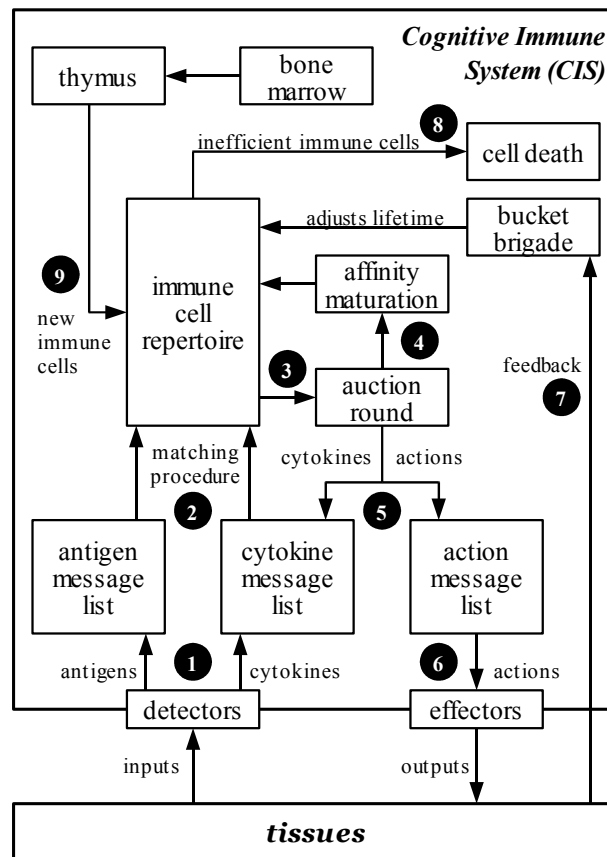


Figure 2: CIS overview

The first step is to clear all message lists and to produce an initial population of immune cells by means of the bone marrow and the thymus. Apart from some structural modifications, the general computational sequence of the CIS is similar to the LCS: (1) The detectors place messages in the corresponding message lists that describe the overall state of the tissues; (2) Each element of the immune cell repertoire is compared to the contents of the antigen and the cytokine message list and the respective affinities are calculated; (3) These potential activity values are taken as a basis for the auction round that selects the immune cells that actually become activated in this round; (4) An affinity maturation mechanism is applied to a certain subset of the active immune cells that is able to improve their antigenic perception; (5) Furthermore, all activated cells publish their respective cytokine and action messages in the corresponding message

lists; (6) If there is enough action support, the effectors perform the suggested actions in the tissues. On the one hand, from these executed actions arise new CIS inputs (1), and on the other hand, the tissues react by delivering feedback to the system. (7) In order to improve the overall behaviour of the CIS this feedback is used by the bucket brigade to adapt the lifetime values of the cells in the immune repertoire; (8) As a result of the evolutionary steps, certain inefficient immune cells are removed from the repertoire; (9) New immune cells are added and the cycle continues at step (1).

## 5 IMPLEMENTATION AND RESULTS

The purpose of the implementation was to create a simulation tool for the cognitive immune system which can be used for experiments. The CIS simulator is implemented in C++ and WinAPI which is used for the interface. The simulator contains classes for the main concepts described in section 4. The interface consists of two windows, one for the parameters by which the parameter values for a simulation run can be preset, and one for the display of the results showing a diagram and some statistics. Each step of a simulation run comprises one internal cycle of the system. The number of steps can be preset by the parameter “duration of simulation”, e.g. 1000. The number of repeated runs can be preset by the parameter “runs”, e.g. 10, in this case the run with the entire number of steps is repeated 10 times.

The CIS simulator provides three simulation variants which can be arbitrarily chosen pushing one of three buttons in the parameter window. By the first variant the normal processing mode of the simulator can be tested, i.e. the classic cycle input - processing - output. One element from the set of patterns is randomly selected and input in CIS as an antigen string. It is processed according to a predefined number of internal cycles. At the end, one element from the list of actions is presented as output and matched against the action in the predefined pattern. Depending on the result a negative or positive reward is generated. The diagram in the result window shows how the accurateness of the result develops over the whole preset number of iterations. The solid line indicates the values of the last run while the broken line represents the average values over the whole number of runs. The statistics present the exact numbers of this development and in addition contains the numbers for the development of the set of immune cells, their average affinity, and their remaining lifetime.

The second variant imitates the immune system. An antigen pattern presented to the CIS is repeatedly processed in the system until the action message list contains an element that fits to the antigen. The maximum number of cycles is limited to ten. The reward is computed according to the number of needed cycles: If only one or two cycles were required, the CIS gets the full reward, for three cycles it gets a reduced reward, and for four or more it gets a punishment increasing till the full punishment for seven cycles. The diagram in the result window shows how the number of needed cycles decreases over the simulation run.

With the third variant the learning capabilities of CIS can be tested. A randomly selected antigen pattern is input in CIS, the internal cycle is processed, and an action is selected from the action message list and matched against the desired action. These steps are repeated until an action is generated that fits to the pattern. After each match at the end of the cycle a reward is given to the system. The diagram in the result window shows how the number of needed cycles decreases, like in the second variant. Figure 3 shows the parameter and the result window for the third variant.

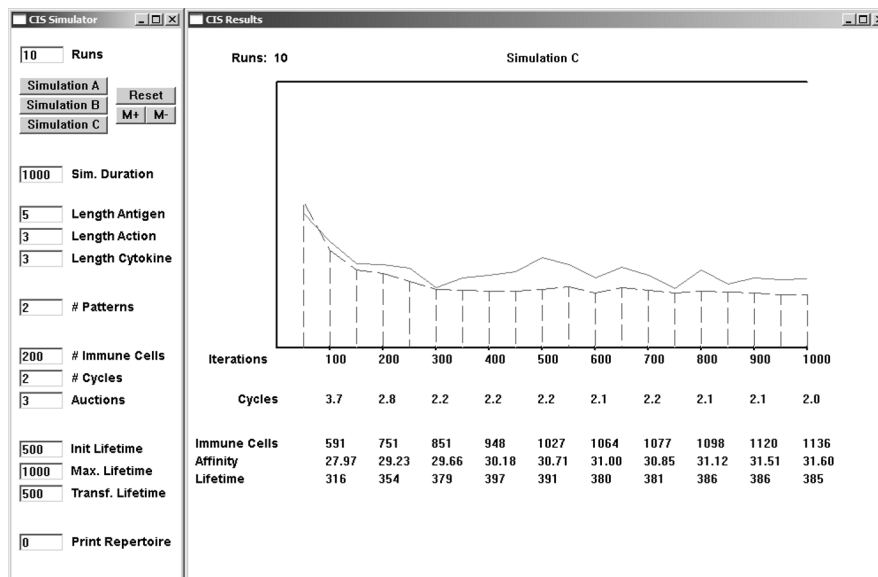


Figure 3: Parameter and result window of the CIS implementation

The parameters that can be preset in the parameter window are the number of cycles, the duration of the simulation run, the length of the strings of antigen, actions, and cytokines, the number of patterns, the starting population of immune cells, the number of cycles, the number of auctions, the initial, maximum, and transferred value of the lifetime parameter, and the print repertoire parameter. By means of this parameter the whole repertoire can be output into a text file at a predefined point in the simulation run. In addition there are the three buttons for selecting one of the simulation variants, a reset button by which a new simulation can be started, and two buttons to switch on or off the memory function of CIS which holds the average values of the current run.

## 6 CONCLUSION

In this paper, we have presented the modification of a classic LCS that can be seen as a model of Cohen's Cognitive Immune Theory. In order to lay the foundations for our modelling approach, we have given a short introduction to the basic elements and mechanisms of the LCS and have described the rule-based system's internal computational cycle. We then have highlighted the characteristic features of Cohen's immune theory, the underlying immune elements and their mutual interactions in time. In particular, here we have concentrated on the immune concepts that can be utilized algorithmically. After that, we have presented the details of our computational model CIS by drawing parallels between the classic LCS and the Cognitive Immune Theory. Where necessary, we have suggested appropriate modifications of the classic architecture and applied mechanisms. The resulting model was termed CIS. As a final step, we have presented our CIS implementation and have explained the parameters that control the overall operations of the system.

As in the case of the classic LCS, the possible field of application for the CIS can be seen in the domain of machine learning and problem solving. In particular, all tasks that involve the context-based processing of signals are in the focus of the CIS (e.g. recognition of noisy patterns). Because of its simple internal representations and general mechanisms the system can be easily adapted to a wide variety of computational problems. Furthermore, the CIS can be used as a general tool for simulating immune learning processes and for examining the question, how specific responses can arise from degenerate perceptions of distributed elements.

## 7 REFERENCES

- [1] Perelson, A; Oster, G.F., 1979, "Theoretical studies of clonal selection: Minimal antibody repertoire size and reliability of self-nonsel discrimination in the immune system", *J. of Theoretical Biology*, 81, pp. 645-670.
- [2] Burnet, F. M., 1959, "The Clonal Selection Theory of Acquired Immunity", Cambridge University Press.
- [3] Jerne, N. K., 1974, "Towards a Network Theory of the Immune System", *Ann. Immunol. (Inst. Pasteur)* 125C, pp. 373-389.
- [4] Cohen, Irun R., 2000, "Tending Adam's Garden: Evolving the Cognitive Immune Self", Elsevier Academic Press.
- [5] Farmer, J. D.; Packard, N. H.; Perelson, A. S., 1986, "The Immune System, Adaptation & Learning", *Physica D*, 22, pp. 187-204.
- [6] Hofmeyr, S. A.; Forrest, S., 2000, "Architecture for an Artificial Immune System", *Evolutionary Computation*, 8.
- [7] Holland, J. H.; Reitman, J. S., 1978, "Cognitive Systems Based on Adaptive Algorithms", In D.A. Waterman and F. Hayes-Roth (eds.), "Pattern-directed inference systems", New York, Academic Press, Reprinted In "Evolutionary Computation. The Fossil Record", David B. Fogel (ed.), IEEE Press, 1998.
- [8] Holland, J., 1992, "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor, 1975, Republished by the MIT Press.
- [9] Nissen, V., 1997, "Einführung in Evolutionäre Algorithmen", Braunschweig, Vieweg.
- [10] De Castro, L. N.; Timmis, J., 2002, "Artificial Immune Systems: A New Computational Intelligence Approach", London, Springer.