A perceptual grounded self-organising lexicon in robotic agents

by Paul Vogt

MSc. thesis for Cognitive Science and Engineering, 1996/1997.

Supervisors:

Dr. Petra Hendriks
Cognitive Science and Engineering,
Rijksuniversiteit Groningen.

Prof. Luc Steels
Artificial Intelligence Laboratory,
Vrije Universiteit Brussel.

CONTENTS

1. INTRODUCTION

In the last century, the research on the origin of language increased from various disciplines like anthropology, linguistics and philosophy. The last few decades this research was given a new impulse from artificial intelligence and lately from artificial life. Most research is focused on the biological and genetical explanation of the evolution of language, which yielded some extreme and diverse speculative hypotheses. The approach that is taken here, however, focuses on a culturally based evolution. This does not mean that we ignore the influence of brain-evolution on the evolution of language, but we think that language *itself* is not genetically evolved, but culturally. I.e. language evolved through means of the social interaction between distributed agents. This research entails experiments on the language formation in robotic agents, and the grounding of the semantics of this language.

In the behaviour-oriented research of artificial intelligence physical robots have been build to investigate the sensory-motor control of these physical bodies. Recently this investigation is focused on the bottom-up approach. I.e. the research on how behaviour can be build up by an agent from its low-level (sensory-motor) control in such a way that the agent autonomously increase its cognitive capabilities. For an introduction to the behaviour-oriented approach of AI see e.g. [38]. There are several methods used in the study of this subject: behaviour-oriented architectures, neural network methods or genetic algorithms. Our approach is based on the behaviour-oriented architectures. Agents directly connect their sensors to motors in order to control behaviour. Architectures that are used may be implemented in subsumption or dynamical system. For details and results on this approach see e.g. [50]. The main question that still needs to be answered is how these robots may increase their complexity, so that they can be classified as cognitive agents. We think that building up a language based on grounded perception is necessary to increase cognitive intelligence [42][51].

Recently more scientists investigated the origination of language on robotic agents. One example of such a research was the one done by Holly Yanco at the AI-Lab at MIT [57]. She used written commands for human-robot communication with one robot. This robot first learned to associate these commands with appropriate actions, and then it tried to teach these associations to another robot in a robot-robot communication. The system was implemented in a neural network using reinforcement learning. The trouble with this system was that human interference was necessary for the formation of the language. So, the robots did not produce the language autonomously, i.e. without the interference of human beings [57]. Another interesting research on robot communication involved imitation [3]. Here one agent followed another in a hilly landscape. The followed robot was the teacher, while the follower was the learner. The teacher expressed words while it was driving in a hilly landscape, and the learner associated these words with its internal representation. The learner used a neural network with associated learning to learn. In principle the learner could learn the right associations, but because there was a delay between the actions of the teacher and the learner, the learner had lots of difficulties learning the right associations [3].

This thesis report on the implementation process of the hypothesis on language evolution that was introduced by Luc Steels (see e.g. [40][41][46]). The idea was to implement the formation of a spatial vocabulary as described in [44]. In this experiment, autonomous agents use self-organisation as a basis for building a lexicon. This lexicon should concern objects in the environment of the agents and their spatial relations (i.e. left, right, front etc.). Language formation is based on the notion of language games [56]. The purpose of the experiment is twofold: (1) To show how a group of distributed agents can build a coherent vocabulary by means of adaptive language games, and (2) to show how autonomous agents may generate distinctions to discriminate between objects in their environment [51]. The basic assumption we make is that the agents already have the mechanisms to communicate, i.e. they know how to communicate, although initially they have no language. The task was only to implement the language games. Language, however, is based on meaning. In the theory described by Steels meaning is represented in the form of distinctive feature sets. These distinctive feature sets can be perceptually grounded by means of the generation and adaptation of features as described in [43].

This report is organised as follows: In the next chapter a brief overview of theories on language evolution will be given. Chapter three will describe the basic mechanisms for language acquisition as introduced by Steels. Chapter four describes the mechanisms for meaning creation. Chapter five defines the physical robots and the environment where they live in. Chapter six discusses how the experiment is defined and proposes solutions to implement the experiment. Chapter seven then discusses the implementation of the language games. Although the intention was to experiment on spatial relations, only naming games [49][51] have been properly implemented. The experimental results are discussed in chapter eight. Furthermore, chapter nine discusses the cognitive and neurophysiological plausibility of the theory of language formation. Finally, chapter ten discusses the conclusions that can be drawn from the whole project, and it also discusses future research.

2. THE EVOLUTION OF HUMAN LANGUAGES

2.1 Introduction

In the latest century many scientists from different areas in science proposed many different hypothesis on the evolution of human languages. These theories are all highly speculative, because we cannot investigate this evolution on very reliable data. The hypothesised theories differ extremely in their proposed fundamentals. Take for example proposed ideas why languages emerged in humans: There are scientists who believe that languages emerged from innate structures in the human brain [7]. Others think that the onset of language evolution was an accident of nature. And still others think it emerged purely on functional grounds (see [12]).

The most well known and influential theory of contemporary scientists is the one suggested by Chomsky (e.g. [7][8]) and some others. Chomsky [8] thinks that language is an innate structure that emerged as a by-product of the evolution by natural selection of other brain structures. Chomsky [8] claims that the innate modular structure of language in the brain itself, called the *language faculty*, did not evolve by natural selection. Although Chomsky has many followers for this theory, for instance (to some extent) S.J.Gould [19], there is also a lot of critique. Most critiques focus mainly on the suggestion that a language faculty could have evolved by means of 'Darwinian' natural selection (see e.g. [14][32]). Another critique, for example, agrees with Chomsky that a universal grammar could not have evolved by natural selection, because a universal grammar does not show enough variety [27]. Variety is one of the fundamental principles of natural selection [10]. Lieberman [27], though, claims that language mechanisms must have been evolved by means of natural selection.

In the next section I will give a brief overview of the claims made by Chomsky and discuss some critiques that has been raised. In particular, I will discuss the critique posed by Pinker and Bloom [32]. In section 2.3 I will briefly discuss some other approaches on the evolution of language, which have been discussed at the Santa Fe Institute workshop on the evolution of human language in 1989 [22]. In section 2.4 I will argue for other approaches that came forward from studies of pidgin and creole languages. Finally, in section 2.5 I will look forward to the approach that we follow at the AI-Lab in Brussels.

2.2 The not-evolved faculty of language

The theory proposed by Chomsky suggests that the ability to acquire and use language is, like all other cognitive functions, structured in the brain in a highly modular fashion [7]. These modules are called the language faculty. Chomsky claims that the language faculty is an innate structure. This innate structure contains the structures of a universal grammar. I.e. the underlying grammatical structures are the same for all human beings, irrespective of their native language. All human beings are born with the same universal grammar, whether they are born in the Netherlands or in China. During the development of a child, parameters of the universal grammar are set according to the language that needs to be acquired [7]. Chomsky [8] furthermore claims, like e.g. Gould [19] that the language faculty has not evolved by a ‘Darwinian’ natural selection, but that it is a side effect of other evolutionary forces, such as increasing brain size. Other yet unknown laws of structure and growth also may have played a part in the evolution of the language faculty. Gould and Lewontin [20], for example, claim that there are other *non-adaptionist* approaches to evolution as opposed to the Darwinian approach.

Although there are many scientists who agree with Chomsky, there are also lots who disagree (for a brief overview see [32]). As I mentioned in the introduction, most critiques involve the claim made by Chomsky that the language faculty did not evolve by natural selection [32]. Most of these critiques however do not address the inexistence of a universal grammar. I.e. they do believe that humans are born with an innate language faculty, but this faculty must be the result of natural selection, see for instance [14][32]. Pinker and Bloom argue, since they recognise the theory of Darwinian natural selection as the most successful theory of evolution, that the language faculty must have been evolved by natural selection [32]. They furthermore raise the objection that "natural selection is the only scientific explanation of adaptive complexity". Lieberman [27], on the other hand, agrees with Chomsky that an universal grammar could not have been evolved by natural selection. Instead he argues that not the Darwinian theory is false, but that the theory of language must be false.

There are also views that there may be a language faculty, but that this faculty cannot bear innate structures of a universal grammar [28]. The argument is that, in case of an injury on brain-structures for language, the human brain has the ability of using another region for language processing [28]. In extreme cases where the left hemisphere has been ablated in infants, the right hemisphere systematically takes over the language functions [17]. Some scientists take this as an argument that the notion of an innate universal grammar is not very plausible [28]. Furthermore, James Hurford [25] rather refers to the language faculty as the *Language Acquisition Device (LAD)*. Although he does not reject the possibility of a universal grammar, this terminology may be more appropriate if we talk about an innate device for language processing. I think, namely, that a 'LAD' may have been evolved by natural selection. The main purpose of such a device may, for example, be to connect the right sensor-motor devises in the brain. Thus resulting in a system that is able to perform speech acts, perceive speech and acquire language by making the right neuronal connections and using a general learning mechanism.

Concluding, the theory of Chomsky is rather controversial. He claims that the language faculty is an innate structure that has evolved as a by-product of the (non-Darwinian) evolution of other brain structures. This view raised many objections by contemporary scientists, but there are also some well-known scientists who do agree with Chomsky. I have mainly discussed the critiques of Chomsky's theory. The most objections concern that a language faculty can evolve through Darwinian natural selection. Others claim that the idea of an innate structure of language must be false. We in Brussels take the latter view, as will be discussed in the next chapter.

2.3 Dynamical Approaches

In august 1989, the Santa Fe Institute held a workshop on the evolution of human languages. In this section I will give a brief review of this workshop which is published in [22]. The idea was that languages evolved by a complex dynamical system. The approach taken was in the field of chaos theory. In this section, I will only focus on the interest of this thesis.

Human language is only a recent development in evolution. Devises for speech and syntax are only available in the human brain for at least 100,000 years [28]. Human language clearly involves some biological components, including the ability to acquire and use words, speech and syntax [12]. According to Darwin, the process of evolution always makes use of old parts, modifying them to perform new functions. Evidence for the ability to produce speech at a high level, i.e. the way that present humans do is found in fossil hominids. The focal tract that makes it possible to produce the speech we can produce, has not been found very much earlier in hominid species than 100,000 years ago [28]. Lieberman does not explicitly deny the existence of innate grammar structures, but he argues that natural selection could have "produced specialised brain mechanisms adapted for syntax" [28]. I think that this does not have to mean that there are innate grammar structures, but rather that the human brain has structures that can adapt to syntax.

Terrence Deacon [12] refers to two extreme and opposite theories of human evolution: (1) "…Language appeared as a result of a single evolutionary accident that yielded a brain so radically changed as to contain all the innate prefigurements of modern language structure." This is the view that Chomsky takes. (2) "… Language is the step-child of a generalised increase in intelligence." According to Deacon, both views are too radical to be true. In his paper, he argues that language evolved by four fundamental principles: (a) Language evolved over a period of more than 2 million years though continuous selection determined by brain-language interaction. (b) Language was the major cause of human brain evolution. (c) Origins of the complex organisation of human brains can best be understood in terms of brain-language co-evolution. And (d), the structures and circuits in human brains that are most altered in human brain evolution must reflect the computational powers demanded by natural languages. [12].

Hawkins [21] concludes in his paper that "functional pressures" have to do with language processing clearly have become biologised in the evolution of *Homo sapiens*. But "not all functional pressures reflected in language universals must be assumed to be innate." Moreover, he argues "that grammatical universals can be explained in terms of non-innate functional pressures". Grammars of all languages may have evolved to reflect the functions that language performs. [21].

Many contemporary studies in language evolution focus on pidgins and creoles [9][35]. Pidgins arise from human communication between two different languages, which together form a cultural society. These languages came together, for example, as a result of colonisation. Creoles, on the other hand, are native languages that arise in the situation where children of 'pidgin' parents grow up in the same cultural society. [9]. In the next section I will discuss the evolution of pidgins and creoles in more detail.

2.4 Pidgins and creoles

As a result from the colonisation period (only less then a few centuries ago) a lot of new cultures arose, including new languages. The evolution of these cultures can be studied with relative high reliability. These studies of pidgins and creoles bring along new data for the evolution of human languages. It appears that pidgins and creoles evolved, and still evolve, on a cultural base, rather than on a biological base. For references in this section I refer to [35].

Although linguistics still has no clear definitions of pidgins and creoles, they all recognise that there is such a group of languages. Until now, there is no definite way to decide whether a language is pidgin or creole, unless reference is made to three criteria: linguistic, social and historical. I already discussed some historical grounds from which pidgins and creoles originate. So, I will focus on linguistic and social grounds.

Linguistics classifies pidgins generally with one main feature: that it shows a drastic reduction of morphological complexity and irregularity. It has a limited vocabulary, an elimination of many grammatical devices, and a drastic reduction of redundant features. Creole languages are built on the pidgin structures and show an increase in syntax complexity, although these languages still show simplification of structures compared to well evolve natural languages.

Differences in pidgins and creoles can be best explained in terms of social backgrounds. Whereas a pidgin language evolves from interaction between two social cultures with different languages, the native people in this new culture build creoles further on pidgin. It has linguistic consequences as well: People who use pidgin also have another language, so they can get by with a minimum of grammatical apparatus. The linguistic recourses of a creole, however, must be adequate to fulfil the communicative needs of human language users.

In pidgins lexicalisation is, as a consequence of the small vocabulary, highly motivated by the needs of the speakers. Whereas ordinary languages have 25-30,000 lexical items, a pidgin language such as Tok Pisin (from Papua New Guinea) has about 1500. Furthermore, the lexicons are mainly build up by the two native languages of the speakers. The grammar of pidgin languages, however, is very redundant. The complexity of the grammar grows as the lexicon grows. Moreover, grammatical structures show little roots from the original languages. As a consequence of the small vocabulary, many semantic items are lexicalised with one word, or with one word for a semantic group extended with an extra lexicalised refinement (see e.g. table 2.1).

Table 2.1 *Lexical comparison of the Tok Pisin word "gras", derived from the English "grass".*

Tok Pisin	English
Gras	"hair"
gras bilong fes	"beard"
Mausgras	"moustache"
gras antap long ai	"eyebrow"
gras bilong pisin	"bird's feather"
gras bilong dog	"dog's fur"
gras nogut	"weed"

We see that Tok Pisin uses *gras* to refer to a semantic feature of "grass", namely that it is 'hairy'. Explicit references to where we find the hair (chin - *fes*, above the eyes - *antap long ai*, dog etc.) are made by other lexical items. If we look closely at the words made in Tok Pisin, we see a close resemblance with English. The way we pronounce *bilong* looks a lot like the English *belong*. The same holds for *fes* vs. *face, antap* vs. *on top, ai* vs. *eye* and *nogut* vs. *no good*.

According to Romaine [35], the pidgin and creole languages appear to evolve as a cultural evolution, which shows many features of the chaos and catastrophe theory in increasing complexity. Chaos theory is an attempt to deal with fluctuations, which reveals in the languages from which pidgins and creoles evolve. The chaos theory suggests treating the evolution of languages as processes, rather than states. Moreover, it is relevant to study the evolution of languages as being open systems rather than closed systems. The catastrophe theory predicts sudden transitions from one state into another as a result of increasing complexity. This kind of transition can be seen in the formation of pidgins and creoles. Both theories have common background in the theory of thermodynamics [33]. According to these theories, dynamical systems that are sensible to many fluctuations converge to a certain (dissipative) structure. In the next chapter I shall explain more about dissipative structures.

2.5 Towards a new approach

In this chapter, we saw that there are many approaches in research of the evolution of human languages. The yielded hypotheses all have one thing in common: they are highly speculative. In this section yet another approach is introduced in this field of research. This new approach, firstly introduced by Luc Steels [40], views the evolution of languages as cultural evolution rather than a biological evolution. Furthermore it is based on selectionistic theories as introduced by Darwin [10], but treated as cultural rather than biological as was proposed by Dawkins [11]. The theory of Dawkins, however, treats *memes* (or ideas/tricks) as the items being evolved by means of, among others, language rather than language itself.

Dawkins [11] introduces memes as analogues of genes, but not containing genetic information, but rather concepts of ideas like, for instance, *the wheel*, *eating bread* or *drinking beer*. Evolution of memes is selected by natural selection processes just as genes are. They are only not ‘thrown’ in a *biological genepool* but rather in a *cultural memepool*. Selection then appears on the best or most useful memes, which are used more often so they become part of the social society. So, the population of memes in the memepool increases, whereas the *bad* ideas die out. As for dissipative structures (see next chapter), natural selection is driven by variations of the items that are sensitive for selection. Finally, the best memes survive and constitute the society.

According to Dennett [13][14], language may well have been evolved in the same way, although he does not make it clear with so many words. He does, however, make clear [14] that meaning may have evolved analogues to the theory of Dawkins. Our approach is that language evolves by means of cultural selective evolution of language in co-evolution of meaning [47]. In the next chapter I will define our approach more precisely.

3. LANGUAGE GAMES AS A BASIS FOR SELF-ORGANISING LEXICONS

3.1 Introduction

In this chapter I will give a more detailed description of the model designed by Luc Steels for the self-organisation of an adaptive language, see for example [40][41][44][46]. I will do so by first introducing the mechanisms proposed. Secondly, I will give a detailed explanation of the language games played by autonomous agents in order to adapt the language. Finally, I will clarify the model with an example.

Language can be defined as *a representational system that is used for communication* [2]. Natural languages have many properties. The most important properties from our point of view are [41]: (1) The community of language users are distributed agents, who have limited knowledge of the language and limited control. (2) Languages are open systems. I.e. agents may enter and leave the community without influencing the language in a great extend. (3) A weak transmission channel, imperfect production and perception, and an inhomogeneous group of speakers conceal the robustness of languages. And (4), languages adapt and evolve to cope with new demands, and to optimise for more efficient and effective communication. The main hypothesis made by Steels [41] is that language is an *emergent* (collective) phenomenon: each agent constructs and follows its own rules, and coherence emerges through self-organisation. Furthermore, the mechanisms of language formation are based on evolution, co-evolution and self-organisation.

As has been argued in the preceding chapter, we think that languages have been evolved under social and cultural pressures. Steels [44] has suggested that there are no innate language structures present in the brain like Chomsky has proposed, but if there are, they must have been evolved by natural selection. The latter suggestion has been argued clearly by Pinker and Bloom [32], and by Dennett [14]. Steels, however, suggests that languages completely evolved through cultural ‘natural’ selection. He proposed three mechanisms that are the basic principles for the spontaneous self-organisation of an adaptive language [44], [46]:

1. Agents adopt word-meaning associations from others, which thus *propagate* in the population.
2. Agents may *generate* a new word and associate it with a decoded feature set.
3. There is a positive feedback mechanism between the success so far in using a word and the selection of that word in a conversation, thus leading to *self-organised* coherence. [46].

Computer simulations with these mechanisms have already shown that the languages that emerged show some main properties of natural languages, which will be discussed in the next section. Furthermore, although the experiments had no purpose to form syntax, agents had expressed expressions with multiple words with arbitrary word order, expressions with more than one meaning (ambiguity), and meanings with alternative expressions (synonymy) [46].

The mechanisms are implemented in language games in which two agents are having a dialogue about a certain topic in a certain context. In language games - a term introduced by Wittgenstein [56], however used somewhat different here - one agent, the *initiator* (or *speaker)*, identifies an agent or object (the *topic*) out of a set of other

agents/objects which constitutes the *context*. Another agent, the *recipient* (or *hearer*) must identify the chosen topic [46]. According to Steels [46], “there are two possible ways to do so: either the speaker points to the topic so that the identification is direct, or the speaker uses language. Language formation and acquisition is only possible when the speaker first uses pointing and then language. When more and more language comes available, purely linguistic means suffice.” This, however, is still a major discussion in the philosophical literature [34].
In the next section, I will define the mechanisms for language formation proposed by Steels. In section 3.3 I will explain how these mechanisms are implemented in language games. In section 3.4 I will give an example of the language formation. Finally, in section 3.5 a summary of this chapter is given.

3.2 The mechanisms needed for language formation

The three mechanisms *propagation*, *lexicalisation* (or *generation*) and *self-organisation* are described in [46] as follows:

> When two agents engage in a language game and the hearer already knows what the topic is (by means of e.g. pointing), then both agents first have to identify which possible sets of features distinguish the topic chosen, from the other objects in the context. These sets are called distinctive feature sets. There could be several distinctive feature sets for one object. The speaker then must choose one distinctive feature set and encode it into an expression. An expression contains one or more words. Words are allowed to be ambiguous and there is a possibility for synonymy. Next the hearer decodes the expression. From the distinctive feature sets the hearer formed he can confirm that the expression encodes one of the expected distinctive feature sets. Or he can infer and possibly adopt new associations between words and meanings. This feedback then enables both agents to adjust their lexicons, and so word-meaning pairs can *propagate* through a population of agents.
>
> This propagation mechanism though is not enough for building a lexicon. Agents also must be able to extend the language whenever the existing language is not adequate. This happens when there are no words to express certain feature sets. Extending the lexicon is achieved by allowing the speaker to create a new word and associate it with a decoded feature set. This is called *lexicalisation*. The creation of new word-meaning pairs happens with very low probability, because the more words exist in a population the longer it takes to reach coherence.
>
> Coherence is achieved through *self-organisation*, in the sense of spontaneous formation of dissipative structures through pressures that are forced by random variations. (Dissipative structures will be discussed in greater detail at the end of this section, *PV*). The fluctuations are caused by the different associations floating around in the population. An agent records how many times a word-meaning pair has been used and how many times it was successful. When meanings need to be encoded, the agent picks the most commonly used successful association. This introduces a positive feedback loop: The more a word gets used, the more successful it will be and therefore the more it solidifies. [46]

These mechanisms bring along the following important properties of natural languages: (1) There is no single agent with a complete view of the language, nor is there a single agent in charge of creating language. We therefore have a fully *distributed* system. (2) The system is *open*. I.e. at any point in time new agents and/or objects are allowed to enter the system without destroying the language. And (3) there may emerge ambiguous and synonimal words and complete coherence is never reached. [46]. The idea behind the development of these mechanisms is that language should evolve to a *dissipative structure* [44].

Dissipative structures evolve through a dynamic interaction in a system that is subject to (large) fluctuation [33]. The fluctuations (or *non-equilibrium states*) are the driving conditions that leads the dynamical process to a certain structure. Due to ever increasing entropy, however, this structure is not a stable state [33]. Entropy is a measure that can be interpreted in different ways. It was introduced in the current sense in the theory of thermodynamics, and it is a parameter that measures the useful exchange of energy of a certain system [33]. In the chaos theory, one can think of "(metric) entropy as a number measuring the time rate of creation of information as a chaotic orbit (or *attractor*) evolves [31]. An attractor is a particular space to which a system converges, it can be compared with a dissipative structure. So, entropy in the sense of language evolution may be compared with the flow of information and representation that the communication produces. Ever increasing entropy leads to new fluctuations in the system, thus keeping the system evolving to the dissipative structure irreversibly [33]. It can, for example, be compared with the way an ant society forms a path.

Dissipative structures have been introduced in the theory of thermodynamics. Consider the following example of a thermodynamic system that is called the *Brusselator* [33] (this example is rather complex, but readable for physical die-hards):

> In some closed system there are two types of molecules, X and Y, which are initially concentrated in the numbers X_0 and Y_0. Molecule X is produced from Y in some process, and conversely Y is synthesised by a reaction of substance A (which synthesises X as well) and reactant B. The system is explored for increasing values of concentrations of B, with A remaining constant. It can be calculated that the system is likely to evolve to a stationary state of concentrations of X and Y, i.e. $dX/dt = dY/dt = 0$, *and concentrations* $X_0=A$ *and* $Y_0=A/B$. If, however, the concentration of B exceeds a certain threshold, the system spontaneously leaves the stationary state (X_0, Y_0), as a result of fluctuations. Now the concentrations of X and Y are oscillating with a well defined periodicity. Whatever the initial conditions were, the system approaches a limit cycle, the periodic behaviour of which it is stable. We therefore have a periodic chemical process - a *chemical clock*. Now we might expect that the concentrations of X and Y just fluctuates so that at a given moment we expect more molecules of X than Y in, say, the left part of the system. Then a bit later we would expect more molecules of Y, and so on. This, however, is *not* what happens with a chemical clock; here the system has only molecules of X at a given moment, then it abruptly changes to only Y molecules. Because all these changes occur with regular time intervals, we have a coherent process.

> If this kind of process has not been observed, nobody would believe it. To change their type just at once, molecules must have a way to 'communicate'[1]. The system has to act as a whole in order to change all at the same time. Dissipative structures introduce probably one of the simplest physical mechanism for communication, while it is formed through self-organisation. *Adapted* from [33].

As we can see, due to fluctuations in a system and by self-organising processes, a system that is in a certain initial state can evolve to a certain (dissipative) structure, which is coherent. We think that language is a dissipative structure as well. So we think that, simplistically spoken, at a particular moment there was a system that carried a wide variation of information, i.e. it was far-from-equilibrium. There was no language, but only the intention and the physical means to communicate. During a cultural process of communication, creation and association, words and meanings emerged throughout the system of agents with large fluctuations. Due to these fluctuations and a self-organising mechanism a (dissipative) structure evolved: language. Because of ever increasing complexity and diversity in meaning and language (due to the openness of the system), the process keeps on evolving.

3.3 Language games

The formal description of the language games given in this section is completely adapted from Steels [46]. It first describes the terminology of the system. Secondly, it describes the algorithm of playing language games. And finally, it defines the selection criteria for the self-organising.

3.3.1 Terminology

In the system there is a set of agents $A = \{a_1,\dots,a_n\}$ and a set of objects $O = \{o_1,\dots,o_r\} \cup A$. Each agent is assumed to have a set of features $F = \{f_1,\dots,f_m\}$. A feature f_i consists of a *(p v)* where *p* is called an attribute(-name) and v is a corresponding value. A *distinctive feature set* $D_{\{oj,B\}}$ is a set of features that distinguishes an object (or an agent) from a set of other objects/agents $B = O \setminus \{o_j\}$ iff $D_{\{oj,B\}} \subset F$ and $\forall a \in B$, $D_{\{oj,B\}} \not\subset F$. There can be several distinctive feature sets, and there can also be none.

A *word* is a sequence of letters drawn from a finite shared alphabet. In my experiments words are created in alphabetic order, i.e. the first word is (a b), the second is (a c), …, (a z), (b a), …. An *expression* is a sequence of words.

A *lexicon* $L \subset F \times W = \{k_1, \dots, k_p\}$, where $k_n = (w_i\ f_j)$ with $w_i \in W$ is a word that is associated with a feature $f_j \in F$. k_n is called an *association*. Each agent is assumed to have its own lexicon, which is initially empty.

There is also a *use-factor* $u(k_n)$ which is the number of times the association k_n is used by a particular agent. *Success-factor* $s(k_n)$ is the number of times that association k_n is used successfully, i.e. when a game ended with communicative success. [46].

[1] This use of terminology is somewhat confusing here, because here the usage of 'communicate' has nothing to do with the experiment that is presented here.

3.3.2 Language games

A language game includes a context $C = \{o_1, o_2, \ldots, o_j\} \subseteq O$, a *speaker* $s \in A \cap O \neq \varnothing$, and a hearer $h \in A \cap O \neq \varnothing$, and a topic $t \in C$. The language games involves the following steps:

1. Both the speaker *s* and the hearer *h* determine the distinctive feature sets $D_{\{t,C\}} = \{D_{\{t,B\}} \mid B = C \setminus \{t\}\}$. It is assumed that both agents share the same distinctive feature sets.
2. The speaker chooses one distinctive feature set $D_j \in D_{\{t,C\}}$ and *encodes* an expression *e* which *covers* D_j.
3. The hearer *decodes* from *e* the feature sets $H = \{H_1, \ldots, H_q\}$ using the *uncover* function.
4. The language game ends in communicative success when $H \cap D_{\{t,C\}} \neq \varnothing$, otherwise in failure.

In the language encoding and decoding processes, the *cover* and *uncover* functions are the most important. They are defined as follows:

$$cover(D,L) = \{e \mid e = \{w \mid D = \cup f_i \text{ with } < f_i, w > \in L\}\}$$

$$uncover(e,L) = \{H \mid H = \cup f_i \text{ with } < f_i, w > \in L, w \in e\}$$

The cover function yields a set of possible expressions. Only one is selected for use in communication, based on two criteria: (1) the smallest expression is preferred and (2) in case of equal size, an expression is preferred that has the best score $m(k_i) = s(k_i) / u(k_i)$.

The success of a language game is determined by the hearer by means of comparing the decoded feature set with the expected distinctive feature set. A language game has three possible outcomes:

1. There are not enough distinctions to identify the topic in this context, i.e. $D_{\{t,C\}} = \varnothing$.
2. The game ends in communicative success, i.e. $H \cap D_{\{t,C\}} \neq \varnothing$.
3. The game ends in communicative failure, which could take many different forms:
   - The speaker s may not have enough words to encode all the distinctive features, i.e. $cover(D_j, L_i) = \varnothing$.
   - The hearer *h* may not have enough associations to decode all the meanings, i.e. $uncover(e, L_h) = \varnothing$.
   - There is a mismatch between expected and decoded meanings, i.e. $H \cap D_{\{t,C\}} = \varnothing$.

These different types of result are criteria for steps that are used for the language formation. This is discussed in the next section. [46].

3.3.3 Rules for language formation

In this section I will discuss the rules that an agent follows in adapting the lexicon. This adaptation is driven by the outcome of a language game. The explanation follows the explanation used in [46].

3.3.3.1 No distinctions are possible

This is the case when the agent was not able to distinguish the topic from other objects in the context (outcome 1 in the preceding section). This should put pressure on the meaning creation process to introduce a new distinction. This mechanism is discussed in detail in chapter 6.

3.3.3.2 The lexicon is inadequate for the speaker

When an agent has not enough associations in his lexicon to encode all the features in the chosen distinctive feature set D_j, the game ends in communicative failure and it is indicated for which features there were no words. Every use of the associations, that were used, are incremented, but not the success. The speaker may create a new word with a certain probability (usually 0.05) and associate it in his lexicon with the non-covered features. [46] This is based on the generation mechanism. The probability factor is introduced in order to decrease the amount of ambiguity.

3.3.3.3 The lexicon is inadequate for the hearer

The hearer may not have enough associations in his lexicon to decode all the meanings from *e*. In this case the game ends in failure and it is indicated for which words no associations were available. Several possibilities can be distinguished:

1. No words at all could be decoded. In this case the hearer associates the expression with one of the feature sets in the distinctive feature sets. Note that this operation may lead to ambiguities because there might be more than one way in which the topic is distinguished from the context.
2. Some words could be decoded while others could not. When there is only one word that is unknown, then the hearer can deduce from the meaning of the known word(s) which distinctive feature set(s) is (are) appropriate to the expression. So, a new association can be made. For the decoded words the use is incremented, but not the success. [46].

New associations are made, thus words propagate through the community.

3.3.3.4 No words are missing

This is the case when both the speaker and the hearer were capable of encoding or decoding the distinctive feature sets with an expression. Different possibilities can arise:

1. There is complete success when the distinctive feature sets expected by the hearer include the one decoded from the expression used by the speaker. Both the use and the success of the association is incremented by the speaker and the hearer.
2. There is success, but it is too general. This is the case when the hearer decoded more possible meanings for the expression. In that case, only the success of the association that was effectively relevant, i.e. the one that resembles the topic best, gets incremented. The use is incremented for all associations used.
3. It may be that the feature set decoded by the hearer is not one of the feature sets that is distinctive for the topic. The success score for the implied association is therefore not incremented. The words in the expression are associated with the appropriate associations in the distinctive feature sets.

The way an agent adjusts the use and/or success scores is the basis for the self-organising selection. If agents have to choose between words that have the same meaning, the most successful ones are chosen, thus leading to the selection of the best association.

These are all the rules that an agent has in order to form a lexicon. Note again that all the rules are adapted from [46].

3.4 An example of the formation of a language from scratch

In this section I will explain the formation of a language using an example that is adapted from [44] and [46], but it focuses on the robot environment. It is an example where a spatial vocabulary is formed. In this experiment two agents are communicating about objects in their environment, including themselves. They can either indicate the object by their names or by their spatial relations [44]. Implementation of this experiment on the robots is the end-goal for the experiment that is described in this thesis.

Objects o_i in the environment can be described by some perceptual features that I will give abstract names like f_1, f_2, … etc.. Their spatial relations from the point of view of an agent also can describe these same objects. I.e. objects can be located by their two dimensional place in space. If it is located, an object can be indicated by, for example, 'front left', cf. the terminology of [44]. The x- and y-co-ordinates in a two-dimensional space can assign an object. The agent's point of view is the origin O = (0,0) the *xy*-plane, the direction that it is facing in the front is defined along the *y*-axis. Each agent autonomously determines the location of an object, and assigns the *xy*-co-ordinates of the object from the agents' own point of view.

Each agent then can assign spatial relations like left, right, front etc. by the following rules:

- Front: $y > 0$.
- Side: $y = 0$.
- Behind: $y < 0$.
- Left: $x < 0$.
- Straight: $x = 0$.
- Right: $x > 0$.

In the simulation experiments that have been done, it is assumed that two agents participating in a language game are standing in front of each other facing one another. This has two important consequences: (1) the other party in a language game is always standing in the *front* and *straight*, from the point of view of both agents. And (2), all other objects have different spatial relations from the point of view of different agents. If, for example, an object has the spatial relations 'front left' from the point of view of agent a_1, then it has, from the point of view of agent a_2, the relations 'behind right'. So, for clarity, the following pairs of opposing relations can be identified: (*front behind*), (*side side*), (*left right*) and (*straight straight*).

In the examples drawn here, I will follow some dialogues that, for simplicity, are held with only one context. The context consists of agents a_1 and a_2, and the objects o_1, o_2 and o_3. It is assumed that all agents have the same feature sets for the objects. The features for the objects are as follows:

o_1: $\{f_1, f_3, f_6\}$, pov-a_1: (*front left*), pov-a_2: {*behind right*).
o_2: $\{f_1, f_4, f_5\}$, pov-a_1: (*side right*), pov-a_2: (*side left*).
o_3: $\{f_1, f_3, f_5\}$, pov-a_1: (*behind right*), pov-a_2: (*front left*).

Here, 'pov-a_i' means 'from the point of view of agent a_i'. In the examples that follow, the agents change their role in the language games randomly from speaker to hearer. In every language game a topic is chosen randomly by the speaker. The speaker also chooses the way it indicates the topic, i.e. either by name or by spatial relation. It is assumed that the hearer knows what the topic is, and that it determines rightfully the way the topic is indicated.

Example 1. This is language game 1. Both agents do not have any associations made in their lexicon, i.e. the lexicon is empty. Agent a_1 is the speaker and a_2 is consequently the hearer. Object o_2 is the topic and the way of communication is by indicating perceptual relations. As can be derived from the feature sets of the objects $\{f_4\}$ is a feature set that distinguishes o_2 from all other objects in the context, so $\{f_4\}$ is a distinctive feature set.

Dialogue 1
a_1: $D_{o2} = \{f_4\} \rightarrow$ (nil)
a_2: $D_{o2} = \{\{f_4\},\{\{side\},\{left\}\}\} \rightarrow$ pov-a_1: $\{\{f_4\},\{\{side\},\{right\}\}\}$
a_2: (nil) $\rightarrow \emptyset$? $\rightarrow$ failure

a_1: create word (a b) and associate it with $\{f_4\}$

Agent a_1 could not encode feature set {f_4} with an expression because the lexicon was empty, therefore it expressed *(nil).* The communication was a failure and agent a_1 created a new word *(a b)*, which it associated with feature set {f_4}. The hearer could do nothing, because the expression *(nil)* was decoded into an empty feature set. It can be seen how the hearer transforms its spatial relations from its own point of view into the point of view of the speaker.

Example 2. In dialogue 17 o_2 is again the topic, with a_1 the speaker. Again a_1 chose to indicate the topic by name. Agent a_1 has one association in his lexicon: *<{f_4}-(a b)>*, and the lexicon of a_2 is still empty. Now we see the following dialogue.

Dialogue 17
a_1: D_{o2} = {f_4} $\rightarrow$ (a b)
a_2: D_{o2} = {{f_4},{{side},{left}}} $\rightarrow$ pov-a_1: {{f_4},{{side},{right}}}
a_2: (a b) $\rightarrow \emptyset$? $\rightarrow$ failure

a_2: associate word (a b) with feature set {f_4}

Agent a_1 could encode {f_4} with the expression *(a b).* The hearer, however, could not decode this word with a meaning, so the language game was again a failure. Now the speaker increments the use for the association *<{f_4}-(a b)>*, and a_2 associates this *(a b)* with feature set {f_4}.

Example 3. In the mean time the system has evolved some more and the lexicons for both agents are starting to form. They now look like this:

L_{a1}: {{<{f_4}-(a b)>,s=2,u=4},
{<{f_6}-(a c)>,s=0,u=1},
{<{right}-(a c)>,s=0,u=0},
{<{behind}-(a d)>,s=0,u=0}}

L_{a2}: {{<{f_4}-(a b)>,s= 2,u=2},
{<{right}-(a c)>,s=0,u=1},
{<{f_6 }-(a c)>,s=0,u=0}}

As you can see both agents have an ambiguity for *(a c)*, this is caused by independently creating the word *(a c)* and associating it with different meanings. In systems where there are more than two agents, ambiguity could have other causes as well.

Now suppose another dialogue where agent a_1 is the speaker and object o_3 is the topic. a_1 chooses to indicate the object by its spatial relation.

Dialogue 38:
a_1: D_{o3}={{behind},{right}} $\rightarrow$ (a d) (a c)
a_2: D_{o3}={{{front },{left}},{{f_3},{f_5}}} $\rightarrow$
pov-a_1: {{{behind},{right}},{{f_3},{f_5}}}
a_2: (a d) $\rightarrow \emptyset$? $\rightarrow$ failure
(a c) $\rightarrow$ {right} and {f_6} $\rightarrow$ {right} $\rightarrow$ success

a_2: associate word (a d) with feature set {behind}

This dialogue was partly a success. Agent a_2 could not decode *(a d)* with a feature set, because this word was not part of its lexicon yet. It could, though, decode *(a c)* with the feature sets *{right}* and *{f_6}*. Since, *{f_6}* is not an element of D_{o3} and *{right}* is, a_2 could infer that *(a d)* must be associated with *{behind}*. Because this dialogue was partly a success a_1 increments the use of <*{behind}-(a d)*>, but not the success; and both a_1 and a_2 increment both the use and success of association <*{right}-(a c)*>. This way the association <*{right}-(a c)*> emerges to a coherent word-meaning pair of the language.

These examples show how words are created and associated. I have not discussed all the rules given in the previous section, nor are they real experiments, although they are examples drawn from real experiments as reported in [44] and [46]. They only have been modified a little bit for a closer correlation with the experiments reported in this thesis, and they are only used to clarify the adaptation processes of the lexicon. In these examples some simplified assumptions have been made. The first one is that there are only two agents in the environment. In the simulations done by Steels there were always more, e.g. twelve which then were also the only objects. Secondly, I assumed that the feature sets were held constant, an assumption also made by Steels, but which will not be true on real robots. The third assumption was that both agents always identified the same topic, which may not be true in real robots. Finally, I assumed that the distinctive feature sets in both agents were the same. This follows from the previous stated assumption that both robots have the same feature sets for every object, and that the topic is the same in both agents. In addition, this will not be true in real robots either.

3.5 Summary

In this chapter the language formation based on adaptive selection mechanisms introduced by Luc Steels [41] has been described. The idea is that language evolves through social interactions between agents and by adaptation driven by selectionistic forces, yielding a language that evolves to a dissipative structure. Steels proposed three mechanisms for language formation: *generation*, *propagation* and *self-organisation*. These mechanisms are implemented in a process of so-called language games.

In a language game, two agents are involved. One chooses to be the speaker, the other is the hearer. Both agents are assumed to share the same context of objects in their environment. The speaker chooses a topic from the context, and uses extra-linguistic means to make this topic clear to the hearer. Both agents then try to discriminate the topic from the other objects in the context, yielding a (shared) set of distinctive feature sets. The speaker chooses one distinctive feature set and encodes this set into an expression. The hearer decodes from this expression a set of features, which it compares with its own set of distinctive feature sets in order to determine the success of the language game. Success is reached if the decoded feature set is an element of the set of distinctive feature sets. Otherwise the game ends in (partial) failure. According to the cause of this failure, the language is adapted. The speaker may create new words, while the hearer may make new associations. For every use of an association, a use-factor for this association is updated. A success-factor of an association is updated if the game where this association is used was successful. If the speaker has to choose between two or more words that have one meaning, the most successful word is chosen, thus increasing the (successful) use of that word. So, the associations that are often used successfully will be used even more often, thus making the system selectionistic.

Experiments in simulations have shown that these mechanisms are sufficient to originate a coherent lexicon from scratch. The resulting lexicons are natural language-like in that it shows some important similarities with human languages: *ambiguity*, *synonymy* and *no complete coherence* are all present in the system. Furthermore, the system is fully *distributed* and *open-ended*. Until now, only the mechanisms for the formation of a lexicon have been discussed. A language, however, also must have semantic features (or meaning). Just as language, we think that meaning should evolve with selectionistic mechanisms [43] in a co-evolution with language [47]. The theoretical model for meaning creation will be discussed in the next chapter.

4. PERCEPTUAL GROUNDED MEANING CREATION

4.1 Introduction

In the linguistic literature, approaches to meaning are classified in three different groups [6]: (1) A 'referential' or 'denotational' view. This viewpoint approaches meaning from the outside of an agent (usually human beings), and it concentrates on the informational significance of language. (2) There is a 'psychological' or 'mentalistic' approach. Theories for this approach look at the inside of an agent and focus on the cognitive significance of language. From this point of view, meaning lies in the internalised representation of their retrievable content. And (3) some theories may be classified as 'social' or 'pragmatic', which focus on the communication as a social activity. According to this view, meaning lies in the way agents use symbols in the course of their interactions with each other. Chierchia and McConnell-Ginet think meaning must be explained using all these aspects. I agree that it is right to investigate meaning from these approaches.

Their point of view, however, is in line with the Chomskian (or Fodorian) idea that meaning resides in universal concepts. I.e. the meaning of a word can be explained in universal components which are innate, and from which all languages draw their lexical labels. The main mistake that they make, I think, is that meaning cannot be represented by universally innate structures, because then there must be too many concepts stored in a limited amount of genetic structures. There are no more than 200,000 structural genes present in the human genome, as opposed to the ca. 30 billion cortical neurones [5]. In addition, if we look at the evolution of brain organisation, we may note that the total amount of DNA per cell does not change from mouse to man, despite differences in brain complexity [5]. So, it can be argued that there are no innate structures of meaning present in the brain.

At the AI-Lab, we think that meaning develops in interaction of an agent with surroundings or its internal states. A concept (being an object or internal state, etc.) is observed by an agent, who constructs an internal representation of this concept using discriminative features. I.e. the agent associates a concept with features that distinguishe that concept from other concepts. Meaning is further classified by adding lexical labels to these features, thus constructing a shared set of meanings in a society of agents. The associations of word-meaning pairs are constructed in fuzzy sets of family resemblance-like features. This is cf. the notion of Aitchison [1] that "for the majority of words, meanings in the mind are fuzzy, not fixed". The way the meaning co-evolves with language in our experiments the same principle emerges, as will be discussed in section 4.3.

As was mentioned in the preceding chapter, words in a lexicon must be associated with a set of features that represent an object. This set of features constitutes the meaning of an object from the point of view of the agent. Instead of implementing a conditional rule-based system, I propose to implement the perceptual grounded meaning creation proposed by Steels [43]. This is conform the idea that meaning and language co-evolve in a group of distributed agents [47]. Like in the mechanisms used for adapting a lexicon, the method that is used for adapting meaning is roughly based on the theory of natural selection. The mechanisms for the categorisation of objects are *generation* and *self-organisation*. Again, a series of games are introduced, only this time these games are called *discrimination games*.

Our basic notion of meaning is the internal representation of a concept. This concept may be either a physical object, a particular relation (e.g. a spatial relation), an internal state of an agent or an action etc. The experiments that I have implemented concern only physical objects, so I will only concentrate on the construction of the meaning of physical objects. Although there are many linguistic and philosophical definitions of meaning, we think that meaning is represented by one or (usually) more distributed sets of feature sets that *distinguishes* a particular concept from other concepts in a particular context. These *distinctive feature sets* need not be the same for every situation in which the concept is discriminated, but they have to be general enough to be used more often in distinguishing the same concept.

Meaning takes many forms depending on the context and nature of the situation concerned. Some meanings - such as light intensities - are perceptually grounded. Others - such as social hierarchies, goals or intentions for actions - are grounded in social relations or in the behavioural interaction between the agent and the environment [43]. A theoretical model is proposed to explain how an autonomous agent may originate new meanings. The model is theoretical in the sense that no claims are being made that it is empirically valid for humans or animals. The goal is to outline and validate possibilities [43].

The next section describes the formal mechanisms for the meaning creation, and discusses which results should be used in the language games. The formation process will be illustrated with some examples in section 4.3. The final section contains a small summary of the described process. Results of the robotic experiments are given in the chapter 8, the examples from section 4.3 are drawn from simulation experiments.

4.2 Discrimination games

4.2.1 Introduction

In the process of language formation, an agent has to distinguish one object or direction from others using sensors and low-level sensory processes, in order to communicate about it. To distinguish certain objects in a coherent fashion, an agent has to obtain features of an object so it can successfully discriminate that particular object from another. The process of discriminating an object is called a *discrimination game*.

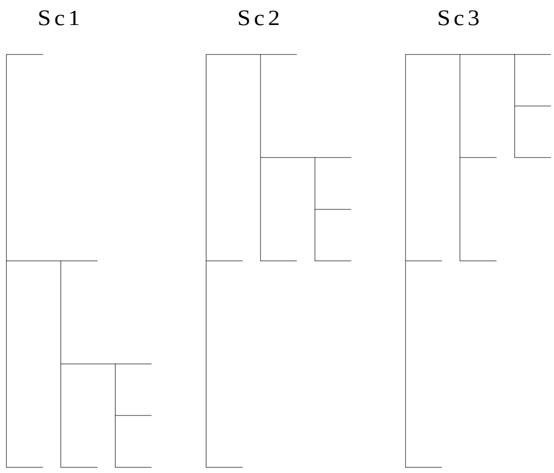


Figure 4.1 *The segmentation of the feature space by the sensory channels.*

In the model it is assumed that there are different objects, which have characteristics that are sensed through sensory channels, which are either derived directly from a sensor or from low-level sensory processes. Sensory channels are designed to characterise certain properties that can be derived from a sensor. These properties could be, for example, the sensor values that are sensed directly. Or they could be properties of a spatial description from a certain point of view in the robot, of internal states like a certain motivation, actuator states. I think one can best compare these feature detectors with neurones that are only sensitive for certain stimuli. Hubel and Wiesel [24] have found neurones that, for example, only respond to stimuli of lines in a specific orientation, such as those neurones that are present in the visual cortex of cats.

In the context of this experiment I will only focus on perceptual properties of objects. These properties may be absolute, like the type of light they emit, or temporal, like the relative place where a particular object is. Meaningful distinctions take the form of features, which decompose into an attribute and a value [43]. According to Steels [43] a feature "is derived by a feature detector which discretisises the continuous space of one sensory channel. The feature indicates that the value of a sensory channel falls within one sub-region of the space" defined by the feature detector (see fig. 4.1). Attributes represent the names of the different features. The value is the outcome of a certain function that is applied to the object by a sensory channel.

The model is based on the hypothesis that meaning emerges from the construction and selection processes that are embedded in the discrimination tasks [43]. Each agent is capable of constructing new features (i.e. new segmentations of the sensory space) and selecting these features by differentiating them from features of other objects in a context. Discriminations are based on one or more features grouped as a *distinctive feature set*. A distinctive feature set is a set of features that discriminates one object from another. There may be more than one distinctive feature set for an object, but there also may be none if there are not enough features to discriminate the topic [43].

In the next sub-section the mechanisms for the discrimination tasks will be defined more concretely. I will define the mechanisms for just one agent, but they are obviously applicable to all agents.

4.2.2 The formal description of discrimination games

This section describes the theory that is fully adapted from [43]. For convenience, I will not refer to the source again in this section.

Let there be a set of objects $O=\{o1,...,on\}$, and a set of sensory channels $S=\{sc0,\ldots,scm\}$. Each sensory channel $sci: O \rightarrow [LB,UB]$ is a function that maps a sensory value to a real value in the interval $[LB,UB]$. LB is the lower bound of sensory channel i, and UB the upper bound. Every agent has a set of feature detectors $FD=\{d_1,...,d_k\}$, where $d_i=< p_i,V_i,\varphi_i,scj >$. Here p_i is a attribute name, V_i is a set of possible values for p_i, $\varphi_i : S \rightarrow V_i$ is a real-valued function, and scj is a sensory channel. The result of applying a feature detector d_i to an object oi is a feature written as a pair $(p_i\ v_i)$, where $v = \varphi_i(sc_j(o_l)) \in V_i$ is the value of attribute p_i. The feature set for an object o_l is now defined as:

$$F(o_l)=\{(p_i\ v_i) \mid d_i \in D\}$$

Two features $(p_1\ v_1)$ and $(p_2\ v_2)$ are distinctive iff $p_1 = p_2$ and $v_1 \neq v_2$. A distinctive feature set is then defined as:

$$D^C(o_t)=\{f \mid f = (p\ v) \in F(o_t) \text{ and } "o_c \in C \text{ either } \neg\exists f'=(p'\ v')\in F(o_c) \text{ with } p=p' \text{ or } \exists f' \in F(o_c) \text{ with } f \text{ and } f' \text{ distinctive}\}$$

There can be several distinctive feature sets for the same o_t and C, or none.

A discrimination game for an agent involves a topic $o_t \in O$, and a context $C \in O \setminus \{o_t\}$. The discrimination game is a success if a distinctive feature set could be found, i.e. $D^C(o_t) \neq \emptyset$, and it is a failure when no such feature set could be found, $D^C(o_t) = \emptyset$. Depending on the outcome of a discrimination game the repertoire of a meaning is adjusted by an agent in the following way:

- The game is unsuccessful, so the agent could not make enough distinctions. There are two ways to remedy this situation:
  - If there are still sensory channels for which there are no feature detectors, a new feature detector may be constructed. This option is preferred.
  - Otherwise, an existing attribute may be refined by creating a new feature detector, that further segments the domain that is covered by an existing attribute.

- The game was successful. In case there are more than one distinctive feature sets, the feature sets are ordered based on preference criteria. The 'best' feature set is chosen and used as outcome of the game. A use-factor is increased for the chosen features. The criteria for choosing the best set are as follows:
  - The smallest set (i.e. the one with the least number of features) is preferred.
  - In case of equal size, the set in which features imply the smallest number of segmentation is preferred. Thus the most abstract features are chosen.
  - In case of equal segmentation, the set of which the features have been used the most is preferred. This ensures that a minimal set of features develops. [43].

Just like in the language games, when discrimination games are played repeatedly, using all the objects in the environment, a coherent system develops from the pressure of the selection and the diversity of the environment. Experiments done by Steels already showed this principle.

Conform the language games defined in the previous chapter, I introduced a success factor. In the system I have implemented, the use factor is increased for *all* distinctive feature sets. The success is only incremented for the *best* distinctive feature set. It is thought that it will increase the efficiency of the selective self-organisation.

In the terminology that I will use in this thesis, a feature detector *d* and a feature *f* are sometimes used through one another. The characteristics of these terms are more or less the same, however, a distinction that I make is that a feature is a *temporal* characteristic of an object, while a feature detector is a *long-term* representation in the system of discriminations. For both feature and feature detector I will denote the attribute name as sc0, sc1,… A *feature* is further denoted with the *value* that is measured with the sensory channel. The *domain* of a *feature detector*, on the other hand, is either denoted as an interval [v0,v1] or, cf. [51], as an extension to the attribute name as sc0-0, sc0-1, sc0-0-0, sc0-0-1, etc.. The extension -0 refers to a refinement of the lower half, and -1 to the upper half.

4.3 An example

Suppose we have an agent who has three sensory channels: *sc1, sc2* and *sc3* and who perceive three objects with the following features:

o1: [sc1:10,sc2:1,sc3:4]
o2: [sc1:7,sc2:34,sc3:5]
o3: [sc1:11,sc2:2,sc3:182]

The objects initially have the following feature detectors:

o1: {sc1,sc2,sc3}
o2: {sc1,sc2,sc3}
o3: {sc1,sc2,sc3}

The agent initially has no feature detectors that are segmented in the domains of the sensory channels. The agent chooses o1 as the topic for the first discrimination game, and the game immediately fails, because all objects have the same feature detectors, so no discrimination is possible. The agent now refines an arbitrary feature detector as follows:

sc1: [0,255] → sc1-0: [0,127.5] and sc1-1: [127.5,255]

The agent refines all sensory channels further until we have, for example the following features attached to the objects (note that, although only refined features are taken, the not-refined features are valid as well):

o1: {sc1-0,sc2-0,sc3-0}
o2: {sc1-0,sc2-0,sc3-0}
o3: {sc1-0,sc2-0,sc3-1}

Suppose we have object o3 as topic, we can find a distinctive feature set that distinguishes o3 from the other objects, namely: {sc3-1}. This feature is not found at any other object, so the discrimination game was a success. Both the use and success of this feature is incremented.

Suppose now, that we have o2 as topic of the discrimination game. There are still no feature sets that can distinguish o2 from the other feature sets, so the feature detectors have to be refined further, for example as follows:

sc2-0: [0,127.5] → sc2-0-0: [0,63.75] and sc2-0-1: [63.75,127.5]

After a while, the given objects can, for instance, be classified as follows:

o1: {sc1-0,sc2-0,sc2-0-0,sc2-0-0-0,sc3-0}
o2: {sc1-0,sc2-0,sc2-0-0,sc2-0-0-1,sc3-0}
o3: {sc1-0,sc2-0,sc2-0-0,sc2-0-0-0,sc3-1}

Now we can distinguish all objects from one another. Suppose the agent plays a discrimination game with object o2. It is clear that the set {sc2-0-0-1} discriminates o2 from o1 and o3, so the discrimination game ends in success. If, on the other hand, we have o1 as topic, the set {sc2-0-0-0,sc3-0} is a distinctive feature set; no other object in the context shares this set, so the game again ends in success.

Until now, the context was a static context. I.e. the objects had the same features over time. Obviously this is not true in real situations. If we look at an object, say a computer monitor, from different angles, we see different features of this object. If we look at it from the front side, we see, for example, a glass-like screen, with some windows in it, we see some icons and maybe characters. But if we look at the monitor from behind, we see some kind of protuberance with cables plugged in. So we observe different sets of features, but we would recognise it in both cases as a computer screen. Although we may mistakenly recognise it as a television set. Our agents must be capable to make similar judgements.

Suppose we have the following context:

o1: [sc1:38,sc2:1,sc3:4]
o2: [sc1:7,sc2:34,sc3:5]
o3: [sc1:11,sc2:2,sc3:182]

And we have, for example, the following sets of feature detectors:

o1: {sc1-0,sc1-0-0,sc1-0-0-1,sc2-0,sc2-0-0,sc2-0-0-0,sc3-0}
o2: {sc1-0,sc1-0-0,sc1-0-0-0,sc2-0,sc2-0-0,sc2-0-0-1,sc3-0}
o3: {sc1-0,sc1-0-0,sc1-0-0-0,sc2-0,sc2-0-0,sc2-0-0-0,sc3-1}

Now the discrimination game yields the following set of discriminative feature sets for o1: {{sc1-0-0-1},{sc2-0-0-0,sc3-0}}. Obviously, the game ends in success. All feature detectors that constitute these discriminative feature sets increments the use factor. Because set {sc1-0-0-1} is the smallest set, this set is used for the language game and the success of this feature detector is incremented, while the others' are not.

Note that, although only {{sc1-0-0-1},{sc2-0-0-0,sc3-0}} are given as distinctive feature sets, combinations of features are also allowed if they are distinctive from any combination of the other objects. So, we may include sets like: {sc1-0-0-1,sc2-0}, {sc1-0-0-1,sc2-0-0}, {sc1-0-0-1,sc2-0-0-0},{sc1-0-0-1,sc3-0}, …, {sc1-0-0-1,sc2-0,sc3-0},…, but not sets like: {sc1-0-0-1,sc1-0} or {sc1-0,sc2-0}, because the first set has two features for one sensory channel and the second is not discriminative. In the example I have not included all combinations of possible distinctive feature sets because the set would be very large.

Suppose now that a certain discrimination game yields the following set of distinctive feature sets: {{sc1-0-1},{sc1-0-1-0},{sc2-0-0-0,sc3-0}}. Which distinctive feature set is the best? The answer is {sc1-0-1}, because this is one of the two sets that has only one element and this feature detector is least refined (i.e. it has only two segmentations, while sc1-0-1-0 has three).

What if the discrimination yields {{sc1-0,sc3-1-0},{sc1-0-1,sc3-1}} as set of distinctive feature sets? In this case, there are two sets both with two elements, and furthermore, the total depth of segmentation is equal for both distinctive feature sets. Now the agent will choose the set that has the best score $m = \Sigma\, s / u$, where s is the success score of a feature detector, and *u* the use. The sum, in this example, is taken for every distinctive feature set over both feature detectors. For the chosen set, both the success and use are increased by 1, thus increasing *m*, while for the other set only the use is increased. If *m* is equal in both sets, then one set is chosen arbitrary, hence the next time this situation occurs, this set is chosen because then *m* is higher for the chosen set.

As we saw in practice, we may find different distinctive feature sets for the same objects in different situations. So, one might argue that it is necessary to introduce a higher label in the hierarchy of a meaning. I think the top label of this hierarchy, at least for physical objects, can be established by language. I.e. all classes of distinctive features that constitute an object can evolve to a unique representation in language. So the highest level of meaning is represented by language, which in turn is represented by its distinctive features. Selective pressures and variation of the environment can classify different distinctive features into one class if they belong to one object. How does this work?

Let us look at a complete language game, including the discrimination. The speaker chooses the best distinctive feature set, which it uses for the language game as described in chapter three. Suppose the speaker expresses (a c), the hearer then tries to decode (a c) into a feature set and compares this set with its own distinctive feature sets. If the hearer fails to decode the expression, it associates the expression with its set of distinctive feature sets. Because there may be more than one set, the different distinctive feature sets may be associated with the same word, thus leading to variation and ambiguity. We see that fuzzy sets are formed which show a family resemblance. The ambiguity causes the lexicon and meaning to cohere, because the different representations in fact mean the same thing. So, if the agent now uses one of these distinctive feature sets to indicate the same object, it has one word for this object. Furthermore, if different associations are used successfully more often, then selection may cause different representations to mean the same. So, ambiguity at this level is not a linguistic ambiguity, but rather an ambiguity at the level of internal representations.

4.4 Summary

In this chapter the process of meaning creation is introduced. The theory developed by Steels is based on two basic mechanisms: (1) the generation of feature detectors, and (2) self-organisation. These mechanisms are the same as those that were introduced for language formation, except that the language formation also has a mechanism for propagation (see previous chapter).

The mechanisms are implemented in a system that ‘plays’ discrimination games. In discrimination games, a topic is chosen from a certain context that consists of several objects. All objects are described by features that are observed by the agent’s sensory channels. In the discrimination game, the agent tries to find a set of features that distinguishes the topic from the other objects in the context, thus yielding one or more ‘distinctive feature sets’. If this set is not empty, the discrimination game ended in success, and the use for all distinctive features is incremented. The success of only the distinctive features that are used in the communication, which constitute the most general set is incremented. This way selection secures that the most useful distinctions survive in the population of distinctive features. If the result of the discrimination game is an empty set, the agent constructs new feature detectors by choosing an arbitrary feature detector and refining this detector by dividing the feature space in two equal halves.

Simulations have shown that agents can construct meaningful distinctions for classifying objects using the above mechanisms. The building blocks for these distinctions are not genetically determined, but emerge from the adaptive selectionistic interaction of discrimination games with objects in the environment of an agent. The resulting representations of objects are distributed features, which are further classified by the language that the agents form.

5. THE EXPERIMENTAL SETUP AT THE AI-LAB

5.1 The Robots

The experiments are implemented on the Cbots, which are developed at the AI-lab. These are Lego robots as shown in figure 5.1. The robots are controlled by the SMB2 (Sensory Motor Board), which is also partly developed at the AI-lab. This sensory-motor board can be programmed in, an especially for behaviour-oriented programming designed Process Description Language (PDL). The robots are equipped with 11 sensors and 8 actuators that I have used. There are three infrared (IR) sensors, two white light sensors, and two light sensors that are modulated at a certain frequency. All these sensors are mounted in the front (see fig.5.2). Furthermore, the robots are equipped with four bumpers, two on the front and two on the back (see large triangles). The robots are also equipped with four IR emitters, mounted in such a way that the robot can emit IR to the front, back, left and right side (small triangles). There are two wheels connected to two separate acting motors. The sensors and actuators are connected in parallel with the SMB2 board. The energy of the robots is supplied by 9.6V 1100 mA NiMH batteries, which are rechargeable in a charging station by direct physical contact through a charging rod mounted on top of the robot and an aluminium plated at the bottom [50]. In this section I give a broad description of the hardware on the robots, which is used during the experiments.

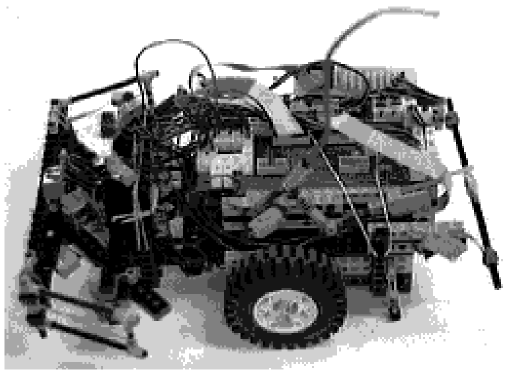


Figure 5.1 *A Lego-robot as is used at the AI-Lab In Brussels.*

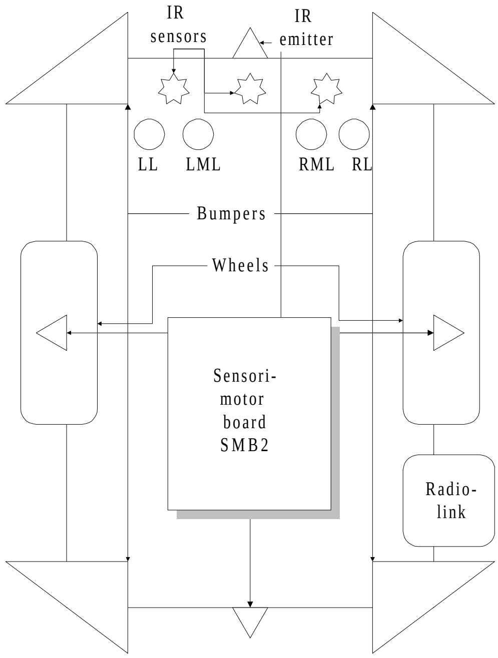


Figure 5.2 *A schematic view of the sensors and motors of the robots as seen from above. The robot is facing north. LL and RL are white light sensors, LML and RML are modulated light sensors.*

The SMB2 is a special designed Sensory Motor controller Board: a microcomputer for the direct processing of the data from the sensors and to the actuators. It consists of a SMB add-on-board, containing I/O chips, bus-controllers and connectors, plugged in a Vesta board which includes a Motorola MC68332 micro controller. The micro controller contains 128k ROM and 256k RAM; and it can run at 16.78 MHz at 5V [54]. It is designed to process software written in PDL.

The robot can be programmed in a C implementation of PDL, which was first developed in Lisp. This program was developed because of the need for a simple programmable device that controls dynamic behaviour of intelligent autonomous agents [37]. The system is born inside the behaviour-based approach to autonomous systems [38] and it is designed to be capable of real-time interaction between dynamic processes inside the agent and the physical processes of the environment [52]. The PDL has its own structure and syntax, although its implementation is embedded in C, thus the C syntax is valid as well. The software sets up a network between quantities that capsulate the state of the network, and dynamic relationships between these quantities which are described by the processes [52]. The quantities set up the network between the sensors and the actuators. The actuators can only be updated by the following statement:

```
add_value(Quantity,DeltaQuantity)
```

which means that Quantity is increased by DeltaQuantity. The quantities are more or less like the normal variables in a common program. The difference is that these quantities are not being updated instantaneously when initiated [52]. This is done at the end of every clock-cycle of the sensory-motor board. PDL is set up to carry out the processes on the sensory-motor board at 40 Hz, but currently moving up to 1000 cycles/sec [49]. I will refer to these cycles as *SMB-cycles*.

The robot is mounted with 2 digital bumpers in the front ant 2 in the back. These sensors are used for *touch-based obstacle avoidance*. 3 analogue infrared (IR) sensors are mounted in the front on the left, the middle and the right. In my experiments they are primarily used to detect a source that emits IR signals (i.e. another robot), but it is also used for *smooth obstacle avoidance*. Mounted on the (top) front of the robot are 2 analogue white light detectors (LDR) for detecting the light emitted by the charging station (see next section for a discussion of the environment). Also 2 analogue modulated light detectors are mounted on the top front of the robot for detecting the competitors in the environment, which emit light modulated at a certain frequency. For both these sensor types there is one on the left and one on the right. The organisation of the robot is schematically given in figure 5.2. All sensors are connected to the SMB2 board by serial cables, and enter the processor in parallel. The actuators on the robot are of course the 2 motors for inducing the movement of the robot. Also 4 IR emitters are mounted upon the robot in order to make itself visible in 4 perpendicular directions for another robot, but they are also used for smooth obstacle avoidance. There is also a display, which can make some phases in the program visibly distinguishable. Furthermore, a radio-link is mounted on the robots.

The SMB2 Radio-link is a radio device that can both send and receive radio signals at a certain wavelength [54]. Therefore we can classify this device both as a sensor and as an actuator. The system was designed to make it possible for robots to communicate with each other at a reasonably fast speed, and also for remote monitoring of the robot's internal state at run-time [54]. It can transmit and receive packets of information which consists of, next to some control and size-indicators bytes, one or more messages. The radio-link can be controlled by PDL running on the SMB2.

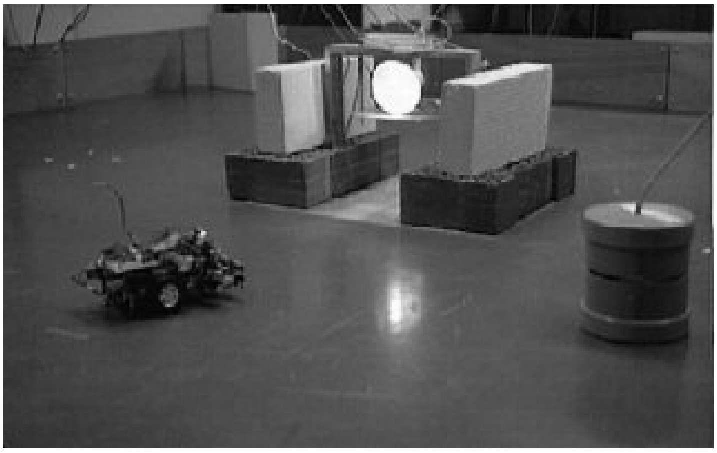


Figure 5.3 The ecosystem is enclosed by walls. There is one charging station (in the middle), and several competitors (like the one on the right).

5.2 The environment

At the AI-Lab in Brussels an ecosystem (figure 5.3) is set up where robots can learn to survive. This was done to do research on the behaviour-based approach to autonomous systems as is described in [38]. The biological background and motivation for the design of this ecosystem was defined by McFarland [30]. In the behaviour-oriented approach co-operation is forced upon the agents by the environment and emerges from the activities of individual agents [39]. In this section I will describe the environment along with the experiments that are designed to explore the behaviour of autonomous agents that need to co-operate in order to be viable.

In order for the robot to survive it needs energy, because the batteries decrease in energy level in time. The robots can recharge the batteries by parking in a charging station (figure 5.4). There is a lamp mounted on the charging station, which can be used by the robots for locating it [49]. The robot can find the charging station using phototaxis. Phototaxis is a dynamical process for orienting towards a light source while driving forward. It is "achieved by the creation of an attracting force field which influences the motor speed dynamics so that the robot turns right when there is less light on the left side and left when there is less light on the right side" [39].
When the robot is parked in the charging station, it recharges its batteries by the supply that is given by the station. In the environment there are also competitors that compete for the available energy. The competitors emit modulated light that draws energy from the ecosystem as well. The robots can attack these competitors by pushing against these cylindrical boxes (see figure 5.5). When

the robots pushes against this box, the lamp diminishes and thus takes away less energy from the global energy flowing into the ecosystem [49]. This diminishing

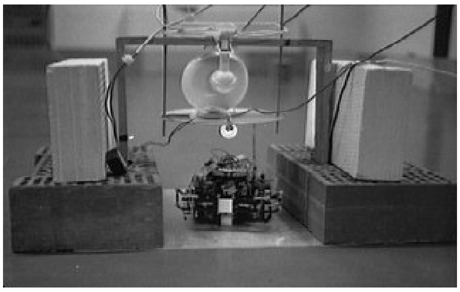


Figure 5.4 *The robot recharging in the charging station. The charging station has a potential difference between two metal plates. The robot has a metal rod on top and a aluminium plate at the bottom to conduct current and thus recharging the batteries.*

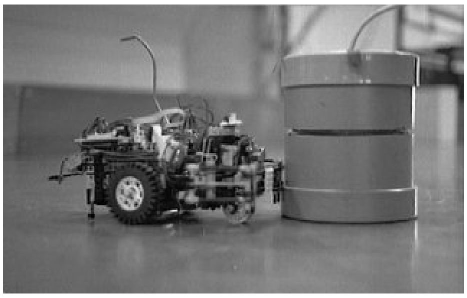


Figure 5.5 *A robot attacking a competitor. It finds the box by modulated light taxis, and by pushing against this box the light will diminish*

is, however, only temporal, so the robots have to keep working in order to survive. The amount of global energy can be set in advance in such a way, that robots need to co-operate in order to survive [44]. Behaviour is learned by exploring different behaviours and coupling these behaviours to a positive feedback mechanism. A selectionistic mechanism is used to select for the best behaviours, which thus would be explored further. [44]. It is thought that the level of co-operation will increase when the robots can communicate [30]. The principle of the ecosystem is shown schematically in figure 5.6.

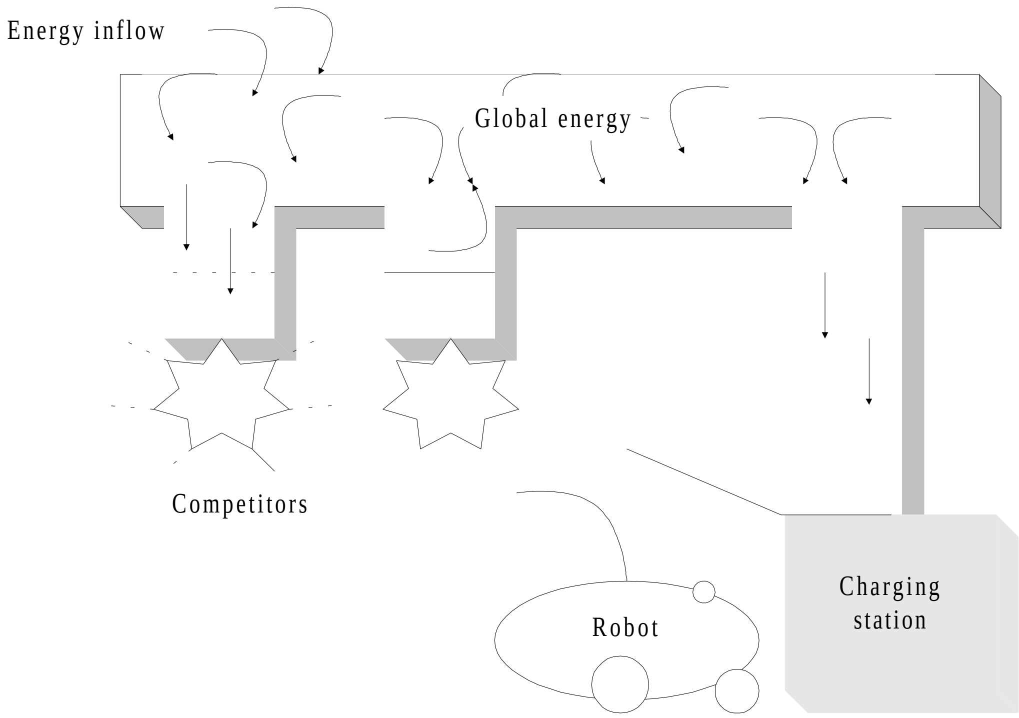


Figure 5.6 *A schematic view of the ecosystem. The system consists of a certain amount of global energy. The robots can consume energy at the charging station. Furthermore, the robots have to compete with the competitors for this energy. They can do so by pushing against the competitors, thus pushing out their light source. If the lights of the competitors are off, they do not consume global energy. The competitors only temporarily turn their light off, so the robots have to keep on working in order to survive.*

6. THE DESCRIPTION OF THE EXPERIMENT

6.1 The outline of the experiment

One of the main purposes of the experiments I have done was to show that the evolution of a lexicon as described by Steels not only emerges by means of simulations, but that it also can emerge in physical systems like real robots. There are many approaches of language games that could have been implemented, but we have chosen to implement the formation of a spatial vocabulary described in [44]. We could, for example, have chosen to implement the naming of agents-experiments as introduced in [41], but then we would have to face the problem of the amounts of robots needed. In order to let this kind of experiment be meaningful, we would have needed at least five robots or so, which were not available. This choice was made because it is one of the simpler variants of the language formation experiments that could be done at the AI-lab in Brussels.

During the implementation, however, we changed our goals. During the formation of a spatial vocabulary, the agents should play naming games as well. Due to time constraints on this project, we ended the project only with experiments containing naming games (see [51]). In naming games, agents form their lexicon by using only names to indicate objects. We chose to combine the naming of agents with the naming of the objects in the environment. Now only two robots sufficed, because the objects already exist in the environment of the laboratory. Note, however, that in the thesis the implementation of the experiment is partly described in terms of adapting a spatial vocabulary. This is because the implementation process was focused on this experiment, only the actual experiments were not carried out.

In the experiment, two robotic agents should form a lexicon about the objects that are recognised by the agents. The objects are indicated either by their name or their spatial description [44]. The mechanisms from which the formation of the lexicon emerges is described in chapter 3. This description raised the first problem, because it is based on already recognised objects, while recognition of objects obviously is no trivial task for real robots. In simulations you can implement these features as a given fact, but in robots these features first need to be recognised. To overcome this problem I could have done various things. For example, I could have implemented the features myself as a rule-based system in the form of: ‘if you see this, then it means that’. Obviously this is a static system that has the ‘intelligence’ of its creator. However, we try to prove that the formation of intelligence of (robotic) agents - be it in terms of meaning, language or behaviour - is a dynamical system. Furthermore, Steels has developed a theoretical system to form meanings based on perceptual grounding in a selectionistic approach similar to the formation of language [43]. Thus I have also implemented this system as described in the previous chapter.

Summarising, the major aim of this research is to check whether the hypothetical model developed by Steels, which is proven to be valid in 'static' simulations, is valid in 'dynamic' real-world experiments as well. The experiments are also built in order to show (1) how agents may develop a shared vocabulary through a series of adaptive naming games, and (2) how agents may generate distinctions to discriminate between objects in their environment [51]. In the next section I will describe the experiment in greater detail, including some problems that I expected in advance. In section 6.3 I will outline the approach I have developed in order to construct the experiment. Finally, in section 6.4 some concluding remarks are given.

6.2 The formation of a spatial vocabulary

6.2.1 A formal description of the experiment

In the robotic experiments a language game is defined as the whole process from making contact until the end of one conversation. In the experiments described here two agents engage in language games about objects by indicating them either by name or by their (two dimensional) place in space. The objects that the agents can name are all the recognisable objects in the environment of the laboratory: the charging station, the competitors and other agents. The spatial relations they use are only rough directions: left, right, front, behind, aside or aligned. As mentioned before, the agents can express sequences of one or more words, where word order is of no importance. The formation of the lexicon needs to be as described in chapter two.

As described in [44], the experiment should involve the following: All robots in the environment are riding around doing their ordinary things, until one decides to communicate. This robot becomes the speaker and makes the robot it wishes to communicate with clear that it wants to communicate. Then the other robot can decide to join the dialogue and becomes the hearer. Both robots that engage in the language game then determine the context, which consist of the objects in their (near) surroundings including themselves. The speaker then chooses a topic from this context and makes the hearer clear what the topic is, using extra-linguistic means (such as pointing). This is necessary, because initially the robots have no other linguistic capacities. If the lexicon is already formed to a great deal, the pointing may not be necessary anymore. When both agents have identified the topic, the speaker expresses this object either by its name or its spatial relation. The hearer decodes this expression in a feature set, determines the successfulness of the conversation and replies the type of success. After a conversation has ended both robots start to do their ordinary exploring until they engage in a new dialogue. This cycle is, in principle, repeated indefinitely.

In order to let the robots derive a coherent context and to determine the spatial relations of the objects in this context from the point of view of the other robot, both robots first need to face each other while standing close to each other. So, when the speaker determines a context from its immediate surroundings (these are typically the objects within a radius of approximately 1.5 m), the hearer needs to derive the same context in order to let the language games run smoothly. Furthermore, when the speaker names an object by referring 'left and behind', the hearer needs to know, where left and behind are from the point of view of the speaker. If both robots are not facing each other, then the hearer (or the speaker) needs to determine the orientation of the other robot so it can determine what direction is left or right from the point of view of the other agent. With current technology and means at the AI-lab the determination of the orientation of another robot, which could be any direction, is a problem. So, two agents that engage in a language game need to stand in front of each other and face one another, before they can determine the context of the game.

Because the robots have only statically placed sensors on their body, usually in front, they can only scan their surroundings by rotating themselves. So, when the robots determine the context, they need to rotate exactly one full circle, in order to end up facing each other again. The discrimination of objects and creation of their semantic representation can be done by the meaning creation described in chapter 4. All the above mechanisms need to be functioning using low-level sensors only (i.e. sensors like infrared, (white) light sensors etc.), because no high-level sensor like camera vision is available at this moment. When the context is derived and a topic is chosen, this topic needs to be pointed at the hearer before the ‘spoken’ dialogue can start.

The language games will be implemented as is explained in chapter 3. Only the features will now be real features instead of pre-programmed features, and thus they can be uncertain with regard to the real-world objects. Furthermore, because they are created by the agents themselves, they will not carry labels like: ‘left’, ‘right’, etc.. But rather, they carry abstract labels, which do not have meaningful names but instead relate to the sensory signals that constitute an object.

6.2.2 Expected problems

Although I already raised some problems that can be expected throughout the implementation of the experiment, I will raise the most influential problems more explicitly in this subsection. Though it may seem that the implementation of the self-organisation of the lexicon will raise the biggest problems, this is not true because the algorithm is already given. And running this algorithm on the robots did not bring up major problems. The biggest problems should rather be expected in constructing the boundary conditions by which the agents should prepare their dialogues and in the recognition process of the objects. It was already identified by Flynn and Brooks [16] that problems that initially seem to be very difficult usually are not, while the most trivial seeming problems usually are. I.e., trivial seeming problems like, for example, the process of finding and facing each other are likely to raise bigger problems than for example adapting a lexicon.

One of the biggest problems that I expected to happen is to let the robots that engage in a language game behave synchronously. For example, if both robots engaged in a language game are looking for each other, then it is better that they do not drive around both, because this will increase the probability that they will miss each other. So, one of the robots should stand still while the other is looking for it. But then: how would one robot know when the other has found it? Or, how do both robots know that they both have identified the topic so they can start to ‘talk’? And how do both robots know when some part of the dialogue has failed and they just have to start over either partially or totally? In the next section I will propose answers to these questions.

Another problem that can be raised at this stage, is how the robots can recognise and find each other. Which sensors should they use, and how should the robots decide that they found each other? How do we resolve the problem of identifying objects and creating meanings? Again, which sensory information do we use? And how does this information need to be ordered?

The next questions I will raise involve what extra-linguistic means should the robots use to clarify the topic of the conversation from the speaker to the hearer. It is, I think most natural to use physical pointing. But, how does the speaker point at an object so that the hearer can identify this particular object as the topic? Is it sufficient to just orient towards this object, or does the robot has to drive towards the object? And how can the hearer determine what object the speaker is pointing at?

And finally, if the preparation of the language game has finished successfully, in what way will the communication continue? I.e., given a topic, how would this topic be indicated (either by name or by spatial relations) and how would the other partner in the communication identify which relation was spoken?

In this subsection I have raised some fundamental questions about how to organise the experiment. In the next section I will discuss the proposed solutions that I thought of initially, whether they worked throughout the development of the experiment or not. In chapter 7 I will discuss the implementation process where the proposed solutions were refined where necessary.

6.3 Proposed solutions

The proposed solutions of the problems raised in the preceding section are the ones that I initially worked out in the development of the experiment. The development of the experiment was more or less done by a means-end analysis. So I divided the problem of implementing language games into smaller subproblems which will be given in the following subsections. These subproblems were then developed further through a process of trial and error, which is discussed in chapter 7. In this section I will try to close the gap between the initial state of the development: a robot with no program, and the goal: playing a language game.

6.3.1 Synchronising the robots

In a pre-study on the implementation of the spatial vocabulary formation done at the AI-lab the problem of synchronisation was already raised as being one (if not *the*) biggest problems of the implementation [18]. Synchronising the robots is a problem due to the parallelism of the system. I.e. all robotic agents run their own program, and the whole system can thus be seen as a parallel program. Therefore the robots cannot have a complete view of the whole environment. Solving this problem needs some very delicate thinking.

One can, for example, naively think that a solution could be found in letting the time be a guide for synchronisation. So we can say: at time t_0 robot 1 (r1) enters state so-and-so, and thus robot 2 (r2) must enter state such-and-such. Obviously, this would be a major mistake for several reasons. First, the robots have no robust time mechanism. All robots have the same computer-board, which has a default clock-cycle of 40 Hz, but if the program that normally is run in the 1/40 second is too large, it exceeds this time-interval and the cycle length will increase. Moreover, both robots engaged in a language game do not simultaneously run the same program, so the mentioned solution will fail. The solution would also fail because one part of the language game (e.g. the process of finding each other) cannot be solved in a fixed time due to the dynamical properties of the environment. Therefore, we cannot say definitely that a certain process will end after a particular time.

The solution then, could be found by creating a finite-state-automaton (FSA) where the robots broadcast a radio-signal at a particular transition in the FSA. A FSA is a virtual machine, which has a finite sequence of possible states. Each state represents a certain process that the robot must execute, and when this process (or state) achieves a certain condition, the machine will make a transition to another state. In the purpose of these experiments it is sometimes necessary that both robots simultaneously make a transition from one state to another. This transition should be caused by the transition of one robot. If this robot broadcasts a radio-signal at the moment it enters a new state, the other robot then can enter the next state as well. Synchrony will thus be achieved by broadcasting radio-signals between the robots during an appropriate transition from one state in the FSA to another. The FSA is defined in a protocol that will be given in chapter 7.

6.3.2 Infrared based robotaxis

One of the next problems that needed to be solved was the decision which sensors to use for recognising and finding another robot. In principle there are a large amount of sensors and methods to choose from. But, because the availability of infrared (IR) sensors at the lab in Brussels, together with the fact that IR can be used as a direct method, the choice was made quite rapidly. It was a decision made on the basis of comparing the pros and cons of the IR emitter and sensor pair in contrast with other systems.

The first consideration in favouring IR was that it was already mounted on the robots, and it was the only system that was not necessary for other tasks in this experiment. The IR module is normally used for *smooth obstacle avoidance*. By smooth obstacle avoidance, obstacles are being avoided when the robot senses a nearby obstacle due to the reflection of IR that the robot emitted. A second consideration was the possibility of using a visual light system, i.e. a light emitter and sensor. This could have been possible, but the charging station in the experimental environment already made use of visual (white) light. So, we had to use coloured light, but earlier experiences with robots distinguishing coloured light were not so good because most coloured lights reveal high noise levels due to the inference with other light sources. Using a polaroid filter was also another option, but appeared to be too expensive. Obviously, camera vision would be the ideal module for doing all recognition. But the camera system at the lab does not work robustly on the robots up to now. Therefore the choice for using IR was made quite easily.

The next choice was the method to be used. The IR module could, in principle, be used in two ways. The first one is to use reflected signals to determine the direction of an object or robot. This causes a lot of problems, because (1) the system only works at short distances, (2) it is very difficult to determine whether the received signals are reflected from a robot or other objects, and (3), the received signals could be received directly from another robot. The latter brings us to the second option. Robots can emit IR, while another receives this signal. The receiver can easily determine the direction in which the other robot is. The robot that is looking for the one that emits IR can turn towards that IR source, and use this source for infrared based phototaxis, or as I called it “robotaxis” or “IR-taxis”.

In order for a robot (e.g. the speaker) to find another robot (e.g. the hearer), I thought of solving the problem as follows (note that the hearer emits IR, while the speaker is not):

1. The speaker turns in the direction of the IR-source until the absolute difference between the two outer sensors is minimal, we call this “IR-orientation”. The speaker now faces the highest level of IR.
2. If the sensed IR level is not high enough when facing the IR-source, then use robotaxis to close in on the hearer. Else the speaker is ready.

3. If the IR level during robotaxis reaches a maximum, then stop and minimise the difference between the outer IR sensors again as in 1.

In these three stages the hearer should be found, but sometimes this may fail as well. This is, however, another problem.

In stage two the speaker uses robotaxis. This is a method derived from phototaxis. In this method the robot will change its motor speed depending on the sensed IR. If there is an IR-source on the left side, then the left sensor value will read a higher output than the right sensor. If the motor values are adjusted accordingly, then the robot can find its way towards the source quite smoothly. I.e., if there is a source on the left-hand side of the robot, the left sensor value is higher than the right one. The robot needs to go left. This is achieved by increasing the speed of the right motor, while decreasing the left motor speed. If now the robot turns too far, the source will appear on the right hand side and the method is used in reverse. In mathematical (PDL) form this process looks like this:

```
add_value(LeftMotor,RoboFactor*(RightFrontIR-LeftFrontIR));
add_value(RightMotor,-RoboFactor*(RightFrontIR-LeftFrontIR));
```

Going through this process, the robot will dynamically home in on the source quite smoothly. [39] IR-taxis is discussed in greater detail in the next chapter.

When the speaker has finished the three stages, it ends up facing the hearer. The hearer though still needs to face the speaker. This can be achieved by means of IR-orientation, as described above under step 1. When both robots face each other, they have to map their surroundings. The method for doing that will be described in the next section.

6.3.3 Perception

In chapter five I have described the characteristics of the robots, together with the experimental environment of the AI-lab. For the purpose of this section I will shortly review the main characteristics. First, the environment consists of a charging station that emits white light, so the robots can find this station using white light phototaxis. The white light sensors, of course, must therefore be used in order to identify the charging station. Secondly, in the environment there are several competitors which all emit modulated light at the same frequency. The robots can find these objects using their modulated light sensors. So this feature can be used to classify this type of object. Similar arguments can be held for using infrared for finding and classifying robots. In this section I will further discuss the characteristics of the sensors as they are measured by the robots during one complete rotation while scanning the surrounding. From these characteristics we can decide how to develop the feature detectors that have been introduced in the former section.

The robots must perceive their environment during a language game by completing one rotation around their axis, while recording their sensors. While they do this they build a map such as is shown in figure 6.1. As can be seen in the figure two sensor values of one type (e.g. IR or white light) intersect. This happens when the robot passes an object. Therefore, I proposed to let the robot decide that there is an object at the place where such an intersection occurs.

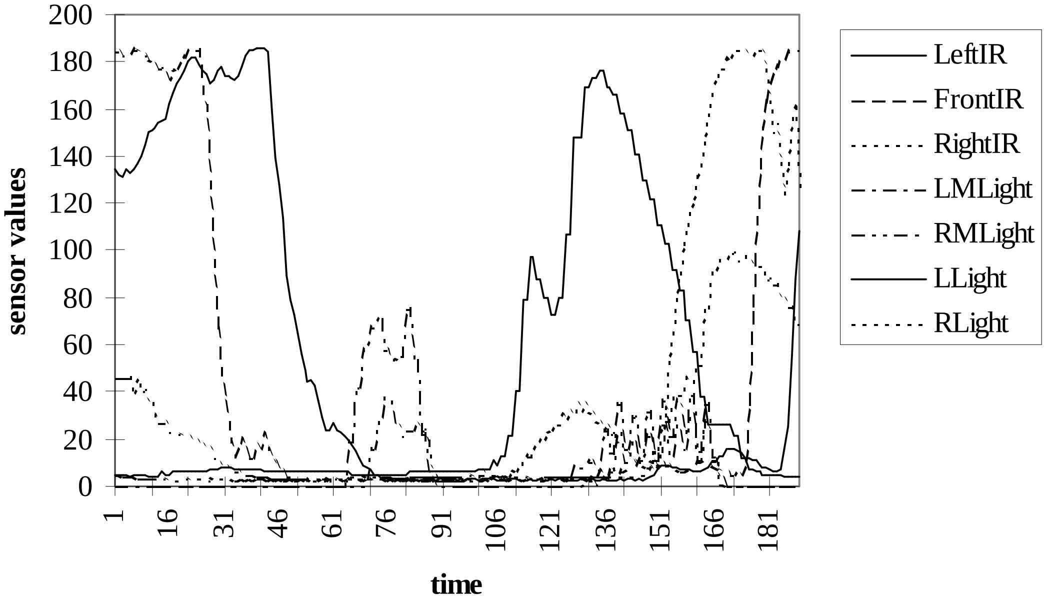


Figure 6.1 *The perception of the surroundings of a robot, rotating one full circle. The rotation starts and ends facing another robot.*

In the next subsections I will describe what is seen for every type of sensor (i.e. white-, modulated light and infrared). In these sections the characteristics of the white- and modulated light are extracted from figure 6.1, while the IR (figure 6.4) was measured in a separate experiment.

6.3.3.1 The charging station

As was mentioned before, the charging station emits normal white light. It normally is mounted with two light bulbs. One with high intensity placed on top of the station, and one with lower intensity placed under the aluminium plate (see figure 5.2). The upper light is meant to allow a robot to find the charging station at larger distances, the lower for shorter distances. This works perfectly well for the phototaxis, but if the robot needs to locate this object it receives two different peaks in the sensed intensities at very different places (see figure 6.2).

As you can see, the left light sensor (LLight) shows two relatively high peaks of intensity. So, the two lights on the charging station are observed as if there are two different objects, when we assume that we pick the maximum peak when we rotate. That the two peaks that occur in this figure were due to the existence of the two lights at the charging station, was clear when I repeated the experiment with one of the lights turned off. We can see that the location of the charging station is approximately at time 160, because there we see an intersection of the two sensors.

So, for constructing a sensory channel for the recognition of the charging station (and more generally for locating white light sources), we could use the difference of the two sensor values as a separate sensor channel, which equals 0 when it is facing the light source. Due to the fluctuations in the values of the sensors, it may well be possible to have more than one intersection within a few, say 10, cycles. To avoid the problem that the robot will recognise multiple objects if there is in fact only one, I let the robot decide that there can be only one object within an interval of ten SMB-cycles. A SMB-cycle is defined as one period of sensory-motor process on the SMB2 board, i.e. approximately 1/40 second.

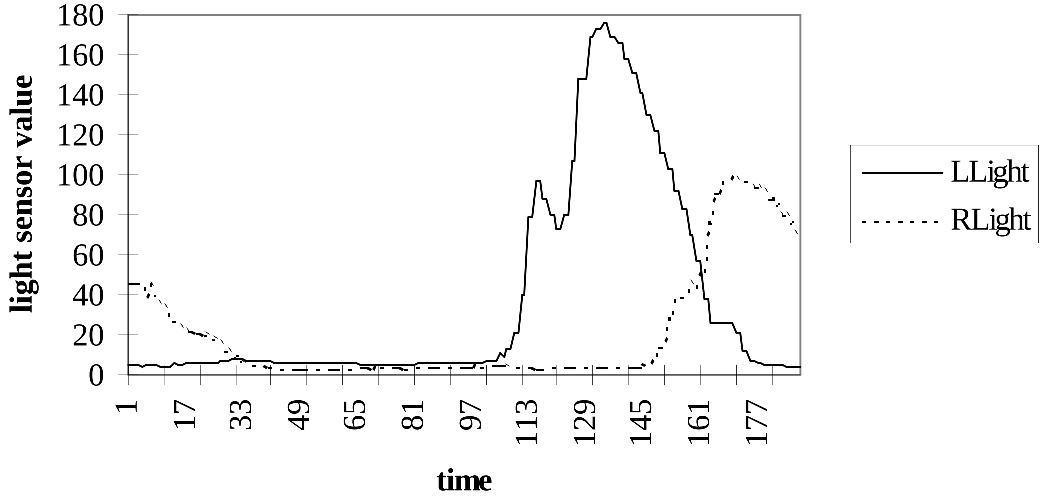


Figure 6.2 *Characteristics of the white light sensors when scanning for the surroundings.*

6.3.3.2 The competitors

In this section I will discuss the modulated light sensor characteristics as I did in the previous section for the white light sensors. As can be seen in figure 6.3, the modulated light sensors have no background noise, i.e. there is no stimulation of the sensors when there is no modulated light, whereas the white light sensors and the IR sensors always have a background noise level.

As can be seen in the second occurrence of a competitor, there may be a lot of fluctuations in the sensed characteristics. These fluctuations are due to the fact that the modulated light sources are narrow field emitters, i.e. they emit their signals in a very narrow beam, so the further you get away from the source, the wider the gap between two emitters that can be seen. This also occurs as the robot is rotating too slowly, e.g. because the energy level is extremely low. So, we measure several peaks for one object. This is also a reason to recognise an object at the intersection of the right- and the left sensors. As we can see, there may be several intersections for one object. These intersections are filtered out by the constraint that there cannot be more than one object within a certain time interval. This may not be very reliable, but it will do, as we shall see later.

Note that in the figure, the area of the right sensor almost overlaps the left sensor completely. When the robot is rotating slower, due to lower energy level, there may not be any overlap, then the object will be missed. In later experiments this shortcoming should be solved.

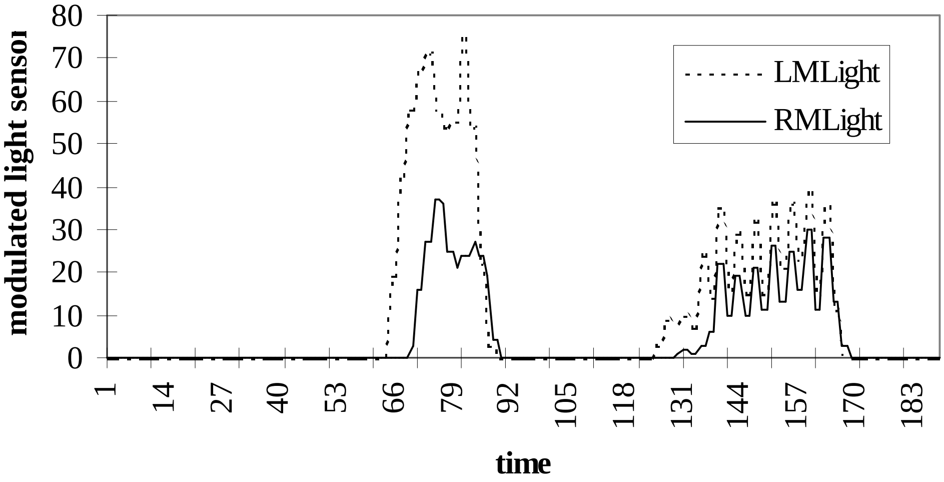


Figure 6.3 The characteristics of the modulated light sensors while scanning the surroundings. Note that there are two competitors. The second is perceived with a lot of fluctuations in the amplitude of the signal. The separate modulated light emitters that are in one competitor in order to emit light in all directions cause this phenomenon.

6.3.3.3 Other robots

As mentioned, the robots can be classified using infrared. The characteristics of the left- and right IR sensors are shown in figure 6.4. Here you see a short period of the scanning robot that rotates 0and passes another robot that emits IR.

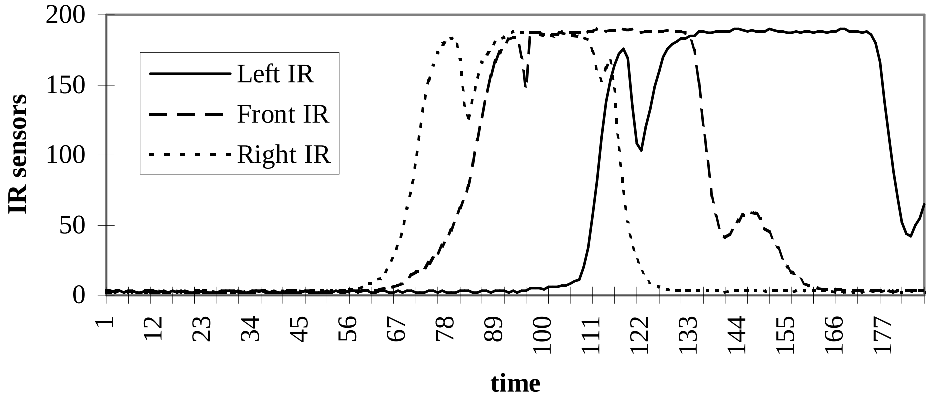


***Figure 6.4** The characteristics of infrared sensors as perceived by a rotating robot that passes an IR emitting robot.*

As can be seen, this figure looks a lot like the former figures in that it shows a peak for both outer sensors, and an intersection on the middle. There are, however, three sensors for the IR on the robot: one on the left front, one on the front and one on the right front. As can be seen, the intersection of the left- and the right sensor occurs somewhere in the middle of the top of the front sensor (approximately at time 113).

The fact, that all the figures shown in these subsections have more or less the same characteristics, has a nice consequence. We can use one method for determining the existence of all objects. The only differences are the type of sensors used. The method used here is analogue to the method used for IR-orientation. If a robot is scanning its surroundings, it senses the superposition of figures 6.2, 6.3 and 6.4. This yields figure 6.1. If the appearance of an object is detected, the robot records the values of all sensory channels as the features of the robot. I proposed to define sensory channels for every type of sensor, with the value of the sensor as if the object was seen right in the front. I.e., the average values of the two sensors of the white- and modulated light sensors, and the front sensor value of the IR are recorded when an object is seen. The time is an indication for the spatial relation of that particular object. The implementation of the perception is further discussed in section 7.4.

6.3.4 Pointing at objects

How can we let a robot point at an object in such a way that another robot can determine what object is pointed at? At first this seemed to me a very difficult problem. The robots that are used do not have arms or fingers to point, nor do they have sophisticated means for determining in which direction another robot is pointing.

Initially I thought of solving this problem by letting the speaker drive towards the object, which it wanted to point at. But then the robots cannot talk about spatial relations anymore, because the spatial situation of the robots has changed. Another option I thought about was letting the object that is most nearby be the topic. But, how can a robot determine which object is nearby if the characteristics for each sensor differ a lot. And, if this determination succeeds for one robot, will it succeed for the other robot as well?

Therefore, I thought to mount four IR emitters in perpendicular directions on each robot. So the speaker can point to an object by orienting towards this object after it was facing the hearer. And the hearer then can determine in which quadrant the topic is to be found by counting the sensed peaks of IR. This method is not very precise, but because there are not so many objects in the environment I expect that this method will work well enough.

6.4 Conclusion

In this chapter I have defined the experiment and proposed how to close the gap between the initial state of the implementation of the experiment and the end-goal of the implementation. I have not discussed the subproblem of discrimination- and language games, because for these subproblems the solutions were already given in chapters three and four. As we shall see in next chapter, not all the solutions given here work out the way it is expected, and they were refined in the process of trial and error.

Summarising, in the experiment two robotic agents must engage in a series of language games. They randomly explore the ecosystem, until of them initiates a language game. The initiator becomes the speaker, while the other the hearer. In order to build a coherent map of their immediate surroundings, both robots have to stand close to each other, while facing on another. After the agents have built a map of the surroundings, they both can play a discrimination game and finally the lexicon formation (including the actual transfer of language) can be held. The approach of the second part of this chapter was to identify the sub-problems of the means-end-analysis, and to propose solutions to close this gap. Many straightforward solutions have been proposed and tried out, as well as some that are not discussed in this thesis, because they were rather irrelevant. The proposed solutions were mainly grounded on small experiments that I have done in order to investigate the characteristics of the robots and their sensors. For clarity reasons I will not report on these irrelevant experiments.

The proposed solutions were the building blocks on which the implementation was based. The next chapter discusses the implementation process in greater detail. In that chapter I will also discuss the cognitive relevance of some of the choices that were made in the implementation process, as well as for some proposals made in this chapter.

7. THE IMPLEMENTATION OF THE LANGUAGE GAMES

7.1 Introduction

Both robots that are participating in a language game first need to find each other. This is necessary, because when they are communicating they both need to identify their surroundings coherently. So, they have to be able to map their own map of the surroundings on the other's. Furthermore, the hearer has to determine what object the speaker points at. In order to be able to communicate about spatial relations it is also necessary that they are facing each other, so they know each other's orientation. This is more or less alike with the standard conventions in converations between human beings, although for different reasons. As we shall see, it is not a trivial task for the robots to find and end up facing each other, mainly because of physical constraints that are inherent to the robots and their surroundings.

Questions could be asked like: How can robots recognise each other? How does one robot find the other? How does one robot know that the other one is looking for it? And how do they both know when they have found each other? Answers to these questions and the like have already been given partially in chapter 5. In this chapter I will comment on the trial and error phase of the implementation of the experiment as it is described in the preceding chapters. I shall not discuss all the details of the development of the experiment, I will only comment on the most important and relevant problems that I have encountered.

In the next section I will describe the basic structure of the protocol that is used in order to conduct a language game synchronously. In the remaining sections of this chapter I will show how the formal protocol evolved. In section 7.3 the implementation of the process for finding each other and facing one another at a close distance will be discussed. In section 7.4 the implementation of the perception and discrimination will be discussed. Section 7.5 will be devoted to the pointing problem. And in section 7.6 I will discuss the implementation of the actual language formation in robotic agents. In section 7.7 I will give some concluding remarks on the implementation of language games. Finally, in section 7.8 I will define the formal protocol as it developed during the trial and error process.

7.2 The protocol

In order to let the robotic agents *communicate*[2] with each other, they must go through various steps. First, the robots have to find each other. As was mentioned before, the robots need to stand close to each other, while facing one another before they can build a map of their surroundings and communicate. So, secondly, when the robots have found each other, they have to face one another. The third step is for both robots to build a map of their surroundings, so that they can determine what objects consist the context of this language game. After they both have built a map of their surroundings they need to end up facing each other again. Now the speaker has to point at the topic, so the hearer can

---

[2] By *communicate* in this sense I mean the final steps of a language game, where the actual words are being processed and transferred from one robot to another.

identify this topic. When both robots have thus determined the topic, they have to distinguish this topic from all other objects in their *field of attention* (or context). When both agents discriminated the topic, they need to end up with a set of distinctive features that they can use for the language formation as is described in chapter three. Schematically we have the following starting point of our protocol for the language games, which is a summary of chapter five:

0. All robots drive around. One robot will initiate a language game, and the other robot may confirm so a language game can start.
1. Both robots have to find each other.
2. Both robots have to face each other.
3. Both robots have to build a map of their surroundings. I.e. they determine the context.
4. The speaker has to point to the object, which is the topic.
5. The hearer has to determine the topic.
6. Both robots heave to discriminate the topic from the other objects in the context.
7. The speaker has to encode an expression from the in 6. derived distinctive feature sets.
8. The hearer has to decode the expression and evaluate the success of the conversation.

Note again that this is only the *basis* of the formal protocol that evolved through the trial and error process that I will discuss in the coming sections. The formal protocol will be given in section 7.8. The protocol is implemented in a FSA as mentioned in the previous chapter. The FSA is divided in two separate FSAs: one for the speaker, and one for the hearer. Initially both agents are in state 0 for both FSAs, which means that the agents explore the environment randomly.

7.3 How to find each other?

As was discussed in section 5.3.2 the robots should find each other using their IR module. The robot that is searching the other robot must not emit IR, while the robot that is looked for must. It was also discussed that IR-taxis (or what I call *robotaxis*) should be implemented for the search. For alignment, the method of *IR-orientation* could be used. IR-orientation is more or less as the same the process of IR-taxis. It minimises the absolute difference of both outer IR sensors, but the forward movement is left out. Remains the question of how to implement these methods, as well as, of course, the question of when a language game would be initiated. I shall start to discuss the latter problem.

7.3.1 The initiation of a language game

I first tried to let the agents decide randomly to initiate a language game, cf. the simulations described in [44] and [45]. I.e. the robots drive around, while every SMB-cycle (one sensory-motor process of 1/40 second) they generate a random number. If this number equals 1, then they send a radio message saying 'communicate'. If the other robot confirms that it received the message, the first one would go into the first *speaker-mode* of the FSA where it emits IR and waits until the other robot finds him. The second robot would go into the first *hearer-mode* and start to look for the other robot using robotaxis. This method of initiation, however, brought along one fundamental problem. Most of the times a language game was initiated, the robots were separated too far from each other. So, they needed too much time to find each other, *if* they found each other at all.

A part of this problem was the character of the IR emitter. This IR emitter emits IR in a narrow field, i.e. the beam of IR that is emitted has a rather small angle (ca. 35 degrees), see figure 7.6. So, it was hard to detect the presence of a robot. Furthermore, when the presence of a robot was detected from a large distance, the hearer sometimes left the region where the IR was sensible. So, the search had to start all over again. In one of the first experiments using this method, it took about 25 minutes before the robots actually played two language games. In this experiment the robots only had to find each other and simulate a language game while facing each other. Another problem was to find a sufficient criterion for the hearer to decide when it was close enough to the speaker, so it should stop. In this experiment this decision was made when a maximum in infrared was reached from the characteristics that can be seen in figure 7.1a. This problem will be discussed in greater detail later in the next subsection.

Having only two language games in 25 minutes is not enough to form a language. Remember that the lexicon we want to be formed may need more than thousand language games. A side-problem is that the robots can only work for a maximum of 40 minutes before they need to be recharged. Therefore I needed to reconsider when to initiate a language game. In the experiment I just mentioned, I noticed that the robots often did *not* initiate a language game while they were close to each other. I thought it would be easier for the robots to find each other if they initiated a language game when they were already close to each other. So, I had to figure out a way to implement this. The problem, then, was how could a robot sense the presence of another robot? If I let the robots both emit IR, the other robot can sense this. But, then the robots also could mistakenly identify the reflected IR from objects, such as the wall, for another robot. This would cause robots talking to the wall consequently.

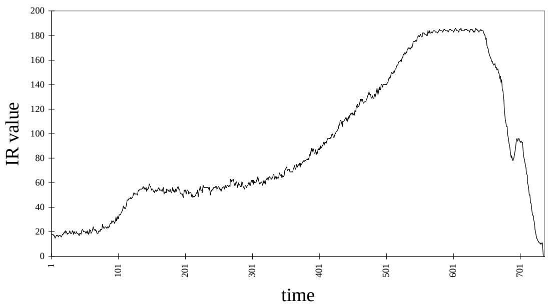


Figure 7.1 The IR characteristic of a robot driving straight towards an IR source starting from a distance of 3.5 meters.

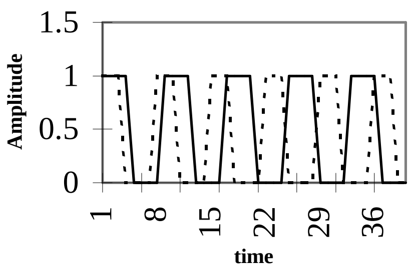


Figure 7.2 Pulses of different frequencies are always out of phase.

What would happen if the robots emit pulses of IR, so one robot can sense the presence of another robot in the period that itself is not emitting IR? In principle this must work, if the pulses of different robots are always out of phase. So, if one robot emits pulses of a time length t_1 with intermediate periods of t_0 where it does not emit IR, then a second robot must send a pulse for a time $t_2 \neq t_1$ (see figure 7.2). In this case both robots send pulses that are always out of phase, so the robots may sense one's presence in the intermediate period. This method, however, brought up another problem.

The hardware of the IR module (i.e. the combination of emitters and sensors) was designed only for smooth obstacle avoidance. I.e. this module is mainly designed to detect reflected IR send out by the robot itself. The sensors have a build in comparator for comparing signals that are emitted and signals that are received. This comparator is modulated at a certain frequency. If you want to sense IR, while not emitting IR, then the received signals go through a filter that is modulated for a frequency as if the robot is emitting IR [54]. But it is not emitting IR! If, for example, a robot is driving towards an IR source, not emitting IR and the modulation turned on, it senses the characteristics as is shown in figure 7.3. These characteristics are difficult to process. Turning off the modulation, when not emitting IR, can solve this problem. This has two consequences: (1) It reverses the characteristics of the sensed IR. I.e. when the modulation is on, high sensor values mean no IR and low values mean the presence of IR, and when the modulation is off, then the opposite happens. (2) If the modulation is turned off (or on), then it takes a while before the high values are relaxed into low values (or the opposite), see figure 7.4. As can be seen, it takes about 12 SMB-cycles before the sensor values are relaxed. So, the robots can only sense IR after this period.

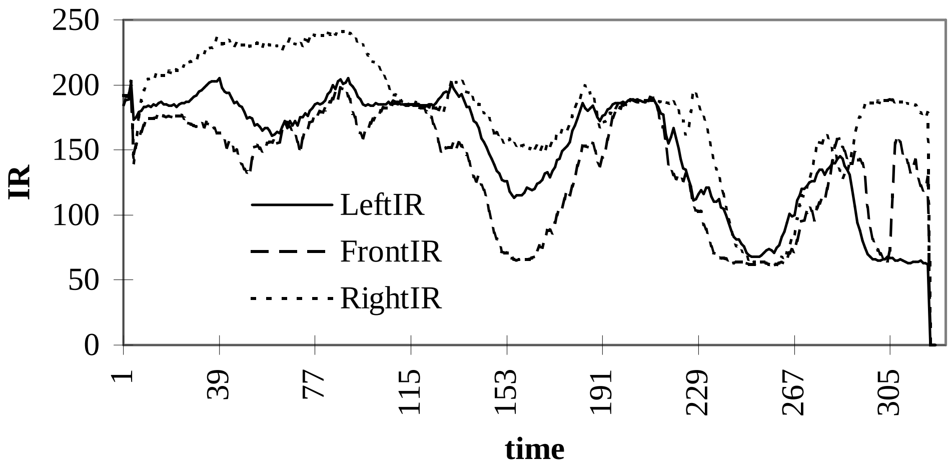


Figure 7.3 *A robot driving towards an IR-source, while the modulation on.*

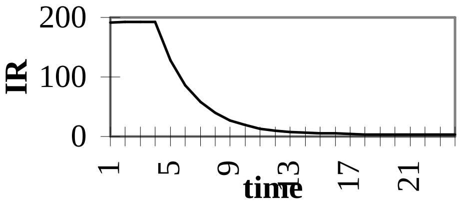


Figure 7.4 *Relaxation of the IR signal after the modulation is switched off.*

This information was important for the design of the pulses to be emitted. The period in which no pulses can be sent (t_0) must be longer than the relaxation time t_{relax} = 12 SMB cycles, for safety reasons I let t_{relax} = 15. I decided to let t_0 = 25, so the robots have 10 cycles to detect IR. To increase the chance that infrared is sensed in this period, the length of the emitted pulses should be longer than t_0, so t_1 = 30 and t_2 = 35. The characteristics for the IR sensors of a robot, while emitting pulses of IR and driving towards an IR source can be seen in figure 7.5. If the robots sense an IR value higher than a threshold value in the detecting period, then a broadcast is sent out for communication. This threshold is set somewhat higher than the noise-level for infrared.

Although the robots still initiate language games at improper times, this method improved the amount of language games significantly. Sometimes a language game is initiated due to reflections of the wall, which arrive at the robot after the relaxation period. But in the same kind of experiment as is mentioned above, a language game can succeed every two minutes. There was, however, one other important difference with the former experiment. This time the hearer did not seek for the speaker, but the other way around. I implemented this, because it was the speaker that sensed the presence of the hearer. Therefore, this robot should home in on the other robot more easily.

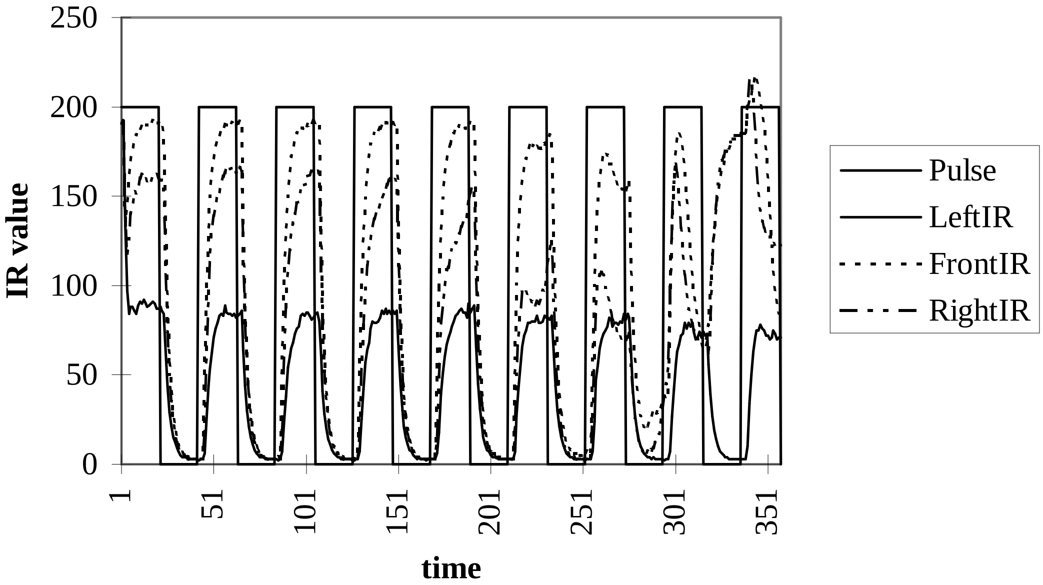


Figure 7.5 *A robot driving around, emitting (square) pulses of IR. Towards the end it closes an IR source, now the sensor values increase in the non-emitting period. Note the short relaxation period after turning off the pulse.*

7.3.2 IR-taxis and -orientation

The homing in on an IR source, i.e. the robotaxis, was a rather difficult task to implement. This was mainly due to some physical problems concerning the narrow field emission of IR. As was mentioned before, the IR emitters only emit IR in a beam with an angle of 35 degrees (figure 7.6). When I started the implementation the robots only had one IR emitter. Another problem was finding a good criterion to decide when the speaker got close enough to the hearer, so it should stop. The IR orientation also brought up a physical problem, which was caused by the asymmetrical set-up of the IR sensors. In this subsection I will discuss all these problems. I will also discuss the whole process of the implementation of these tasks. I.e. the process that starts after the initiation of a language game and ends with both robots facing each other at a short distance.

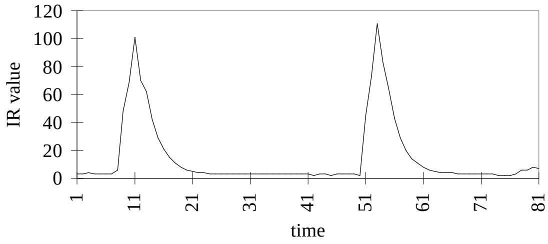


Figure 7.6 *The angle of an IR emitter as is spotted by an observing robot. The robot that emits IR is rotating, the two different pulses represent 360 degrees.*

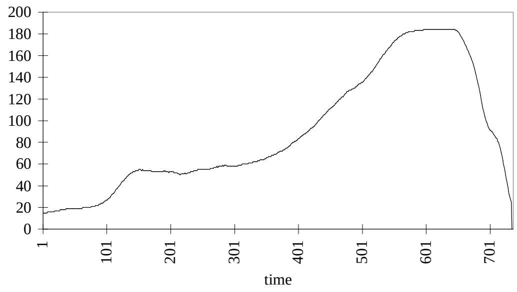


Figure 7.7 *The same figure as fig. 7.1, but now the signals are averaged over a walking period of 15 units. Note that the y-axis represents the IR value.*

I start the discussion with the implementation of the IR-taxis. The mathematical form of the IR-taxis has been given in the previous chapter. Initially the robots were equipped with only one IR emitter that was mounted on the front. This emitter emitted a beam of 35 degrees, with highest amplitude in the middle and decreasing amplitude towards the edges of this beam (figure 7.6). This figure shows what a robot senses with its front IR sensor, when another robot, that emits IR to the front, is rotating. This rotation starts with the back of the robot facing the observing robot and it completes more than one cycle. It is not difficult to imagine that this set-up of IR emitters did not yield a nice performance. I would rather have the robots emitting IR uniformly in all directions. I.e. that they emit IR at equal strength in all directions.

This could, for example, be done as follows. Mount an IR emitter in such a way that it emits upwards. Above the emitter, a cone of a highly reflective material should be mounted, so that the IR is spread into all horizontal directions. I have not implemented this, because of several reasons. It probably would be very difficult to make such a device, so that it would work properly. Moreover, this probably would cause a high loss of the strength of the signal, so it would decrease the distance in which the presence of a robot can be sensed. As can be seen in figure 7.1, direct IR can already be sensed from a distance of ca. 3 meters in ideal circumstances. Although, it may be interesting to test the device I described above in a future experiment, I decided just to mount more IR emitters. In order to cover a region of 360 degrees at a reasonable strength I could divide the 360 degrees in parts of 25 to 30 degrees, and mount IR emitters in all these parts. That, however, would need (for parts of 30 degrees) 12 IR emitters, while there are only 8 output ports for IR on the SMB2 board. A device with 12 (or even more) IR emitters so that they are connected to one cable only could be constructed. This, however, would cause a new problem. This device, either using serial or parallel connections, would need a lot more power than the batteries can serve. So, I had to settle with a smaller region where the IR had influence. I started to use only three emitters. One emitting to the front, the other two emitting side-wards, while all were mounted on the front of the robot. This already increased the performance significantly. Note that at the time I was working on this problem, I had not decided yet how to implement the pointing, as I reported in section 5.3.3.

In figure 7.1, a robot is driving straight towards an IR source starting at a distance of 3.5 meters. When the robot gets closer to the other robot, a maximum in the IR-level will be reached at a distance of ca. 30 cm. At first this seemed to me to be a perfect criterion to stop. I.e. the robot should stop when the maximum of IR is reached at a certain threshold. As can be seen in figure 7.1 there are a lot of small fluctuations in the IR-level. For calculating the maximum in the IR characteristics such fluctuations are not wanted, because they may cause the maximum to be a local one. Therefore, I let the robots calculate the temporal (or walking) average of the IR to filter out these fluctuations. I.e. I calculated the average of the last 10 SMB-cycles. Then we get the characteristics as in figure 7.7. Of course, this method again caused some problems. The biggest one I observed occurred when a robot entered the influence of the IR too close to the other robot. Then a maximum cannot be reached, because, when it further closes in on the robot, the IR is only decreasing.

I mistakenly used the wrong characteristics of the IR-taxis. As was mentioned, figure 7.1 and 7.7 shows the characteristics of IR when a robot is driving *straight* towards an IR source. When using IR-taxis, however, a robot does not drive in a straight line towards an IR source. It rather drives towards the source in a zigzag movement as in figure 7.8a. Figure 7.8b then shows the resulting IR characteristics.

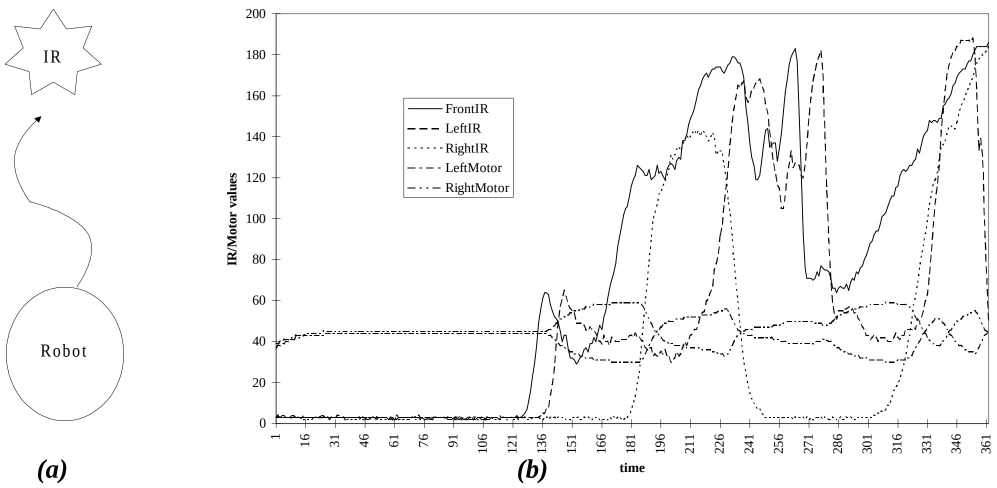


Figure 7.8 (a) *The zigzag movement a robot makes when using IR-taxis in driving towards an IR-source.* ***(b)*** *The IR-characteristic when closing in on an IR-source, together with the motor values (fluctuating around 45). As can be seen, this is a total different graph than in fig. 7.1.*

As can be seen, the zigzag movement of the robots yields a totally different characteristic than the one shown in figure 7.1. The characteristics are measured in a robot that started the IR-taxis from ca. 3.5 meters with an edge of ca. 30 degrees to the right side (measured from the axis along the IR beam) and facing the intersection with the IR beam at ca. 2 m. from the source. At approximately time 120 (fig.7.8b), the robot enters the influence of the IR. There are now more maxims as a result of the zigzag movement. The robot dynamically minimises the difference between the left- and the right- IR sensors. Due to the forward movement, the robot can cross the centre, thus increasing the difference again. The criterion of stopping at a maximum of IR, obviously, is not valid anymore. So, I chose to set a threshold for the front value of infrared. Although, sometimes this value is exceeded at a relative large distance (see point 220 in figure 7.8b), this method worked quite well in practice.

Later, when I worked on the implementation of the pointing process, I mounted four IR emitters in perpendicular directions, with the centre in the geometrical centre of the robot. IR is emitted to the front, the back and both perpendicular sides. This improved the performance of the search process a lot. Now the speaker could also find the other robot approaching from aside and the back, while earlier it could only approach the other robot from the front.

Directly after a language game is initiated, the speaker already orients towards the hearer in order to speed up the IR-taxis. If this is not done, the robot could loose the influence of the IR by driving initially in the wrong direction. The process of IR-orientation is first to determine the direction in which the IR-level is highest. The robot then turns towards this direction, calculating the absolute difference between the two outer IR sensors every SMB-cycle. If this absolute difference reaches a minimum while the front sensor reads a value higher than the noise-level, then the orientation is completed (see fig. 6.4). The speaker repeats this process, when it got close enough to the hearer by robotaxis. This latter process is unnecessary most of the times, because the speaker already ends up facing the hearer. Sometimes the speaker finishes robotaxis not facing the hearer. When the speaker faces the hearer, the hearer uses the same process for orienting towards the speaker. At this transition, the hearer changes its IR modulation. Therefore the hearer has to wait until this transition has relaxed as shown in fig. 7.4 before it starts to rotate. Every time such a transition is made, the robots have to wait until the IR-level has relaxed.

As mentioned before, the process of IR-orientation brought up a physical problem in the design of the robot. The outer IR sensors were mounted on the robot asymmetrical. This caused the orienting robot consequently finish orienting some 30 degrees late. It took me quite a while before I identified the cause, but when it was identified it was solved quickly. I changed the design of the robots in such a way that the sensors were mounted symmetrically. This problem will be further discussed in section 7.4.2. It may be that these kind of problems caused, for example, human eyes to evolve into symmetrical devises as well. It seems that natural selection solves these kind of problems. I think that such an argument can defend some of the critique for the (re-)design of the robots by human developers who take over the roll of natural selection. This, however, is a philosophical discussion that I will not discuss further here.

Summarising, the development of the process, in which robots need to find each other and end up facing each other at a close distance, was a very difficult one, if not the most difficult problem of the whole project. I solved it by letting the robots emit pulses of IR and when one robot senses the presence of another robot a language game is initiated. The robot who first perceives the other, becomes the speaker, while the other one becomes the hearer. The speaker then initially orients towards the hearer using what we call IR-orientation. It then closes in on the hearer using IR-taxis, and then finally orients towards the hearer again. When the speaker finishes this process, it broadcasts a message 'aligned' to the hearer, who then orients towards the speaker using the same method. If the hearer finished this orientation, then it broadcasts 'aligned' to the speaker, so the both robots can transfer into the next state of the FSA. Here the process of perception starts which is described in the previous chapter. In the next section I will continue with the phases that follow the discrimination processes. In that section it is assumed that both robots determine what objects are in their neighbourhood. The speaker randomly chooses a topic from its context and points at it, so the hearer can determine the topic as well.

7.4 The implementation of the discrimination games

7.4.1 Introduction

For the implementation of the discrimination games I have constructed some sensory channels which yield sufficient information to discriminate the objects that need to be recognised in the context of the experiments. Arguing that they speed up the evolution of an agent, so we get a base from which the theory can be tested can defend the intelligent construction of these sensory channels by the developer. The purpose of this experiment is not to investigate the evolution of hard-wired sensor channels, like for instance the eye, but it is constructed to investigate the evolution of language and meaning creation. Therefore it is assumed the agent already evolved the necessary receptors (or feature detectors).

One of the most fundamental questions I encountered, was how to represent the objects in a singular way, so these representations could be used as features for language games. In other words: how do we represent the classification of objects, and what kind of representations is useful for the language games. This question came forward due to the fact that in the different situations, i.e. in different points of view for the robots, there appeared different sensor values for the same objects. So, the classification would become rather ambiguous. Also, the theoretical model in [43] presupposes that the features of objects being given are fixed, which of course is not true from different points of view. The model of language games assumes that the features that represent the objects and that the agents use to create the lexicon are fixed throughout time. Both assumptions are obviously not true. In section 4.3 I have discussed that these assumptions do not really matter, because the system can deal with these variations of real-world representations as an inherent characteristic of the proposed mechanisms.

In the next section I will describe the development of the actual perception. Section 7.4.3 discusses the implementation of the mechanisms in greater detail. In section 7.4.4 I report some important conclusions on the implementation of the meaning creation.

7.4.2 The sensory channels and perception

As was mentioned before, the robots that take part in a language game will make one rotation around their axis, while scanning the surroundings in order to derive the context of a discrimination and language game. While scanning their surrounding, they are looking for an intersection of the outer sensor values as was described in 6.3.3. At the moment that such an intersection occurs, a record is made for all the sensory channels. I have constructed sensory channels for the modulated light -, the white light -, and the infrared sensors. In this section I will first discuss some problems that I encountered during the implementation of the scanning process on the physical behaviour of the robots. After that I will define the sensory channels more precisely.

During the implementation of the scanning process, when the robots have to rotate around their axis for one circle, I encountered some problems that I will discuss here. Finding criteria for deciding when to stop was a hard problem during the implementation for two reasons. The first one was that the robot tended to stop when it faced the charging station or sometimes the wall. The heat of the light source caused the IR level to increase a little bit, especially when the robot was relatively close to this source. Although measurements of this level gave me the information that it was not higher than ten, I decided to set the threshold to fifty. This because the other cause of the heightening in IR level was the reflection of IR emitted by the other robot from the walls, when the language game was done near the wall. The reflection of IR could increase the IR level significantly to even values higher than 100. Setting the threshold to 50, however, seemed to work quite well.

The second problem I faced was that, after I had dealt with the former problem, the robot always stopped too late. I.e. it almost structurally stopped about 20 to 30 degrees too late. I first thought it was due to the fact that in the same SMB-cycle where the robot stopped, the robot already executed the first discrimination game, which caused the default 25 msec of the board cycle to extend. Maybe this was true, but after delaying the first discrimination game, no significant improvements were found. Then someone told me that it could be caused by the fact that the infrared sensors were not mounted symmetrically. The front of the IR sensors are metal boxes with unequal sized sides mounted on an equal sized rectangular plate. The metal boxes have their sensors in one corner and all boxes are mounted on the plate in one way, so that they cannot be mounted symmetrically (see figure 7.9a). Carefully executed measurements showed me that indeed the intersection occurred about 25 degrees late. Thus I changed the robot in such a way, that the sensors are mounted approximately symmetrically (figure 7.9b). It increased the performance significantly. The slight error in symmetrically of the sensors was no problem in the other experiments of the lab, because for the obstacle avoidance the exact place of the sensors is not very important. This is a beautiful example of problem solving in the area of robotics, where it often is more useful to solve a problem directly with the physics of the system, instead of adjusting or creating complex algorithms, as was discussed by Flynn and Brooks [16] and indicated by Steels [38].

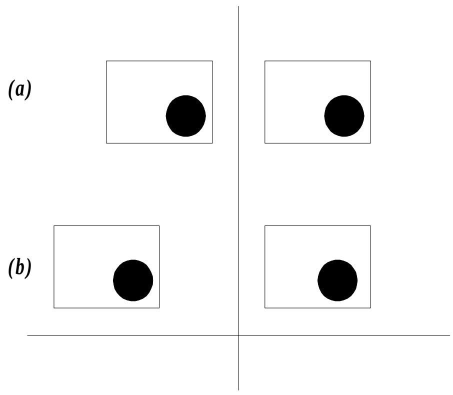


Figure 7.9 (a) *The IR-sensor boxes are placed symmetrically, but the (black) 'eyes' are not.* ***(b)*** *The configuration has changed so that the 'eyes' are symmetrical.*

The sensory channels are evaluating the input continuously, during the scanning period. Only when an intersection of one of the three sensor-pairs is sensed, the system decides that there is an object. This decision is only made *if* the centred sensor value exceeds a certain threshold, because at noise-levels there may be intersections as well. The centred sensor value for the IR sensor is the front sensor value, for all other sensors it is the average value of both outer sensors. So, it is as if the object is 'seen' in the front. The threshold represents the upperbound of the noise-level that a particular sensor senses. If an object is 'seen', then all the sensory channels are recorded and associated with this object as a feature. This feature gets the value of the centred sensor value. So:

$$\mathtt{Feature}_j(o_i) = sc_j(o_i)$$

At the same time a record is being updated for the spatial relation of this object. To be more precise: the position is recorded as number of the SMB-cycle from the start of the scanning process. When the circle is completed, the relative directions of the objects are calculated in x- and y-co-ordinates. Although it might be possible, no indication is made on the distance of the object. The sensory channels for the spatial relation then looks like this:

```
sc-xi=cos(2π(Positioni/FullCycle))
```

and

```
sc-yi=cos(2π(Positioni/FullCycle))
```

Here sc-x_i/sc-y_i is the sensory channel on the x/y-axis (as in normal xy-graphs) for object number i, Position_i is the position of object i in SMB-cycles, and FullCycle is the number of SMB-cycles for a complete rotation. This way we can assign spatial relations, cf. Steels [44] as follows:

```
left: sc-x < 0, right: sc-x > 0, align:  sc-x = 0,
front: sc-y > 0, behind: sc-y < 0, aside: sc-y = 0.
```

So these functions give a spatial description of all objects i that are in the field of attention.

Translating the above into the terminology of section 4.2, a feature detector $d_k = <p_k, V_k, \phi_k, sc_j>$. Here p_k is the attribute name. I will give the attributes names cf. [48] and [51], as sc_j, sc_j-0, sc_j-0-0 or sc_j-0-1 etc., where sc_j stands for sensory channel j and every -0 or -1 suffix denotes a further segmentation of sensory channel j in the lower region (-0) or in the upper region (-1) of the domain of the former feature detector. V_k is a set of possible values for the function ϕ_k. In the experiments I have done all ϕ_k are **I**, the identity function. So, a feature is a pair (p_k v), where $v=sc_j(o) \in V_k$ is the centred sensor value for sensor channel j and object o. I have connected the sensory channels to the sensors as follows:

```
sc0 - white light sensor
sc1 - modulated light sensor
sc2 - IR sensor
```

Because of the close resemblance defined here I may sometimes use the terms feature and feature detector intertwined.

7.4.3 The implementation of the mechanism

The implemented mechanisms for playing the discrimination games is roughly the same as given in [43], although a few changes have been made. In this subsection I will describe the process of the discrimination games as I implemented them.

If the scanning process is completed the robot will have enough data to play a discrimination game. It has located several objects that I will call the *objects in the field of attention* or the *context*, cf. [43]. The discrimination games were implemented according to the principles stated in [43]. The algorithm that I have extracted from this paper and adjusted for the physical robots is given below. Although the algorithm can be used for classifying the spatial features as well, it is implemented for the classification of the perceptual grounding of objects only. The algorithm is as follows:

1. Choose the topic of the discrimination game (dg).
2. For all the objects in the field of attention create features for all the sensory channels, and connect them to the feature detectors (d_i) that represent that feature. I.e. if the feature value of an object satisfies the conditions of a d_i, then this d_i is associated with this feature, else it is not.
3. Extract the distinctive feature sets ($D=\{D_j\}$) according to the definition given in section 2.2.
4. If $D=\emptyset$, then the dg was a failure and the d_i's need to be adjusted according to rule 1 in section 2.2, else the dg was a success and the robot will choose the best distinctive feature set D_b (in case of more than one set) according to rule 2.

As mentioned in section 7.4.2 I sometimes intertwine the terms feature and feature detector. In fact I use the feature detectors directly as features, so the distinctive feature sets consists not of real features but mere feature detectors. Although the term feature refers to a temporal property and a feature detector is a fixed relation. The representations of distinctive feature sets are implemented as arrays of identifiers, which refer to the feature detectors. The feature detectors are also represented in arrays, but in a multidimensional array. In this array all information of the feature detector is stored, such as upperbound, lowerbound, reference to the particular sensory channel, the use and success scores.

The algorithm I used is somewhat different than is described by Steels in [43]. In his algorithm only a use factor is recorded, while I chose to record a success score as well. I think that this will increase the efficiency of the selection process, because now successful features will be used more often and the pruning of unsuccessful features (as a *forgetting* mechanism) can be made more efficient. Every time a feature is selected as (part) of a distinctive feature set, the score for use is increased. Because a feature may be an element of several distinctive feature sets, it may well be possible that in one discrimination game the use of a feature is increased by more than one. If a distinctive feature is used in the language game, i.e. its associated words are expressed, and then a success score is increased.

7.4.4 Conclusions of the implementation process

The implementation of the discrimination games was relatively easy, although there were some difficulties, such as implementing the physical process of scanning the surrounding. The choice of representation of the features and the implementation of the process to derive the distinctive feature sets was also quite difficult. It is very important to design the right representation for your data so references in different functions can be made easier. I chose to design only feature detectors to store data, while the distinctive feature sets were mere a reference to those feature detectors. Choosing the right design of sensory channels made the implementation easier than expected, because if you have the proper data to process, the mechanism of playing discrimination games is rather simple.

I only implemented discrimination games for perceptual grounded meaning creation. I did not implement these games for classifying spatial relations, although the same algorithm could be used. The distinctive features that classify spatial relations, however, classify meaning at a different level. This is because spatial relations are only temporal features of an object, and should therefore be treated as such. I.e. there should be a clear-cut distinction between features of spatial relations and perceptive features of objects. So that an agent can choose what kind of relations it will express, and if we do not separate this distinction in the discrimination games, the features would interfere with each other.

Future work on the implementation of discrimination games should deal with these spatial relations. There also is a need for discriminating internal states of an agent, as well as perceptive relative features such as the speed of another agent and other changes in the environment.

7.5 Pointing at the topic

In chapter three, where I explained the process of language games, I may not have made clear enough the importance of pointing. Nor have I done this in section 6.3.4. The process of pointing strongly influences the formation of a lexicon in the way that we are investigating. Especially, when the objects are not lexicalised yet. How can the speaker make clear to the hearer what it is 'speaking' about, when the hearer has no word for an object? For physical objects, the only way this can be made clear is by physically pointing at an object, or by using language in order to describe an object. This is the way we, human beings, normally do when we introduce a new (visible) meaning. There are, however, some philosophical objections to pointing [34], but I will not discuss them because they go beyond the purpose of this thesis. Our autonomous robots initially are not sufficiently lexicalised that they can use language to point out a new meaning. Therefore they need to point at an object physically. When objects are lexicalised, they need not to be pointed at anymore. Moreover, when the language becomes large enough, it may be possible to point

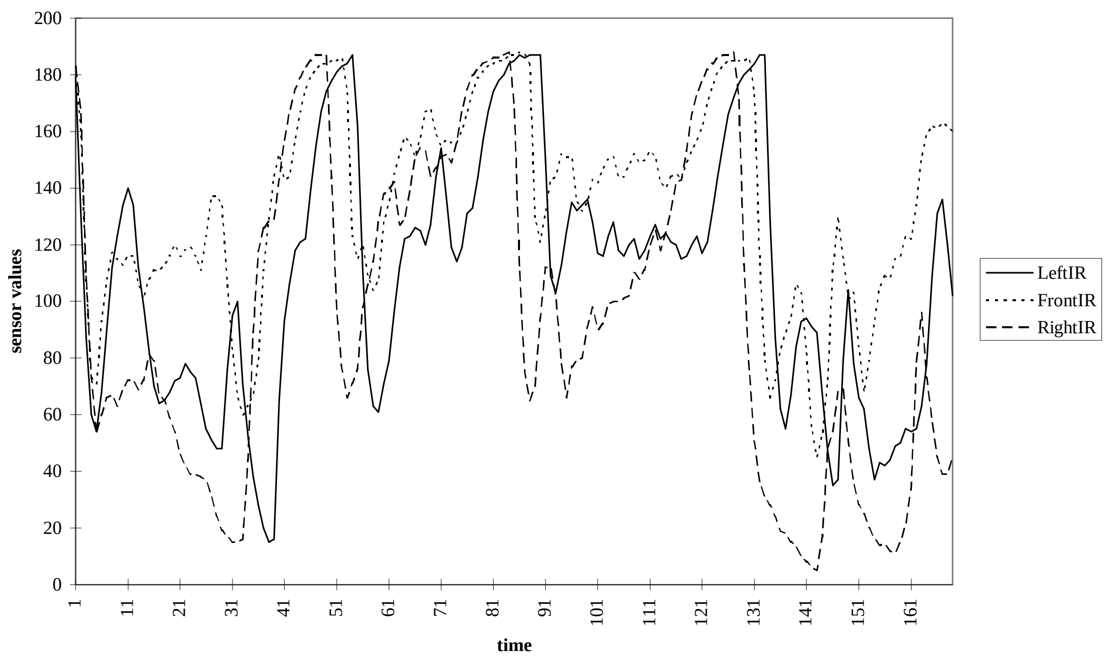


Figure 7.8 *The observation of the pointing process of a pointing robot that rotates 360 degrees. Note the many intersections of the left- and right sensor signals. Only the three highest intersections (approximately at 51,83 and 125) are the relevant ones.*

at meanings using language only. This may be necessary when we want to experiment in the future on language formation about internal states of a robot. In this section I will first discuss the pointing at objects that are no agents. In the end of this section I will discuss the pointing, when one of the participants of the language game is the topic.

Human beings have several ways of pointing at an object. We can use, for example, our fingers to point at an object or we simply look at an object. The hearer in a human conversation can determine the topic by watching the speaker point. For this we have a very sophisticated visual system. This, however, the robots do not have, nor do they have fingers or arms that they can extend in order to point at something. I needed to invent another way of pointing. I have thought about a lot of things, some of which I have already discussed in section 5.3.3. There I decided to mount IR emitters in four perpendicular directions, with its geometrical origin in the middle of the robot. Then a robot could, in principle, point at an object by simply orienting towards this object, analogue to the looking at an object that we humans use to point. Assuming that the speaker initially faces the hearer, the hearer can observe how many peaks of IR pass, and thus determine in which quadrant the topic must be. In our ecosystem, there are not so many objects, so the hearer can determine the topic with a relatively high probability of success.

Above, I mentioned that the process should work 'in principle'. I used the words 'in principle', because it took a long time between the idea and the implementation, and then it took even more time before it finally worked reasonably. What was the problem? Figure 7.8 shows the characteristics that the observing robot 'sees' when the pointing robot makes one complete circle, starting and finishing facing the observer. As can be seen, there are a lot of intersections (or peaks) between the left and the right sensor values. Note also that there are four intersections at more or less equal distances (at times 48, 81, 125 and at the end). At these places the sensor values are relatively high (near 180). These intersections are the relevant ones, i.e. they represent the passing of a beam of IR. The other intersections probably come from the interference of the adjacent beams of IR. Therefore, the hearer should only count the intersections that exceed a certain threshold. Now we end up with the problem, how to set this threshold. If we use a fixed threshold, then the problem arises that this threshold may be too high if the robots are not standing close enough in the conversation. On the other hand, if we lower this threshold, then, if the robots are standing too close to one another, it may well be possible that other irrelevant intersections will be counted as being relevant. So, this threshold should depend on the situation in a certain language game. As can be seen in the figure, the relevant intersections occur when the value of the font IR sensor exceeds 90% of the highest IR value that is sensed. So, before the pointing starts, the hearer determines the height of the front IR sensor when the speaker is facing the hearer. The threshold, then, is set at 90% of this value. This way the irrelevant intersections are not counted. The number of counted intersections when the speaker starts to 'speak' is represented as the quadrant of the speaker.

The next problem to overcome is the mapping of the quadrant that is pointed at from the point of view of the speaker, into the one from the point of view of the hearer. This mapping is not trivial, as is illustrated in figure 7.9. Note that the robots determine their own quadrants clockwise, as they both scanned their surroundings clockwise. The mapping is implemented as the following *look-up table*.

Quadrant speaker	Quadrant hearer
1	3
2	4
3	1
4	2

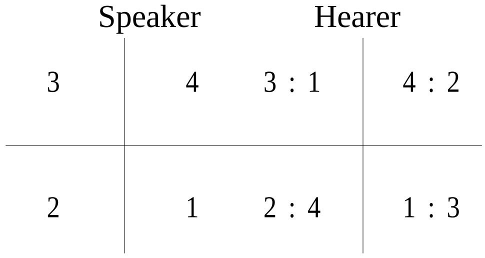


Figure 7.9 *The transformation of the quadrants for opposite agents.*

When the hearer determined the quadrant that the speaker pointed at, it still must determine the topic. There are three possibilities: (1) There is exactly one object in the domain of this quadrant. Then there is no problem, this object is the topic. (2) There is more than one object in this domain. In this case, the agent will choose the first object it encounters as the topic. This may cause an uncertainty in the language game, but the pressures that are on the selectionistic method of language games will converge the lexicon to cohere. And, (3) there is no object in the quadrant. In this case we could do two things: either we let the hearer determine the topic as being the object that is closest to the specified quadrant, or we could decide to let the language game be classified as a failure. I decided to implement the first option, because of the same argument that made the decision in (2). Furthermore, this method of determining the topic, i.e. looking for the object that is closest to the quadrant, probably yields a better than chance result.

Until now, I discussed the pointing at objects, but we also want the robots to be a possible topic of a language game. I implemented the pointing at the agents as follows: (1) If the hearer is the topic, then the speaker only emits IR before it starts to converse. The speaker is already oriented towards the hearer, so it does not need to orient again. (2) If the speaker is the topic, then it will not emit IR, but it starts the transmission of the expression right away. So, if the hearer 'hears' the speaker 'speak', then the hearer is the topic if there are no intersections counted and it senses IR. The speaker is the topic if no IR is sensed at all. Now that both robots determined the topic, they can really start the conversation. Issues on the implementation of this process will be discussed in the next section.

7.6 The conversation

In the simulations that have been carried out (e.g.[41][44][45]), a language game was played by two agents that lived in one computer. In these simulations it was assumed that both agents participating in a language game had determined the same context and topic. It was also easy to transfer from a 'speaker's task' in the communication to a 'hearer's task', because these transitions were carried out by a global program. Now the agents reside in different computational devices, thus making the same system, described in the simulations, a parallel program. Here there is no global program, nor does the assumption hold that both agents will determine the same context and topic. This assumption, however, is being made in this experiment as well! Although the assumption will not hold in every language game, it is assumed that it will hold in a sufficient amount of language games, in order to form a coherent lexicon. The mechanisms that are implemented for the conversation were already described in detail in chapter three and will not be discussed here.

The implementation of the language formation initially seemed to be no problem, though I forgot to implement it for multiple word expressions. I first implemented the language games in a simulation, where I assumed only one-word expressions. Then I copied this simulation onto the robots with the necessary adjustments, and this seemed to work. The process of decoding and encoding expressions, though, is much more complicated for multiple word expressions (without a particular word order) than for one-word expressions. The parallelism of the system and the ever-increasing complexity of the source-code made it worse. And because the robots do not have any viability mechanisms implemented, the robots will 'die' after half an hour on the average. A robot can also fall out because of some technical problems.

As mentioned before, the formation of a coherent lexicon may take a thousand or more language games. In order not to loose the already formed feature sets, the lexicon and other vital information to language formation, one must record this data regularly. So, one could initiate the robots with this data and start a new run of experiments without any loss of information. I first tried to use the radio-link to transmit this data to a monitor program, but due to noise and other unsolved problems this did not work properly. Therefore I decided to connect a serial cable to every robot, thus recording the language games and system that evolved. Although this made the experiments more difficult to carry out, it also had the advantage that language games are recorded more precise, because the radio-link is unreliable.

In the following sections I shall not discuss all the boring details of the problems I encountered; I will only discuss some important decisions I had to make. I first will discuss the speaker part of the implementation. And in subsection 7.6.2 I will discuss the hearer's process. In 7.6.3 some concluding remarks will be stated and some more general problems will be discussed.

7.6.1 The speaker's task

As mentioned in chapter three, the speaker has to encode a given distinctive feature set into an expression, which represents the topic of the language game. The first question I encountered was what representation to use. Although this was a general problem, I will discuss it in this section. The second question was how and when a new word-meaning pair association is created. The third question was, if there is more than one word that represents a feature of the distinctive feature set, how to choose the best word. The final question was how to implement the formation of multiple word expressions.

The representation I chose was in fact a rather simple one. All items are stored in an array, of which the identity number (ID) refers to the lexicon entry of a word-meaning pair. A word is represented by an arbitrary string of two characters from the alphabet, separated by a blank space. In the interface it is represented as, for example, *(a b)*. The meaning is represented by an integer value that points to the entry of the feature detector that represents that meaning. I think this kind of representation may be found in the brain as well. I.e. there may be a place in the brain where you can find a representation of a word, which is coded as (or pointing towards) a certain set of characters from the alphabet. This representation (like neurones) than have a connection with a certain meaning, i.e. the connection is made from the neuronal lexical entries, to the neuronal representation of the feature detector. The lexicon also represents the use and success scores. The neuronal plausibility of these representations will be further discussed in chapter nine.

The representation of features is different from the one used by Steels [44]. He associates lexical entries with feature sets that contain features, which represent the sensor values. I, on the other hand, make a short cut here; I directly associate the words with feature detectors. As pointed out earlier, the distinction between feature detectors and features is not very clear-cut.

Although it may be argued that feature detectors (or features) need to be classified further, I think this classification is only necessary for level formation. Level formation is a classification, based on the (syntactic and semantic) functionality or structure of a word-meaning pair [47]. Classification of the meaning that will be lexicalised in my experiments is done by the lexicalisation itself. I.e. the words for certain feature detectors cause the lexicon to be a higher level in the hierarchy of meaning. Due to ambiguity, different sets of feature detectors may arise that are associated with one word, but that also refer to the same meaning, thus leading to coherence, as was argued in chapter four. This, however, may not always be true, and so lexical ambiguity enters the lexicon, which is a common feature of natural languages. Once again, co-evolution of meaning and language is a main principle of perceptual grounded language formation.

If the speaker has determined a distinctive feature set, it searches its lexicon for word-meaning pairs that match one of the features from this set. If the lexicon is already somewhat matured, there may arise synonymy, i.e. there are more words for one meaning. In this case, the most useful word needs to be selected. So the word that has the highest score of success/use will be used in the expression. When a word will be expressed, the use is incremented immediately. So, when the hearer replies the evaluation, only the success score has to be incremented if necessary. If there is no association matched with a feature, then the string *(nil)* is being expressed. A new word is created immediately, i.e. with a certain probability. In the simulations of Steels [41][44][45], this probability was set at 0.05. This probability is kept low in order to slow down the emergence of ambiguity.

The probability is implemented using a random generator that generates numbers between 0 and 20 and when this number equals 1, a new word is created. By this probability at 0.05 (or so low), it is assumed that the system does not crash or ‘dies out’ during the evolution. Our physical robots, on the other hand, do crash due to energy deficits or other technical problems. During some long experiments I noticed that it took very long before a second word was created. I reasoned that due to the crashes, the random-generator had to start all over again, making the probability of word creation even lower. For instance, when I experimented with a run of some thirty language games (which is quite common), a particular robot may have been the speaker for, say, fifteen times. It also may be possible it tried to create a word ten times, but the constraint of probability caused it not to. The same property can hold for several runs in which no words are being created. Summing up these runs, the probability may decrease to, for example 0.01 or even lower. Therefore I had to increase this probability. I intuitively estimated that a probability of 0.2 would suffice.

As mentioned before, the implementation of multiple word expressions caused some problems. These problems arose, however, mainly in the hearer’s task and will be discussed in the next subsection. The speaker encodes one by one the words that it will express from the distinctive feature set. At the time the words are encoded, and the best one has been chosen, this word is transmitted by the radio-link. This whole procedure is processed during one SMB-cycle, so all radio messages that are transmitted at the end of this cycle constitute the expression. At the same time as the message is formed, the use is incremented if and only if a word is expressed.

7.6.2 The hearer's task

The implementation of the hearer's part of the conversation was much harder. This was mainly due to the fact that the decoding process is much more complex than the encoding process. Especially, the evaluation of the successfulness of the language game appeared to be quite complex. The hearer decodes the expression into (usually more than one) feature sets, which it then has to compare with its distinctive feature set. Furthermore, the hearer has to look for that distinctive feature set, which resembles one of the decoded feature sets most. So, the hearer has to search what feature set has the highest correlation between the decoded set and the expected set. This yields a computationally large search tree. If the lexicon becomes really large this problem may be intractable. Different variables for the different feature sets and the different types of outcome did not make the task easier. In this section I will discuss this problem of the implementation in greater detail, starting with an overview of the task that had to be implemented. I do not address the process of updating the use/success scores, because this is already discussed in chapter three.

If the hearer gets an expression from the speaker, and the hearer already determined the topic, then it has to decode this expression. If the expression is empty, then the language game ends in failure, and the failure is reported to the speaker. The speaker had already determined that the game was a failure and possibly created a new word. If the expression is not empty, the hearer needs to search its lexicon in order to find all associations that match the words used in the expression. This way the hearer builds a set of features that represents the expression from the hearer's point of view. The yielding set, then, has to be compared with the set of distinctive feature sets. The hearer must search the distinctive feature set that most closely resembles the feature set that it had decoded. Three things can happen now: (1) If there is not any element of the decoded feature set in one of the distinctive feature sets, then the game ends in failure. (2) If these two sets are equal, then the game ends in success. (3) If the cross section between the decoded feature set and the set of distinctive feature sets is not empty, but there are not some features missing, then the game ends in a partly success.

The resulting behaviour of the first two cases was quite easy to implement. In case (1), if the set of distinctive feature sets is not too large, then the hearer had to associate the words with all the features in the distinctive feature sets. If the set of distinctive feature sets is too large, then too much uncertainty and ambiguity would enter the lexicon, so no associations are being made. (2) If the game ends in success, no associations have to be made. The last option (3), however, was much more difficult to implement. It has to be determined which distinctive feature set(s) most closely resembles the decoded set. Then all the words that do not have associations must be associated with all the distinctive features that do not have associations yet. Problems were: (a) how to find the distinctive feature set(s) that correlates best with the one that was decoded. And (b) to make sure that all new associations are only made once. Problem (b) has been resolved, but problem (a) still has some shortcomings. It should be the case that only one distinctive feature set will be used for association, whereas up to now, usually more than one distinctive feature sets is being associated.

As you can imagine implementing such a search was quite difficult and complex. I will, however, not discuss these problems and their implemented solutions further, because the algorithm has already been given in chapter three, and the implementation does not differ much from that algorithm.

7.6.3 Some general remarks

In implementing the conversation I encountered some more problems than were mentioned in the previous subsections. The most fundamental problem was when to do what? I.e. at what moment should a robot do what process. When, for example, should the hearer determine the topic? I have mentioned in the introduction of this section that it was assumed that this was already done, but in fact it is somewhat more complicated.

If the speaker encodes the expression, then the hearer still does not know what the topic is. In practice, the speaker first points at the topic, then it plays a discrimination game, and then it encodes the expression. Instead of transmitting a *synchronising broadcast* like e.g. ‘pointed’, the speaker broadcasts the fact that it finished pointing by expressing the encoded expression. So, only when the hearer receives this expression, the hearer can determine what the topic is. When the hearer has determined the topic, it plays a discrimination game and then it decodes the expression.

Another example is that when the speaker finishes pointing, it must stand still before it can play a discrimination game and encode the expression. The process of playing a discrimination game and decoding the expression may take some computational power of the processor, which in turn may increase the time-interval between two SMB-cycles. Thus the robot would turn too far. Here this may not be such a big problem, but when, for example, alignment is required, this may cause rather big problems.

7.7 General problems and the main program structure

Before I will present the final protocol, I will discuss some general problems that also arose during the implementation. In this section I will also discuss the main structure of the program I developed. The protocol will be given in section 7.8.

7.7.1 General problems

The hardest problems that arose during the implementation revealed itself during the implementation of the search for a robot before or in the initial stages of the language game. Initially I had a lot of technical problems with the IR module. At first I did not know that modulation differences had to be set in the PDL code. Then there arose the problem that when I turned all the IR emitters on, the system crashed. For some unresolved reason, it appeared that it is not always possible to turn on all the emitters (including the ports that were not in use) at once, especially when the other processes became more complex. So, I first changed it so that all emitters were turned on one by one, but in order to emit pulses of IR, the emitters needed to be switched on and off at the same time. So, I changed the program in such a way that now only the necessary emitters are turned on. Until now this did not cause any problem. Other users of the

SMB2 board have observed the problem before, and it probably has something to do with the limits of processing in the kernel.

A next delicate problem that I had to deal with throughout the implementation process was about the same that was mentioned in section 7.5.3. When do the robots have to do what? With this problem I do not mean the ‘big’ relevant tasks, but rather the more trivially looking ones, like doing more processing when the robot decided to stop, causes problems. In section 7.3.1 I already mentioned the problem that when a robot switches from emitting IR to sensing IR, it will take some time before the modulation differences are died out. I.e. if a robot emits IR, it has high sensor values for not sensing IR and vice-versa; the IR module needs some time to relax between a transition of emitting and sensing. In the FSA, one must carefully design such transitions in order to let the robots sense reliable information.

The radio-link also caused some problems. As was mentioned in chapter four, the radio-link transmits radio signals in an unreliable mode. This means that it is uncertain whether a transmission arrives at its destiny, but when it arrives, that the message is reliable. There is no reliable method to check the arrival of a radio-signal. When, for instance, the receiver confirms the arrival by transmitting a radio-signal, the problem starts again, and again, and again … So, I had to implement some method to get a robust system.

All other processes that are directly connected to the dynamically changing environment in one way or the other are unreliable. I.e. the processes that make use of sensory-motor interactions may fail anytime. A process that fails often, for example, is the search of the speaker for the hearer, causing both robots, in principle, to stay in the current state indefinitely. Therefore, I have built in a timer that causes the robot to leave a certain state after a particular time. This time length depends on the time in which a process can be carried out. The search for a robot, for example, takes more time than the scanning process. I have implemented a default time period in which most processes can be done. Only, the search process takes longer than this default period, and therefore I increased this period appropriately. If a process does not succeed within this time period, the language game fails. I thought about methods of how to let only the particular process fail, thus not needing to start all over again, but I did not find any solutions. Maybe transmitting a broadcast for failure and a new call for communication may help to keep the system playing language games, which brings me to another problem. How can we keep up as many language games as possible.

As was mentioned before, the system may need more than a thousand language games before the language converges to a coherent structure. As I mentioned in section 7.3.1, I already succeeded to increase the frequency of a language game to every two or three minutes. This result could be better. This can be done by playing another language game right after one has finished. So, every time a language game has finished, the hearer broadcasts a request for communication, which the former speaker may, or may not confirm. Confirmation depends only on the success of the broadcast. The hearer then becomes the speaker and the other way around. I first though it would be sufficient to re-enter the FSA again after they had scanned the surroundings. So, halfway the protocol. This, however, turned out to cause asynchrony in the implementation of the program, because not all constraints were satisfied (like, e.g. the alignment constraint initial to the pointing). So, I decided to start the process all over again. This way, the robots may play more than ten language games in a row, taking turns in the role of speaker and hearer. There may also be only one, due to the uncertainties of the system.

7.7.2 The main structure of the program

In constructing the several default processes of the program I encountered only a few problems. These concerned mainly the order in which the several procedures must be processed. Another problem concerned the conditions under which the robot can enter either the speaker or the hearer mode. In this section I will briefly discuss these problems.

The program is organised such, that it can simulate parallel processes. There are mainly processes for motor-behaviour, language adaptation, perceptual grounding, and communication control. Motor-behaviour is driven by motivations that are calculated by the default and communication processes. The basic behaviours are forward drive, stop, rotate, IR-taxis, pulse IR, smooth obstacle avoidance, and touch based obstacle avoidance. The motivations are first calculated at the 'top' default process. They are also calculated by the communication control, which is the finite state automaton for both the speaker and the hearer. The motivations calculated by the communication control inhibit the default behaviour. In addition, the communication behaviour is inhibited by the touch based obstacle avoidance, which also calculates behaviour. In my implementation, motivations are binary relations: zero when they are off, and one when they are on. When, for example, a forward driving robot bumps onto an object, then the motivation for touch based obstacle avoidance is activated. This motivation then is activated for about 2 seconds, so the robot is able to drive backwards and turns away from the object before. During this period, the forward drive is inhibited, i.e. turned off.

In order to inhibit default motivations, the inhibiting motivations must be calculated after the inhibited ones, because, although in theory the processes are calculated in parallel, in practice the processes are still calculated in serial. So, the main structure is to first calculate the default motivations, then the communicative motivations, which inhibit the default, and finally motivations for touch based obstacle avoidance.

In this chapter I have already discussed the communicative motivations in detail, and they will be repeated in the next section. The motivations for touch based obstacle avoidance, which are initiated by the bumper sensors, are discussed briefly above. So, now I will discuss the default behaviours briefly. There are, besides touch based obstacle avoidance, three default behaviours: forward drive, smooth obstacle avoidance and IR pulsing.

1. *Forward drive*: This is a dynamical behaviour, where the robot always minimises the difference between the actual motor speed and the default motor speed. This assures the robot to ride at default speed. The motivation during default processes is always 1.
2. *Smooth obstacle avoidance*. In this behaviour, the robot uses active infrared to avoid objects. It emits IR (in pulses), and when it receives reflected IR signals, the robot will turn away from the object. As was mentioned before, the IR characteristics are the reverse when IR is on, compared to when the IR is off. So, IR-taxis has the reverse effect, when the IR characteristics are reversed. I.e. instead of driving towards an IR source when the sensed IR values are increasing from a default low value, the robot will drive away from an IR source (e.g. an object that reflects IR) when the IR decreases from a default high value. (Check yourself with the formula given in section 6.3.2) The motivation for this behaviour is set to 1 in default mode. Side effects are

that when the robot receives IR from another robot it will already turn towards it, and that it does not work in periods where no IR is emitted.

3. *Touch based obstacle avoidance*. If the robot, despite the smooth based obstacle avoidance, bumps into an object, the touch based obstacle avoidance behaviour is activated. There are basically three possibilities: (1) If the robot bumps into an object on the front and only one bumper (either left or right) is touched, the robot inhibits all other motor behaviours and starts to drive in reverse for one second. Then it turns away from the wall for ½ second, after which the behaviour dies out and all other behaviours are restored. (2) If both front bumpers are being pushed, the same behaviour as (1) is started, except that the robot will turn away in a random direction. This is implemented in order to resolve the possibility that the robot gets stuck in a corner. And (3) if the robot bumps into an object from behind, it only gets a small forward impulse, thus driving forward again.
4. *Pulse emission*. When the robot is in default mode, the robot emits pulses of IR (see figure 7.5). This is done at different frequencies in different robots. It is calculated that when IR must be on, the motivation will be set to 1, else it is set to 0. When it is 1, the IR is set to maximum only once, and vice versa, the same happens with the modulation level.

The execution of the motor-behaviours has to be ordered as well. The stop command, for example, must inhibit all other behaviour. The IR-taxis works best if it is executed before forward drive. Rotation can only be executed when all other behaviours, except stop motors, are inhibited. This inhibition is assigned to the motivational variables when rotation is activated. As mentioned, the touch based obstacle avoidance must inhibit all other motor behaviours in order to drive backwards long enough and turn away from the obstacle. Otherwise, the forward drive behaviour causes the robot to repeatedly bump into the obstacle.

In between the default mode and the execution processes is the *communication control*. Hereto I designed a FSA for both the speaker as the hearer. The states of the FSA are called motivations as well. At default, both motivations are set to 0, so the default mode as described is executed. The default process determines when an agent should transfer into a particular communication mode. If the agent has broadcast a request for communication, which is replied by the other agent, then the motivation to go into the speaker mode is set to 1. And if the agent receives a request for communication, it 'confirms' the other agent and will enter the hearer mode at level 1. The communication controls are processed after the default motivations. If the robots have a motivation for either the speaker or hearer mode, the FSA will process the commands in the particular state of the automaton. When the system arrived at a certain condition, the agent will proceed in the next state of the automaton. The FSA will be summarised in the protocol that is given in the next section.

Although all processes are carried out in serial, they are virtually implemented in parallel. Motivations of a particular behaviour can inhibit other motivations of the same behaviour according to certain *strength* of the particular motivation. This way we have a parallel working system. Schematically the architecture of the software is summarised in figure 9.1.

7.8 The final protocol for playing language games

Until now I did not discuss all the details of the implementation process, because that would cause a chapter of maybe more than a hundred pages. I did, however, address the most important steps that needed to be taken in order to get a smooth working parallel system. We started in section 7.2 with an initial base of the protocol about what needed to be done in a language game. In the following sections this protocol was more formally refined, so we now are able to present the resulting protocol. Schematically the protocol now looks like this:

PROTOCOL FOR A LANGUAGE GAME

All robots drive around and emit IR pulses each at a different frequency. When one robot ‘sees’ another, it enters the speaker mode.

SPEAKER:

- Broadcasts ‘communicate’ request.
- On confirmation turn IR off and approach hearer by IR-taxis and -orientation.
- When aligned, broadcast ‘aligned’. IR on.
-
-
- When receive ‘aligned’, IR off and scan surroundings.
- When finished scanning, broadcast ‘scanned’. Turn IR on.
-
- When receive ‘scanned’, choose a topic from scanned objects.
- Point the topic with IR on by orienting towards the topic.
- Discrimination game.
- Encoding of the best distinctive feature.
- Transmit the derived expression.
-
- Update type of success
- Finished.

HEARER:

- Confirms and enters hearer mode. Stops driving and turns IR on.
-
- When hearer receives ‘aligned’, start IR-orientation to speaker. IR is off.
- When aligned, broadcast ‘aligned’ and turn IR on.
-
-
- Turn IR off and scan surroundings.
- When finished scanning, broadcast ‘scanned’.
-
- Determine quadrant and thus topic.
-
-
- Discrimination game.
- Decode expression, evaluate success and transmit result.
- Finished, ask for new conversation.

These protocols both work in parallel at the same time. Radiobroadcasts will secure the system is synchronous. In order to be visible for the other robot and to be able to perceive, turns are being taken in emitting IR. We can see that the basic scenario as was given in section 7.2 is being followed. To increases the amount of language games, the hearer will ask for a new conversation at the end of each language game.

In this chapter I discussed the implementation of the experiments, which took me approximately four and a half months. The implementation of processes which made basically use of the IR module, i.e. all processes until the pointing is completed, were most difficult and took me about three months altogether.

8. EXPERIMENTAL RESULTS

8.1 The two experiments

In the course of the project I have done several experiments on the formation of a lexicon. In this chapter I will report on the results of two of those experiments. As was mentioned before, the experiments only concerned two robots. The robots did not play language games on spatial relations as was intended in the first place, they only played naming games. Word order was not important in these experiments. In both experiments, the protocol given in the preceding chapter was executed in every language game. The first experiment, however, did not concern the perceptual grounding, and the only objects that could be lexicalised were the robots themselves.

So, in the first experiment the robots engaged in a language game. They had to find each other so that they were facing each other. Then both robots scanned their surroundings, but did not play discrimination games. This was to have an indication of how the robots performed in executing the protocol. The speaker chose the topic, which was either the speaker self or the hearer. The distinctive feature sets were given by {*self*} or {*other*}, each from the robot's own point of view. Pointing was done as described in the preceding chapter, i.e. emission of IR when the hearer is the topic and no emission when the speaker was the topic. The language formation was done as described before.

In the second experiment the possible context did involve the other objects in the environment. The robots did play discrimination games as described in chapter 6. The objects were only indicated by name, i.e. only naming games were held. In the next section I will report on the first experiment. Section 8.3 reports on the second experiment. Conclusions are given in section 8.4.

8.2 The naming of agents

I have conducted a series of experiments where the robots were playing language (or naming) games, concerning only the robots themselves. These experiments were held to show that the language formation worked on the robots. Although the lexicon is not perceptual grounded, the results are interesting. It first showed that the system was robust and that the games went on continuously. However, not every language game succeeded to execute the given protocol completely, i.e. complete alignment was not always reached and the robots did not always scan their surroundings properly. The system was lexicalised in a small amount of naming games. In this section I will briefly discuss the results this experiment yielded.

Topic identification:
The topic of the naming games was derived with a coherence of 100%. This was due to the fact that there were only two possible objects in the environment, which were pointed in a simple way. If the hearer sensed IR, it had itself as the topic. Otherwise, the speaker was the topic. As we shall see, this caused no ambiguity in the lexicon.

Examples of the naming games:
After 10 dialogues I recorded this dialogue with robot r1 as speaker, and r2 the hearer:

```
Dialogue 10:
Topic r1: self
Topic r2: other
Encoded expression r1: (a b)
Decoded expression r2: nil
Failure
Association r2-other == (a b) 0/0
```

As we can see, r1 already has created a word for *self*, it was the first time this expression was encoded by r1. r2 successfully determined the topic, and it associated appropriately the expression with *other*.
Another example:

```
Dialogue 14:
Topic r1: self
Topic r2: other
Encoded expression r1: (a b)
Decoded expression r2: other
Success
```

We see the same dialogue structure, r1 is speaker and topic, r2 is the hearer. This time the dialogue is a success.

The final lexicon
After 37 dialogues the lexicons of both robots are evolved as follows:

```
The lexicon of r1:
r1-self == (a b) 16/15
r1-other == (a c) 3/3

The lexicon of r2:
r2-other == (a b) 15/15
r2-self  == (a c) 4/3
```

As we see, the lexicons converged into complete coherence without ambiguity of synonymy. This is mainly due to low complexity of the experiment (only two agents and objects) and to the complete coherence of the topic identification. Ambiguity, however, could have entered the lexicon if r2 had created word (a b) for *self*, before it associated (a b) with *other*.

We can also see that, for the same reasons just mentioned, the use/success rate is extremely high. The robot that associated a word secondly has a 100% score, while we can see that the robot that created the word has a use factor one point higher than the success. Other experiments showed the same result. This would cause the use/success rate to converge to 100%.

Other comments
We saw that in 37 dialogues both robots successfully adapted a name for both agents. These dialogues were held in less than 25 minutes. In this time there were, however, more onsets for a language game. These games already failed in the preparation stage of the language game. Failure occurred in all sub-tasks. Most of the times these were inappropriate calls for communication (more than 50% of the failures). Inappropriate calls for communication are made when the robots are separated significantly at the time of initiation. These inappropriate calls were probably due to reflected IR signals that are perceived at the end of a 'non-emission' period. In these cases the radio

broadcasts usually do not arrive at their destiny, so the initiator soon discovers its error, because no confirmation arrives. The initiator waits only 1.5 seconds for confirmation, and when it does not receive a confirmation, then it broadcast ‘wall’ to the other agent in case the initiator mistakenly sensed the wall instead of a robot. This way the robot that already entered the hearer mode can continue exploring. Initial alignment and the perception tasks also failed during this experiment. This only caused a language game to end, without influencing the language formation.

Although these experiments were set-up to have a rather simplistic formation of a language, we have seen that under these relative ideal circumstances adaptation of the lexicon in robotic agents occurs quite rapidly. The circumstances are relative ideal, because there are only two objects, the topic identification is very reliable, and there is no perceptual grounding. In the next section we shall see that increasing the complexity of the experiment, and thus decreasing the reliability of some sub-tasks, makes lexicon formation much harder.

8.3 Grounding experiment

This section reports on the results of an experiment on grounding an adaptive language. The experiment was one of a series of experiments that all partially failed in one way or another. The reported one was the best one of the series. I will report on different parts of this experiment, each with their specific results. Future experiments should combine the best results, thus leading to better results. The experiment was carried out in one and a half day, which lead to two and a half hour of actual robotic action. Results of this experiment are also reported in [51].

This small amount of time was due to: (1) Technical problems, where robots stopped working. (2) Recharging of robots (after half an hour work, the batteries needed to be recharged for half an hour). And (3) the way I measured the language games: I measured the language games using two serial cables to connect the robots to my PC, thus recording the internal states of a robot during the language games. This took extra time, because when the robots are exploring the environment, the serial cables had to be disconnected. In the effective experimental time I measured 201 language games and ca. 140 discrimination games. This difference is due to the fact that the agents do not play discrimination games when they have themselves as the topic. Unfortunately, only the first 55 language games were reliable.

The experiment is divided in the first part, where the grounding, and thus the language formation was most reliable, and a second part, in which the pointing was more reliable. About the first part I will report on the results of these language games, which are already very interesting. Due to problems with the pointing process in this part, the topic identification was very unreliable. Reliability of the topic identification was best in the last part of the experiment. So, there I will report on the coherence in topic identification. In this part, however, the meaning creation was unreliable because of an error that entered the source code, so the language formation was unreliable as well. Furthermore, I will also report on the coherence in derived context over this part. Finally, I will report on some general results.

8.3.1 Language formation and grounding

Here I will report on the first 55 successfully executed language games, i.e. those language games that resulted in a transfer of communicative information (words and non-words). During these language games, the robots were able to create the basic perceptual feature tree that is used in the language. Words are also coherently lexicalised for the agents themselves. No words or features are forgotten yet. The topic identification for the agents succeeded at approximately 31%, while for other objects this was approximately 2%. Here is an example of the grounding process in agent r2, starting with discrimination game 1. (Note that the sensory channels are numbered as defined in section 7.4.2).

```
Discrimination game 1 by r2
Objects r2:
      o0: [25] [sc0:11,sc1:3,sc2:46]
      o1: [80] [sc0:11,sc1:1,sc2:3]
      o2: [162] [sc0:2,sc1:7,sc2:188]
Topic r2: o2
Failure r2. No feature sets
New feature detectors r2: r2-sc1 [0,255]
```

As we see, robot r2 perceived 3 objects in its surroundings, o0, o1, o2. The topic of the discrimination game is o2. Object o2 is detected at time/position 162 with values 2 for sc0, 7 for sc1 and 188 for sc2. The discrimination game is a failure, because there are no feature detectors yet, but it leads to the construction of a new feature detector r2-sc1. r2-sc1 is a feature detector on sensory channel 1, i.e. modulated light, and it expects a value between 0 and 255, which is the domain of this sensory channel.

The next discrimination game is somewhat further in time, all sensory channels are associated with a feature detector.

```
Discrimination game 4 by r2
Objects r2:
      o0: [13] [sc0:70,sc1:7,sc2:136]
      o1: [38] [sc0:8,sc1:1,sc2:3]
      o2: [79] [sc0:5,sc1:23,sc2:12]
      o3: [215] [sc0:1,sc1:1,sc2:183]
Topic r2: o3
Feature sets r2:
      o0: {r2-sc0,r2-sc1,r2-sc2}
      o1: {r2-sc0,r2-sc1,r2-sc2}
      o2: {r2-sc0,r2-sc1,r2-sc2}
      o3: {r2-sc0,r2-sc1,r2-sc2}
Failure r2. No distinctive feature sets
New feature detectors r2: r2: sc1-0 [0,127.5]
      r2-sc1-1 [127.5,255]
```

Now r2 has feature sets for each object, but because all feature detectors work in the same domain for every sensor, no distinction can be made. sc1 is divided in two equal regions, thus constructing new nodes of the discrimination tree.

Already at discrimination game 10 the first success is recorded. The discrimination game involves two objects:

```
Discrimination game 10 by r2
Objects r2:
      o0: [225] [sc0:32,sc1:7,sc2:2]
      o1: [256] [sc0:1,sc1:1,sc2:151]
Topic r2: o1.
Feature sets r2:
      o0: { {r2-sc0,r2-sc1,r2-sc2},
            {r2-sc0,r2-sc1-0,r2-sc2},
            {r2-sc0,r2-sc1,r2-sc2-0},
            {r2-sc0,r2-sc1-0,r2-sc2-0} }
      o1: { {r2-sc0,r2-sc1,r2-sc2},
            {r2-sc0,r2-sc1-0,r2-sc2},
            {r2-sc0,r2-sc1,r2-sc2-1},
            {r2-sc0,r2-sc1-0,r2-sc2-1} }
Distinctive feature sets:
      { {r2-sc2-1},{r2-sc0,r2-sc2-1},
        {r2-sc1,sc2-1},{r2-sc1-0,r2-sc2-1},
        {r2-sc0,r2-sc1,sc2-1},
        {r2-sc0,r2-sc1-0,r2-sc2-1} }
Success r2. Best distinctive feature set:
      {r2-sc2-1}
```

As we see, there are already many combinations of feature sets for both objects. All combinations of such features that distinguish the topic o0 from o1 constitute the set of distinctive feature sets. There are six such feature sets; the best distinctive feature set is r2-sc2-1, because this is the smallest set. We see that already in an early stage the robot is able to successfully discriminate objects. r2 increases the use parameter of all the features that are in the set of distinctive feature sets, but it only increments the success of r2-sc2-1. From this point on, the language formation can evolve. Before this was not possible, because there were no distinctive feature sets to lexicalise.

Now that the feature detectors are refined further, it is interesting to see how the lexicon evolves in co-evolution with meaning. Let us look at such an early naming game.

```
Dialogue 19
Speaker r1. Hearer r2.
Objects r1:
      o0: [17] [sc0:3,sc1:0,sc2:193]
      o1: [91] [sc0:3,sc1:62,sc2:3]
      o2: [192] [sc0:3,sc1:0,sc2:191]
Topic r1: self
Distinctive feature sets r1:
      { {r1-self} }
Objects r2:
      o0: [243] [sc0:2,sc1:0,sc2:186]
Topic r2: o0
Distinctive feature sets r2:
      { {r2-sc0,r2-sc2},
        {r2-sc0},{r2-sc2},
        {r2-sc0-0},{r2-sc2-1},{r2-sc2-1-0} }
Encoded expression r1: (a b)
Decoded expression r2: ∅
Failure. r2 associates (a b) with {r2-sc0},{r2-sc2},
            {r2-sc0-0},{r2-sc2-1},{r2-sc2-1-0}
```

Here we see a typical example of incoherence in the perceived context. r1 perceived three objects. The first and the last are the same object, because the decision for recording both objects fell on the IR channel, implying the object to be a robot, but there is only one robot. This happened more often. In the beginning of the perception, a robot senses an interval in infrared that is caused by the opposite robot. When this happens before time/position 10, then it is assumed that it is the other robot, otherwise this may not be true. The same phenomenon may occur when both robots communicate near a wall or an object. Then it is well possible that reflected IR is perceived as an object. Furthermore, r1 perceived a competitor (o1), while r2 did not. This may be caused by r1, which might have been in between r2 and the competitor. r2 only perceived o0, which represents r1.

Although this incoherence in context appeared by coincidence, it had little influence on the language formation. r1 chooses itself as the topic, and r2 was able to identify r1 as the topic. As we can see, r1 already created a word for *self*, namely (a b). In turn, r2 could not decode the expression in a set of features and the language game ended in failure. Therefore it associated (a b) with all the distinctive feature sets it derived from o0. What we do not see is that r1 increments the use for (a b), but not the success. It increments both the use and success of distinctive feature r1-self. r2 increments only the use of all distinctive feature sets, because no set is used for communication.

After a while, the language gets somewhat matured, and we encounter a dialogue as the following:

```
This is dialogue nr 43
Speaker: r2. Hearer: r1.
Objects r2:
      o0: [138] [sc0:2,sc1:0,sc-2:183]
Topic r2: self
Distinctive feature sets r2:
      {{r2-self}}
Objects r1:
      o0: [12] [sc0:1,sc1:0,sc2:186]
      o1: [185] [sc0:2,sc1:12,sc2:188]
Topic r1: o0
Distinctive feature sets r1:
      { {r1-sc0,r1-sc2-1-0-1-1-1-0},
        {r1-sc0,r1-sc2-1-0-1-1-1-0-1},
        {r1-sc0-0,r1-sc2-1-0-1-1-1-0},
        {r1-sc0-0,r1-sc2-1-0-1-1-1-0-1},
        {r1-sc2-1-0-1-1-1-0},
        {r1-sc2-1-0-1-1-1-0-1}}
Encoded expression r2: (a b)
Decoded expression r1: {{r1-self},{r1-sc2-1},{r1-sc2-1-0},
                  {r1-sc2-1-0-1},{r1-sc0},
                  {r1-sc2-1-0-1-1-1-0},
                  {r1-sc2-1-0-1-1-1-0-1}}
Success
```

This time the dialogue was a success. Although, as we saw in the preceding dialogue, r1 also has the word (a b) for itself, it is able to decode the expression rightfully. r1 now increments both the use and success of (a b) associated with feature r1-sc2-1-0-1-1-1. This feature is chosen to participate in the conversation, because it resides in both the decoded feature set and the distinctive feature set of one element, *and* it is the feature that is least refined in the discrimination tree. The success for this feature in the discrimination tree is incremented as well, whereas the use is incremented for all the features in the set of distinctive feature sets. Note the distinction between the use/success between the lexicon and the feature detectors.

After 55 language games, the lexicon of both robots looks like this: (we see the features associated with a word, and their use/success)

```
The lexicon of r1:
r1-self == (a b) 10/1
r1-sc2-1 == (a b) 1/1
r1-sc2-1-0 == (a b) 3/1
r1-sc2-1-0-1 == (a b) 0/0
r1-sc2-1-0-1-1-1-0 == (a b) 1/1
r1-sc2-1-0-1-1-1-0-1 == (a b) 0/0
r1-sc0 == (a b) 0/0
r1-self == (a c) 0/0
r1-sc1-0-0-0 == (a d) 0/0

The lexicon of r2
r2-self == (a b) 14/3
r2-sc2 == (a b) 2/0
r2-sc2-1 == (a b) 2/0
r2-sc1-0 == (a b) 4/0
r2-sc2-1-1 == (a b) 0/0
r2-sc1-1-0 == (a b) 2/1
r2-sc2-1-0-0 == (a c) 1/0
```

As we can see, both robots have associated (a b) with *self*, but it appears that r2 starts to win (a b) with *self*. Both robots also associated (a b) with features that can represent the other robot. Furthermore, we can see that r2 successfully associated (a b) with a feature for high values of the sensory channel for modulated light. This must have happened in a dialogue where the topic was wrongly identified by one of the robots. Because in a language game, both robots assume they identified the same topic, the expression may be associated wrongly. r2 somewhere associated (a b) with sc1-1-0, and in another game it decoded the expression (a b) of r1 with sc1-1-0, thus leading the language game to a success. We can also see that r1 associated (a c) with *self*, while r2 coupled this word with a feature for r1. Finally, r1 uses (a d) for a feature that represents a competitor.

The discrimination tree that is built in this period by r1 now looks like this: (note that this time the score are presented as success/use)

```
Features and success/use r1
r1-sc0 [0,255] 0/46
      r1-sc0-0 [0,127.5] 0/30
            r1-sc0-0-0 [0,63.75] 0/15
            r1-sc0-0-1 [63.75,127.5] 0/0
      r1-sc0-1 [127.5,255] 0/0

r1-sc1 [0,255] 0/29
      r1-sc1-0 [0,127.5] 4/45
            r1-sc1-0-0 [0,63.75] 2/42
            r1-sc1-0-1 [63.75,127.5] 0/3
                  r1-sc1-0-0-0 [0,31.875] 3/48
                  r1-sc1-0-0-1 [31.875,63.75] 0/0
                        r1-sc1-0-0-0-0 [0,15.937500] 0/34
                        r1-sc1-0-0-0-1 [15.9375,31.875] 0/4
                              r1-sc1-0-0-0-0-0 [0,7.96875] 0/16
                              r1-sc1-0-0-0-0-1 [7.96875,15.9375] 0/0
      r1-sc1-1 [127.5,255] 0/0

r1-sc2 [0,255] 1/84
   r1-sc2-0 [0,127.5] 2/21
   r1-sc2-1 [127.5,255] 2/26
      r1-sc2-1-0 [127.5,191.25] 3/26
      r1-sc2-1-1 [191.25,255] 0/0
         r1-sc2-1-0-0 [127.5,159.375] 0/4
         r1-sc2-1-0-1 [159.375,191.25] 0/26
            r1-sc2-1-0-1-0 [159.375,175.3125] 0/0
            r1-sc2-1-0-1-1 [175.3125,191.25] 0/26
               r1-sc2-1-0-1-1-0 [175.3125,183.28125] 0/0
               r1-sc2-1-0-1-1-1 [183.28125,191.25] 0/8
                  r1-sc2-1-0-1-1-1-0 [183.28125,187.265625] 2/13
                  r1-sc2-1-0-1-1-1-1 [187.265625,191.25] 0/0
                    r1-sc2-1-0-1-1-1-0-0 [183.281250,185.273438] 0/0
                    r1-sc2-1-0-1-1-1-0-1 [185.273438,187.265625] 0/13
```

As can be seen, the robot has built up a tree of feature detectors. A lot of them are not used at all, and also a lot was not successful. These feature detectors will be forgotten after a while. The evolution of successful features can be seen in figure 8.1

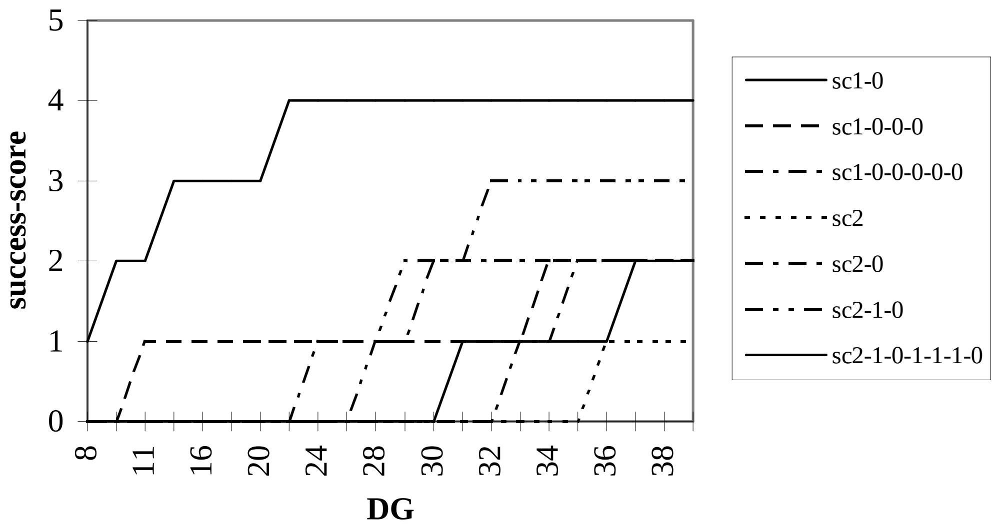


Figure 8.1 *This is a plot of the successfulness of certain features of r1 over a period of 42 discrimination games. The features that are relevant become more stable.*

The feature space covers the objects in the environment rather coherently, i.e. the features that are used successfully usually represent the same object every time they are used (table 8.1).

Table 8.1 *The coherence between the features and the objects in their environment for robot r1 after 42 discrimination games.*

Charging station	Competitor	Robot
sc2-0: 1	sc1-0 : 3 sc1-0-0-0: 3 sc2-0: 1 sc1-0-0-0-0-0: 2	sc1-0: 1 sc2-1-0: 3 sc2-1-0-1-1-1-0: 2

We see that the charging station had not been the topic often, so it is difficult to say something about this object. The other objects, however, all use specific features, although we can see some overlap. sc1-0, for example has been used three times for delineating the competitor, but only once for the robot (i.e. r2). All other features are very specific.

8.3.2 Topic identification and context coherence

As mentioned before, the first part of this experiment revealed reliable data on the meaning and lexicon formation. The lexicon formation was essentially reliable on the lexicalisation of agents. The last 90 language games, on the other hand, revealed more reliable data on the topic identification, which, of course, is very important for the lexicon formation. In this section I shall report on this topic identification and the coherence in the context between both robots. The latter does not always succeed, as we already saw in the preceding section, but if we look at the sufficient conditions for context coherence, we see that this figure is rather high.

As I already reported in chapter 7, topic identification by the hearer is rather difficult. The speaker by means of orientation towards the topic points at the topic. Meanwhile the hearer perceives a graph like in figure 7.4. Due to the errors in alignment, i.e. the robots do not start pointing while completely aligned, the height of the threshold is not determined well enough. It may also be possible that some of the relevant peaks in the graph are not high enough to exceed the threshold. In the last runs of the language games, the threshold was determined somewhat better, so the result, although still not astonishing, increased.

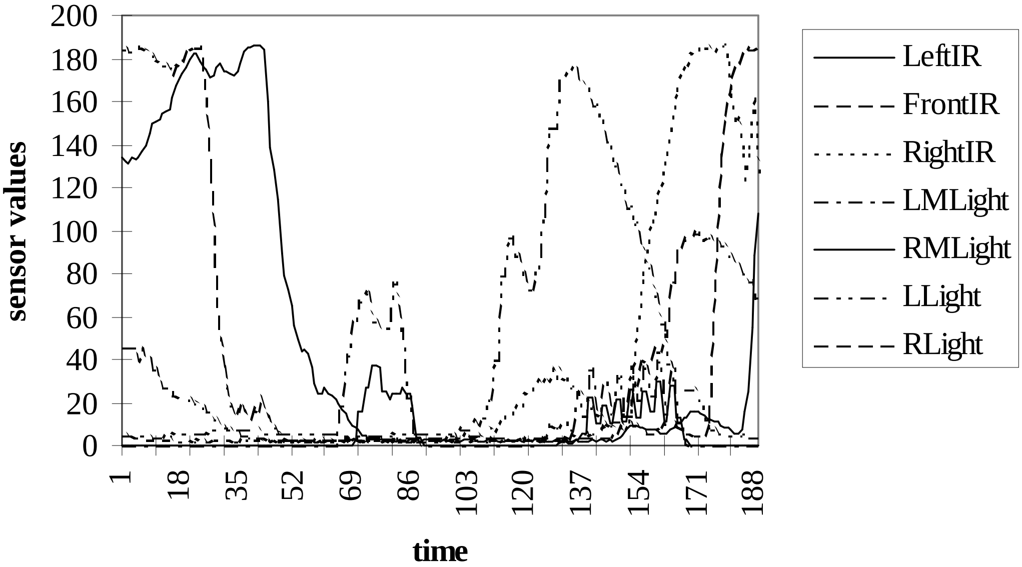
As we saw in section 8.2, the coherence in the topic, with one of the agents as the topic, in naming games for agents only was 100%. In this experiment this yield was decreased drastically to 31%. This probably was due to the fact that the hearer may have perceived some intersections, while the other robot was standing still. At least, because there are many more objects that could be the topic, the uncertainty increases. Recognition of the other objects as being the topic was 19%. Reasons why these results are this low, still have to be found. Research on that is currently been carried out.

As we can see in figure 8.2, both robots engaged in a language game perceive their surroundings very differently. Though, in the last 90 language games the coherence in the perception of the context was very high. The coherence of the context is defined as $m_{tot}(C_s \subseteq C_h)$, i.e. a measure of the rate in which the context of the speaker is the same as the context of the hearer. This coherence is sufficient to have a successful language game, because the hearer perceived all the objects the speaker can choose as a context. It was calculated as follows:

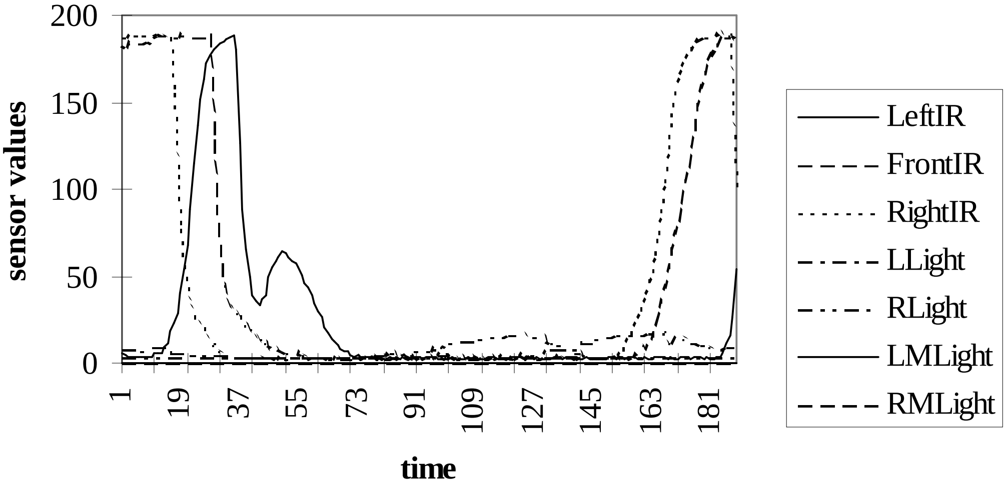
$$m_{tot}(C_s \subseteq C_h) = \frac{1}{N}\sum_N m(C_s \subseteq C_h)$$

$$where: m(C_s \subseteq C_h) = \begin{cases} 1 & if\, C_s \subseteq C_h \\ \dfrac{n(C_s \subseteq C_h)}{n(C_h)} & otherwise \end{cases}$$

Here C_s is the context of the speaker, C_h the context of the hearer and $n(X)$ is the number of objects that satisfy the particular condition. N is the number of language games being played. The sum is taken over all language games. Following this equation, $m_{tot}(C_s \subseteq C_h) = 0.83$. This is a pretty good result, considering the different viewpoints of the robots.



A



B

Figure 8.2 *The perception of two robots engaged in a language game. It is clear that both robots perceive different sensor characteristics. In figure A we see, for example, clear peaks of modulated light, whereas we do not see this in figure B. But, if the robot of figure B is the speaker, then it may have a sufficient perception to succeed in a language game.*

8.3.3 Some general remarks

Finally, I would like to discuss some results of the experiment that were not discussed yet. In a period of 2 ½ hours of effective experimenting, 201 language games were held. This comes down to 1.34 language game per minute. This result is rather high, compared to the amount of language games I initially hoped for: 0.5 per minute. When the robots start a language game, they may play a series of 10 language games in a row. This amount could be higher though. In between these language games (i.e. *within* a series of e.g. 10 language games), a lot of time is lost. The robots start their language game all over again, after another language game has ended. This may cost some time, because the robots initially appeared to be out of phase when a new language game should start. This problem also could be overcome, when synchronously start a new game making sure that they transfer to the state next after the perception. So the robots do not have to find each other again or scan their surroundings. This may save a lot of time, thus increasing the amount of language games. This is necessary because the robots may need more than thousand language games before the lexicon converges towards coherence.

In between the separate series of language games, the robots explore the ecosystem. When a robot receives IR signals that reflected from an object in the non-emitting period, the robot asks for communication. This happened in the experiment at least as much as there were language games. From the moment that a robot asks for communication, it will wait for 1½ second for reply. If the robot does not receive reply within that 1½ second, *or* when it does not perceive IR anymore (which means that the reflections came from an object), then it broadcasts a message that it sensed the ‘wall’ and starts to explore again. This property, however, is not a big problem. It takes only a second and sometimes contact has been made, with coincidentally successful communication.

If contact is made, the success-rate for proceeding the language game until the end was pretty high: 86.6%. The unsuccessful encounters are not included by the 201 language games. So, if the system were more stable in finishing a language game, the amount of language games would already have been ca. 235 in 2½ hour. If we can increase the amount of language games otherwise, for example by increasing the series of language games in one place, I think we can even double this amount. So, finally we may have a thousand language games in maybe 5 hours or so.

8.4 Conclusions

The overall results of the language games in the second experiment were a bit disappointing, in that the experiment revealed some small but significant errors in the system. The pointing for example did not work properly. Initially sometimes the discrimination games yielded too many distinctive feature set, which increased the search tree for the best set unnecessary. Although the results of the discrimination games were reliable, I adjusted the algorithm slightly so fewer distinctive feature sets revealed. This, however, brought up another problem, so the results of the remaining dialogues were not reliable anymore for the lexicon formation. The time schedule for this project started to run out, so these errors still need to be filtered out before new experiments can start. This is currently been done.

On the other hand, the results were successful in the sense that they do show that a lexicon may be formed using this method, and that future results must increase the coherence in the lexicon. We saw that coherence emerges between the objects and their relevant features. Coherence also started to emerge in the lexicon for agents. This was more obvious in the first experiment, although that experiment had no grounding. New experiments have to be carried out to increase the overall yield of the experiment.

The results of the discrimination games showed some promising results for the grounding problem. The agents are able to create discrimination trees themselves, and use selection for adaptive classification of features associated with objects. Thus building classes of features that discriminate can represent the meaning of objects.

9. COGNITIVE PLAUSIBILITY OF THE THEORY

9.1 Introduction

In this chapter the cognitive plausibility of the theory described in the preceding chapters will be discussed. The chapter will be divided in two parts. The first part discusses the architecture by which the robots are communicating. A link will be made with human architectures for language processing and other processes. The second part discusses the neurological plausibility of the selectionistic theory of language formation. The neurological plausibility is discussed by means of a theoretical research on the construction of a connectionistic framework capable of language formation according to our model. This section is a part of a paper that I have written, which will be submitted for publication in the future [55].

The cognitive architecture of language processing and -acquisition is rather well studied in both neurolinguistics and psychology. In the next section a schematic view of the architecture of the communicating robots will be given. This architecture will then be compared with cognitive architectures of the brain. This comparison will be brief, and does not intend to make claims that language is processed in humans the way it will be sketched. Furthermore, no discussion will be given on the neurological implementation.

The neurological plausibility of the theory will be discussed as follows: First, I start with looking at self-organising neural network theories that emerged mainly from artificial intelligence. I will discuss that ‘ordinary’ neural network theories will not suffice to selectively adapt languages. Secondly, the theory of neuronal group selection, introduced by Edelman [15], will be discussed as a possible candidate for the selective adaptation of language. And finally, a neuronal architecture is proposed for the implementation of the theory.

9.2 The cognitive architecture of the communicating robot

In this section the architecture of the robots will be compared with cognitive architectures that can be found in the physiological literature like [26]. The architecture is divided in modules that are thought to process different functions. All the functions that are processed in the robot in different modules can also be found (more or less) in different brain areas. In this section I will sketch the comparison without thorough motivations. The comparison is made more or less intuitively on the basis of knowledge of brain structure. The physiological data given in this section can be found in every overview of neurophysiology, see e.g. [26]. No further references will be given here.

The robots at the AI-Lab are built to investigate a behaviour-oriented approach of AI [39]. The sensors and actuators are connected in parallel to the ‘brain’ of the robot. The brain of the robots consists of the SMB2 sensory-motor board as described in chapter five, in which the program is written in a ‘Process Description Language’ PDL. PDL is capable of connecting sensory input with motor output in dynamical processes. These processes are implemented in parallel, so that it builds up a network of processes.

The network of processes that are build up in this project is schematically given in figure 9.1. As we can see, the whole system is divided in several processes that act in parallel. The architecture clearly distinguishes the sensors, the central processor and the actuators. This is conform the literature on robotics. It also closely resembles the organisation of cognitive

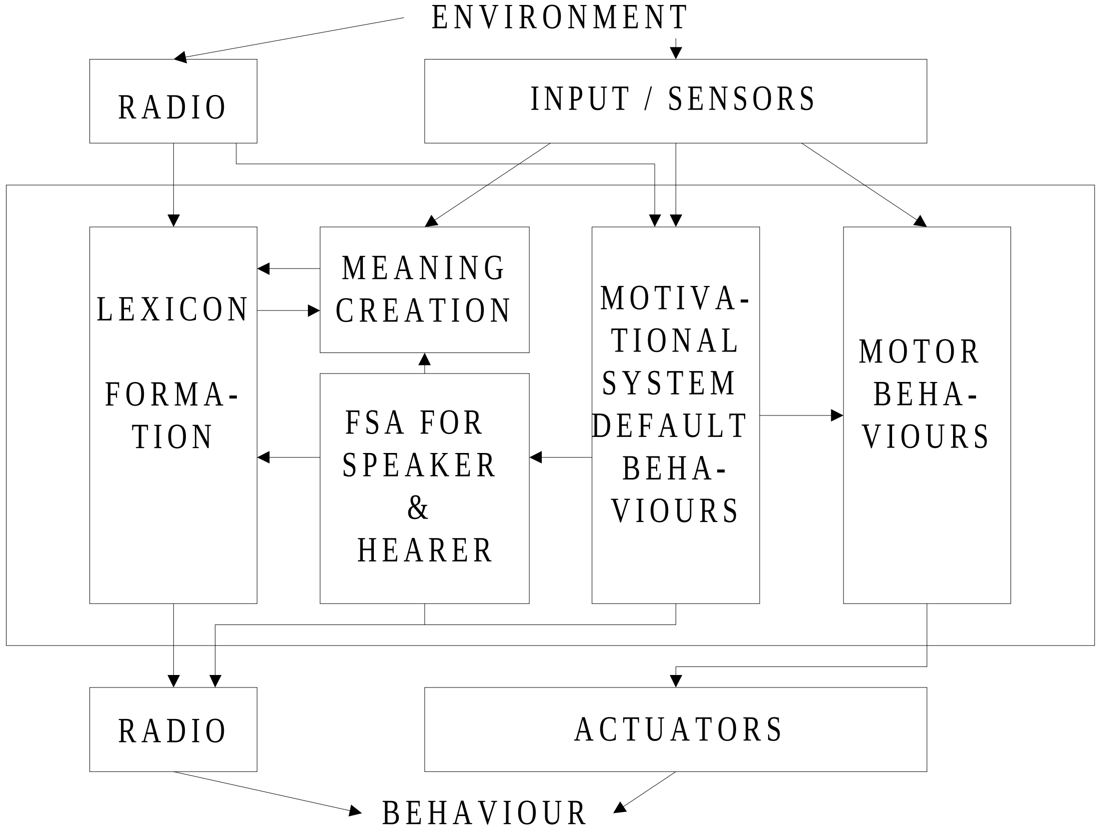


Figure 9.1 *The schematic architecture of the robots. On top we see the radio receiver and other sensors. The large box contains the program that is divided in several processes, which act in parallel and are interconnected. The processes are connected with external actuators, which in turn cause the behaviours.*

processes as they are described in the psychology literature (e.g. [29]). The sensors, including the radio-link, are connected with several modules that are each responsible for different cognitive capacities. In the figure, the radio-link is divided in two modules: the upper module can be classified as a sensor, and the lower one as an actuator. The radio-sensor is given separately from the other sensors, because the different functions of the radio in the communication system, as opposed to the perceptual receptors.

The radio-sensor is connected to the ‘default motivation’ module and the ‘lexicon formation’ module. The default motivation module calculates the motivation for default behaviours. Normally it is not directly driven by auditive stimuli, in the way it is here. This is inherent to the physical nature of the robot, which has a radio-link that may be compared with the auditive receptors. In our experiment the radio-link also takes over some vision-like functions in order to behave synchronously with its environment (i.e. other agents). Although the connection is not drawn in the figure, the radio is also directly linked to the FSA for the same reasons as mentioned above.

This module can be compared with brain modules that are specialised for language processing and -acquisition. This module is schematically given as one unit, but it may be subdivided in more, parallel and serial working, units such as a unit for language production (encoding) and one for language understanding (decoding). Such functionalities may be found in the brain as the Wernicke area and Broca’s area.

Another unit that may be found in the language formation is one for the storage of the lexicon (or the memory). This latter 'module' is thought to highly distributive in nature in the brain. It also must have clear connections with the meaning. The way that this may be stored is discussed in the next section.

The lexicon module is in two directions connected with the meaning creation module. The directions represent the two-way flow of information. The meaning creation module gets its input from the sensors in order to discriminate objects. The resulting distinctive feature set is matched with the lexicon in the speaker's task. In the hearer's task, the lexicon is matched with the stored meaning, yielding a feature set that is compared with the earlier derived distinctive feature sets. In our brain, meaning may be created in very distributed areas. Perceptual grounded meaning, however, starts from (or maybe before) the visual cortex. The discrimination process may reside in an area comparable to the visual cortex. Memory storage of meaning, on the other hand, is thought to be organised throughout the brain, but mainly in the associative cortex. Like the lexicon module, the meaning creation module also gets information from the FSAs for the hearer and the speaker causing the onset of processing.

The FSAs are shown in one box for simplicity, although they are separated for processing the speaker's tasks and the hearer's tasks. In this module, motivations are being determined for motor actions, as well as in what state of the automaton to be in. The module also initiates the language formation and meaning creation processes. One may see the separate procedures to which a state refers as different modules in the organisation, thus in fact being parallel processes. Although for clarity reasons the link is not shown in the figure, the motivations for motor actions are being sent to the motor behaviour module. The links that go to the radio transmitter module serve to broadcast signals for synchronising the agents. There is physiological and psychological indication that planning and motivational control is processed in the frontal cortex. In this brain area there are connections with many areas in the brain.

The default motivation module determines, as mentioned before, the motivations for default behaviour. This module also determines whether the agent should enter the speaker or hearer mode. Therefore, it has connections to the FSA module as well as the motor behaviour module. The default motivation module may also be compared to areas in the frontal cortex for the same reasons as the FSAs.

The motor behaviour module integrates the different motor behaviours, like IR-taxis, forward drive etc., and outputs the motor commands. All the different behaviours are in fact processed in parallel, although virtually, because the hardware is still not capable of actually processing in parallel. The integrated output values for motor actions are being sent to the motors directly. Motor processes in the brain are distributed as well, but it is thought that the cerebellum processes the production of motor action. The cerebellum receives efferent signals from different receptors to process stimuli directly and indirectly. The resulting actions are motivated by intentions that are thought to be derived in the frontal areas. A structure, that can be seen in the robot as well.

The motors, as well as the radio transmitter, can be seen as straightforward actuators. Although it is not mentioned here, the IR emitters are also actuators, and their processes can therefore also be classified in the appropriate pathways. I.e. their motivations are being calculated in the appropriate modules, the integration of their motivation and action commands are being processed in the motor behaviour module, which outputs the commands to the actuators.

In the robotic system, we can also distinguish between short-term memory and long-term memory from different perspectives. The clearest perspective is from the communication side. In a language game, objects are being perceived and firstly ‘stored’ in the short-term memory. The distinctive feature sets yielded from the discrimination games are also stored in the short-term memory as a reference to the features that are stored in the long-term memory. It is thought that the short-term memory also refers to long-term representations by activation in the hippocampus. The agent also ‘remember’ what has been expressed in a language game, while it determines the success. The long-term representation of meaning and lexical items occurs through repeated use of these items, which may be compared with long-term potentiation. Long-term potentiation is though to be responsible of the long-term representation of memory in the brain.

As argued in this section, the functional architecture of the robots may be compared with the functional architecture of the human brain. Comparisons are made on the basis of standard introductions to brain sciences as [26]. Although not every separate pathway in the robots is followed, they may work partly in parallel and partly in serial like the brain does. Parallelism in the robot is, however, only simulated. The SMB2 board receives its input and transmits its output in parallel, but the processes are actually been carried out in serial, although the processes are programmed in parallel. In this section the architecture of the robots is discussed in terms of its functional modality. The next section will discuss the possible functional implementation on the neuronal level.

9.3 The neuronal architecture

In this section the neuronal architecture of the language formation will be discussed. It focuses on the neuronal plausibility of the theory proposed by Steels on selectionistic language formation. The discussion should also be valid for the meaning creation, as well as other approaches to selectionistic systems capable of adapting other behaviours, as discussed for example in [47]. Moreover, the resulting discussion should be valid for the origination of cognition (or intelligence) in general, like it is discussed in [42]. There are more views on the theory of neuronal group selection [4], but they will not be discussed in this chapter.

This section first extracts the features of the language formation that a connectionistic framework should learn (section 9.3.1). Then it is discussed why the standard neuronal network theories are not likely to be adequate of adapting languages selectionistic in the way Steels has proposed (section 9.3.2). Section 9.3.3 briefly summarises the neuronal theory introduced by Edelman [15] as a more promising model. In section 9.3.4 a hypothetical model for language formation will be introduced. And finally, in section 9.3.5 some concluding remarks are being given.

9.3.1 What are the important features for a neural network?

The first sub-question that I will answer is: what do we want the neural network to learn? My main concern is how the mechanisms of the language formation, investigated here, can be modelled in a connectionistic model. The system of language games should be, I think, a hybrid system of some procedural systems that will do at least the following: (1) Choose a topic, (2) extract perceptual features, (3) form distinctive feature sets, (4) encodes - and (5) decodes expressions, and (6) adapt to new associations. In the remaining part of this chapter I will concentrate on processes (4), (5) and (6). In this section I will try to extract the main features of these processes that are necessary for the construction of a neural network. The network that needs to be constructed resides in every agent. It should be trained during real-time interaction with its environment. Let us first focus on what needs to be learned.

The neural network needs to be trained to encode the distinctive features from the lexicon and to output an expression if the agent is the speaker. If the agent is the hearer, then this same network needs to be trained to decode a set of features from the expression. So we need a network which is capable of processing reciprocally, although not both at one time. Furthermore, the network needs to have a learning mechanism which simulates the natural selective mechanisms proposed by Steels [46]. This is the most important feature of the network, because it is the way the system is evolving in a real-time fashion. The training should yield a network that represents the lexicon, together with the use and success scores of every entry.

So we need a network that is adequate for doing the following things:

1. In the speaker's task the network gets as input a series of features $\{f_1,...,f_n\}$, and the network should map these features to a series of words $\{w_1,...,w_m\}$ that it encoded.
2. In the hearer's task the network gets as input the words spoken by the speaker, and the network should map these words to get as output the set of features it decoded from the expression.
3. There must be a mechanism for the evaluation of the type of success.
4. Every time a language game had taken place, the network has to be trained according to the rules given in the previous section. This way the network should adapt to new associations, and construct the selectionistic strength of these associations.

Four types of adaptations should be involved:

1. When a new word-meaning (w-m) pair is created by the speaker, or a new association is made by the hearer, then the network should learn this new w-m pair with a success factor s=0 and a use factor u=0.
2. When there is a mismatch between the distinctive feature set and the uncovered features, then only the use factor u needs to be updated for every word that has been used. The agents also make new associations with s=u=0.
3. In case of success, the network needs to reinforce for every word that was used successfully both the use- and the success factor.

Here I should comment on a few things. First, the determination of success should be done in a hybrid subsystem of the system. Secondly, the use- and success factors should be implemented somewhat like weight factors. And finally, the learning mechanism should be implemented as one or more function(s) that will increment the use- and success factors as if they were populations. This learning mechanism should also be capable of storing new w-m pairs.

I think we now have outlined what the network should do. In the next section I shall try to identify the processes stated above with processes of well-known neural networks.

9.3.2 Problems with ordinary neural networks

In identifying a suitable ‘ordinary’ neural network for the selectionistic approach of self-organising languages, I got stuck on the fundamentally different nature of most learning mechanisms used in these ‘ordinary’ networks. This was because they all work highly distributed in nature. With ‘ordinary’ neural networks I mean networks that are commonly used for unsupervised learning: self-organising networks which use for example *Hebbian learning*, *Competitive learning* or *Information Theoretic models*. In these networks the neurones are updated during learning as single neurones, while their representation of the concepts are distributed. So, these concepts can hardly represent a population.

Although many of these networks may be capable of simulating the task (i.e. they all may be capable of adapting a self-organising vocabulary), the mechanisms they use to map the input onto the output are too different from the mechanisms introduced by Steels. The principles of ordinary neural networks are totally different from the algorithm used by Steels. If we let the weights of the network represent the success score values, we should have a one-to-one mapping of word-meaning pairs which is not normal in distributed networks. In ordinary networks, the representation of the information-carriers of a word or concept is highly distributed throughout the system. The representation of concepts (or word-meaning pairs) used by Steels are less distributed. If one, however, strongly opposes to the idea of letting the synaptic weights of the network represent the use/success-scores, one would get a fundamentally different representation of the vocabulary and their use/success scores. The opposition to the proposal to let these scores be represented somehow by synaptic weights, thus representing w-m pairs is very much arguable. The critiques can say that the distributed representation of concepts is a fundamental feature of connectionistic systems [36]. Due to the mentioned differences in representation, the ordinary networks are not adequate to adapt the language in the selectionistic way as proposed. So, it is not likely that we can find an ordinary neural network that is adequate for the formation of a language as hypothesised by Steels. We need a neural network that *is* adequate for representing populations of a concept in order to be selectionistic in the sense of Darwinian models.

Therefore I will discuss another type of connectionistic model. Steels [42] is referring to Darwinian neural networks, introduced by Edelman [15], for explaining the biological plausibility of his theory. Haykin [23] explains the Darwinian model as a selective learning model that presupposes that the nervous system operates by way of natural selection. This natural selection is taking place in the brain and during the lifetime of an animal. According to this theory, the basic operational units of the nervous system are not single neurones but rather a group of interconnected cells. Darwinian selective learning is different from the learning algorithms commonly used in neural networks. I.e. it assumes that there are many subnetworks, and that only the subnetworks with the desired response are selected during the training process [23].

In the next subsection I will explain the principles of the Darwinian selective learning method in greater detail. After that we are supposed to be ready to formulate an implementation of a self-organising vocabulary in such a network, which I will do in section 9.3.4.

9.3.3 Theory of neuronal group selection

In this section I will give a summary of the main principles of the theory of neural group selection as is proposed by Edelman [15]. Although Edelman does not really discuss the subject in detail, I will focus on the interest of this research: language. The theory of neuronal group selection tries to bring up another view of the theories that have been exploited on the neuronal workings of the nervous system, and especially the brain. In contrast to other theories Edelman tries to explain the information processes in the brain not by actions of single neurones but by the dynamically organising of individually variant groups of neurones [15]. These groups then behave like populations as in the theory of natural selection in evolution proposed by Darwin [10].

The theory makes three fundamental claims. In the words of Edelman:

> "(1) Diversification of anatomical connectivity occurs epigenetically during development, leading to the formation of primary repertoires of structurally variant neural groups... (2) A second process occurs during postnatal behaviour though epigenetic modifications in the strength of synaptic connections within and between neuronal groups. As a result, combinations of those particular groups whose activities are correlated with various signals arising from adaptive behaviour are selected. This selection occurs within the primary repertoire, and it results in the formation of a secondary repertoire consisting of functioning groups that are more likely to be used in future behaviour. Neurones in neuronal groups are populations, and repertoires form higher-order populations. And (3) coherent temporal correlations of the responses of sensory receptor sheets, motor ensembles (like speech), and interacting neuronal groups in different brain regions occur by means of re-entrant signalling. Such signalling is based on the existence of reciprocally connected neural maps. These maps link the secondary repertoires that emerge dynamically as a result of the selective behaviour of the selective developmental events and the synaptic selection mentioned above, and their re-entrant interactions maintain spatiotemporal continuity in response to real-word signals." [15].

The formation of the *primary repertoire* during the development of a species is thus co-ordinated by genetically based foundations, but it is also evolved through means of natural selection. The result of the formation of primary repertoires is a configuration of initial local maps, which can be functionalised during the formation of the secondary repertoires. I will not discuss the development of the primary map, because in an AI application the developer should do this formation. So, for our purpose this process is not necessarily explainable in biological terms.

The formation of the *secondary repertoire* is more interesting, because it explains the modification of the network at the synaptic level. Unlike most ordinary theories of neural networks - where the synaptic weights are modified either post-synaptic or pre-synaptic -, the synapses are modified at both post- and pre-synaptic levels. Post-synaptic modifications are fundamentally short-term modifications, and are caused by correlated activation of different dendritic synapses in one neurone. Post-synaptic modifications form the primary basis for neuronal group selection. Pre-synaptic modifications are essentially long-term modifications and determine the functioning of a whole neurone. These modifications are caused by fluctuations in the strength of the long-term potentiation of the neurone, and regulate the level of neurotransmitter release. The modifications on neurones in a group are strongly correlated to each other, due to lateral connections which causes overlap and competition. Furthermore, they are strongly correlated to signals that re-enter the local map by re-entrant connections. The formation of groups is, thus, a dynamical interaction between competing groups and their input and re-entrant signals.

The re-entrant maps are to correlate signals to and from functionally different local maps, as well as within the same local map. This correlation is to keep the spatiotemporal organising of functionalities in order. I.e. the neuronal organisation of procedures and categories is kept in order by re-entrant correlations [15]. The whole system of a certain functionality, like the information processing of language, is structured by a set of (more specified) local maps. These local maps, together with their re-entrant connections, constitute a global map.

According to the proposals of Edelman, representations in a global map (i.e. the whole system for processing a certain type of information) are procedural. This means that memory in the system constitutes a procedure of recategorization in which particular output may be achieved repeatedly by many different (degenerate) combinations of group activity. “Memory is stored as a result of alterations of synaptic strengths. It leads to connections of whole systems or populations of neuronal groups responding to unique features, with separate populations of groups acting to correlate features and yielding more or less invariant response.” [15]. The collection of global mappings with such memorial properties provides the main basis for generalisation and categorisation, enhancing the possibility of adaptive learning as such maps are linked to hedonic centres (i.e. the centres that provide a basis for the assignment of value). [15].

Summarising, we can state that the theory of neuronal group selection offers a neural theory that constitutes several local maps. These maps are functionally defined by the dynamical formation to groups of neurones through selective modification of their synapses in response of input- and re-entrant signals. Moreover, these neuronal groups are represented as populations. The whole system is organised in a set of global maps in which the concept of memory is represented as a procedure of categorisation and generalisation. The system is capable of interacting with other species in order to evolve a whole society by the means of natural selection. A schematic summary of the theory is given in figure 9.2.

In my aim of constructing a neural network system adequate for adaptive lexicon formation, we should construct a global map consisting of several local maps in the primary repertoire. Each map should have its own functionality and should be connected with other maps for re-entry. The rules for updating synaptic weights within a local map and between re-entrant maps should be twofold: (1) pre-synaptic and (2) post-synaptic. This makes the system selective in the Darwinian sense. In the next section I will propose such a network in a simplified way.

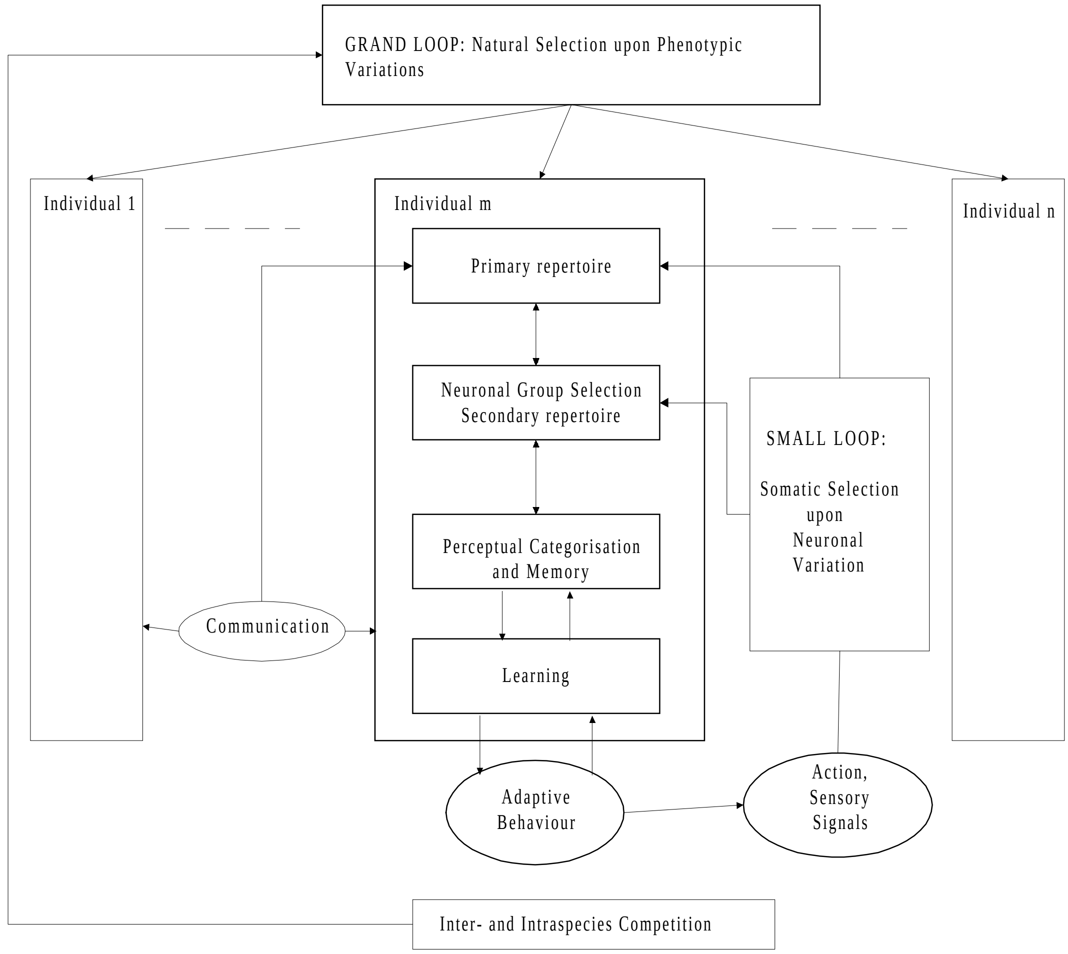


Figure 9.2. *A schematic illustration of some interactions between evolution (the GRAND LOOP) and various developmental constraints imposed by epigenetic events and by somatic neuronal group selection (SMALL LOOP). Adapted from [15].*

9.3.4 Language formation through neuronal group selection

In this section I will propose a model based on the theory of neuronal group selection for the adaptive lexicon formation through self-organising. I will do so by introducing a simplified architecture of neuronal maps (figure 9.3). Each map is thought to represent a part of the lexicon (formation). The maps can interact with each other by re-entrant connections. In all maps the neuronal groups represent information. In the model I propose, each agent would be able to adapt a language through interspecies interaction. Furthermore, this model could be used for similar systems like adaptive meaning creation, phonetic formation, level-formation and for the formation of intelligence in general, as proposed in [42] and [47]. Like in our robotic experiments, only meaning creation is integrated. Thus, the model will be a general-purpose neural network. The model is developed in analogy with the models introduced in [15].

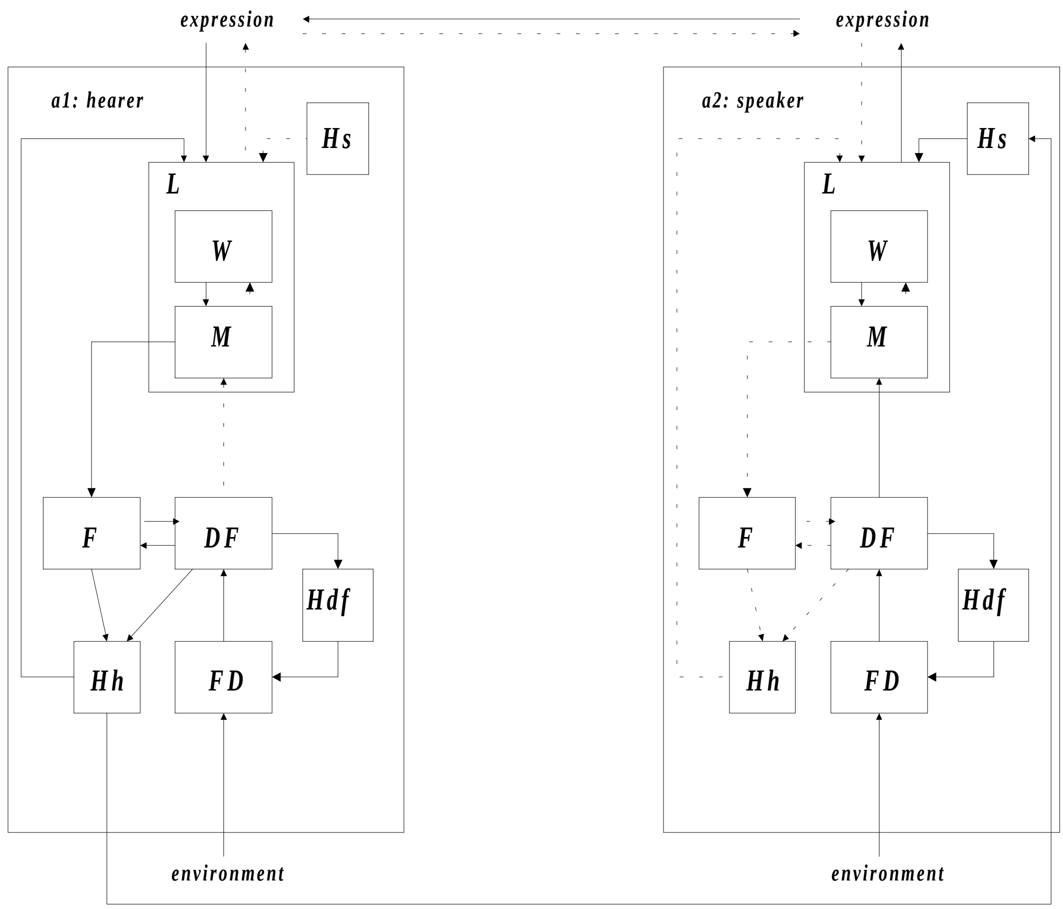


Figure 9.3 *The architecture for a neuronal group selection based language formation in two agents. Each agent has a global map that consists of local maps for the lexicon (**L**) that is divided in words (**W**) and meaning (**M**), the distinctive feature sets (**DF**), a decoded feature set (**F**) and a map for feature detectors (**FD**). Three hedonic maps (**Hh, Hs, Hdf**) are also interconnected. In a language game between speaker and hearer, the continuous lines are active, while the dashed lines are inactive.*

As we saw in paragraph 9.3.1, we need a network that: (1) Is capable of encoding a distinctive feature set for the right choice of an expression. (2) Is capable of decoding an expression in a distinctive feature set And (3) will allow the system to adapt a lexicon on a selectionistic manner. Hereto I propose a set of eight local maps. Figure 9.3 gives a schematic view of the proposed architecture. This figure represents two agents that are participating in a language game; the maps that I propose to involve the language formation are shown in each agent. Continuous lines are active in an agent according to its role in a language game, as shown in the figure. The discontinuous lines are inactive.

I propose two maps (W and M) for the storage of w-m pairs, and the en- and decoding of these pairs. These maps represent the lexicon, and can process information reciprocally. Re-entrant connections serve the adaptation of the language. Another map will have a long-term storage of feature detectors (FD). From this map a temporal storage of distinctive feature sets can be determined (DF). A hedonic system (H_{DF}), i.e. a system that assigns a value to a certain process, is connected to these perceptual maps in order to adapt meaning as in [43]. These three maps form the core of perceptual grounding. Another map in the proposed architecture is active when the agent is a hearer. This map has a short-term representation of decoded features (F). This map serves to compare, together with another hedonic map (H_h), these decoded features with the map of distinctive feature sets. The results of this hedonic system re-enters signals to the maps that form the lexicon, in order to adapt the lexicon appropriately. This hedonic map also sends a signal through interspecies communication to another hedonic map (H_s) that is active in the speaker (fig.9.3).

The configuration of maps, as shown in figure 9.3, results in a global map that must be capable of perceptual grounding and lexicon formation in a selectionistic way. All maps, except maybe the hedonic systems, must be large enough to represent certain concepts, like words for instance, as neuronal groups. These maps must have feedforward connections with other maps in order to select associations by self-organisation, thus forming a lexicon. Lateral connections between neurones in a map must cause the self-organisation of group selection and competition. Constructing excitatory connections to nearby neurones and inhibitory connections to neurones further away can accord for this. Group formation then emerges by way of synaptic modifications. As mentioned, the post-synaptic modifications will serve the short-term group selection. Long-term potentiation of certain neurones will cause pre-synaptic modifications, which, in turn, cause successful connections to become long-term representations.

In the remaining part of this section I will describe the system in more detail by explaining the processes during the playing of a language game. Without further reference, I will refer to figure 9.3 during the explanation; this will make the process clearer.

When two agents engage in a language game, they first have to delineate the context of the conversation by means of perception. Therefore, the objects in this context are perceptual grounded. So, both agents look in their surroundings, which causes certain neuronal groups in the FD map to be active. These groups represent the features of the objects. Then both agents determine the topic of the language game (this is not shown in the figure). If the topic is determined, the FD will send its activated signals to the DF, where discrimination of objects takes place. The H_{DF} will determine the success of the discrimination. If this is not successful, then re-entrant signals to the FD will cause the FD to refine their feature detectors in new groups. The groups that are activated successfully in the DF, make connections to map M in the speaker (a2). This will activate neuronal groups in M, which represent the features that are activated by the DF. It may well be possible that maps DF and M can be integrated into one map, but for clarity reasons I have divided these maps. In case this is possible, however, map F also needs to be integrated so an internal process in the resulting map can compare the activated groups. I will not discuss this integration further, instead I will continue the language game. The neuronal groups that are now active in M will send excitatory signals to W. In W these signals cause associated neuronal groups to be activate. Groups that represent the largest populations now will be most activated. A process of *the winner takes all* now causes the speaker to select the most successful used association, which it expresses to the hearer.

When the hearer receives the expression, this will cause the right neuronal groups in W to become active. Activation of entries in W will, in turn, activate associations made with M, yielding a set of active neuronal groups in F representing these features. Re-entry with DF from F, will compare these maps in terms of equal features. This comparison yields a signal to hedonic system H_h, which evaluates the type of success of the language game, and will send appropriate re-entrant signals to W. By means of normal communication, these signals are also sent to H_s in the speaker, which, in turn, will send appropriate re-entrant signals to map W of the speaker. These re-entrant signals now must selectively adapt the lexicon of both agents appropriately, according to the mechanisms explained in chapter 3.

Although I have not worked out this adaptation in detail, I will outline the steps that need to be taken in order to form a lexicon. Where possible, I will also make suggestions how this may be implemented.

1. The speaker could not encode the distinctive feature set. A new word-meaning pair needs to be created. This may be implemented by exciting a region in map W that has not been used yet. Or a region that already turned out to be unsuccessful, thus degenerating the association it represented. This region then, somehow, must represent a word. At the same time connections are made with meanings that are still active in map M. This association must represent a very small population, so the group must be small. This group can be formed by small post-synaptic modifications to the association by excitatory re-entry into M. Small excitation of lateral connections within W can pre-form the group that will respond with the word it formed.
2. The hearer could not decode the expression in a feature set. The hearer needs to make a new association with the expression of the speaker that it has in W and the active features in M. This may be done by an excitatory signal from the active group in W to the active groups in M. Causing post-synaptic change in M. The speaker, in turn, has to increment the use factor. This raises a fundamental question that I have not answered yet. Are use and success populations separately represented in a group, or are they integrated in a group? I suggest the latter solution may suffice. In that case, the re-entrant signal from H_s to W and M may be inhibitory, causing the strength between the connections to decrease.
3. There was a mismatch in meaning. The hearer makes a new association in the same way as described above. Both the hearer and the speaker increment the use but not the success. This may also be implemented as mentioned above.
4. The language game was a success. Both agents send excitatory signals to the still active associations. Thus, the group will increase due to post-synaptic

modifications in the lateral and feedforward connections within and between the neuronal groups in both maps W and M.

Some comments may be in place here. (1) If an association will be used often, while there is no success, inhibitory long-term potentiation will cause this association to die out by pre-synaptic modifications. (2) On the other hand, if the association is used successfully for a longer time, then long-term excitatory potentiation will increase the neuronal group population by pre-synaptic changes. (3) I think the proposed inhibition for increasing the use and not the success, will yield the same result as updating them separately. Separate updating of the use and success would require two different channels, or groups, for every association. I think this would not be so efficient.

Summarising, I have proposed a hypothetical architecture for the selective language formation as in [46], based on the theory of neuronal group selection introduced by Edelman [15]. This is cf. the suggestion made in [42]. The architecture is hypothetical in the sense that it is theoretically based on the theory of neuronal group selection; no experiments have been carried out. The model consists of a global map connecting eight local maps together in order to be capable of selective language formation. Both pre- and post-synaptic modifications and re-entrant connections between local maps carry out group selection. Evaluation of successfulness of conversations is processed by hedonic systems, as suggested in [15]. The evolutionary processes of the language will be driven by interspecies interaction, causing cultural selection.

9.3.5 Conclusions and future research

In this section I have done theoretical research on the neuronal plausibility of the hypothesis on the origins of language as proposed by Steels [46]. Though, a lot of research still has to be done on the self-organising adaptive language formation in symbolic experiments [47]. These experiments are currently been carried out at the AI-lab for the formation of a lexicon, syntaxis and phonetics on both robotic and software agents. A lot more research also needs to be done on the neuronal plausibility of the hypothesis described in [46]. This section may be a first and small attempt. Implementation of the neuronal group selection for language formation is a main attempt for future research in this area.

Due to the unary methods of synaptic adjustments in ordinary neuronal theories, combined with their views on neurones as single units and the lack of re-entrant signals, these theories do not seem very promising in modelling Steels' method. Therefore I have argued that the theory of neuronal group selection [15] is a more promising model. This theory proposes groups of neurones as populations. These groups could represent word-meaning pairs as competitive entities in an agent. In the evolution of a coherent language in a society of agents, natural and cultural selection will be the selective force on the formation of these neuronal groups. The theory describes three fundamental principles for selectionistic adaptation of behaviour (like, for example, language): (1) The formation of a primary map, which is due to epigenetic selection. (2) The formation of a secondary map by means of inter- and intraspecies interactions, and both pre- and post-synaptic adjustments. And (3) re-entrant connections between functionally different but dependent neuronal maps.

According to the theory of neuronal group selection, I have proposed a model that should be adequate in evolving a self-organising language as an emergent principle through selection. The developer should carry out the first fundamental principle of neuronal group selection. He/she must develop the architecture of maps, together with the primary re-entrant connections and the rules for synaptic modifications. The second principle will emerge through the processes of playing multiple language games and the rules that are governed by the theory of neuronal group selection. The language formation by means of language games are both inter- as intraspecies processes. Thirdly, re-entrant connections will be formed initially by the developer as the primary principle. They will be strengthened or weakened by the whole process of interaction.

I have given a rough outline of a connectionist model that could integrate the theory of neuronal group selection with the hypothesis of the selectionistic adaptation of language. Both theories, however, have not even been proven. It may, for example, turn out that some other neuronal theory may be adequate to serve the language formation. The model I proposed must be worked out in greater detail before implementation can be possible and experiments can be carried out. If, however, the conclusions made by Edelman prove to be valuable, then this theory seems to be a promising one in order to test the hypothesis made by Steels.

9.4 Conclusions

In this chapter the cognitive plausibility of the system has been discussed on two different levels: (1) on the functional brain-behavioural level, and (2) on the neuronal level. Under (1), the robotic architecture has been discussed and compared with cognitive architectures of brain-behaviour models. And under (2) the theory of language formation is compared with the theory of neuronal group selection introduced by Edelman [15].

We saw that, although the robots are fundamentally different from human beings, the functional architecture of the robot looks like the functional architecture of the human brain. The architecture of the robots, however, is much less complex than the human brain, which is natural, since humans still have much more complex cognitive capacities that any robot. But we could see that the capacity of the robots to communicate and form a lexicon made the architecture more complex and, qua structure, more or less similar to that of the human brain.

In section 9.3 the theory of language formation is described in terms of neuronal group selection. We saw that that theory was capable of adapting a language on a selectionistic base. There is, however, much critique on the theory of neuronal group selection. One of which argues that Edelman’s theory is not Darwinian in that there is no form of replication, which is one of the main principles of Darwin’s natural selection [53]. The proposed theory in the former section may be integrated with another theory of group selection that is proposed by Changeux et.al [4]. I think their theory does resolve this problem better.

In general, we could conclude that the system that evolved in the course of this project has a biological plausibility, because the system has properties of the human brain. And it could be implemented in a connectionistic model that, at least to some extent, explains the neurological mechanisms of the brain.

10. CONCLUSIONS

This thesis described the implementation of a self-organising grounded lexicon formation as was introduced by Luc Steels (see e.g. [40][41][44][46]). Experiments are carried out with this implementation and the results have been reported (see also [48][51]). At the beginning of this thesis, a summary is given of the most influential theories of the evolution of human languages, as well as a detailed description of the approach that is been studied at the AI-Lab in Brussels. A detailed description of the experiments and their implementation has been given in the middle section of this thesis. Results have been discussed in chapter eight. Finally, the preceding chapter discusses the cognitive plausibility of our approach. In this chapter I will give some concluding remarks on the experiments that have been done. I will also give a summary of future research that is still needed on the robotic implementation of language formation.

The approach that we follow in the research on the origins of language is a selectionistic one in a cultural sense. It is based on three basic mechanisms: *generation*, *propagation* and *self-organisation* [46]. The robots ground the language themselves, according to the theory described by Luc Steels [43]. This grounding, also, is selectionistic. The basic mechanisms of this grounding are the same as for language formation, except the *propagation*, which is not used in grounding. The robotic agents thus ground the meaning of objects individually. The formation of the language then evolves culturally. All the mechanisms were implemented in robotic agents, in order to enable them to communicate (or engage in a language game). This ability is our main physical assumption [51]. We did not focus on the question how communication itself may arise.

So, summarising, our goal was to implement a communication system that had to ground language from scratch. The resulting experiment addresses two fundamental question on the origins of cognition [51]: (1) How can a set of perceptual categories (i.e. a grounded ontology) arise in an agent without the assistance of others and without having been programmed in. And (2) how can a group of distributed agents which each develop their own ontology through interaction with the environment develop a shared vocabulary by which they can communicate about their environment.

In chapters six and seven we saw that the implementation of the communication processes was extremely difficult. The communication, which is described as a language game, is conditioned by a lot of physical constraints. In contrast with simulations, as described in e.g. [43][46], this experiment has to deal with the physical world. A physical world that, although the environment is closed, continuously changes. Changes in temperature, light circumstances, physical states of the agents etc. can influence the environment enormously. The dynamical approach of the sensory-motor behaviour, though, is capable of dealing with these constraints. The developed protocol, which was defined in chapter seven, also deals with constraints that are put on the experiment.

The hardest part of the implementation was the search by the robots for each other at the onset of a language game. The synchronising of the distributed agents was also very difficult. The perception and lexicon formation, although not easy, was less difficult because the algorithms were more or less given. The search of another agent uses the active IR-module, and the method of IR-taxis and -orientation. Synchronising is achieved by sending radio signal whenever a certain transition in the other agent has to take place. The robots rotate around their axis in order to build a temporal map of their (immediate) surroundings. Discrimination games and naming games are used for the grounded language formation.

The implementation yielded a system of two agents that could be in three modes: regular exploration mode, speaker mode and hearer mode. In the speaker mode, the agent first searches the hearer, then it scans its surrounding, after which it grounds meaning for the topic and decodes an expression. The hearer, on the other hand, first waits until the speaker has found it, then it aligns, scans the surroundings, determines the topic, grounds its meaning and encodes the expression. Finally the hearer determines the success of the language game. This system is robust in the sense that it jumps to the regular exploration mode, whenever a particular state in the other modes fails. And it does fail sometimes.

The experiments revealed failures in (1) the initiation of a language game, (2) the IR-taxis and -orientation, (3) the map building, (4) topic determination, (5) discrimination of objects, (6) the formation of language, and (7) the synchronising by radio-link. So, failure occurred at all levels of the system. Failure (1), however, occurred most of the times. This was caused by the method by which robots initiate communications, as well as the failures in making radio contact. This directly brings us to the follow up in terms of number of failures. The radio contact did not always work properly. This may be due to the build-in unreliability of the system. Next, failure (4) occurred quite often, which is probably due to the uncertain relations of the IR module in terms of reflections and noise of the IR. The same reasons can be brought up for failures (2) and (3) which are the next in row. Failure (6) mainly occurred as a 'by product' of failures (4) and (7), as well as some small errors in the implemented algorithm. The latter reason is also cause of tiny failures in (6). Despite the mentioned failures, the system works well enough to show that a grounded ontology, as well as a shared vocabulary arose in robotic agents during the experiments.

Although we think that the results of these experiments are an important step forward in the ALife approach towards the origins of cognition, a lot of work still needs to be done. We are currently working on (1) the refinement of this system, (2) the actual implementation of the formation of spatial categories, (3) the grounding and lexicalisation of other objects, situations, actions and internal states, and (4) the integration with experiments that uses vision as the primary sensory experience [51]. Details for the first two experiments are already discussed in great deal in this thesis. The third option concerns initially, the grounding of action. There the robots have to match observed actions with internal states. The fourth experiment is currently being developed. The idea is to use two cameras attached to a PC-network as the two agents. The camera can move and follow a robot. The camera agents then identify the topic as being the object that is closest to the robot. The language games can thus be held.

ACKNOWLEDGEMENTS

The experiments reported here are done at the VUB AI Laboratory in Brussels for my masters' thesis project for Cognitive Science and Engineering at the University of Groningen. First, I would like to thank Tony Belpaeme, Andreas Birk, Peter Stuer and Dany Vereertbrugghen of the AI-Lab for helping me with the technical problems, as well as for the creation and maintenance of the robotic environment. Secondly, I would like to thank Bart de Boer, also AI-Lab, for his advice on writing a thesis in English and our discussions on the evolution of languages.

Furthermore, I would like to thank Prof. Dr. Luc Steels of the AI-Lab for his extremely good advice, especially on the language formation and protocol definition. I would also like to thank Dr. Petra Hendriks of Cognitive Science and Engineering for her clear advice on linguistic literature research, and her critical remarks on the early drafts of this thesis. This working visit was partly financed by an EU Erasmus fellowship.

I would also like to thank my family for their support along my career as undergraduate student.

REFERENCES

[1] Atchison, J. (1994) Words in the mind: an introduction to the mental lexicon. 2nd edition. Blackwell Publishers Inc. Cambridge Ma.

[2] Bickerton, D. (1995) Language and human behaviour. University of Washington Press.

[3] Billard, A. and K. Dautenhahn. (1997) The social aspect of communication: a case study in the use and usefulmess of communication for embodied agents. Submitted to ECAL97.

[4] Changeux, J.-P. and S. Dehaene. (1989) Neuronal Models of Cognitive Functions. In: *Cognition 33 (1989)*, pp. 63-109.

[5] Changeux, J.-P., T. Heidmann and P. Patte. (1984) Learning by Selection. In: *The Biology of Learning*, P. Marler and H.S. Terrace (eds.), pp. 115-133. Dahlem Konferenzen 1984. Springer-Verlag

[6] Chierchia, G. and S. McConnell-Ginet (1990) Meaning and grammar: an introduction to semantics. The MIT Press.

[7] Chomsky, N. (1980) Rules and representations. In: *The behavioral and brain sciences* (1980),3, pp. 1-61. Cambridge Univ. Press

[8] Chomsky, N. (1982) The generative enterprice: A discussion with Riny Huybrechts and Henk van Riemsdijk, Foris.

[9] Comrie, B. (1992) Before complexity. In: [22].

[10] Darwin, C. (1859) On the origin of species. Reprinted in 1964 by Harvard Univ. Press.

[11] Dawkins, R. (1976) The selfish gene. Oxford Univ. Press.

[12] Deacon, T.W. (1992) Brain-language coevolution. In: [22].

[13] Dennett, D.C. (1993) Conciousness Explained. Penguin. London.

[14] Dennett, D.C. (1995) Darwin's dangerous idea:evolution and the meanings of life. Penguin. London.

[15] Edelman, G.M. (1987) Neural Darwinism. Basic Books Inc., New York.

[16] Flynn, A.M. and R.A. Brooks (1989) Battling Reality. AI-Memo 1148, MIT.

[17] Geschwind, N & Galaburda, A (eds.) (1984) Cerebral dominance: The biological foundations. Cambridge, MA: Harvard University Press.

[18] Goethals, O. (1996) Praktische Implementatie van een Zelforganiserende Woordenschat. Licenciaatsverhandeling VUB 1995/1996.

[19] Gould, S.J. (1987) The limits of adaptation: Is language a spandrel of the human brain? Paper presented to the *Cognitive Science Seminar, Centre for Cognitive Science*, MIT.

[20] Gould, S.J. and R.C.Lewontin (1979) The spandrels of San Marco and the Panglossian paradigm: A critique of the adaptationist programme. In: *Proceedings of the Royal Society of London 205*. pp. 281-288

[21] Hawkins, J.A. (1992) Innateness and funcion in language universals. In: [22].

[22] Hawkins, J.A. and M. Gell-Mann (eds.) (1992) The evolution of Human Languages. Proceedings volume XI of the Santa Fe Institute Studies in the Sciences of Complexity. Addison-Wesley

[23] Haykin, S. (1994) Neural Networks. Macmillan College Publishing Company, Inc., New York

[24] Hubel, P. and T. Wiesel. (1959) Receptive fields of single neurones in the cat's visual cortex. In: *Journal of Physiology* 148. 57-591

[25] Hurford, J.R. (1992) An approach to the phylogeny of the language faculty. In: [22].
[26] Kandel E.R., J.H. Schwartz and T.M. Jessell (eds) (1995) Essentials of neural science and behaviour. Appleton & Lange, Norwalk,Connecticut.
[27] Lieberman, P. (1984) The biology and evolution of language. Harvard University Press.
[28] Lieberman, P. (1992) On the evolution of human language. In: [22]
[29] Matlin, M.W. (1994) Cognition. Forth Worth. Harcourt Brace.
[30] McFarland, D. (1994) Towards Robot Cooperation. In: Cliff, Husbands, Arcady Meyer, and Wilson (eds.). *From animals to animats. Proceedings of the Third International Conference on Simulation of Adaptive Behavior*. The MIT Press, Cambridge Ma.
[31] Ott, E. (1993) Chaos in dynamical systems. Cambridge Univ. Press
[32] Pinker, S. and P. Bloom (1990) Natural language and natural selecion. In: *Behavioral and brain sciences* (1990), 13, pp. 707-789. Cambridge Univ. Press
[33] Prigogine, I. and I. Strengers (1984) Order out of Chaos. Bantam Books, New York.
[34] Putnam, H. (1975) Mind, Language and Reality, Philosophical papers volume 2. Cambridge University Press.
[35] Romaine, S. (1992) The evolution of linguistic complexity in Pidgin and Creole languages. In: [22]
[36] Rumelhart, D.E. & J.L. McClelland. (1986) Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol.1. MIT Press, Cambridge, MA.
[37] Steels, L. (1992) The PDL Reference Manual. VUB AI-Lab Memo 92-05.
[38] Steels, L. (1994a) Artificial Life route to Artificial Intelligence. In: *Artificial Life Journal 1(1)*, pp 89-125
[39] Steels, L. (1994b) A case study in the behaviour-oriented design of autonomous agents. In: *Proceedings of the third Simulation of Adaptive Behavior Conference*. The MIT Press, Cambridge Ma.
[40] Steels, L. (1996a) Self-organizing vocabularies. In: Langton, C. (ed.) (1996) *Proceedings of Alife V*, Nara Japan.
[41] Steels, L. (1996b) Emergent Adaptive Lexicons. In: Maes, P. (ed.) (1996) *From Animals to Animats 4: Proceedings of the Fourth International Conference On Simulation of Adaptive Behavior*, The MIT Press, Cambridge Ma.
[42] Steels, L. (1996c) The origins of intelligence. To appear in: *Proceedings of the Carlo Erba Foundation conference on Artificial Life*. Fondazione Carlo Erba. Milano. 1996
[43] Steels, L. (1996d) Perceptually grounded meaning creation. In: Tokor, M (ed.) (1996) *Proceedings of the International Conference on Multi-Agent Systems. AAAI Press*, Menlo Park Ca. pp 338-344
[44] Steels, L. (1996e) A self-organising spatial vocabulary. In: *Artificial Life Journal, 3(2)*.
[45] Steels, L. (1996f) Discovering the competitors. In: *Journal of Adaptive Behavior 4(2)*. 1996.
[46] Steels, L. (1996g) The Spontaneous Self-organization of an Adaptive Language. To appear in: Muggleton, S. (ed.) *Machine Intelligence 15*. Oxford University Press, Oxford.

[47] Steels, L. (1997a) Synthesising the origins of language and meaning using co-evolution, self-organisation and level formation. In: Hurford, J., C. Knight and M. Studdert-Kennedy (ed.) (1997) *Evolution of Human Language*. Edinburgh Univ. Press. Edinburgh.

[48] Steels, L. (1997b) Constructing and Sharing Perceptual Distinctions. In: van Someren, M. and G. Widmer (eds.) (1997) *Proceedings of the European Conference on Machine Learning*. Springer-Verlag, Berlin.

[49] Steels, L. and R. Brooks (eds.) (1995) Building Situated Embodied Agents. The ALife route to AI. Lawrence Erlbaum Ass. New Haven.

[50] Steels, L., Stuer, P., Vereertbrugghen, D. (1996) Issues in the Physical Realisation of Autonomous Robotic Agents. VUB AI-Lab AI Memo 96-02.

[51] Steels, L. and P. Vogt (1997) Grounding adaptive language games in robotic agents. Submitted to ECAL97.

[52] Stuer, P. (1996) Using PDL on the CBot: an introduction. Draft.

[53] Van Belle, T. (1997) Is Neural Darwinism Darwinism? In: Artificial Life, Vol 3,1 1997. MIT Press, Cambridge, MA.

[54] Vereertbrugghen, D. 1996. Design and Implementation of a Second Generation Sensor-Motor Control Unit for Mobile Robots. Licenciaatsverhandeling VUB 1995/1996.

[55] Vogt, P. (1997) The neuronal plausibility of a selectionistic evolution of an adaptive language. Draft.

[56] Wittgenstein, Ludwig (1967) Philosophische Untersuchungen, Frankfurt: Suhrkamp.

[57] Yanco, H. (1992) Towards an Adaptable Robot Language. In: *Proceedings of the AAAI Fall Symposium on the Applications of Artificial Intelligence to Real World Robots*, Boston, MA.