Design and Performance of Pre-Grammatical Language Games

JORIS VAN LOOVEREN

Artificial Intelligence Laboratory
Vrije Universiteit Brussel

Proefschrift voorgelegd voor het behalen van de
academische graad van doctor in de wetenschappen.

Acknowledgements

DOCTORAL THESISES have, by definition, one single author on their cover. After all, only one person can earn their degree with the same work. Everybody knows however that the work carried out by the applicant never stands on its own. On the one hand, the scientific work is always framed within a larger body of work, in which there is a need or a desire for that specific work to be carried out, for example to solve a smaller problem within the framework, so the framework can be carried further, or because it seems to be a promising line of research to pursue.

On the other hand, the researcher is not isolated from the world. He functions in the world; interacts with colleagues, family and friends, and is influenced by them. Some of these influences, such as the interactions with his colleagues, will have a large impact on the research, while interactions with family and friends have not so much a direct impact on the scientific work, but nevertheless are crucial to make the researcher feel good and allow him to do his scientific work as well as possible.

Of course, I am no exception to this rule. In total, I spent about ten years at the VUB and in Brussels, four years as a student, and six years as a researcher at prof. Steels' Artificial Intelligence lab. During that time, I have seen many people come and go, both at the university, and elsewhere in my pursuit of other activities.

First of all, I want to thank my supervisor, prof. Luc Steels, for the opportunities he has given me. I have worked at both of his laboratories, the AI-lab at the VUB in Brussels and the Sony Computer Science Laboratory in Paris. I have been able to go to many places and meet many interesting people, through conferences and through the Talking Heads project. I have had the honor to work together with many people that, thanks to their different backgrounds, each have had different, interesting points of view on many issues: Tony Belpaeme, Joachim De Beule, Bart De Vylder, Bart Jansen, and Dominique Osier. Apart from these current colleagues, over the years we also had a steady supply of Dutchmen: Bart de Boer, Edwin de Jong, Paul Vogt, Jelle Zuidema, and Tom ten Thij, and an extended German presence in the form of Andreas Birk, Holger Kenn and Thomas Walle. At CSL, Frédéric Kaplan and Angus McIntyre have been essential to get our scientific projects going.

At the personal level, there are a great many people who helped, by being who they are, get me to where I am now: my "running mates" from the Brussels

Athletics Club, my teachers and co-students at the photography course, etc., and not in the least, the old, but stable values in my circle of friends: Peter and Bénédicte and their wonderful kids, Bart and Sofie, and Zeger. And finally, without my parents, who supported my studying at the university, and my two sisters, all of this would simply not have been possible.
Of course, not only relationships between people make for a good PhD thesis. It may be a sobering thought, but without monetary support which ensures that one can concentrate on the research, such an undertaking would be next to impossible. The work reported in this thesis has been financially supported by several institutions. I started doing research in Paris, at Luc Steels' Sony Computer Science Laboratory. After my four-month stint at Sony CSL, I came back to the VUB to work first as a researcher, then as a teaching assistant. Finally, on January 1st, 2001, I became a full-time researcher again on a four-year scholarship of the IWT (Instituut voor Innovatie door Wetenschap en Technologie, Vlaanderen).

Brussels, March 24th, 2005

Joris Van Looveren

Abstract

Past efforts in the study of language and its evolution have tended to focus on an individual's language capacity, and tried to understand in detail how this speaker/hearer's language capacity (LC) works. This was done e.g. by presenting people with sentences containing special cases and exceptions of specific rules, and judging their reaction to them. While this is a valid approach, it ignores many aspects of language that may be relevant to a global picture of how it works. Additionally, when the daunting complexity of the LC became apparent, it has been proposed that it must have evolved genetically, in analogy to the complexity of what evolution has accomplished in nature. This view has become widespread throughout the linguistic community.

Recent research, especially on the evolution of the human LC, has taken more of a bottom-up approach, by attempting to identify core features of language, and thinking about the order in which these features must have become available for language. Initially, the linguist Bickerton proposed two stages, a simple protolanguage and modern language, with a genetic transition between the two. More recently his colleague Jackendoff proposed a much more detailed schema with many milestones that must have been reached at some point during the evolution of language.

Techniques developed in other areas of science are also being applied more and more to language. Of specific interest here are game theory (from economy) and dynamic systems (physics), because they are specifically geared towards systems with many small components, and the interactions between them. Language can be viewed as a prime example of such a system, with many individuals that interact, and create a language in this way.

Computer science offers a method that permits us to actually test theories based on this view of language in a very elegant way: multi-agent simulations. Individual language users are modeled as agents, which each have the ability to produce or interpret an utterance. The agents are then allowed to interact repeatedly according to a fixed protocol (a language game), describing objects and events that occur in their environment. During such a series of interactions, the agents develop utterances to express their meanings, and ultimately develop fully usable communication systems that cover the environment.

This thesis describes three multi-agent models of three different linguistic communication systems, which correspond roughly to several of the milestones in Jackendoff's proposed schema. The agents have different cognitive capabilities in each model: in the first model, the agents are capable only of expressing simple meanings using utterances the contain only one word. The second model extends this with compositional meanings and the capability to use several words in one utterance to allow the agents to produce more complex utterances. The agents in the third model are able to express meanings that contain references to several objects and/or events (and the relations between them) through utterances that contain syntactic structure.

Each model is evaluated against a number of criteria to see how it performs: basic communicative success, but also more qualitative measures such as lexicon size, lexical coherence, and degrees of homonymy and synonymy. It is shown that the same type of dynamics that work in the simplest model to create and maintain a stable and adaptive lexical inventory, scale up to more complex environments and agent LCs; in the second model to multi-word utterances, and also to syntactic rules in the third model. Even though the models become more complex at every step, their performance in terms of communicative success remains high. Communication becomes more efficient, in the sense that while the linguistic mechanisms become more complex, the agents are able to express more using smaller lexicons. The fact that all models perform well and that they become more efficient according to criteria relevant to communication and cognitive capacities lends support to the hypothesis that language evolved in small steps rather than in one leap, as proposed earlier in linguistics.

In the three models, efficiency is measured using global measures. In a fourth model, we show that they can also serve as internal pressures in the agents that could guide the evolution process between stages. This model is a hybrid version of the two first models, in which the agents can choose between two strategies to use when they create an utterance. Both strategies are subject to pressure based on the agents' cognitive limitations and performance in communication. Experiments with the hybrid model show how agent-internal pressure on the strategies can lead to global coherent behaviour, where all agents agree on the communication strategy to use. Several efficiency criteria are looked at. We have at this point found two selection criteria that work reliably; however they depend on strategy selection being done probabilistically instead of deterministically like the mechanism used for lexicon lookup does. So while it seems that the same mechanism that is used for the evaluation of words and other linguistic constructions, can also be used to evaluate whole communication strategies, we have not yet identified the "ultimate" selection pressure. This result shows how the transition from simple to more complex communication system can have taken place, without needing recourse to genetic evolution.

Samenvatting [1]

In het verleden waren taal- en taalevolutiestudies dikwijls gericht op de individuele taalcapaciteit van taalgebruikers, om te proberen om in detail te begrijpen hoe deze taalcapaciteit werkt. Dit werd onder meer gedaan door mensen allerlei uitzonderingsgevallen voor te leggen, en hun reactie hierop te bestuderen. Dit is een goede benadering, maar ze gaat voorbij aan allerlei aspecten die relevant kunnen zijn om een globaal beeld van taal en zijn werking te krijgen. Bovendien werd er, toen de ingewikkeldheid van de taalcapaciteit duidelijk werd, geponeerd dat die genetisch moest geëvolueerd zijn, naar analogie met de complexiteit van de flora en fauna die in de natuur ontstaan zijn door evolutie. Dit standpunt is vervolgens wijd verspreid geraakt in de linguistische gemeenschap.

In recent onderzoek heeft men, specifiek met betrekking tot de evolutie van de menselijke taalcapaciteit, een andere benadering gekozen. Daarbij wordt geprobeerd om de belangrijkste kenmerken van taal te identificeren, en na te denken over de volgorde waarin deze kenmerken in de taalcapaciteit moeten verschenen zijn. In eerste instantie stelde de linguïst Bickerton twee stadia voor: een eenvoudige "prototaal" en moderne taal, met een genetische overgang tussen de twee. Recenter heeft zijn collega Jackendoff een veel gedetailleerder schema voorgesteld, waarin verschillende mijlpalen voorkomen die taal (en de taalcapaciteit) tijdens zijn evolutie heeft moeten bereiken.

Technieken die ontwikkeld werden in andere domeinen van de wetenschap worden ook steeds meer toegepast op taal. Van specifiek belang in deze context zijn speltheorie (uit de economie) en dynamische systemen (uit de fysica), omdat deze speciaal gericht zijn op systemen met veel componenten en de interacties tussen deze componenten. Taal kan gezien worden als een mooi voorbeeld van zo'n systeem, met veel individuen die interageren, en door deze interacties een taal creëren.

Binnen de informatica bestaat een paradigma dat toelaat om theorieën gebaseerd op deze benadering van taal op een elegante manier te testen: multi-agent simulaties. Individuele taalgebruikers worden gemodelleerd als agents, die elk talige expressies kunnen produceren en interpreteren. Deze agents interageren dan herhaaldelijk volgens een vast protocol, waarbij ze objecten en gebeurte-

[1] Zie ook (Vogt, de Boer en Van Looveren, 2000), (Steels, 2000) en (Belpaeme en Van Looveren, te verschijnen) voor uitgebreidere beschrijvingen van dit (en aanverwant) werk in het Nederlands.

nissen uit hun omgeving beschrijven. In de loop van zo'n reeks interacties ontwikkelen ze expressies om de betekenissen uit te drukken die ze vinden, tot ze uiteindelijk een volwaardig, bruikbaar communicatiesysteem hebben dat hun wereld dekt.

In deze thesis worden drie multi-agent modellen beschreven van drie verschillende linguistische communicatiesystemen, die ruwweg overeen komen met een aantal van de mijlpalen in Jackendoffs schema. In elk model hebben de agents specifieke cognitieve capaciteiten: in het eerste model kunnen ze enkel eenvoudige betekenissen uiten met expressies die uit één enkel woord bestaan. Het tweede model breidt dit uit met complexere, samengestelde betekenissen en de mogelijkheid om expressies bestaande uit meerdere woorden te gebruiken. De agents in het derde model kunnen betekenissen uitdrukken die referenties bevatten naar meerdere objecten en/of gebeurtenissen (en de relaties ertussen) in expressies die syntactisch gestructureerd zijn.

Elk model wordt volgens een aantal criteria geëvalueerd om na te kunnen gaan hoe het presteert: eenvoudig communicatief succes, maar ook meer kwalitatieve maten zoals de grootte van het lexicon, de coherentie van het lexicon, en de mate van homonymie en synonymie. We tonen aan dat hetzelfde soort dynamiek dat in het eenvoudigste model werkt om het lexicon te organiseren, ook werkt op grotere schaal: in het tweede model voor samengestelde expressies, en in het derde model ook voor syntactische regels. Ondanks het feit dat de modellen complexer worden in elke stap, blijft hun performantie in termen van communicatief succes hoog. De communicatie wordt efficiënter, in de zin dat hoewel de linguïstische mechanismen complexer worden, de agents toch meer kunnen uitdrukken met kleinere lexicons. Het feit dat alle modellen goed blijven werken en efficiënter zijn met betrekking tot een aantal communicatief en cognitief relevante criteria steunt de hypothese dat taal in kleine stappen geëvolueerd is, eerder dan in één grote stap, zoals in de linguistiek voorgesteld geweest is.

In de drie bovenstaande modellen wordt de efficiëntie gemeten aan de hand van globale maten. Dit wil zeggen dat de meetinstrumenten inzage hebben in de interne toestanden van alle agents. In een vierde model tonen we aan dat ze ook kunnen werken als interne selectiecriteria die het evolutieproces tussen de stadia kunnen sturen. Dit model is een hybride versie van de eerste twee modellen, waarin de agents kunnen kiezen uit twee strategieën om expressies te produceren. Beide strategieën zijn onderworpen aan de selectiedruk opgelegd door de cognitieve beperkingen van de agents en hun performantie in communicatie.

De experimenten met het hybride model tonen hoe agent-interne druk op de strategieën kan leiden tot globaal coherent gedrag, waarbij alle agents dezelfde communicatiestrategie gebruiken. Verschillende efficiëntiecriteria worden onder de loupe genomen. We hebben op dit moment twee selectiecriteria gevonden die betrouwbaar lijken te werken. Ze steunen echter op het feit dat de communicatiestrategieën probabilistisch gekozen wordt, i.p.v. deterministisch zoals het opzoekmechanisme van het lexicon doet. Desalniettemin lijkt het er toch op dat hetzelfde mechanisme dat instaat voor de evaluatie van woorden en andere linguistische constructies ook gebruikt kan worden om hele communicatiestrategieën te evalueren. Dit resultaat toont hoe de transitie van eenvoudige naar complexere communicatiesystemen plaatsgevonden kan hebben, zonder genetische evolutie nodig te hebben voor de verklaring.

Contents

1 Introduction **1**
1.1 The Problem . . . 2
1.2 Methodology . . . 4
1.2.1 Multi-Agent Systems . . . 4
1.2.2 Successive Stages . . . 5
1.2.3 Self-organisation . . . 7
1.3 Language Games . . . 9
1.3.1 Agents . . . 11
1.3.2 History . . . 13
1.4 Evaluation of the Models . . . 15
1.4.1 Communicative Success . . . 16
1.4.2 Lexical Coherence . . . 16
1.4.3 Lexicon Size . . . 16
1.4.4 Synonymy and Homonymy . . . 17
1.5 Outline of the Thesis . . . 17

2 Simple Naming Game **19**
2.1 Related Research . . . 20
2.2 Simplest Experiment . . . 22
2.3 Semantics . . . 23
2.3.1 First Experiment . . . 24
2.3.2 Discrimination Game . . . 24
2.3.3 Algorithm . . . 27
2.3.4 Discussion . . . 27
2.4 Form . . . 29
2.5 Variants . . . 31
2.5.1 Stochasticity . . . 31
2.5.2 Talking Heads . . . 33
2.6 Summary . . . 37

3 Simple Naming Game: Experiments **39**
3.1 Meanings Are Referents . . . 39
3.1.1 Naming Game Algorithm Details . . . 39
3.1.2 Communicative Success . . . 41
3.1.3 Coherence . . . 42

3.2 Discrimination Game 44
3.2.1 Discrimination Algorithm Details 44
3.2.2 Naming Game Algorithm Details 46
3.2.3 Success 46
3.2.4 Coherence 46
3.3 Coherence Revisited 50
3.3.1 Vocabulary Coherence 54
3.3.2 Other Types of Coherence 55
3.3.3 Ontology Coherence. 56
3.4 Talking Heads 56
3.4.1 Communicative Success 56
3.4.2 Coherence 56
3.5 Summary 60

4 Multi-Word Naming Game **63**
4.1 Related Research 64
4.2 Semantics 65
4.2.1 Discrimination Game 65
4.2.2 Predicate calculus 65
4.3 Form 69
4.3.1 Explicit Subform Boundaries 69
4.3.2 No Explicit Subform Boundaries 74
4.4 Evolutionary Transition 75
4.4.1 Game Result 77
4.4.2 Lexicon Size 77
4.4.3 Lexicon Expansion 78
4.5 Summary 78

5 Multi-Word Naming Game: Experiments **81**
5.1 Basic Experiments 81
5.1.1 Discrimination Game Algorithm Details 81
5.1.2 Predicate Semantics Algorithm Details 81
5.1.3 Multi Word Naming Game Algorithm Details 82
5.1.4 Communicative Success 82
5.1.5 Coherence 83
5.1.6 Lexicon Size 83
5.1.7 Efficiency 84
5.2 Competition between Words 88
5.2.1 Synonyms and Homonyms 90
5.3 No Explicit Subform Boundaries 90
5.4 Evolutionary Transition 93
5.4.1 Dual Strategy Algorithm Details 93
5.4.2 Results 93
5.4.3 Discussion 97
5.5 Summary 99

6 Simple Syntactic Naming Game 101
6.1 Related Research 102
6.2 Decreasing Ambiguity: Illustration 105
6.2.1 Meaning of an Utterance 105
6.2.2 Composite Meanings 106
6.2.3 Context 106
6.2.4 Syntax 108
6.3 Semantics 109
6.3.1 Specialists 110
6.3.2 Combining specialist data 111
6.3.3 Processes 112
6.4 Form 115
6.4.1 Rules & Constructions 115
6.4.2 Example 119
6.4.3 Implementation 119
6.5 Discussion 121
6.5.1 Categories 121
6.5.2 Syntactic Devices 122
6.6 Summary 123

7 Simple Syntactic Naming Game: Experiments 125
7.1 Basic Experiments 125
7.1.1 Predicate Semantics Algorithm Details 125
7.1.2 Simple Syntactic Naming Game Algorithm Details 126
7.1.3 Communicative Success 127
7.1.4 Coherence 129
7.1.5 Lexicon Size 132
7.1.6 Grammar Use 132
7.2 Word Competition 134
7.2.1 Synonyms and Homonyms 136
7.3 Summary 136

8 Conclusion 139
8.1 Models 139
8.2 Results 140
8.2.1 Individual Models 140
8.2.2 Global Interpretation 141
8.3 Future Research 142

A Measures 145
A.1 Communicative Success 145
A.2 Coherence 147
A.2.1 Lexical Coherence 148
A.2.2 Grammatical Coherence 151

B Selection Pressure: Experiments 153

List of Figures

1.1 Succession of the models described in this thesis. 5
1.2 Different layers in agents and the population as language transfers from generation to generation. 6
1.3 Jackendoff's proposal for incremental steps in the evolution of language (Jackendoff, 2002). 8
1.4 General structure of a "cognitive module." 12

2.1 Semiotics in the basic experiment. 23
2.2 Example of two discrimination trees. 25
2.3 Pseudo-code for the discrimination algorithm. 28
2.4 Semiotic triangle in experiments with semantics. 29
2.5 "Pipeline" of the single-word utterance system. 30
2.6 Discrimination trees as developed by the agents in the Talking Heads experiment. 36
2.7 *Talking Heads* installation layout; schematically (i) and actual (ii) . 36
2.8 An example of the distortion faced by the Talking Heads software; the whiteboard is parallel to the "camera plane." 36

3.1 Success and coherence in a 10-agent, 10-object experiment (5,000 games; averaged over 10 runs). 43
3.2 Success and coherence in a 20-agent, 10-object experiment (15,000 games; averaged over 10 runs). 43
3.3 Success and coherence in a 20-agent, 20-object experiment (30,000 games; averaged over 10 runs). 47
3.4 Success and vocabulary coherence in a 10-resp. 20-agent experiment, with 7 objects in the context (50,000 resp. 100,000 games; averaged over 10 runs). The fact that the success curve is severely jagged in both graphs even after being averaged over 10 experiments, is due to the short intervals over which communicative success has been measured. 47
3.5 Competition between words for meaning WIDTH in a population of 2 agents. 49
3.6 Competition between words for meaning HEIGHT in a population of 5 agents. 49

3.7 Competition between words for meaning GRAY-LEVEL in a population of 10 agents. 51
3.8 Competition between meanings for word "bee" in a population of 2 agents. 51
3.9 Competition between meanings for word "gedale" in a population of 2 agents. 52
3.10 Competition between meanings for word "soopeeki" in a population of 10 agents. 52
3.11 Percentual occurrence of meanings (most frequent meanings on the left) in a 10-agent, 7-segment experiment (50,000 games). . . . 57
3.12 Success and ontology coherence in a 10- resp. 20-agent, 7-segment experiment (50,000 resp. 100,000 games; averaged over 10 runs). . 57
3.13 Communicative Success in the Talking Heads experiment. 58
3.14 Rate of appearance of new agents in the Talking Heads experiment. 58
3.15 Meaning-word coherence in the Talking Heads experiment. . . . 59
3.16 Evolution of the lexicon for the meaning *green* in the Talking Heads experiment. 60
3.17 Percentage of interactions covered by the most frequently used meanings in the Talking Heads experiment. 61

4.1 An example of a generation tree. Note that the operations in the tree use the actual LISP representation. 67
4.2 Pseudo-code for the generator. 70
4.3 "Pipeline" of the multi-word utterance system 72
4.4 Pseudo-code for interpretation and production 73

5.1 Rate of communicative success measured over 1000 language games played by two multiple word agents. 83
5.2 Success and coherence in a 10-agent experiment (20,000 games; averaged over 10 runs 84
5.3 Lexicon Sizes for Single-Word and Multi-Word experiments (both 10 agents, 7-segment meanings, 35,000 games; averaged over 10 runs) 85
5.4 Efficiency in a multi-word agent. 86
5.5 Efficiency in a single word agent. Note that both curves coincide in this graph. 86
5.6 Number of meanings and words in a single-word naming game with 10 agents (10,000 games; averaged over 10 runs). 87
5.7 Number of meanings and words in a multi-word naming game with 10 agents (20,000 games; averaged over 10 runs). 88
5.8 Word competition in a 2-agent and a 5-agent experiment; meaning WIDTH in both cases. 89
5.9 Word competition in a 10-agent experiment; meaning GRAY-LEVEL. 89
5.10 Meaning competition in a 10-agent experiment; word "lopoo". . . 92
5.11 Lexicon size evolution in the single-form multi-word naming game. 92
5.12 Game Result—Absolute Calculation. 95

5.13 Game Result—Positive and Negative Feedback. 96
5.14 Lexicon Size—Absolute Calculation. 96
5.15 Lexicon Expansion—Positive and Negative Feedback. 97
5.16 Lexicon Expansion—Negative Feedback Only. 98

6.1 Schematic representation of process dependencies in the process engine. 113
6.2 Pseudo-code for the process engine. 114
6.3 "Pipeline" of the simple syntactic system. 120

7.1 Communicative success in a Simple Syntactic Naming Game (averaged over 10 experiments). 128
7.2 Percentage of single-word interactions that was successful (averaged over 10 experiments). 128
7.3 Fraction of failures in `introduce-new-words`. 129
7.4 Form-meaning coherence in a 3-agent experiment. 130
7.5 Form-meaning coherence in a 5-agent experiment. 131
7.6 Grammar coherence in a 3-agent experiment. 131
7.7 Grammar coherence in a 5-agent experiment. 132
7.8 Evolution of lexicon size. 133
7.9 Evolution of grammar use. 133
7.10 Competition between words for BOX in a 2-agent population. . . 135
7.11 Competition between words for TINY in a 5-agent population. . . 135
7.12 Competition between words for MOVE in an 8-agent population. . 138

A.1 Communicative Success. 146
A.2 Communicative success. 146
A.3 Success in all games and multi-word games. 147
A.4 Coherence data from the Talking Heads experiment. 150
A.5 Grammar coherence in a 5-agent experiment. 152

B.1 Game Result—Absolute Calculation. 154
B.2 Game Result—Positive and Negative Feedback. 155
B.3 Lexicon Size—Absolute Calculation. 156
B.4 Lexicon Expansion—Positive and Negative Feedback. 157
B.5 Lexicon Expansion—Negative Feedback Only. 158

Introduction

NO OTHER animal species seems to have a communication system of the same level of complexity as human language. There have been several attempts to teach human language to man's closest relatives, the great apes (Terrace, 1987; Savage-Rumbaugh et al., 2001). However, they seem to have stranded at vocabularies of a few hundred symbols, and a few grammatical rules that are applied with some level of consistency. What subset of language the animals learned, they learned at the expense of considerable effort on both the trainers' and the animals' part. Little spontaneous learning seems to have taken place, in contrast to human children for whom learning to speak comes naturally and seemingly effortlessly. To be sure, the fact that the animals were able to learn a subset of language in itself is impressive, but what they learned is still nowhere near the complexity of the language humans use daily and effortlessly. Additionally, even if some animals or species would be able to learn language, it does not explain why they do not do so spontaneously.

Because language seems to be so unique and defining to humans, one of the big, open problems of linguistics is how language actually came to be. The primary form of language is spoken language; written language was developed only comparatively recently. Consequently, the early history of language is not traceable, and when written language emerged, language had already become mature. This has not stopped many researchers from attempting to explain the origins of language anyway. Many a theory has been launched in the past one or two decades. The authors of these theories have come up with several ways to try to counter the lack of factual evidence they faced.

Several theories focus on precursors to language, upon which modern language could have been built: imitations of sounds occurring elsewhere in the world (onomatopoeia), involuntary sounds that come with expressions of emotion, music, or a more structured use of gestures. These hypotheses usually go (intentionally or not) by names that are not very confidence-inspiring, such as the "ding-dong" theory, the "bow-wow" theory or the "pooh-pooh" theory. The problem with this kind of theories is that they are impossible to verify; they cannot be proved right, but they cannot be proved false either. From a scientific point of view, such theories are not very interesting.

Another string of theories picks a factor of early human life and proposes that it acted as a catalyst for language to emerge; in turn language would strengthen the factor that caused it to emerge, and in that way secure its persistence and development. An example of such a theory is the theory that language and conversation replaced (physical) grooming as a way to more efficiently create and maintain bonds between people, to allow for an increase in the size of the groups in which humans live (Dunbar, 1998). While this is an interesting theory backed up with evidence, the fact that it puts the burden of language origins on a single aspect of human life makes it likely to be incomplete.

Still other theories try to reverse-engineer the history of language by going back one "generation" of language at a time. Already in the 19th century, the Neogrammarians developed techniques to reconstruct the phonetic system from a parent language based on the phonetic systems of its daughter languages.[1] The Neogrammarians used this method to establish the relations between comparatively young languages, e.g. to reconstruct how the Romance languages developed from their common ancestor, Latin. Recently, some people have tried to take this back much further, trying to reconstruct the "very first" language based on core lexical items from languages from all over the world (Ruhlen, 1996). However, this approach assumes implicitly that the users of the languages that are reconstructed in this way have the same grammatical, semantic and pragmatic competences as contemporary language users, and hence does not allow us to go back to the time *before* these capabilities were fully developed in the species.

These are just a few of the attempts that have been made to shed light on the history and origins of language, and to cope with the lack of factual evidence. However, several fundamental issues have not been addressed satisfactorily by these theories, and these issues define an agenda for our research.

1.1 The Problem

We want to find an explanation for the gap that exists, and for which no empirical evidence is available, between animal communication systems and modern language. Somehow, a competence to learn a complex system like modern language must have arisen along the way. There are two important dimensions along which the discussion has been developing: the nature/nurture debate, and whether language evolved gradually, or as the result of a sudden change.

The presence of an innate endowment for language has been a much-debated topic. Chomsky (1965) speaks of a *language acquisition device* that embodies a "universal grammar," which is a device that contains the algorithms for language processing fully laid out, with just a few parameters to set based on linguistic input in the beginning of a child's life. Others have argued that no special cognitive apparatus is needed at all to learn language; general purpose learning algorithms already in place for other cognitive tasks can do the trick.

As for the gradual/sudden debate, the linguist Bickerton (see e.g. Bickerton, 1990) argued for a protolanguage stage before modern language, which already went beyond animal communication systems. Initially, his aim was to argue for a sudden transition from protolanguage to modern human language. However, based on Bickerton's work, Jackendoff (2002) offers a much expanded set of

[1] See Lehmann (1967) for a collection of important writings by the Neogrammarians.

hurdles that language must have taken at some point during its development.
We think that the discussions along both of these dimensions do not exclude each other. Maynard-Smith and Szathmáry (1995) in evolutionary biology, for example, argue for a biological endowment for language, and for a gradual evolution of it. In any case, in order to have a complete picture of the origins and development of language, both debates need to be settled.
In this thesis, we do not take a position in the debate on biological endowment, but rather take an opposite approach. We try to imagine what mechanisms are needed to reach a certain level of linguistic competence, and once these mechanisms are known, it can be decided in how far they are specific to language or not.
In the gradual/sudden debate, we choose the side of continuous evolution of language, independent of its substrate. This is the basic assumption that underlies the work in this thesis. In order to support this position, there are several issues that all need to be addressed in a satisfactory way:

1. Communication systems of all intermediate complexities have to be possible. Moreover, they must be successful, according to a set of criteria that embody the constraints to which the communication systems must conform.

2. There must be a path that connects each communication system to the next one. Moreover, every "next system" must be more successful than the previous one according to the criteria.

3. The population of language users must be able to make each transition on its own. That is, there should be no central direction of the evolution, or in other words, the pressures that steer the evolution must be local to the language users.

Attempts at providing insight in these issues have been done; for example, Jackendoff (2002) tries to answer the second question by sketching a road map that contains different possible milestones along the path from communication systems of a complexity common in the animal kingdom to human language.
In this thesis, we will try to provide a number of more extensive starting points to answers for each of these questions. More concretely, we offers a detailed explanation of three different computational models, of increasing computational and linguistic complexity and competence. These models can be considered to be "snapshots" of three points in time along the path between animal communication and modern language. The models are each evaluated against a number of performance criteria, such as communicative success, lexical coherence and lexicon size. These three models can be seen as a (partial) answer to the first question.
Additionally, the path from the first model, the Single-Word Naming Game (chapters 2 and 3) and the second model, the Multi-Word Naming Game (chapters 4 and 5), will be studied using a hybrid model. In this model, the language

users can “choose” to use either strategy, according to selection pressures that use only local information, and originate inside the agents themselves. This model can be seen as a (partial) answer to both questions two and three.

The following sections will explain in detail what the methodology is that is used for all the models and the assumptions behind it (section 1.2), describe in more detail the language game framework (section 1.3), and explain what the criteria are for evaluating the models (section 1.4).

1.2 Methodology

The method we adopt in this thesis is *synthetic construction* (Steels, 1997, 1998b). Instead of being able to rely on empirical evidence (such as fossils in archaeology) and building theories based on them, the lack of empirical evidence prompts us to build models of what could have happened, and use those as a starting point for developing a scenario of the events as they may have unfolded themselves.

Of course, the models we build are not without foundation, despite the lack of empirical evidence. Their design is defined by several considerations.

1.2.1 Multi-Agent Systems

Language is spoken in groups. Children that are born into a community always learn the language that is spoken in that community; it seems inconceivable that children could choose *not* to learn the language that his or her community members speak. The community aspect of language seems to be a core phenomenon of it.

The way in which this social aspect can be embedded in a model, is by shaping it as a multi-agent system. The philosophy of multi-agent systems is that different, autonomous entities perform tasks to arrive at a global optimum. This maps nicely on the concept of language as a sociological phenomenon.

The thesis offers four models of communication systems in different stages of complexity: a model in which the language users exchange single words, a model in which the language users are capable of joining several words and their meanings together to talk about the same referent, and a third model in which the language users are capable of using a limited form of syntax to structure their utterances even more, and talk about relations between different referents. The fourth model is a combination of the first two, where the agents decide which strategy they use to communicate.

Of course, nowhere is it claimed that these models accurately represent past stages of language. Rather, they serve as “proofs of concept” that such intermediary communication systems are viable, and that moreover, each system can represent a genuine improvement over earlier, less complex systems with respect to a number of criteria, that we deem important for a communication system.

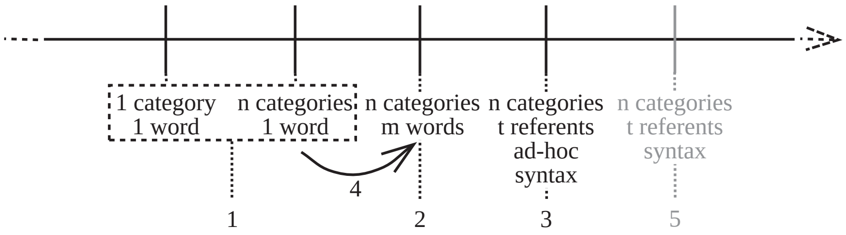


Figure 1.1: Succession of the models described in this thesis.

1.2.2 Successive Stages

Each of the three models can be viewed as a snapshot of a moment in time during the evolution of language, where the agents have a specific set of cognitive abilities, which in turn results in a communication system with specific features. Figure 1.1 shows how the three models succeed each other, and briefly what the innovations are that make them different. In the figure, the models are grouped on the basis of their syntactic capabilities, where "syntactic" refers to the structural properties of the utterances produced. Hence, model 1 embraces both the case in which single categories are assigned to words and the case in which composite categories are assigned to words. Models 3 and 5 on the contrary are not grouped together, because the step from ad-hoc syntax to "real" syntax represents a major leap (see section 6.5.1). In fact, model 5 would implement fully-functional grammar. We do not concern ourselves with this model in this thesis, but see (Steels, 2004).

The use of the terms "successive stages" could point both to a gradualist view of the evolution of language, with evolution progressing gradually, and the models being snapshots, or to a more punctualist view (Eldredge and Gould, 1972), with periods of stability and periods of fast change, where the models would represent the status quo in longer periods of stasis.

Language Evolution

In fact, before committing to either view, we should be clear on what "evolution of language" means exactly. Often "evolution" is interpreted on a biological level. This resonates well with linguists, because language would seem to be the product of a biological system: the brain. In particular, linguists have postulated the existence of a dedicated "Language Acquisition Device" in the brain. This LAD is a black box incorporating a Universal Grammar, which is a generalised grammar that can be configured to any particular grammar for a human language, by setting a number of parameters to true or false. The perceived complexity of such a device led to a scenario in which language came about in a catastrophic way, possibly even as a result of just a single gene change (Bickerton, 1998). Bickerton based his ideas on the observation that, in several instances, a fully complex "modern" language (a creole) seemed to have been

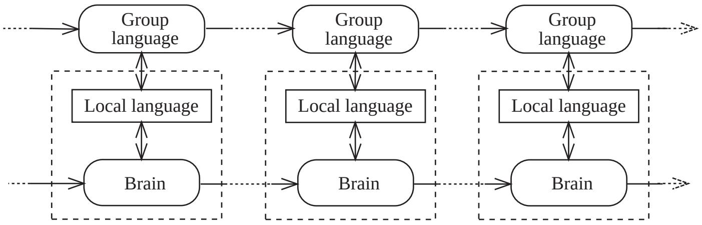


Figure 1.2: Different layers in agents and the population as language transfers from generation to generation.

created from a communicatively challenged pidgin.

The LAD hypothesis suggests that all individuals learn exactly the same language. It subscribes to the view of an ideal speaker-hearer, which knows "the language." But in fact, "the language" is not the product of one single individual. Rather, it is some sort of common denominator of all the individual languages, which encompasses a core that is virtually identical in all the individual languages, and feathers out into lesser used constructions and jargon that are understandable only to increasingly smaller subsets of the population of language users. Consequently, the language at the group level is to some extent detached from what happens at the individual level, and language can be considered to be evolving in its own right (Mufwene, 2002).

The interaction between the group language and the individual languages goes both ways. While the group language is based on the individual languages, whenever a new individual is born into the population it will have to reconstruct its individual language from the group language because it has no access to the contents of the other individuals' brains. Figure 1.2 shows this graphically.

The fact that brain and language are decoupled opens up the possibility that, while both the brain (on a biological level) and language (on a cultural level) evolve, they may do so at different speeds. It is conceivable that, while individual brains develop faster, language develops slower, because no communicational pressure requires language to develop faster. Conversely, language may develop faster while brain development stagnates, when the brain has cognitive capabilities that could be recruited for language without biological change. Lastly, individuals with different cognitive capabilities may share the same communication system, and communicate successfully, despite possible different internal representations.

Another implication of the weak coupling between the two is that the evolutionary regimes at the biological and the cultural level may havebeen totally different. Language may have evolved gradually, while biological evolution may have been punctuated; it may have been the other way around; or both may have evolved gradually or punctuatedly together. More importantly, we will not be able to predict how either evolved from knowing how the other evolved.

As a result, we will not attempt to answer the question of whether "language evolution" was gradual or not. Even when the level at which we are speaking is specified, the available evidence does not allow conclusions either way. Therefore, we will use the term "language evolution" only on the more general level of denoting the shift or the tendency of language to become more sophisticated over time, with respect to a certain set of selection criteria.

Milestones

Even if we cannot at this point paint a global image of how language on the one hand and its biological substrate on the other hand evolved over time, we can still try to pinpoint a number of crucial phases that language users must have passed through at some point during the evolution of language.

Undoubtedly without realising it, Bickerton (1990) was the first to make a proposal in this direction. Although his two-stage scheme was designed to argue for a catastrophic event leading to the transition between protolanguage and modern language, it provided the inspiration for Ray Jackendoff (2002) to follow up on Bickerton's scheme by proposing a fairly detailed set of milestones that language must have reached at some point in its development. The schema is reproduced in fig. 1.3.

The three models around which the thesis is built, map relatively well onto three milestones in Jackendoff's proposal. Briefly, the Single Word Naming Game model (SNG) represents a holistic communication strategy, that uses arbitrary symbols and permits large lexicons. It corresponds roughly to two stages in Jackendoff's diagram: *use of symbols in a non-situation-specific fashion*, and *use of an open, unlimited class of symbols*. The Multi-Word Naming Game (MWNG) model represents the transition to a compositional strategy, where each utterance is the conjunction of the forms and meanings of each part. This model corresponds to the *concatenation of symbols* step in Jackendoff's diagram. Finally, the Simple Syntactic Naming Game model augments the simple compositional strategy of the MWNG with more complex ways to structure an utterance. These new ways to structure an utterance correspond more or less to Jackendoff's *use of symbol position to convey semantic relationships*.

1.2.3 Self-organisation

Looking again at fig. 1.2, we can say that the group language *self-organises* from the individuals' lexicons. The classic example of a group phenomenon where the interactions between the members of the group cause global behaviour is the example of self-organising ant behaviour. A colony of ants is capable of efficiently coordinating their food transports, despite the fact that ants are very simple insects and that there is no coordinating ant (Goss et al., 1989). How do they do this? Upon close inspection it turns out that ants *need* only a few simple behaviours to cause this globally complex behaviour. (1) When an ant roams around looking for food, it leaves behind a chemical trace of pheromones. (2) An ant that comes across pheromones when it roams around, will be attracted

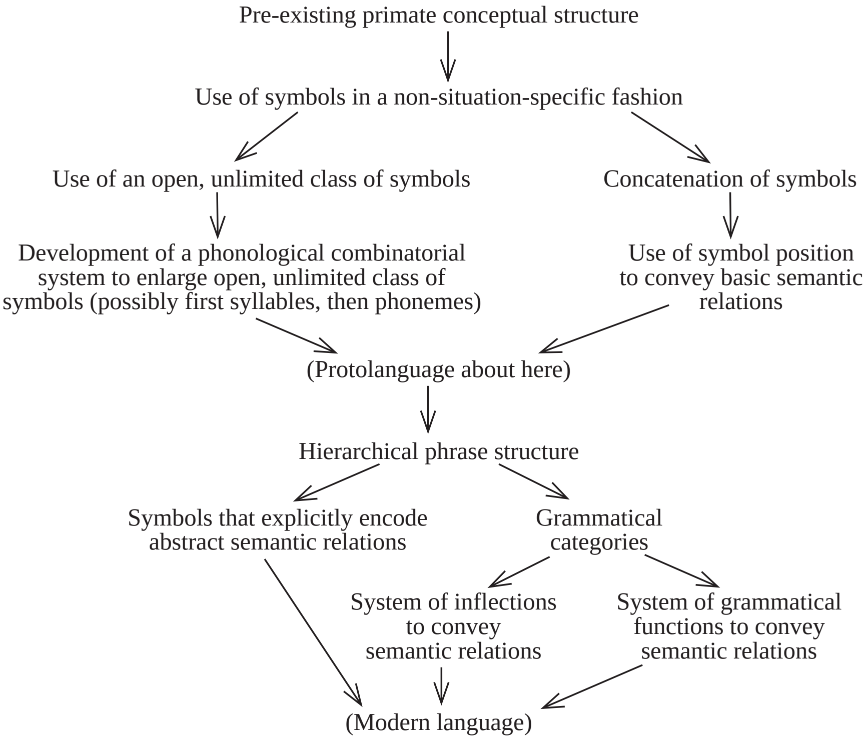

Pre-existing primate conceptual structure
Use of symbols in a non-situation-specific fashion
Use of an open, unlimited class of symbols
Concatenation of symbols
Development of a phonological combinatorial system to enlarge open, unlimited class of symbols (possibly first syllables, then phonemes)
Use of symbol position to convey basic semantic relations
(Protolanguage about here)
Hierarchical phrase structure
Symbols that explicitly encode abstract semantic relations
Grammatical categories
System of inflections to convey semantic relations
System of grammatical functions to convey semantic relations
(Modern language)

Figure 1.3: Jackendoff's proposal for incremental steps in the evolution of language (Jackendoff, 2002).

by the pheromones, and follow the pheromone trail with a certain probability. (3) The more pheromones a path has, the higher the attraction will be. These simple three simple behaviours cause the paths to the food to become richer and richer in pheromones, so that more ants will be attracted to them and follow them. These ants in turn deposit pheromones, which further strengthen the paths. It is remarkable that the strongest paths also turn out to be the most efficient ones.

The global behaviour is thus not caused by any individual ant, but by the simple, individual behaviours of all the participating ants. The whole is more complex than the sum of the behaviours of the individual ants. This type of self-strengthening interaction between the behaviours of the individual members of a group is called *self*-organisation, because there is no central entity coordinating the members. It is the simple, individual behaviours that lead to the global, "intelligent" behaviour of the group.

For language, when a new individual enters the population, it will monitor the group language that is in place, and try to extract the features such that its own attempts at communication will be as successful as possible. If we now imagine a situation in which there are individuals but no group language, those same mechanisms will make sure that the individual languages expand and converge towards each other. Self-organisation here means thus that the individual actions that an individual takes to make its own language conform to the group language also produce a coherent group language when there is none.

One question that the models have to answer is then: what are the simpler behaviours that are at work in each individual? The models we build serve on the one hand as concrete implementations of theories about the behaviours of individuals, but also (assuming the models work as expected) as a reinforcement of the theory. When the models exhibit behaviour that corresponds to the behaviour of natural language in similar circumstances, this increases the probability of the theoretical assumptions being correct.

1.3 Language Games

The models presented in this thesis have all been developed within a specific type of multi-agent model: *language game* models (Steels, 1996a). Language games are formalized interactions between individuals in a population, cfr. Axelrod (1984); the individuals are modeled as *agents*. The task of the agents in a language game is for the speaker to accurately describe a referent from the environment in which the agents live, and for the hearer to use the speaker's utterance to identify this same referent in the environment. The goal of a series of language games is to accomplish this task solely using linguistic means.

A language game proceeds briefly as follows. A speaker and a hearer are chosen from a population of simulated language users. The speaker selects a topic from the environment, finds a meaning that distinctively describes it, and produces an utterance that encodes this meaning. The hearer hears the utterance, tries to decode it into its meaning, and based on the meaning found, points out the

0	Select a speaker and a hearer from the population.	
	Speaker	**Hearer**
1	Perceive the environment as a set of objects $E_s = \{o_{s,1}, \ldots, o_{s,n_s}\}$.	
2	Choose a topic $t_s \in E_s$.	
3	Find a meaning m_s for t_s.	
4	Construct an utterance u for m_s.	
5	Utter u.	
6		Perceive the environment as a set of objects $E_h = \{o_{h,1}, \ldots, o_{h,n_h}\}$.
7		Reconstruct the meanings for u: $M_h = \{m_{h,1}, \ldots, m_{h,p_h}\}$
8		Calculate score $s_{h,t}$ for4 each meaning $m_{h,t}$.
9		Find the meaning m_h with the highest score.
10		Retrieve the referents $R_h \subset E_h$ filtered by m_h.
11		If $\#R_h = 1$, then $r_h \in R_h$ is proposed as the topic.
12	If $r_h \equiv t_s$, declare *success*.	

Table 1.1: General language game protocol.

object from the environment that it thinks is the topic. The speaker will then agree or not, which determines the outcome of the game. Table 1.1 gives a more formal description of the protocol.

This specification still leaves a lot of room for variation when actually implementing language games. For example, there are countless ways to choose two agents from the population. One could take all agents in order from the population, or randomly, or one could distribute the agents in a space and have frequent interactions within clusters of nearby agents and few interactions between agents of different clusters. There are many ways in which the agents' perception could be implemented; choosing a topic can be done in many ways, etc. Some of these degrees of freedom are (or turn out to be, later) less important to the model, while others can be critical.

These types of variations allow the experimenter to retain a considerable degree of freedom to explore the aspects of language he finds most interesting, while still adhering to the global philosophy of using language games.

1.3.1 Agents

The focus in our models is on the semantic and syntactic capabilities that individual agents need to reach a certain level of communication. Therefore, we assume that each agent already has certain capabilities (Steels et al., 2002), and do not concern ourselves with how these capabilities can have evolved.

We assume that the individuals in our models have a desire or drive to communicate. We do not ask the question why the individuals would want to communicate. This question has been (and is being) researched in other work, such as (Cangelosi, 2001; Quinn, 2001).

Similarly, we do not at this point ask ourselves how a phonetic coding can arise. Again, other researchers are working on this (de Boer, 2001; Oudeyer, 2003), and we assume our agents have such a system at their disposal.

Also, we assume that the agents have a non-linguistic way of communicating, namely pointing, to draw attention to an object in the environment.

Cognitive Capabilities

The capabilities that we vary in the different models presented in the thesis include a perception module, which takes sensor values recorded from events in the outside world (such as camera images), and extracts useful information from them. For example, a camera image will be segmented into salient regions, and for each of these regions a set of features will be measured to describe/represent it, such as colour, position in the image, size, etcetera. Of course, in simulation the image route need not be taken explicitly; the region or object descriptions may be generated directly.

Another module, the semantic module, will accept this data, and try to extract a description that is unique for one of the perceptual elements. This one element is called the topic.

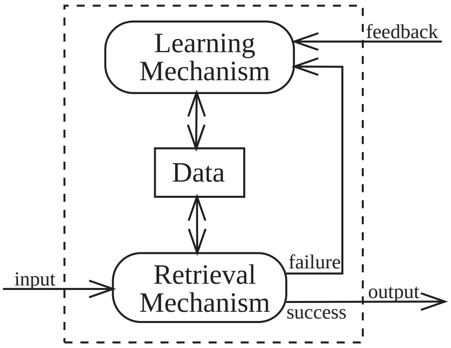


Figure 1.4: General structure of a "cognitive module."

Finally, this meaning description is converted into an utterance, which will be processed by another agent.

This brief description of the structure of the agents in our experiments are what we usually call the "cognitive capabilities" of our agents. We have no intention of claiming that our agents are capable of complex reasoning or any other difficult mental tasks, but we do believe that things like perception, meaning creation and utterance creation are cognitive activities even if they are performed automatically and unconsciously.

Modules

In terms of implementation, the modules described above all share the same basic construction plan. Every module revolves around a data structure that represents the module's internal data. In the case of a lexicon for example, the internal data structure is simply the list of associations between words and meanings that the lexicon module knows at that specific point in time. There are two mechanisms that use or modify this data structure: a retrieval mechanism, which always tries to find the best solution according to the input data, and a learning mechanism that can add to or modify the data structure. Figure 1.4 shows the general layout of a module. As an example, the retrieval mechanism for a lexicon might simply be a linear search that finds the best word for a certain input meaning (in a simple, single-word naming game), or it might use complex combination criteria and several searches in the data structure to compose a set of words that most accurately covers the input meaning. Crucially, the retrieval mechanism must also be capable of working "in reverse:" the lexicon must for example be capable not only of looking up words for a certain meaning, but also the meaning(s) associated to some word.

The learning mechanism is triggered whenever the output of the retrieval process is deemed inadequate by the process(es) that use(s) the data from the module. If a rendering process decides that the output from the lexicon module is not good enough because it could only partially render an utterance for a meaning, the lexicon will be signalled to extend its data structure with a new association for the part of the meaning that was not covered. Likewise in interpretation, if the lexicon returns a certain meaning for some word, and interpretation fails, the lexicon may be prompted to change the association, or at least bias the interpretation of that particular word to some other meaning.

1.3.2 History

The language game framework already has a long history at the AI-lab of the Vrije Universiteit Brussel. It began with the development of three different, separate models at more or less the same time: the Discrimination Game, the Naming Game and the Phonetic Imitation Game.

The Discrimination Game (DG) (Steels, 1996b) was geared specifically towards studying the development and acquisition of perceptual distinctions, or more concretely, how an individual could learn to categorise the objects in its environment. It studied the agent-internal phase of meaning creation, and the influence of the environment in that process.

The purpose of the Naming Game (NG) (Steels, 1996a,c) was to study how a group of individuals, modeled as a multi-agent system, could reach a consensus about a set of conventions. The conventions in question were names for the agents themselves—hence *Naming Game*. The NG thus studied the process of translating meanings into utterances, and building a reliable and coherent system for exchanging these utterances between different agents.

The Phonetic Imitation Game (PHIG) (de Boer, 1997), finally, studied how agents can build a system for exchanging these utterances. People use spoken language, and this requires translating utterances into a form that carries well through air. The agents thus need to agree on a system of sounds. De Boer built a system in which the agents produce and imitate vowel sounds, and shows that the agents agree on systems that look remarkably like real vowel systems.

These three experiments conceptually cover a large part of linguistics, with models for several key aspects of language. In each of these areas, they attempted to show that factors outside of those that were covered by mainstream linguistics, such as the dynamics of the interactions between the agents, can play an important role in developing language. Especially the DG and the NG spawned several threads of further research.

Meaning

One of the problems with meaning and making sense in artificial systems is the *symbol grounding problem* (Harnad, 1990). In symbolic systems, the ultimate meaning of what goes on in the system is not in the system itself, but in the mind of the researcher observing the system. Therefore, systems need to be *embodied*, i.e. have a "body" that can interact with an external world, and provide perceptual feedback from this world. The world will then contribute to shape the system's internal representations, and conversely the system can use the structure it finds in the external world to its own advantage.

In order to solve the symbol grounding problem in the context of the Discrimination Game, it was implemented on a robot that roams around in an environment that it can perceive. The values of the robot's sensors are used to represent the sensory channels that the discrimination game uses to categorise, and the meanings that result from the DG are used by the robot to make decisions about the actions it will take in its environment, e.g. avoid obstacles or follow another robot (Steels and Vogt, 1997; Vogt, 2000; Steels and Kaplan, 2002). The Talking Heads experiment (see also section 2.5.2) also represents an instance of embodiment, based on the Discrimination Game.

The DG evolved also into other, more sophisticated approaches to construct a sensible categorisation of the world. De Jong and Vogt independently developed different algorithms (the Simple Prototype method and the Adaptive Subspace Method, respectively). They were compared in (de Jong and Vogt, 1998).

Another method of categorisation was used by Belpaeme: he used Radial Basis Functions in his colour categorisation experiments (Belpaeme, 2002; Steels and Belpaeme, submitted). This method is more akin to prototype-based methods.

Finally, not developed separately but as part of models in which the agents can use complex categories and utterances, a limited form of predicate calculus is used as semantics. This type of semantics will be explained in detail in chapters 4 and 6 of this thesis.

Form

The Naming Game too has given rise to several variants and further developments. The basic model did not use a separate meaning step; the meanings were stored with the words in the lexicon and directly comparable to the objects in the environment. This made it possible to focus specifically on the dynamics of lexicon learning in a population, without the added complication of agreement on meaning.

Later, the NG was integrated with the DG to incorporate meaning in the system (Steels, 1998a). This gave rise to a number of variants of the basic NG model. Vogt's experiments with robots (see above) included the NG and used actions as meanings.

Experiments were also done with models in which several components were replaced by stochastic variants, which simulated uncertainty in communication. For example, it is not always clear to which object a speaker points when it points to the topic it chose for the language game (Steels and Kaplan, 1998). The Talking Heads were the most advanced version of this type of naming game, sporting a version of the naming game that was augmented with perception through real cameras, a population distributed spatially over several sites, and stochastic pointing through camera pan and tilt motion (Steels, 1999).

Going beyond single-word utterances, the Multi-Word Naming Game allows agents to use several words to describe complex meanings. This model is explained in detail in chapters 4 and 5. Another model that contains compositionality (but not grammar) is described in (Neubauer, 2002). In that model, meanings are triplets describing colors in the CIE LUV color space.[2] They can be described using one word that encompasses the whole triplet, or several words describing parts of the triplet, where the unfilled parts of the triplet associated with each word are wildcards left open to be filled in by the other words. Several triplets with wildcards can be merged to form a complete instance of a colour. (See also section 4.1 for a more in-depth description of this model.)

The most recent development of the NG includes syntax: agents can make syntactic rules, and use them to order the words in an utterance. This ordering signals the semantic role that the meanings of the words in the utterance, and allows the agents to efficiently refer to several referents in the same utterance. This model will be looked at in more detail in chapters 6 and 7.

1.4 Evaluation of the Models

The models in this thesis are positioned as "successive." This means that there must be some way to order them. In general, we have been using the concept of "complexity" as the ordering principle: Simple Naming Game utterances are simpler than Multi-Word Naming Game utterances, and Multi-Word Naming Game utterances are in turn more simple than Simple Syntactic Naming Game utterances. However, we should be more precise as to what complexity entails.

Suppose that in the very beginning, language users had a small repertoire of alarm calls and other emotional utterances. At some point, their cognitive abilities may have increased to the point at which they wanted to name other objects. This would have meant increasing the size of the call inventory that holds the associations between vocalisations and semantic representations.

When the number of new lexicon entries is not large, it may be sufficient to just add the new entries to the existing lexicon. In that case, we have a *quantitative* increase of the lexicon. At some point, it may not be possible any more to add large amounts of new entries to the lexicon. At that point, some kind of *qualitative* change is needed in the way the language user creates its utterances, so that it can say more with a lexicon of the same size.

The times when qualitative changes occur, mark the transitions between the different models. Several measures will be used to show the effect of the transitions on the agents' linguistic performance. Note that, in line with previous comments about "evolution" in a linguistic context, it can be hard to pinpoint exactly when this is. Some individuals may make a transition at other times than others, and there may be oscillation between different cognitive strategies before a pressure is strong enough to yield a clear winning strategy.

Another aspect of the evaluation of the models and the communicative strategies that they represent is that the pressures involved may change.

[2] The Comité Internationale de l'Éclairage standardized a number of spaces for colour representation, among which the LUV space, which is three-dimensional.

1.4.1 Communicative Success

Communicative Success is a very basic measure but also a very telling one. Without going into the details of the actual communication system that the agents in an experiment are developing, it simply measures the number of successful interactions that occur within a specified number of games. This gives a percentage of the number of games that were successful versus the number of games that were not successful.

By itself communicative success does not tell everything, because it merely indicates whether the agents are communicating well or not. It does not give much insight in the structure or quality of the communication systems that arise. However, in our experiments, the agents start out with empty lexicons, building a communication system from the ground up. We can use the Communicative Success measure to track the performance of the system from the beginning. In this way we do not only have an abstract number saying how well a population of agents is communicating at a certain moment in time, but we get a consecutive series of values that we can also use to study how fast the agents become better communicators, whether the system is stable, etc.

1.4.2 Lexical Coherence

In contrast to Communicative Success, Lexical Coherence does look at the internal structure of the communication system the agents build up. An instructive way of looking at (lexical) communication systems is by comparing the lexicons of the individual agents. All associations in these lexicons have a strength assigned to them, which makes it possible to determine which word each agent prefers when a meaning is lexicalised in several ways.[3] These preferences can be compared across agents, and by weighing the preferences of all meanings, a measure of lexical coherence can be calculated.

High coherence indicates that the agents have well-matching lexicons; however, as de Jong (2000) notes (p. 118), a high value for coherence does not necessarily mean that communication will be successful or reliable. If the agents would use the same word for every referent, they would show a high coherence value, but they would hardly be able to communicate. Therefore, coherence is always shown together with communicative success.

1.4.3 Lexicon Size

An important hypothesis that is implied by the criteria against which the different models in this thesis are being judged, is that the lexicon becomes "better organised" as the model becomes more complex. This means that it is able to fulfill requests to lexicalise meaning better with less resources. One of the ways

[3] This works because in our experiments, association strengths are used as absolute scores. In other experiments, notably some experiments on the influence of stochasticity on the model (Steels and Kaplan, 1998), association strengths have been used as a probability of selection. In these experiments, it is not possible to always know which association an agent will choose.

in which this is possible is by needing less entries for the same lexicalisation capability. (Note that we include the retrieval mechanism with the lexicon here. It is of course this mechanism that will need to perform better.)
The lexicon size measure will measure just this. By measuring the lexicon sizes produced over several experimental runs of the different models, we can compare the results. This allows us to see if, on average, the more advanced models indeed perform better that the simpler ones.

1.4.4 Synonymy and Homonymy

Synonymy and homonymy are characteristic of human language: several different words can refer to the (almost) same meaning, and the same word can have different meanings. In real language, both synonymy and homonymy are fairly subtle concepts: words may mean the same, but still exhibit subtle differences in other aspects, e.g. the context in which they are applicable: "boss" vs. "chief" vs. "supervisor," etc. Similarly, words that sound the same do not necessarily have the same origin: "site" vs. "sight," etc.
In our experiments, these subtleties are absent, but nevertheless we can observe the same phenomena, because words can shift meanings when interpreted wrongly by a hearer, or when accidentally generated more than once. By calculating synonymy and homonymy levels from the agents' lexicons, we can compare the communication systems across experimental runs, and given similar experimental conditions, across models.

1.5 Outline of the Thesis

Chapter 2 describes the Simple Naming Game (SNG) model. The SNG was the first experiment that was done at the AI-lab of the Vrije Universiteit Brussel to shed light on the group dynamics and agent-internal dynamics of language (lexicon) formation. Starting with the initial version, the mechanisms of the agents and the experiments themselves have been refined, and several versions of the Simple Naming Game have been built, testing resilience to noise (in the communication channel, in the extra-linguistic channel,...), testing different types of meaning generators (direct referents, discrimination games), etc.
Chapter 3 details the experiments that have been done with the Simple Naming Game. First, communicative success and coherence are measured for the basic experiment. The use of semantics and the associated weakening of the coupling between referents and meanings introduce complications for measuring coherence, which are explained and illustrated. Also shown are examples of how the dynamics of word competition work.
Chapter 4 details the Multi-Word Naming Game (MWNG). The MWNG model exists in two variants. The syntactic component, which assembles utterances as they are being produced or disassembles utterances as they are being interpreted, remains the same in both versions of the model, but the semantic

component was changed. Initially, the discrimination game was used as the semantic component, but this was substituted by a more complex semantic component based on a limited form of predicate calculus.

This chapter also presents the hybrid model in which agents are capable of using the communication strategies from the SNG and MWNG models. The model is introduced, along with an explanation of the selective pressures that have been examined.

Chapter 5 gives details about experiments done with the Multi-Word Naming Game. Communicative success and coherence are examined and compared with the results of the Simple Naming Game. The size of the lexicon is also compared with the size of the lexicon in the SNG. This chapter also presents the experiments with the hybrid experiment.

Chapter 6 presents the Simple Syntactic Naming Game, a first venture into the domain of syntax. The agents in these games are not only capable of combining several words into a single utterance, but also of coordinating the different words in an utterance. At the moment, the only constraints they can use involve word order, where the order of words determines the role that different referents play in the scene that is being described.

Chapter 7 then gives the results of the experiments that have been done with the Simple Syntactic Naming Game. These experiments are mainly measurements of communicative success, and an analysis of the games that fail. There are also coherence and lexicon size measurements, and graphs that show the competition between words for a meaning in the simple syntactic naming game.

Chapter 8 presents the conclusion, along with suggestions for future work.

Appendix A contains a definition of the most common and basic measures used in the experiments. In order to be able to provide consistent and comparable measurements across experiments, the measures need to be well defined. In the past, these measures have usually been defined informally in the texts. The appendix tries to define them more formally.

Finally, appendix B contains the graphs for all the experiments done with the hybrid model.

Simple Naming Game

"Apple!"

MANY COUNTRIES have institutions that monitor their languages and produce dictionaries, grammars, thesauri, etc. Generally these documents are descriptive, documenting the language, but sometimes these institutions diverge and try to prescribe how the language community *should* use the language. The most well-known example of such an institution must be France's *Académie Française*, which vigorously tries to protect the French language from any foreign influence, frenchifying words in some cases and inventing new words or reusing old words in other cases. Iceland has a similar institute, which tries to preserve Icelandic in its "medieval" state by finding Icelandic alternatives for new technical and other terms. The Icelandic language has not changed much since the Middle Ages, and the Icelanders, proud of their language, try to keep it that way.

However, most of the world's languages function perfectly well without such institutions. In fact, the word *language* usually refers to *standardized* languages, but languages spoken by illiterate cultures, or simply the different dialects of a standardized language, are languages in their own right. For these languages, it is even more obvious that they flourish (on a local level) without any outside help or control whatsoever.

Consider also that individuals are very creative users of their languages, both consciously and unconsciously: new words and idioms regularly find their way into mainstream language, and language itself changes constantly: most languages are quite different now compared to 200 years ago, although it can be argued that old spoken language is (or would be) much easier to understand than old written language is to read.

No; language is clearly driven and innovated by its individual users rather than a central entity controlling these users, and the mechanisms that are responsible for reaching and maintaining unity in this distributed system have only recently come under investigation.

The *Simple Naming Game* (SNG) represents the first foray into the domain of language modeling at the VUB AI-lab. The goal of this model is to study the type of dynamics of language outlined above, and more precisely to study the mechanisms that allow human language to organize a consistent lexicon, without needing recourse to a central lexicon that defines the words which individual language users use.

Briefly, the Simple Naming Game models a group of individuals that interact to give each other names. In other words, the "environment" in which the agents live is formed by the agents themselves. The agents always interact one-to-one, so imagine every interaction to be a children's guessing game of the form:

> Speaker: "He is called *Bizowa*."
> Hearer: "I think you mean *him*." (pointing to expected topic)
> Speaker: "Yes." or "No, *him*." (pointing to actual topic)

These games are implemented "without meaning," because the referents (the agents themselves) are referenced directly from the agents' lexicons.[1] A more elaborate version of the game uses other objects as referents for the meaning. In these games there is "meaning" in a more linguistic sense, since referents are categorised first on the basis of their features:

> Speaker: "It is *greenandbig*."
> Hearer: "I think you mean *that*." (pointing to expected topic)
> Speaker: "Yes." or "No, *that*." (pointing to actual topic)

As we will find out in the remainder of this chapter, although the differences between the two games seem small, they have a major impact on the design of the agents in our model.

More concretely, the task of the individuals in the model is to link symbols to (internal representations of) entities in the external world. According to Jackendoff (2002) in his scheme for the development of language (fig. 1.3), this is the first major step needed on the way to modern language: "use of symbols in a non-situation-specific fashion." Since the size of the individuals' lexicons in the model is not explicitly bounded by some hard limit, it also models the following step: "use of an open, unlimited class of symbols."

2.1 Related Research

Building communication systems has been of interest for more than a decade already. In robotics research and multi-agent research in general, where agents have to cooperate to solve a specific task, there has been an interest in communication to improve cooperation for the task at hand. In other research, like in this thesis, multi-agent systems are used as the vehicle for experimenting with communication systems.

Early computational research into the inner workings of language and cognition was carried out on a symbolic level. Meaning was represented using forms

[1] This was an implementational choice. Strictly speaking, these games have meaning, but because meanings and referents are the same in this model (in contrast to what meaning is understood to be in linguistics) we consider it to be without meaning.

of predicate logic, e.g. (Woods, 1968), and syntax was represented using symbolic rules that were processed by parsers to arrive at the meaning implicit in an utterance, e.g. (Winograd, 1976). These symbolic representations were then processed by reasoning systems to produce a reaction. These early systems developed into very advanced systems that could apply complex knowledge to situations described in narrative texts, and were able to answer advanced questions about events taking place in a specific domain, e.g. (Schank, 1984).

More recently, computational linguistics has shifted from attempting slow but complete semantic understanding of input texts to faster but less deep understanding of the input. The idea is on the one hand that, for a specific task at hand, full parsing may not be needed, and on the other hand, combining the result of several less-accurate methods (with known weaknesses) may yield a better overall result. Buchholz and Daelemans (2001) for example describe a system that uses *shallow parsing* to find answers to a query on the World-Wide Web by parsing one-by-one the results of a simple Google search.

Computers have also come to be used to research language as a complex dynamical system. Liljencrants and Lindblom (1972) applied a model from physics to vowel systems, and were able to predict vowel systems of specified sizes by minimizing the energy in the vowel system. The realism of the results of their optimisation procedure showed that their main point (vowels in specific vowel systems are optimally dispersed within the acoustic space) was correct. They were the first to show that vowel systems could (and should) also be looked at as a whole, instead of just looking at individual vowels.

Hurford's simulations (Hurford, 1989) were the first "multi-agent" simulations of language, in which a population of individuals with linguistic competences could arrive at a shared lexicon. He contrasted three different language acquisition strategies (strategies for children to learn language from their parents): whether production and interpretation should be treated separately, giving two possibilities (produce with the hearer's interpretation in mind or not) or whether production and interpretation should not be treated separately. Hurford's conclusion is that it is most efficient to not treat production and interpretation separately.

Many people are interested in the origins of communication, where lexicon-only communication presents a good minimalistic communication model, without needing to implement more complex communication systems. Often, the model is primarily a model of a cooperation task, where several agents need to develop a cooperation pattern to solve a task. Communication can be a tool that facilitates cooperation between agents, a goal of the experiments can be to study the differences in performance between systems without communication and systems with communication.

Yanco and Stein (1993) describe an experiment with two robots, where the follower robot must perform the same actions as the leader robot. The vocabulary of the language is fixed beforehand, and the set of meanings (actions) too. Werner and Dyer (1991) did an experiment in which the agents use a recurrent neural network for language processing. The weights of this network are genetically coded, so that the communication system actually evolves genetically. MacLennan (1990) did similar experiments. His agents were modeled using state machines; the genes coded for the transition tables of the agents. In general, these experiments show that cooperation on a task with communication is better than cooperation on a task without communication.

Cangelosi and Parisi (1996) and Cangelosi (2001) describe experiments in which the primary task of the agents is to survive by looking for food which they need to boost their energy levels. Contrasting games with and without communication shows how communication can help the agents in solving their task. (de Jong, 1998) and Smith (2002) describe similar experiments.

The experiments by de Jong (de Jong, 2000; de Jong and Steels, 2003) use almost the same setup as the experiments described in this thesis. The detail that is different is in the selection of the context. In de Jong's experiments, the context always contains all objects in the agent's environment. This is also the case in some of our experiments, notably the simpler ones, but in the more complex ones such as the Talking Heads experiment (see (Steels, 1999) and section 2.5.2 in this work), the context is a subsampling of the set of all objects in the environment. This is done partly because of practical considerations, and partly because of the reflection that humans do not contrast their topic of conversation with *all* objects in their surroundings; rather, there is a subset of objects that are more salient that others and that are used as the background for discrimination.

De Jong introduces several measures for gauging the performance of the models. These measures define a benchmark for the quality of the communication system developed by a population of agents in an environment. Apart from the standard measures also used in this thesis, he defines and uses other measures, which are discussed in more detail in section 3.3.2.

Meanwhile, language games are also being used in practical applications, where the paradigm is used to negotiate an ontology before actually exchanging information. An example is e.g. Avesani and Agostini (2003).

2.2 Simplest Experiment

The simplest version of the naming uses no independent semantics, but uses referents immediately in the lexicon. In a computer simulation, this is trivial to do. Suppose the context is $\{o_1, o_2, o_3\}$; in that case, a lexicon would contain references to o_1, o_2 and o_3 directly:

Word	Meaning	Strength
...	...	...
box	o_1	0.6
cup	o_2	0.4
chair	o_3	0.5
...	...	...

Schematically, this situation can be visualised as in fig. 2.1. This picture shows the relationship between words and their associated meanings. "Meaning" and

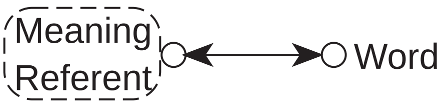


Figure 2.1: Semiotics in the basic experiment.

"Referent" are placed together and grouped by a dashed box to show that in these games, they are essentially the same. This figure is based on the notion of a "semiotic triangle," originally introduced by Ogden and Richards (1969) and first applied to the Simple Naming Game by Steels (Steels, 1999; Steels and Kaplan, 1999).[2] In the more complex models of language we will study later, the semiotic relationships will become more complex.

The direct link between referent/meaning and word makes this model ideal for studying the dynamics of word competition in the population and in the agents' lexicons. Word competition arises when different agents invent different words for the same referent/meaning. In order to arrive at a coherent lexicon, the same word should be used by all agents in the populations. Which word is chosen depends on the result of a negotiation phase, in which the agents monitor how well each word does. (Of course, there is no explicit negotiation; agents decide which words to use based on the success they have in communication.) Another way of looking at the negotiation phase is to view it as competition between the words.

Although being the simplest version of the naming game, it is also the most tractable one in terms of computational complexity. For this reason, it is the version that was used most in experiments with large populations and over long periods of (experimental) time.

Several publications deal with the dynamical systems aspects of the Simple Naming Game, but the most detailed one, that also precedes the research presented in this and the following chapters, is Kaplan's PhD thesis (Kaplan, 2000), which backtracks to even simpler models than this one to explore the rules used in the system.

2.3 Semantics

The agents' overall task in a language game is to verbally describe topics in a way that contrasts them to the other objects in the background. There are two sides to solving this task: when an agent assumes the role of speaker, it must be capable of *producing* such a description. When the agent is the hearer, it must be capable of *decoding* such a description an applying it to the external world.

In both cases, the semantic module plays a vital role. In production, the semantic module finds the unique features of the topic, and produces a semantic description that can be transformed into an utterance. In interpretation, it takes

[2] Semiotics is the study of signs and sign systems.

the candidate-meanings that result from decoding the speaker's utterance, and applies them to the world.
The following sections will describe the different forms of semantic descriptions that have been used in the Simple Naming Game model. In the language game protocol outlined in table 1.1 they cover steps 3 and 7.

2.3.1 First Experiment

The very first naming game, described by Steels (1996c), was not the simplest version, which was described above in section 2.2. It actually had a form of meaning: it used objects and their features as meanings for words. Every object has a value for each of a list of channels. The combination of a channel and its value for a specific object is called a *feature*. For example:

o_1:	(WEIGHT HEAVY) (COLOUR RED) (SIZE MEDIUM)
o_2:	(WEIGHT LIGHT) (COLOUR RED) (SIZE LARGE)
o_3:	(WEIGHT HEAVY) (COLOUR GREEN) (SIZE LARGE)

A topic object can be distinguished from other objects by comparing the values of the features. In the example, o_1 can be distinguished from o_2 by the (WEIGHT HEAVY) and (SIZE MEDIUM) features. It can be distinguished from o_3 using the (COLOUR RED) and (SIZE MEDIUM) features. In order to distinguish o_1 from both o_2 and o_3 at the same time, one feature is not sufficient. It takes a combination of two features to distinguish it, and as it happens, any combination of two features will work in this case, e.g. (WEIGHT HEAVY)(COLOUR RED). One can imagine that with more complex backgrounds, this will not be the case any more. The sets of features that distinguish the topic from the objects in the background are called *distinctive feature sets* (DFSs).
These DFSs are used as meanings for the words, and only words (not meanings) are exchanged between agents. The agents both know the topic of the language game, the speaker produces an utterance and the hearer checks if its interpretation of the utterance matches with the topic the speaker pointed out. These DFSs were the predecessors of the discrimination game, which is described in the following section.

2.3.2 Discrimination Game

The purpose of the discrimination game is to categorise the topic. Unlike in the first experiment, where the features and their values were symbolic, the discrimination game works with real-valued input. It was described for the first time by Steels (1996b). A further development from the Distinctive Feature Set system prototyped in the very first version of the Naming Game, it replaces symbolic values for features with real values (between 0 and 1), and introduces *feature detectors* that cover parts of the $[0 - 1]$ range. It also introduces a learning algorithm that can create new feature detectors on-the-fly, when the agent's current detectors are not sufficient for creating a meaning.

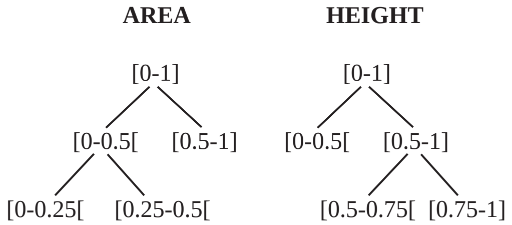


Figure 2.2: Example of two discrimination trees.

Feature Detectors

Like the "symbolic" discrimination game of the first experiment, the real-valued discrimination game has a number of *channels* that it can perceive: area, size, horizontal position, etc. Every channel is quantified for each object in the context using a value between 0 and 1. All values are normalized between the lowest and highest values that appear on that channel. This has important consequences, because values for a certain channel will be different depending on the context.

In order to find distinctions between objects, the objects have to be compared. Of course it is possible to directly compare the values different objects have for a certain channel. However, directly comparing the values is a recipe for failure, because the values depend not only on the objects themselves, but also on the accuracy with which they are measured, and this is frequently not 100%.

An equally simple but more robust way to compare values is to use *feature detectors*. Every feature detector covers a certain part of the input domain, which in the discrimination game always ranges from 0 to 1. Whenever an object's value for a specific channels falls within the domain of a feature detector, it becomes active for that object. Objects can then be distinguished by making sure that there is at least one feature detector that becomes active for one of the objects only.

Feature detectors are organised hierarchically in *discrimination trees*. Figure 2.2 gives an example of what discrimination trees look like. The leftmost tree contains feature detectors for the channel AREA. Note that feature detectors can overlap: for an object which has value 0.15 for the AREA channel, all three feature detectors that cover 0.15 will become active in this example. (Of course, since all values are between 0 and 1, the top feature detector, covering the whole range, is not very useful in practice, since it will always become active.) The rightmost tree shows similar subdivisions for the channel HEIGHT.

In order to describe an object, it may be necessary to combine feature detectors from different discrimination trees. Like the symbolic features of the previous section, we call these *feature sets*, with distinctive combinations being *distinctive feature sets*, as above.

Discrimination

While feature sets have a mostly descriptive character, they can be put to use in discrimination when we realize that it is possible to formulate, for a given object in a context composed of other objects, feature sets in which all features become active only for that specific object.
Suppose we have two objects, one having an area of 0.2 and the other having area 0.4. Looking at the discrimination tree in fig. 2.2, for the first object two feature detectors will become active:

(AREA [0-0.5[)
(AREA [0-0.25[)

For the other object, two other feature detectors will become active:

(AREA [0-0.5[)
(AREA [0.25-0.5[)

This means we can use feature detector (AREA [0-0.25]) to distinguish the first object from the second, and (AREA [0.25-0.5[) to distinguish the second object from the first. Note that we cannot use (AREA [0-0.5]) to distinguish either object from the other, because this feature detector is active for both objects.
As in the symbolic discrimination game, this filtering effect can be used stepwise: having a context of 10 objects, and a feature detector that filters out 5 of them, we can find the feature detector that will filter out the most of the leftover objects, to make the set of remaining objects as small as possible. One can continue adding feature detectors until all objects except for the topic have been filtered out. The resulting set is again called a *distinctive* feature set.
As an example, suppose we have three objects with the following characteristics, and we want to find a distinctive feature set for object 1:

o_1: (AREA 0.4)(HEIGHT 0.6)
o_2: (AREA 0.4)(HEIGHT 0.8)
o_3: (AREA 0.6)(HEIGHT 0.3)

In the AREA channel, we can eliminate o_3 using feature detectors (AREA [0-0.5[) and (AREA [0.25-0.5[). For both choices however, o_1 and o_2 remain in the same class, so we need a feature detector from another channel to separate the two. Using the HEIGHT channel, and more specifically, the feature detector (HEIGHT [0.75-1]), this can be accomplished, resulting in distinctive feature sets:

(AREA [0-0.5[)(HEIGHT [0.75-1])
(AREA [0.25-0.5[)(HEIGHT [0.75-1])

More generally, what the discrimination game does is produce categories that contain one or more objects. The goal of a discrimination game is to produce a category in such a way that the set of objects in the category is a singleton, containing only the topic.

Learning

Of course, in the above examples it is assumed that all the needed feature detectors are actually present in the system. That is often not the case, especially in the beginning of an experiment: the agents start out with empty feature detector repertoires, and build them as needed. If an agent reaches a point where it is not able to build a distinctive feature set, it will refine one of its discrimination trees.

A discrimination tree is refined by selecting the leaf node (feature detector) that contains the topic, together with one or more other objects. The range of this node is then split in two equal parts. For example, supposing (AREA [0-0.25[) is not a distinctive solution in the previous example, it will be split in (AREA [0-0.125[) and (AREA [0.125-0.25[). In the original implementation, a discrimination tree was chosen randomly, but other strategies are possible to make the learning more efficient.

2.3.3 Algorithm

Figure 2.3 details the discrimination algorithm by providing pseudo code. Generally, the algorithm favours combinations of "more general" features to very specific single features.

The algorithm first tries to find discriminating single feature detectors at a certain level in the discrimination trees. When it fails to find discriminating individual feature detectors, it will try to make combinations of feature detectors at that same level, until it either encounters a distinctive feature set, or the possible combinations at that level are exhausted.

If no solution was found and it is possible to descend further in the trees, the algorithm will then descend one level in the trees and search again. If it cannot descend further in any tree, it will stop without having found a solution.

2.3.4 Discussion

The discrimination algorithm detailed above provides a very elegant solution to the problem of categorising input that is continuous along a number of dimensions. It possesses the most important property of a meaning generator in our system: given the same circumstances or context, it will generate the same meaning.

The algorithm has a number of weak points as well. For instance, imagine a channel *shape:* there is an endless array of different types of shapes, which is impossible to codify in the interval $[0 - 1]$. One could try to change the interpretation of the channel somewhat, e.g. by using the number of vertices as the "shape" criterion, however even there the number of vertices can potentially become very large, so the problem of mapping onto the interval $[0 - 1]$ remains. Another technique for coding complex characteristics would be to spread the information in such a characteristic over several simple channels, however this too would quickly become cumbersome.

```
discriminate(DiscriminationTrees,Topic,Background)
  $D = \{d_k \mid d_k = \text{root-node}(DiscriminationTrees, k\}$
  do
    CombinationLength ← 0
    // Refinement
    for all $d_k \in D$
      $d_k$ ← continuation($d_k$,value(Topic,$k$))
    // Find combinations
    do
      CombinationLength ← CombinationLength + 1
      Combinations ← combine($D$,CombinationLength)
    until ((Combinations $\neq$ null) or
           (CombinationLength > max(#$D$,maxCombinationDepth)))
  until ((Combinations $\neq$ null) or
         no more refinements possible)

combine(List,CombinationLength)
  if (CombinationLength= 0) then
    return null
  else if (CombinationLength= 1) then
    Result ← null
    for all $el_k \in$ List do
      Result ← append(Result,make-list($el_k$))
    return Result
  else
    Result ← null
    for Counter= 0 to (length(List −1)) do
      CombinationFragments ←
           permute(List [Counter+1... length(List)],
           CombinationLength-1)
      for all $pf_k \in$ CombinationFragments do
        Result ← append(Result,
                        append(List [Counter],$pf_k$)
    return Result
```

Figure 2.3: Pseudo-code for the discrimination algorithm.

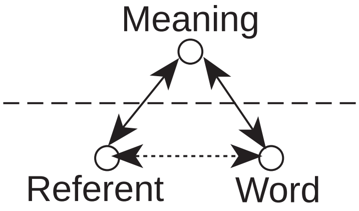


Figure 2.4: Semiotic triangle in experiments with semantics.

Another weak point is scalability: the search process in itself is fairly efficient, looking first for simpler ways to discriminate before trying combinations of feature sets. However, when the number of channels grows, the number of combinations that will potentially need to be searched grows exponentially.

Given these limitations, it would be fairly safe to say that the cognitive relevance of the discrimination algorithm, at least as a general algorithm, is fairly limited. Nevertheless, it could still be relevant as a special-purpose algorithm in certain domains (see section 6.3.1), in terms of computer modeling, and maybe cognitively as well. Also, a number of variants of the basic discrimination algorithm have been implemented that address the weaknesses of the basic algorithm, see e.g. (de Jong and Vogt, 1998).

Figure 2.4 shows a semiotic triangle for language games that include meaning. The dashed horizontal line indicates that the meaning is agent-internal, while the referent and the words are external to the agent. Contrary to the basic SNG, then, there is no direct relationship any more between referents and words. This has important consequences for the naming game, in the sense that success in communication is now not only dependent on the quality of the lexicon, but also on the quality of the meaning pump that categorises the referent into agent-internal meanings. This will become evident in the experiments in the next chapter, and more specifically in the coherence measure.

2.4 Form

"Syntax" in the simple naming game is simple: a speaker agent can at any time only utter one single word to describe a topic. Nevertheless, if we regard "syntax" not only as the external structure that can be found in an utterance, but as the mechanism that causes an utterance to become structured the way it is, we have to take into account what happens through time from the moment an agent starts learning words until its lexicon stabilizes at the "population lexicon."

An agent cannot assume that whatever it hears will be correct. Therefore, it needs a way to make hypotheses, evaluate them for a period of time, and eliminate them if they turn out to be bad or strengthen them when they are good. Concretely the lexicon is a list of triples, that each are composed of a form (a

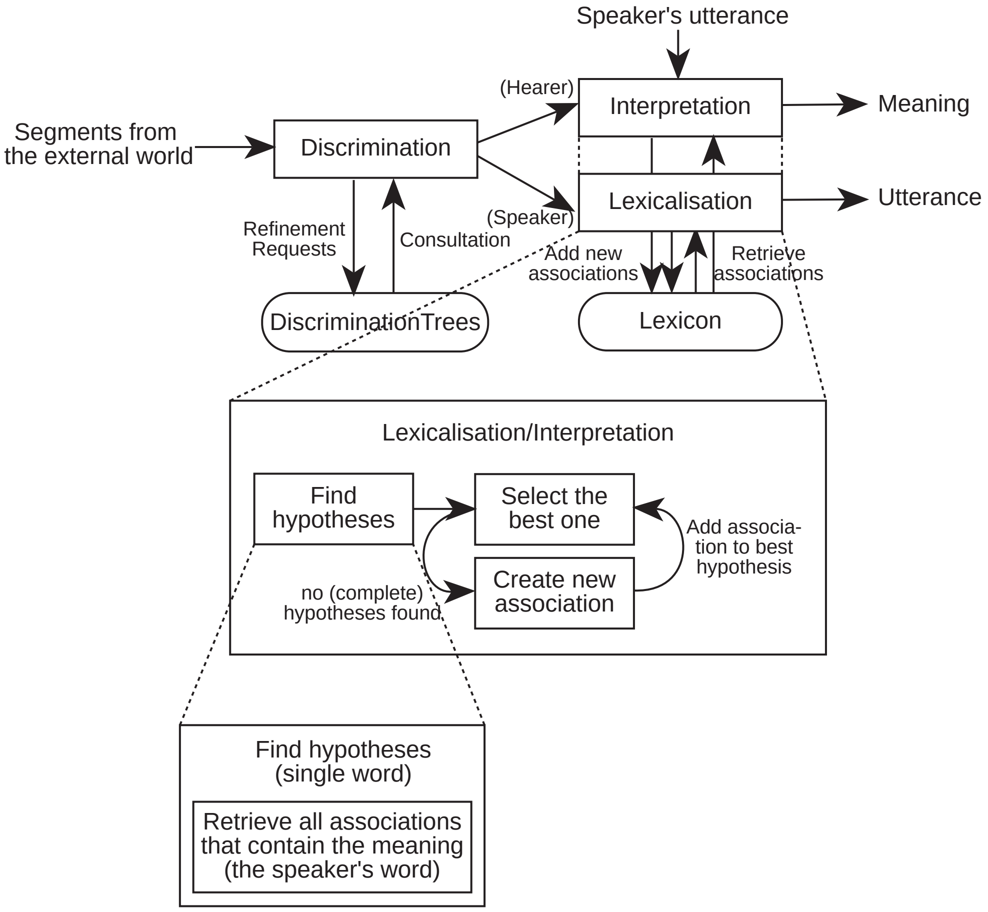


Figure 2.5: “Pipeline” of the single-word utterance system.

word), a meaning (a feature set), and a real number that represents the strength. The lexicon lookup algorithm can search the list based on the form, to retrieve the associated meanings, and based on meaning, to retrieve the associated forms. Even in the single-word case, it is not trivial to decide when to store a new word-meaning association: make too many false assumptions, and the lexicon may never converge; make too few assumptions, and you may miss many good ones. (By consequence successful lexicon learning also depends heavily on good joint-attention and external feedback mechanisms.)

Production

Production is simple: the only thing that has to be done is looking up the discrimination game’s meaning in the lexicon. Since only one word is allowed, we need a complete match. If several associations are found, the one with the highest strength is picked.
If no matching association(s) is (are) found, the agent may produce a new one

and add it to the lexicon.

Interpretation

Interpretation is simply the reverse of production: the agent hears a word, and looks it up in the lexicon. This results in a (possibly empty) set of associations. The agent again selects the association with the highest score, and checks with the context which objects activate the complete distinctive feature set.

If there are no such objects, the meaning was obviously wrong and the lexicon should be updated. If there is exactly one object, there is a good probability that this will be the topic. If there are several objects, the meaning was perhaps not wrong, but at least not discriminative enough. In this case too, the hearer should update its lexicon.

Learning

There are two different modes of failure in the simple naming game: either the agent does not have a certain association between a word and a meaning, or the agent has it, but it is not the strongest one (as compared to associations with the same meaning in production, or associations with the same word in interpretation).

In the first case a new association should be added—in the production case, using a new word (normally randomly generated), in the interpretation case, using the word the speaker used. In the second case, the score of the existing association should be increased.

An important mechanism that works in combination with adding associations or increasing the strengths is that of *lateral inhibition*. Not only are successful associations rewarded and unsuccessful ones punished, but the strengths of competing associations are decreased so that their likelihood of being selected in the future decreases as well.

Figure 2.5 gives a graphical illustration of the steps involved in producing and interpreting an utterance in the single-word naming game.

2.5 Variants

2.5.1 Stochasticity

A potentially key aspect of cultural evolution in humans (and other living animals) has been ignored in the basic model of lexical evolution presented above and in earlier publications: stochasticity. Everything in the models presented above is perfect: there is no uncertainty in any phase of an interaction, except in the lexicon itself. In reality however, uncertainty is present in many aspects of communication:

Transmission. Human language is mostly transmitted using sound. It has to coexist with other sounds in all circumstances except the most controlled laboratory circumstances. Of course, human language never evolved under laboratory circumstances—coexistence with and discernability from other sounds were issues from the beginning. Needless to say, these other sounds and noises take their toll on linguistic communication as the ratio of noise vs. language increases, and even when the ratio is favorable for language, potential for misunderstanding always exists.

Pointing. The extralinguistic communicational exchanges that serve to supplement language, for instance pointing to some referent to instruct a hearer when it failed to induce the correct referent from the utterance, are also prone to misinterpretation due to all kinds of noise.

Even when there is no "obstruction of the channel," it is quite possible for a hearer to incorrectly interpret the 3D-information given by the pointing. It might e.g. make an error in estimating the angle of the speaker's arm, thereby guessing that the referent pointed to by the speaker is a different one from the one the speaker actually points to.

There are many more areas involved with communication that are prone to inaccuracy induced by noise. Even though in none of these domains 100% certainty can be achieved, language exists, and most of time humans manage to communicate successfully.

The upshot of all this is that, if our models are to be taken as serious models of aspects of linguistic communication, they too must be able to deal with these uncertainties, and possibly even turn them to their advantage.

Extensions of the basic model that deal with uncertainty are described by Steels and Kaplan (1998). They introduce three types of stochasticity in the basic Simple Naming Game model: (1) stochasticity on non-linguistic communication, where the hearer cannot be 100% sure if the object the speaker appears to point to is the actual object the speakers intends to point to, (2) stochasticity on form, where the hearer cannot be 100% certain that what he appears to hear is what the speaker actually said, and (3) stochasticity of the lexicon, where the agents' lexicons are not stable, but subject to periodical random changes of the association strengths.

Generally, for all types of stochasticity, they conclude that some stochasticity is tolerable or even beneficial, but too much stochasticity prevents the agents from even reaching a stable state, or disrupts an existing stable state.

De Jong and Steels (2003) study stochasticity in production, where speakers may occasionally select another association from their lexicon than the strongest one when lexicalising a meaning. Like Steels and Kaplan, their conclusion is that modest amounts of stochasticity actually help improve the quality of the final communication system. Larger amounts of stochasticity degrade the system, to the point where no stable system can be formed.

2.5.2 Talking Heads

Philosophy

The first version of the Talking Heads experiment was conceived already during the initial stages of the language evolution research at the AI-lab, in 1998. It was already very advanced: it combined perception through motion-tracking cameras with the discrimination game, which served as input for the language formation algorithms (Belpaeme et al., 1998).

Using the experience of this early effort, the experiment was redesigned using off-the-shelf components, and an advanced version of the experimental platform used for the simulation experiments, called Babel (McIntyre, 1998). The Talking Heads experiment was set up as a publicly accessible experiment which was run at two different occasions. The first exhibition/experiment ran from July 1999 through October 1999. In the experiment, there were 3 permanent sites (at the "Laboratorium" exhibition in Antwerp, in Paris at Sony CSL and at the VUB in Brussels) set up as shown in fig. 2.7 with two pan-tilt cameras looking a whiteboard (a) on which geometrical shapes are pasted (b). Figure 2.7 (ii) shows the actual setup at the AI-lab at the time of writing (December 2004). The second run of the experiment started at the end of January 2000 and lasted until August 2000. The setup of the sites was the same as in the first experiment but they were located at different places (at the UvA in Amsterdam, in the UK in galleries in London and Cambridge, at Sony CSL in Paris and at the VUB in Brussels).

Through a public web site, anyone could create new agents, introduce them into the system, send them to the different sites of the Talking Heads network, and even teach their agents new words.

Goals

The goal of the Talking Heads experiments was twofold. On the one hand it was a public relations effort, to make the experiment and the associated research known to a large audience. On the other hand, with several installations in different places of the world at the same time, it also represented a great scientific opportunity: it became possible to have very large, dynamic populations, for which the interactions could be distributed over the different servers.

Another important goal was to do embodied experiments. Embodiment refers to the fact that experiments should not only be done in simulation, but also in a real environment, where the agents of the experiment are under the influence of the environment with all its quirks and uncertainties. Conversely, the environment can also contain structure that can be exploited by the agents, and reflected in its perceptual and conceptual systems. Finally, embodiment solves the grounding problem. See also section 1.3.2, and (Steels and Vogt, 1997; Zuidema and Westermann, 2003).

All interactions done were logged in a central database, along with data about when and where the interaction took place, etc. During and after the experiments, different measures were calculated. Some were displayed in (almost) real time on the web site, such as the communicative success. Others showed interesting trends post-experiment. Some of these experiments will be detailed in the next chapter. See also (Van Looveren, 2001a,b).

Experimental Setup

World The agents' world was formed primarily by the whiteboard in front of their cameras. However, they did not merely take images of the whole whiteboard (fig. 2.7). Instead the cameras were zoomed closer, which meant that they could perceive only a fraction of the whiteboard at once.

They had to use their pan-tilt capability to move the cameras around to see different parts of the whiteboard. In order to coordinate, the agents used a common coordinate system on the whiteboard, which they knew how to convert to the coordinate system of their own camera.

There were experimental setups in several different places, which were all connected to a central server through the internet. The agents thus had the possibility to move from one place to another. In terms of hardware and software, the setups were the same. The difference between the setups was in the configuration of the whiteboard: the curators of the different installations were allowed to, and did, configure the whiteboard to their own liking. From the agents' points of view, this meant that the environment in the different places was different: their environment was precisely what was on the whiteboard. This in turn meant that word/meaning associations that were useful at one location might not be useful at another location, which in turn meant that an agent that arrived at a new location might have to learn new word-meaning associations to be able to discriminate the objects at that location.

Agents The Talking Heads experiment was the first time that the software components from different experiments were integrated in one package.

Perception In previous models, perception was fully simulated, but the Talking Heads experiment used cameras to capture images from an external world.

When an agent participates in a language game, it uses one of the two cameras of the system to capture an image. In this image, "areas of interest" are detected using a simple region growing algorithm: the pixels of an image are scanned consecutively until one is found that differs significantly (according to some threshold) from the one last scanned. At that point, a new segment is started, and all neighbouring pixels that resemble it are added to it. This is continued until no more pixels can be added to the segment. At this point, the segment is closed, and the algorithm is repeated. This is done until all pixels in the image have been assigned to a segment, or to the background. In order to eliminate most of the noise, the background and all segments smaller than a certain threshold are ignored. For every segment, a fixed set of features is evaluated to arrive at a description of the segment (horizontal position in the image, vertical position in the image, height, width, area, red, green, blue, lightness). Every feature is measured, and normalised to the interval [0,1]. (See e.g. Sonka et al. (1996), chapter 5 for a more detailed explanation of region growing algorithms.)

A problem in this experimental setup is the fact that the two cameras are located in different positions with regard to the that they observe. This is shown schematically in fig. 2.7 (i-a), and fig. 2.8 gives an impression of how differently the whiteboard is perceived in the Talking Heads installation shown in fig. 2.7 (ii).

In the Talking Heads, this problem has not really been solved. The cameras are calibrated to use a common coordinate system on the whiteboard, which they can use to convert coordinates to and from, but this only crudely approximates reality. For example, when the speaker communicates to the hearer the area of the whiteboard it is looking at, it gives to the hearer the center coordinate of that area. The hearer will convert the coordinates and move its camera to that position, but the area captured by the hearer will be distorted compared to the area captured by the speaker. Nevertheless this approach works fairly well provided the cameras are not too close to the whiteboard (or not too far apart).

Semantics The Talking Heads model uses the Discrimination Game to generate its meanings, as described in section 2.3.2. From the segments retrieved from the perception, one is chosen to be the topic. The goal of the discrimination game is to find a unique description for the topic; in this case, this means a category composed of one or more feature detectors based on the values returned by the perception.

Figure 2.6 shows an example of the type of discrimination trees the agents in the Talking Heads experiment develop. The "empty" slots (with only a small dot) denote channels that have not been developed; the others show discrimination trees of varying levels of complexity.

Syntax The Talking Heads uses the Simple Naming Game syntax as described in section 2.4, i.e., a speaker utters a single word, which the hearer must learn to understand. The words the agents create are created such that they can be easily pronounced by the host computer's text-to-speech system, but this is only used for output; the speaker and hearer merely exchange the symbols.

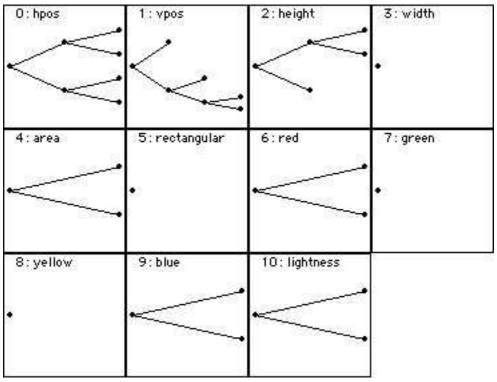
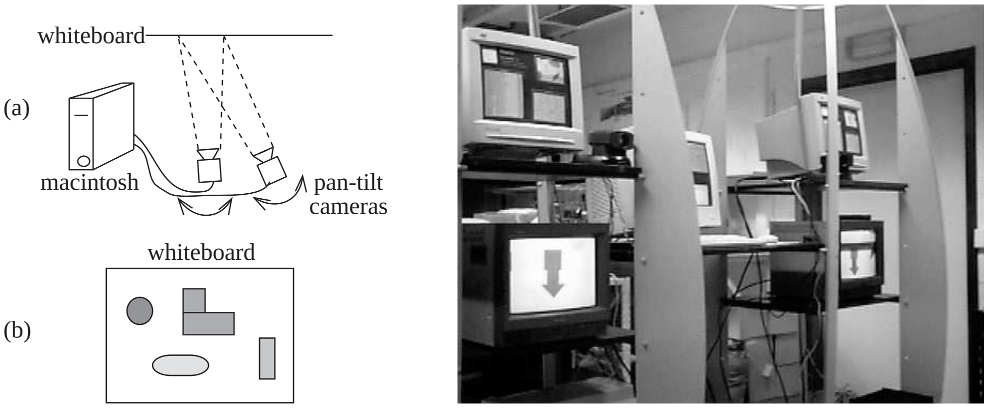
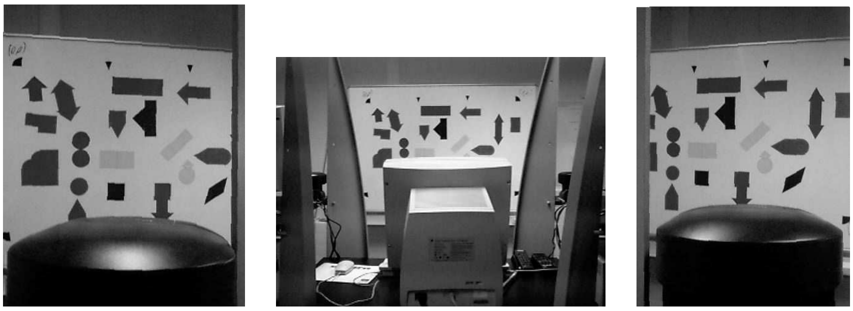


Figure 2.6: Discrimination trees as developed by the agents in the Talking Heads experiment.



(i) (ii)

Figure 2.7: *Talking Heads* installation layout; schematically (i) and actual (ii)



(body 2) (body 1)

Figure 2.8: An example of the distortion faced by the Talking Heads software; the whiteboard is parallel to the "camera plane."

2.6 Summary

This chapter introduced the first model of the thesis. In this model, agents can produce and interpret utterances composed of one single symbol. The semantic mechanism of the discrimination game was explained, and the syntactic mechanisms.

Two variants of the model are discussed: one in which several deterministic mechanisms are replaced with stochastic ones that introduce noise, which is more realistic compared with natural language. The second variant is the Talking Heads experiment.

The Talking Heads experiment allowed for the first time to do complex experiments with the model at a very large scale. Also, the Talking Heads provided a vehicle for experimenting with an embodied variant of the model, where the perceptual data was captured in the real world.

Simple Naming Game: Experiments

THE PREVIOUS chapter described the basic Simple Naming Game model, and a number of different variants of it that have been built over the last several years. This chapter groups together a set of experiments that have been done on several occasions with the SNG model. Section 3.1 will first show the results of experiments in which there was no separate meaning level (i.e. referents are used as meanings). Section 3.2 contains the results of experiments in which the Discrimination Game was used to generate meaning. Section 3.3 details a number of observations that have been made about the coherence measure in the course of the experiments, and finally section 3.4 describes results from the Talking Heads experiment.

3.1 Meanings Are Referents

Figures 3.1–3.3 on page 43 show results of experiments that have been done with the basic version of the Simple Naming Game. In this version, there is no semantics; the focus is on the emergence of the lexicon. The curves shown on the graphs are averaged over 10 runs in all cases. Two measures were used: communicative success and coherence. In both cases the standard deviations were calculated, but in all graphs only the standard deviation for coherence is shown; in all cases, the success stabilises at 100%, and the standard deviation becomes zero.

3.1.1 Naming Game Algorithm Details

Word Creation and Storage Probabilities

These two parameters govern the ease with which new associations are learned. Learning occurs in two circumstances: when the agent is the speaker of an interaction and has no association for the meaning to be expressed, it may invent a new word and associate it with this meaning. The probability with which this happens is set using the *Word Creation Probability*, which is a number between 0 and 1. Too low a value will cause many failures and hesitation to adapt to new circumstances, e.g. new objects entering the agents' environment. Too high a value will cause a proliferation of new words, possibly inhibiting coherence to arise. The default value for this parameter is 0.1, i.e. in 10% of the cases where a speaker cannot lexicalise a meaning, it will create a new word and word-meaning association.

The *Word Storage Probability* (a value between 0 and 1) regulates the ease with which a hearer will assimilate a word-meaning association it does not know yet in its lexicon. When an interaction fails because the hearer does not understand the speaker, the speaker will point out the topic. The hearer can then derive its own meaning for the topic, and associate the speaker's word with it. Setting this parameter to a very low value will inhibit the spread of lexical conventions in the population, while too high a value will cause the hearer to accept anything it hears, which could overflow its lexicon with useless associations. The default value of this parameter is 0.75, so that the hearer will learn a new association in 75% of the cases in which it is not able to find the correct topic by itself.

In both cases, the experiments reported here use the default values for these parameters.

Association Strength Updates

After every interaction between two agents, the agents use the result of the game to update the strengths of the associations that were used during the interaction. When the interaction succeeded, the association between meaning and form should become stronger, and conversely when the interaction failed the association should become weaker.

The actual implementation associates two numbers with each association: a usage counter, and a success counter. When an interaction is successful, both counters are increased; when an interaction fails, only the usage counter is increased.

These two numbers are only used internally; to the rest of the program, the strength appears as a single number, through a conversion function. Again, several strategies to compute a single strength value (between 0 and 1) are possible. Implemented are:

Default is simply success divided by use. However, if the association has been used only a few times, its strength will be zero, even if already used successfully.

$$\mathit{strength} = \begin{cases} 0 & \text{if } \mathit{use} < 5; \\ \mathit{success}/\mathit{use} & \text{otherwise.} \end{cases}$$

Failure $\mathit{strength} = \mathit{use} - \mathit{success}$ (This is only used for testing purposes.)

Use Only $\mathit{strength} = \mathit{use}$

Success Only $\mathit{strength} = \mathit{success}$

Population Size	Context Size	Games
10	10	1500
20	10	5000
20	20	12000

Table 3.1: Time needed to reach 100% communicative success in all three experiments.

In all experiments reported here, the default function was used.

Another mechanism at work is that of *lateral inhibition*. When an association is successful in a certain context, it will be rewarded per the formula described above. Often however, there will be competing associations, which associate the meaning with another word (speaker case), or the word with another meaning (hearer case). The fact that one of the associations was successful, allows us to conclude that these competing associations were not successful, and by consequence they can be punished (inhibited).

3.1.2 Communicative Success

In all three cases, communicative success reaches 100%. The experiments thus all reach their primary goal of building a reliable communication system. However, not all setups do this in the same time frame. Experiments with the same parameters are fairly consistent, but both the size of the population and the size of the context (set of objects) have an influence. Table 3.1 shows for the three experiments how many interactions it takes to reach total communicative success.

The data indicate that both doubling the population size and doubling the context size increase the time to reach 100% sucess more than twofold. Doubling the number of agents needs a threefold increase in interactions, and doubling the context after that needs another 2.5 times as many interactions.

Of course, the experiments done were of the simplest possible kind: a fixed population in a fixed context. Introducing new agents with empty lexicons upsets the stability of total success, because interactions with new agents will fail until they have established their lexicons. On the other hand, taking out experienced agents does not change the stability.

Introducing new agents and suppressing old agents creates a population *flux* that is more like a real population, were children are born not yet knowing their mother tongue(s), and old, linguistically experienced people die. Steels and Kaplan (1998) have done detailed simulations about (among others) this aspect of population dynamics. Their conclusion is that a linguistic system can remain stable even if the population that sustains it is dynamic, as long as the agent flux remains within "reasonable" bounds. When the flux becomes too high, the language breaks down because more words are invented than learned, and the associations between words and meanings cannot propagate fast enough in the population. It is difficult to quantify this though, as it is highly dependent on many different variables, such as the initial population size, the size of the context, the probability with which agents will accept new associations in their lexicons, the magnitude of the increases and decreases that are applied to the associations in case of success or failure, etc.

3.1.3 Coherence

All experiments have, after stabilisation, coherences in the 90% range. This indicates that there is a large amount of overlap between the agents' "active" lexicons (the associations they prefer to use themselves when they have to lexicalise a meaning).

The fact that coherence does not reach 100% also indicates that there are meanings where different agents prefer different lexicalisations. However, since the communicative success does nevertheless reach 100%, this does not seem to impair the communicative abilities of the agents.

The reason for this is that agents are allowed to have more than one association for each meaning or word in their lexicons: this allows the hearer to understand a speaker even if it does not prefer the same word for that subject. These lexicon entries are "passive:" the agent knows them and understands them, but does not use them actively when it needs to lexicalize a meaning. Even so, agents are not required to remove them from their lexicons.

In a sense, the ability to have "passive" lexicon entries is both a curse and a blessing: on the one hand, it allows agents to entertain several hypotheses at the same time for a meaning or a word, and take their time before "deciding" which one to keep. On the other hand, it leads to communication systems that are less than perfect from a structural point of view. Despite that, however, the agents do reach their main goal of communicating reliably.

Other language game protocols might be less forgiving: imagine a game for instance, in which the agents are both "speakers," both know the topic (for example through pointing) and must each supply the word they prefer for the topic. If they give the same word, the game is successful, if not, the game is a failure. In such a protocol, a coherence of 100% would be required to reach 100% communicative success.[1]

We can see from the above experiments, that the basic mechanism in itself is sound. When agents with no lexicons are allowed to interact about a number of topics, words will be invented, there will be a phase in which agents cannot communicate perfectly yet because of disagreement or ignorance about the other agents' use of words, but through the interactions the agents quickly converge on a single lexicon.

[1] This protocol is called the "compatibility game" and was used in several early versions of the naming game.

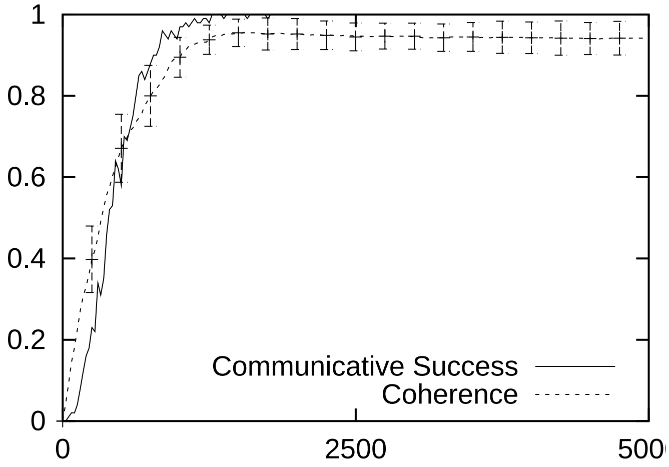


Figure 3.1: Success and coherence in a 10-agent, 10-object experiment (5,000 games; averaged over 10 runs).

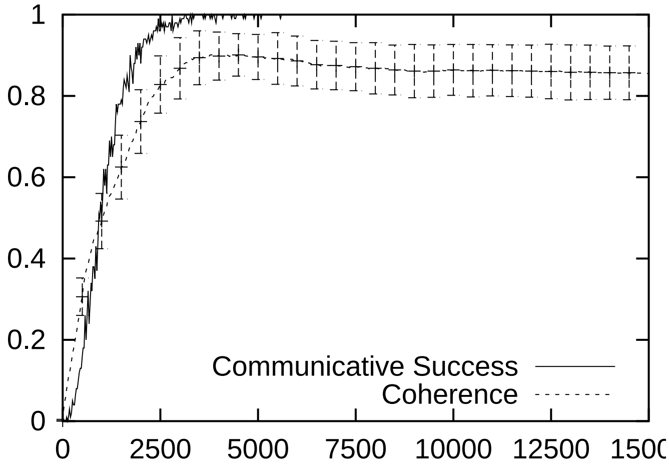


Figure 3.2: Success and coherence in a 20-agent, 10-object experiment (15,000 games; averaged over 10 runs).

3.2 Discrimination Game

The games the agents have to play in these experiments are rather more complicated than in the previous experiment, because there is no direct coupling any more between the actual referent and the meanings stored in the lexicon. In other words, meanings can be applicable to different referents, and referents can be expressed using different meanings (in the same or in different contexts). Another consequence, since meanings are kept internally within the agent, is that different meanings can adequately describe the topic in a certain context. This in turn means that agents can associate a word with different meanings, and still communicate successfully, until the word is used in a context in which one of the meanings cannot be used for the topic. If that occurs often enough, the score of the second meaning may drop and the agent using this meaning may be forced to learn an association between the word and the "correct" meaning.

For example, if in an experiment the meanings BIG and YELLOW are usually either both discriminative or both not discriminative (and lexicalised using the same word by different agents), then the agents in the experiment will not notice that for some of them the word in question has a different meaning than for the others. Only when enough cases arise in which either one or the other is discriminative will they notice it, and (need to) differentiate between the two.

Figures 3.4 (a) and (b) show graphs of two experiments done with the simple naming game with discrimination. For every experiment, two curves are shown: the success curve, and the vocabulary coherence curve.

3.2.1 Discrimination Algorithm Details

Channels

Discrimination is done along a number of input dimensions. For each object detected in the input, a value is computed for each input channel, such that each object is described by a series of feature-value pairs. Several of the channels refer to or relate the object with the dimensions of its bounding box, the smallest horizontally-oriented square that surrounds it.

In all simulated experiments, the following set of channels was used:

BBX bounding box X-coordinate

BBY bounding box Y-coordinate

BBH bounding box height

BBW bounding box width

GRL gray level of the topic

RATIO area of the bounding box actually covered by the topic

AREA area of the topic

The Talking Heads experiment used a more elaborate set of perceptual channels:

HPOS bounding box X-coordinate

VPOS bounding box Y-coordinate

HEIGHT bounding box height

WIDTH bounding box width

AREA area of the topic

RECTANGULAR a number indicating the rectangularity of the topic

RED red component of the average colour of the topic

GREEN green component of the average colour of the topic

YELLOW yellow component of the average colour of the topic

BLUE blue component of the average colour of the topic

LIGHTNESS general brightness of the topic

The `red/green` and `yellow/blue` channels represent a colour coding inspired by the opponent channels theory of human colour perception (Hurvich and Jameson, 1957).

Refinement strategy

When discrimination fails, this is a signal that the discrimination module should learn. Concretely this means that it should extend its discrimination trees, to increase the probability that it will find a distinctive feature set when it has to discriminate within the same or a similar context in the future.
As a rule, the discrimination algorithm will only expand one tree when it fails. There are several possible ways to select the specific tree to expand:

random this simply selects a tree at random.

strongest this selects the tree which was capable of discriminating the most objects from the topic, on the theory that it only needs a little extra work to produce a good discrimination.

weakest this selects the tree which discriminated the least objects from the topic, on the theory that it might become better with only a little extra work.

situated this method combines selection of the strongest discrimination tree and selection of the most abstract discrimination tree: among the strongest trees, it will favour the one that is the least developed.

The *situated* expansion selector is the default one, and has been used in all experiments reported in this thesis where the discrimination game is used as the semantic component.

3.2.2 Naming Game Algorithm Details

The naming game algorithm used in this model works as detailed in the previous section.

3.2.3 Success

The success rate rises much more slowly than in the previous experiments, and seems to reach a maximum at around 80%. Obviously, communication has a harder time establishing itself in these more complicated experiments as in the previous, simpler ones.

When looked at more closely, it turns out that the games that keep failing are games in which a complex meaning is used to describe the topic. Oftentimes, the meanings have not been lexicalised before. This means in turn that a new association needs to be inserted in the lexicon, with a new word. Such games fail always, because the hearer cannot know the speaker's new word.

It seems that agents continue to create new meanings throughout the experiments, so that they never really come close to the performance level of the game that directly references referents. The cause of this would seem to lie with the discrimination game, which always starts a full search process when it needs to produce a meaning. This could lead to "new" solutions also in cases where a previously derived meaning would do the job as well.

3.2.4 Coherence

Calculating coherence in the same way as before, based on lexicon entries, calculates coherences between meanings and utterances (instead of between referents and utterances), because the lexicon contains meanings instead of referents. Figure 3.4 shows the results of two experiments in which this type of coherence was measured.

Competition between words. The graphs in this section (and in corresponding sections in the experiment chapters on the other models) show the "competition" between different words for the same meaning that arises as a consequence of the population dynamics of the multi-agent model.

Both the individual competition between words for a specific meaning and overall competition are discussed. Individual competition provides a closer look at how the coupling works between the low-level mechanisms of repeated

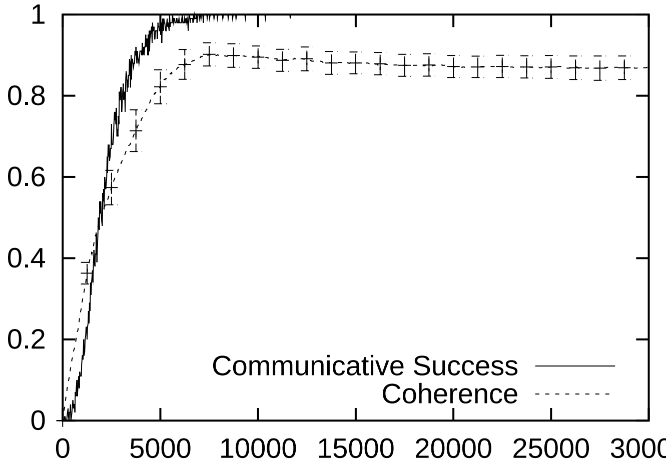
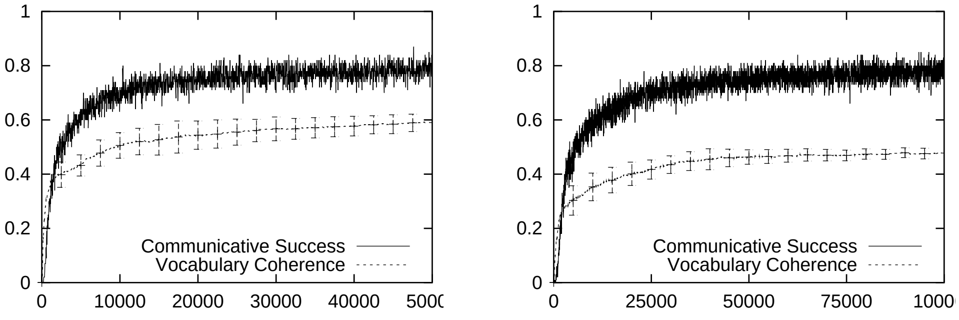


Figure 3.3: Success and coherence in a 20-agent, 20-object experiment (30,000 games; averaged over 10 runs).



(a) (b)

Figure 3.4: Success and vocabulary coherence in a 10-resp. 20-agent experiment, with 7 objects in the context (50,000 resp. 100,000 games; averaged over 10 runs). The fact that the success curve is severely jagged in both graphs even after being averaged over 10 experiments, is due to the short intervals over which communicative success has been measured.

interactions between agents and the strength mechanism in each agent's lexicon. It becomes clear how they work together to reach a stable state that promises a succesful communication system.

When measuring overall competition, we are interested in the (average) number of words per meaning that are created for all meanings, by all agents, during an experimental run. It gives an idea of how easily agents create new words. Partially this depends on the parameters of the experiment: the probability that a new word is created when an agent has to produce an utterance for a meaning it does not know a word for, and the probability that an agent will learn a new word if it hears it but does not know it. It also depends on other factors, such as the size of the population of agents (i.e. the probability that 2 agents will interact), and ultimately also the frequency with which each topic is selected, which is determined randomly.

Figure 3.5 shows a typical example of word competition in a two-agent population. Generally, when a word has been accepted and is used by many agents in a population, it will have a high strength in the lexicons of all of these agents. A simple measure to show the "global" strength of an association in a population is then simply to show the sum of the strengths of that association for each agent in the population. The figure shows that there are two words, one invented by each agent. One of the words, in this case "kiri," is used only once but not picked up by the other agent, leading to a global score of zero. The other word ("gafiku") is learned by the hearer and will be the preferred word for both agents. In the figure, the final strength is 1.3 (the highest possible global strength for an association in a two-agent population would be 2), which means that it is at least somewhat accepted by both agents: in one agent, the maximum score is 1, so it has a score of at least 0.3 in the other agent's lexicon. However, more likely would be a more balanced score of about 0.6–0.7 in both agents.

Reaching the maximum strength takes time, because the speed with which it is reached (if it is reached at all) is dependent on the frequency of the meaning. For a frequently occurring meaning, the agents have many opportunities to increase the coherence; for less frequently occurring meanings, these opportunities are rarer. (See also section 3.3.)

Also, when an agent knows different meanings for the same word, some of the interactions in which it is used may be successful, while others may fail. This accounts for the fact that the curve for "gafiku," even though it is the only accepted word for that meaning, does not monotonically increase from the moment it starts gaining prominence for that meaning.

For larger populations, the situation is more complex, as fig. 3.6 shows for a 5-agent population. Four different words compete with each other, only after about 9000 games does one word gain the upper hand. And even then, its global score less than four, which means that not all agents have reached full coherence for that word (the maximum strength in this case is 5).

Figure 3.7 shows an example of competition in a 10-agent population. Five words compete, but only two of them actually get a non-zero strength. The others are created and used only a few times (the strength remains 0 until the

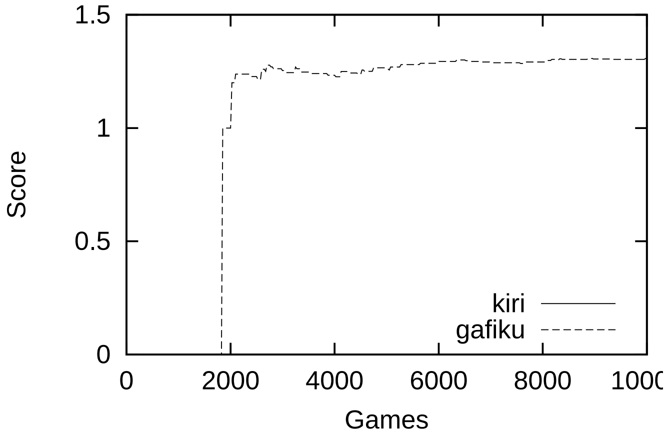


Figure 3.5: Competition between words for meaning WIDTH in a population of 2 agents.

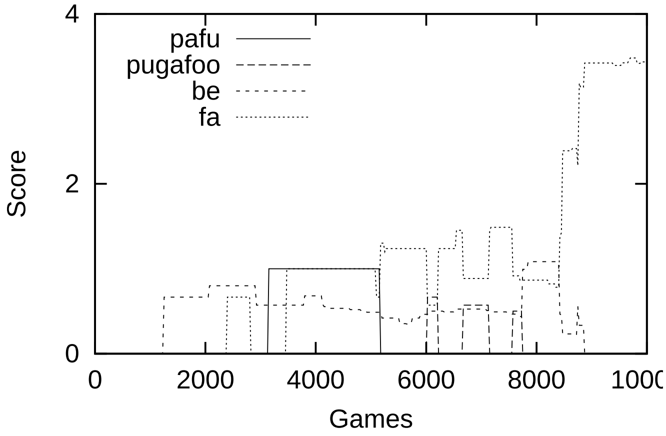


Figure 3.6: Competition between words for meaning HEIGHT in a population of 5 agents.

Population	Number of words per meaning	Standard deviation
2 agents	1.19	0.50
5 agents	1.59	0.91
10 agents	2.17	1.59

Table 3.2: Average number of words per meaning (cfr. synonyms).

word has been used at least five times). From 2500 games on, the word "meelifee" rises steadily, but at about 6000 games, the word "tar" is introduced, and it takes off immediately. Since the global strengths of both words are only 1.5 and 3.8, respectively, it is clear that the competition has not been settled. Both words are still relatively weak, which means that it will probably take quite a number of interactions still before a definite stable state is reached.

From table 3.2 we can deduce that in general, in such small populations there is very little tendency towards synonyms. A population of two agents actually represents a special case: each agent is always either the speaker or the hearer, which means that each time a new word is invented, both agents will be able to register it immediately. In these experiments, this happens with a certain probability, so it is not certain that both agents will always know all words that have been invented. Generally, in larger populations the spread of new conventions will be much more difficult. Therefore, the larger the population becomes, the higher the number of words per meaning is. Nevertheless, also in these larger populations, coherence reaches good values, so even if they do not converge on a single word for each meaning, stable states arise in which only very few words are still in use for each meaning.

3.3 Coherence Revisited

The coherence measure as described above is adequate for the simple Naming Game experiment. With the introduction of meaning, however, the definition of the coherence measure becomes less clear. Should we calculate lexicon coherence (i.e. coherence between meanings and forms)? Or should we instead focus on referent-form coherence? And what about referent-meaning coherence, does that yield interesting results? The existence of these different relations between form, meaning and referent were acknowledged in (Steels and Kaplan, 1999), but their implications on the coherence measure was not.

This section will first discuss a number of factors that influence the result of the coherence measure, and proceed to discuss the different types of coherence that can be measured in the more complex experiments.

The way coherence is calculated in our experiments, is by looking at the possible values for a variable (for instance the referent), and for each of these finding out what (1) are the possible values for it (the different utterances agents prefer to use for that referent), and (2) what the "distribution of preference" is of these values (i.e. how many agents prefer each possible value). Combining the results then yields a single number that reflects the amount of coherence present in the population. The difficult part here is how to combine the different results. A number of factors will have a repercussion on the final coherence value:

Weight of each referent. The basic coherence measure simply averages the individual coherence values for each referent. This is correct if the referents all occur approximately with the same frequency.

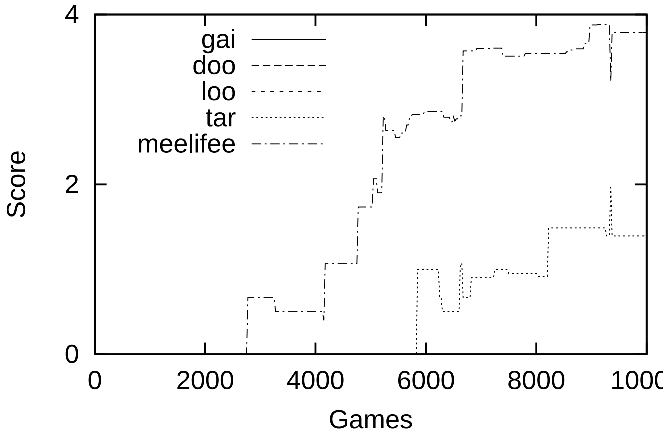


Figure 3.7: Competition between words for meaning GRAY-LEVEL in a population of 10 agents.

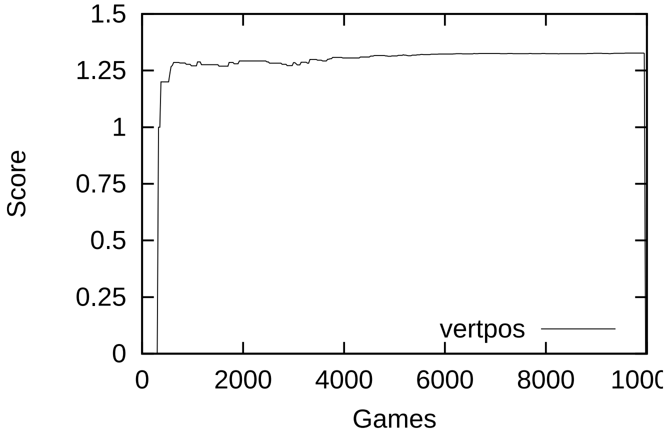


Figure 3.8: Competition between meanings for word "bee" in a population of 2 agents.

Population	Number of meanings per word	Standard deviation
2 agents	1.81	2.19
5 agents	1.82	2.07
10 agents	1.95	2.33

Table 3.3: Average number of meanings per word (cfr. homonyms).

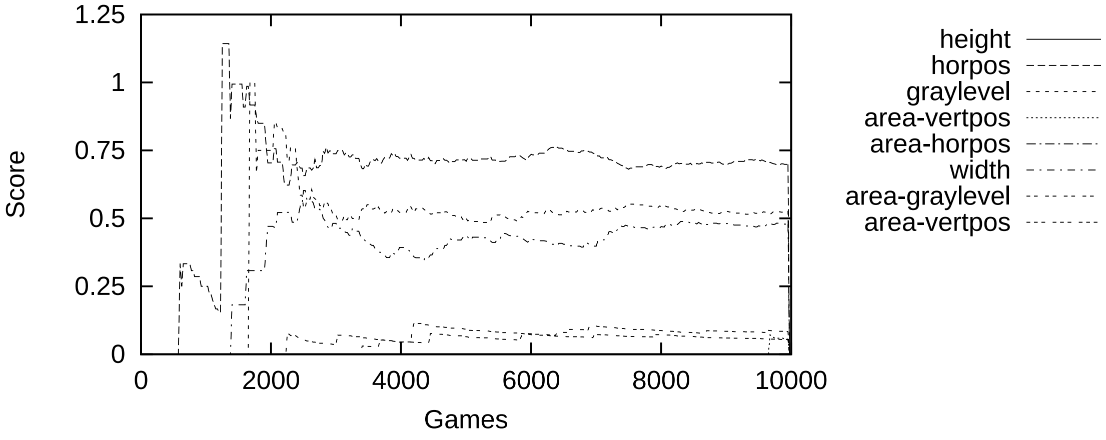


Figure 3.9: Competition between meanings for word “gedale” in a population of 2 agents.

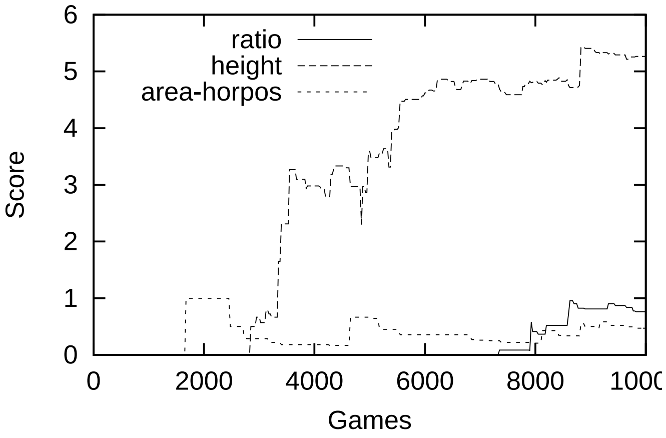


Figure 3.10: Competition between meanings for word “soopeeki” in a population of 10 agents.

This is not correct when the distribution of the referents is skewed: often-used referents will tend to show higher individual coherence than less-used referents. Assuming we want to compute a coherence value that reflects the coherence of the language as it was actually used rather than a value that reflects a simple static comparison of lexicons, the coherence value will differ depending on the relative frequency with which referents occur, precisely because they are used more often and have a greater influence on the strengths in the agents' lexicons. If the individual coherences are combined using a simple average, the often-used references will have comparatively less weight in the final result, and little-used referents will have comparatively more weight in the final result. Consequently, the final coherence value will be lower than the actual "communicative" coherence.

Weight of each utterance. The reasoning about referent frequency carries over to utterance use. Again, in the basic game all utterances are taken to have the same weight in the final "individual" coherence of a referent. In practice, this is not necessarily so. Utterances are preferred by agents, and agents do not necessarily communicate equally often. If some agents communicate often and others not, the preferred utterance of one of the former agents has a higher influence on the coherence for that referent.

Presence of unused meanings in the lexicon. These meanings will often have only one (or very few) words associated to them, which means they generally have a coherence of about 0.5. This may either increase or decrease the coherence value, depending on the coherence of the frequently-used meanings.

Apart from these factors, the fact that there is now an intermediate meaning step between the referents and the utterances introduces a number of specific issues:

- Different agents may use different meanings in the same context.
- The same agent (or different agents) may use different meanings for a referent in different contexts.

The meaning represents a "hidden" variable that is not taken into account when calculating referent-utterance coherence: the referent is first "converted" into a meaning, which is then converted into an utterance. Hence, the 1-to-1 correspondence we would like to find becomes much more difficult to achieve. Combined with the above notes about the relative weights of the "individual" coherences, this all means that the coherence measures become a relatively less reliable indicator of lexicon quality as the complexity of the language games increases.

This does not make coherence irrelevant however. For one thing, it is possible to measure different types of coherence: referent-meaning coherence and meaning-utterance coherence. Additionally, even with a hidden variable in between referent and utterance, good communication still requires a 1-to-n correspondence, where n approaches 1 *in each specific context*. That is, in different contexts the property or feature that sets a referent apart from the other referents in the context is different, but we still want consistency in the utterance used for a certain property or feature.

There is one final aspect that has a deciding influence on the coherence that a population of agents can reach. This is the protocol of the language games themselves. Currently, a game is successful when the hearer *understands* the speaker. Combined with the fact that no entries are removed from the lexicon, even when their strengths have become zero due to underuse or irrelevance, this means that the global vocabulary essentially stops converging when all agents have correct entries for all possible utterances. Preferences for each agent may differ, but since all agents understand all possible utterances, those will not necessarily change any more. (Of course, because strengths are still adjusted, favorable changes may still occur when chance decides so, but the system stops pushing towards a coherent lexicon.)

3.3.1 Vocabulary Coherence

Vocabulary coherence is the analog of the plain coherence measure used in the simplest experiments: it measures how coherent the lexicons are. Although vocabulary coherence is relatively low in both graphs, it can be seen that it is still increasing towards the ends of the graphs, with the standard deviation becoming smaller, which in turn suggests that the vocabularies are becoming more similar.

The coherence measure does not only consider the agents' "active" vocabularies: since no associations are deleted from the lexicons, all associations made from the beginning of the experiments are retained. This includes the associations that are not used any more because they were superseded by new associations, in terms of meaning or form.

Also, a bias in occurrence frequency of the meanings can cause this. Often used meanings may have higher coherence than hardly-used meanings, precisely because of the fact that they are used often. However, using the standard coherence formula (section A.2) these meanings are weighted equally, so that the lesser-used meanings (and their associated forms) account for a disproportionate amount in the final coherence number. (See also the Talking Heads experiment, section 3.4.2.)

Figure 3.11 shows the distribution of meanings in a 10-agent experiment. In this experiment, for each meaning used by the speaker a counter recorded the number of times that that meaning was used. In 80% of the games, the 5 most frequent meanings were used, the next 5 most frequent meanings account for another 10%, and all the other meanings appear in the remaining 10% of games.

3.3.2 Other Types of Coherence

Vocabulary coherence specifically measures the coherence between words and meanings. However, with the introduction of meaning (see fig. 2.4 p. 29) it becomes possible to measure other kinds of coherence too:

Form-Meaning and Meaning-Form Coherence The latter of these two types of coherence corresponds to vocabulary coherence as described above. The former type has not been measured continuously during an experiment, but the graphs shown above of words per meaning and meanings per word for different settings show for individual words how coherent they are, and the corresponding tables show the calculated values at the end of several series of experiments.

Form-Referent and Referent-Form Coherence These types of coherence quantify in how far each word is always used for the same referent, and in how far the each referent is always lexicalized using the same word. These measures have not been implemented for the models in this thesis, because an extra complicating factor in these models is the fact that the context is different from game to game. In other words, the background from which the topic has to be discriminated is different. This means that, even when all other variables are kept the same, in one context an object may be referred to using a word for meaning m_1, while in another context that same object would be referred to using a word for a different meaning m_2.

In models where the context remains the same, these measures can be very revealing about the performance of the model. For example, de Jong and Steels (2003) show that a combination of two these measures defines a *perfect* communication system (a system where referents are mapped one-to-one to words, without synonyms and homonyms), and go on to apply the measures to a variant of the naming game model where the context always contains all objects in the agents' environment. Since any topic objects are always discriminated against *all* other objects, the best meaning to use will always be the same, so that a one-on-one mapping indeed is ideal. De Jong calls these measures *specificity* and *consistency*, respectively. Vogt uses specificity and consistency in similar circumstances (Vogt, 2000, 2002; Vogt and Coumans, 2003).

In principle, the measures could be adapted to work with changing contexts as well. Since the objects in the context are always a subset of the objects in the environment, there is only a finite number of different possible contexts. So, by calculating the specificity and consistency with regard to each different context and averaging all the results, a global specificity number can be obtained. However, when the number of objects in the environment increases, the number of possible contexts increases even faster, which means that the number of data points per context become fewer. Even for moderately large environments, the result would be a large number of contexts with only one or a few data points per context.

Meaning-Referent and Referent-Meaning Coherence These two types of coherence quantify the use of different meanings for a referent or the number of referents that a meaning can refer to. They are subject to the same context-dependence as the previous two types of coherence. These types of coherence have not been measured to date.

3.3.3 Ontology Coherence.

Ontology coherence is a bit of an outsider in this list of coherence measures. It measures similarities, like the other coherence measures, but not of a mapping between two variables. Instead, it measures the similarity between the discrimination trees of the agents in the population.

In the experiments shown in fig. 3.12, ontology coherence rises very quickly (within 200 games even in 100,000-game experiments with 20-agent populations) to 85% and stays there. There also seems to be very little variation between experiments. Thus, the agents' discrimination modules very quickly learn enough distinctions to be able to produce a meaning for a topic in every game.

3.4 Talking Heads

3.4.1 Communicative Success

Figure 3.13 shows the evolution of communicative success of the course of the first Talking Heads experiment (Van Looveren, 2001a,b). In the experiment shown, communicative success is on average 60%, and fluctuates a lot.

An important reason for this is that new agents arrive in the system all the time (see fig. 3.14). While they are learning the language they will cause communicative failures that in turn cause the global communicative success to decrease. Additionally, new words invented by the new agents may "contaminate" the lexicons of the (older) agents they interact with. Also, older agents that disappear from the population take with them their knowledge of the language. These agents appearing and disappearing from the population create a flux of agents in the population. Simulation experiments have shown that too high a flux rate can make the language collapse (Steels and Kaplan, 1998), but in this experiment a core language emerges and remains stable, which indicates that the transmission to new agents is efficient enough. The next section shows that about 10% of the language (the lexicon for 30 meanings, out of 300 in total) is stable and shows high coherence, while for the other meanings there is no meaningful coherence.

3.4.2 Coherence

In the Talking Heads experiment, the above remarks about coherence apply (and actually prompted the analysis). The use of robotic bodies and different physical sites introduces many different types of stochasticity.

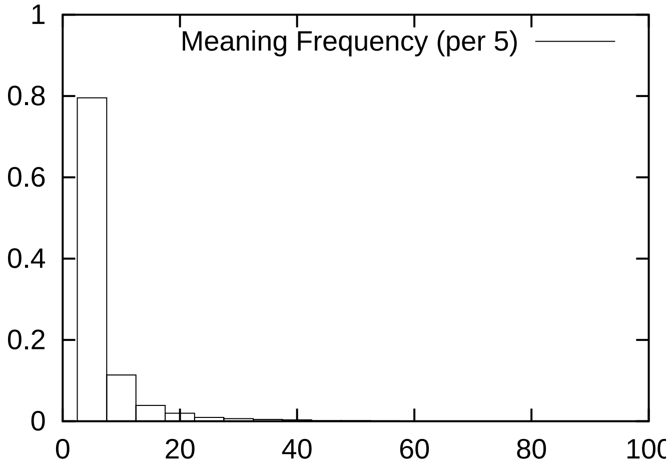
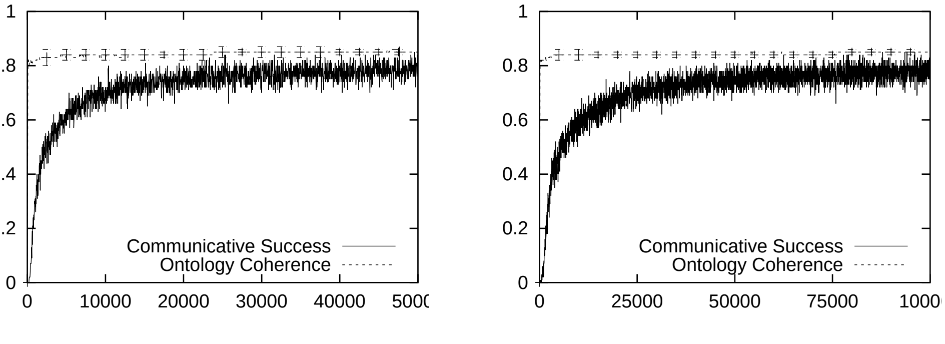


Figure 3.11: Percentual occurrence of meanings (most frequent meanings on the left) in a 10-agent, 7-segment experiment (50,000 games).



(a) (b)

Figure 3.12: Success and ontology coherence in a 10- resp. 20-agent, 7-segment experiment (50,000 resp. 100,000 games; averaged over 10 runs).

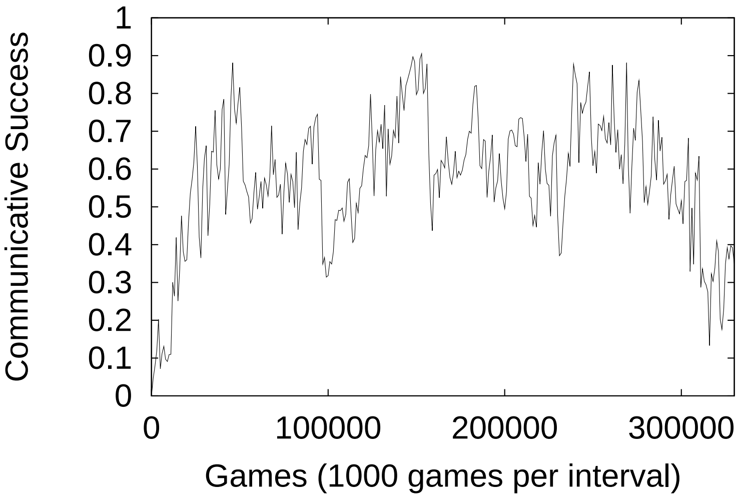


Figure 3.13: Communicative Success in the Talking Heads experiment.

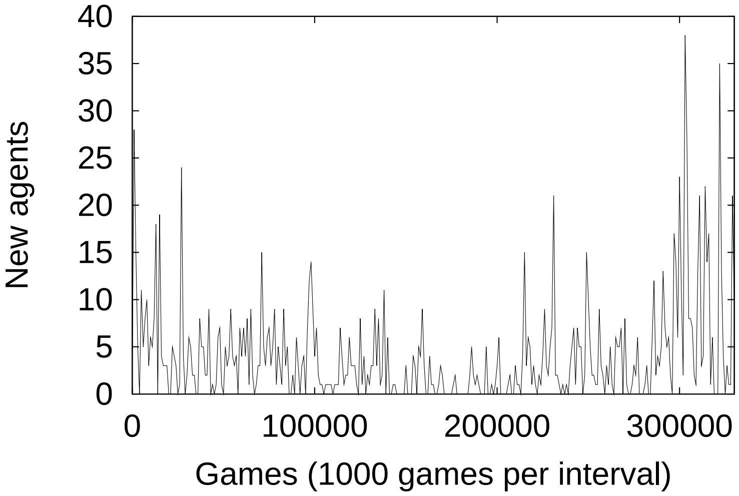


Figure 3.14: Rate of appearance of new agents in the Talking Heads experiment.

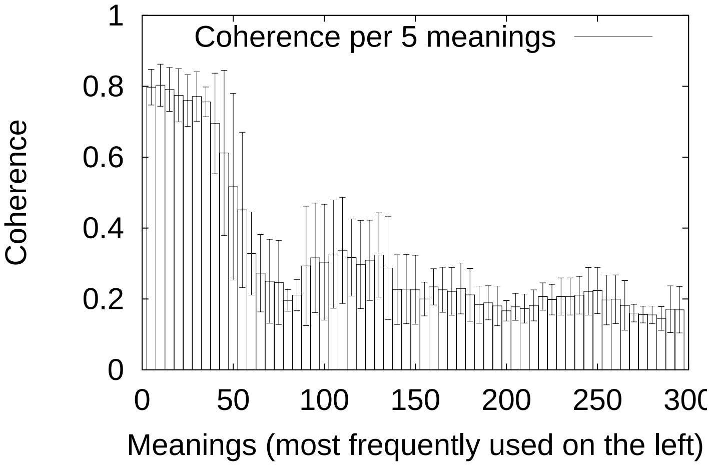


Figure 3.15: Meaning-word coherence in the Talking Heads experiment.

As stated above, in principle it is possible to calculate three types of coherence. Unfortunately, in the Talking Heads experiment calculation of meaning-object or word-object coherence is not possible, because there is not enough information in the database to reconstruct the referents of the interactions after the fact. When meaning-word coherence is calculated in the standard way, averaging over all meanings, the Talking Heads experiment scores a mere 43.2%. This is not very much compared to the 90–100% found in the basic simulation experiments, suggesting that indirect reference to objects results in much worse performance (which would be corroborated by the low success strengths). However, figure 3.15 shows meaning-word coherence in an alternative way. Every bar in the graph shows the average coherence for 5 meanings, with the error bar showing the standard deviation. The meanings are sorted according to their frequency of use in the interactions; the most used meanings are on the left side. It can be seen clearly that for the most frequently used meanings, coherence (almost) reaches the levels achieved in the basic language game experiments. Only the meanings that are less frequently or almost never used, have a very low coherence.

Figure 3.16 shows the meaning-word coherence for the meaning *green* throughout the experiment. In the beginning when no word is dominant yet, many words are competing with each other. Later on, the word *kazozo* becomes dominant, and remains the preferred word for the rest of the experiment, except for a short peak when other words momentarily become more successful.

Figure 3.17 shows how many interactions are covered by how many of the most frequent meanings. It can be seen that in 98% of the interactions, one out of the 50 most frequent meanings is used. This confirms that there is a small number of meanings that are used very often. The extra meaning selection step that is performed by the Talking Heads agents introduces a lot of meanings that are

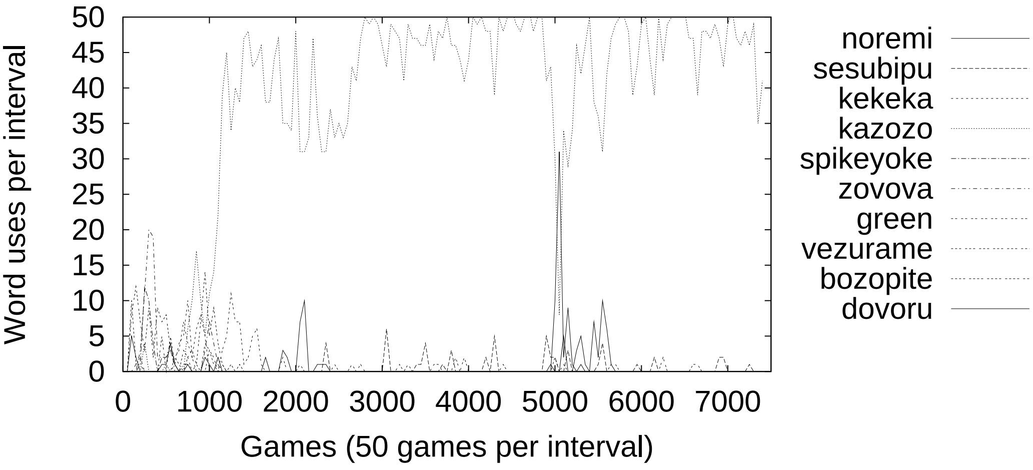


Figure 3.16: Evolution of the lexicon for the meaning *green* in the Talking Heads experiment.

used only once or very few times, which is not the case in the simulation experiments. Since the agents are not capable of removing unused associations from their lexicons, they remain in the lexicon for the duration of the experiment.

3.5 Summary

This chapter described the experiments done with the basic Simple Naming Game model.

- The basic experiments showed good communicative success and vocabulary coherence. This shows that the basic mechanisms for lexicon organisation work.
- As the experiments become more complex, in this case by introducing an explicit meaning layer instead of simply storing the referents in the agents' lexicons, several changes occur. The direct coupling between meaning and referent disappears, which impacts communicative succes and vocabulary coherence. Also it becomes possible to measure different kinds of coherence.
- We also showed how competition between words works by showing the evolution of words for specific meanings over the course of an experiment.
- We saw that one needs to be careful with blindly applying measures. The Talking Heads experiment showed that the basic coherence measure was fooled into giving lower number because it did not take into account the

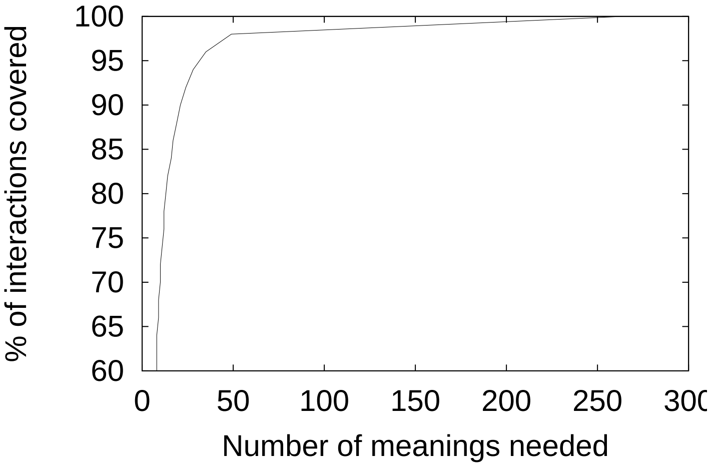


Figure 3.17: Percentage of interactions covered by the most frequently used meanings in the Talking Heads experiment.

relatively higher coherence for words for oft-used meanings and the relatively lower coherence for less-used meanings. Actually, the Talking Heads experiment performed much better than anticipated on the basis of the initial results.

Multi-Word Naming Game

"Big Truck!"

The Simple Naming Game model implemented a very basic form of communication: the capacity to exchange one single word at a time. What sets this system apart from most similar animal communication systems, is that the meanings can be very complex categories, that the association between meanings and forms is arbitrary, and that there can be as many meaning-form associations as needed.

Even though in the Simple Naming Game the lexicon can be large, there will still be limits to the linguistic complexity that can be achieved using an SNG-like communication system, and on the other hand the advantages of being able to use several words to describe a referent are obvious: the lexicon can be smaller (or one can express more meanings with a lexicon of the same size), and more importantly, the language user would be able to deal with unknown meanings without having to learn a new word-meaning pair every time.

Imagine, for example, an agent having the following lexicon:

Word	Meaning	Strength
...	...	...
bizowa	(AREA [0-0.5[)	0.6
tanaka	(HEIGHT [0.75-1])	0.7
...	...	...

If the agent uses the single-word communication strategy from the previous chapters, it would only be able to lexicalise and interpret the meanings (AREA [0-0.5[) and (HEIGHT [0.75-1]). An agent using a communication strategy that can combine words and meanings will also be able to understand, without any extra lexical knowledge, the utterance "bizowa tanaka" (or "tanaka bizowa") and infer that its referent is described by the meaning (AREA [0-0.5[)(HEIGHT [0.75-1]). It would also be able to lexicalise this meaning itself even if it never learned or created an association for this meaning.

This chapter presents a model in which the limitation that utterances contain only one word is dropped. At first sight one could hypothesise that a multi-word utterance is in fact simply several consecutive single-word utterances, with the implicit assumption that they refer to the same referent. But this begs the question: the different single-word utterances are not independent

any more, and the agent will need a mechanism to compose and decompose utterances and meanings and work with the parts. In the SNG, matching the meanings stored with lexical items to the meaning that is to be lexicalised is binary: either there is a complete match, or there is no match at all. In the MWNG, the search mechanism must be able to make partial matches. Similarly, in interpretation the MWNG must be capable of combining the lexicon items that are retrieved from the lexicon, and in case of failure, it must be able to extract the part of the meaning that was not covered and make a new lexicon item with it.
In his tentative plan of language evolution, Jackendoff (2002, see also fig. 1.3 p. 8) proposes two milestones to follow the ability to use symbols in a non-situation-specific fashion. On the one hand, there is the ability to use large numbers of symbols. It is not only necessary to be able to manipulate symbols independently of the situation in which they were created or commonly used, it is also necessary to have enough memory for large quantities of such symbols, and the ability to retrieve them efficiently and detect analogies etc. A model that goes in this direction was studied in the preceding two chapters. Parallel to this step is another step called "Concatenation of symbols." Jackendoff specifically refers to non-grammatical concatenation of symbols, where the symbols are related through the context. This model covers both milestones.

4.1 Related Research

The key concept that describes the difference between the SNG and the MWNG is *compositionality*: the meaning of an utterance is no longer holistic; it is a combination (function) of the meanings of the parts of the utterance. Often it is assumed that compositionality requires some form of syntax, but even if agents are not explicitly using syntax to structure multi-word utterances, the different meaning parts need to be combined in some way. (In our case, the combination function is always simply "and.")
A model in which the agents used two-symbol utterances was developed and described by Crumpton (1994). The model was a direct extension of MacLennan's model (MacLennan, 1990) (see section 2.1), in which the agents performed two "turns" instead of one. A turn involved emitting a signal, doing an action, possibly both (the text is not clear on this issue) or doing nothing. In this model, like in MacLennan's original model, the set of possible symbols is fixed, and the set of actions is also fixed. The population of agents thus has to evolve a successful mapping between these symbols and actions. Like in MacLennan's model, the agents are represented using state machines, and are evolved through a genetic algorithm in which the state transition tables are the genotype. One could say that in Crumpton's model, there is already an implicit form of syntax, because the order of the symbols seems to matter. Crumpton concludes that the agents in his model did to some extent use two-symbol signals, but that attempts to improve the model with e.g. new learning rules, were not successful.
The only other model that we know of in which compositionality has been used without immediately making the step towards syntax is that of (Neubauer,

2002). This model is inspired on Belpaeme's research on colour categorisation (Belpaeme, 2002). In this model, agents perceive colours from a context, and have to learn to differentiate between the colours. The agents create categories of colours and associate these to words. Categories are represented as CIE LUV triplets.

A difference between Neubauer's and Belpaeme's models is that in Neubauer's Model, the agents can generalise over the categories they learn. Rather than specify specific values for all three dimensions of a category, they can specify only one or two, and leave the remaining dimension(s) undefined. When an agent learns a new category, and it corresponds in any of the dimensions with a known category, it can decide to extract the common part into a category that represents that part and has wildcards in the other dimensions. Categories in which complementary dimensions are defined (e.g. $\langle 0.75, *, 0.3\rangle$ and $\langle *, 0.43, *\rangle$ can be merged to form the fully-defined category $\langle 0.75, 0.43, 0.3\rangle$). This in turn can give rise to multi-word utterances. Neubauer notes that multi-word utterances only arise in structured environments, i.e. environments in which the distribution of the colours sampled from the environment is not uniform. When the distribution is uniform, no multi-word utterances arise.

4.2 Semantics

4.2.1 Discrimination Game

The first version of the MWNG used the discrimination game as its meaning source, in exactly the same form as the Simple Naming Game. A distinctive feature set such as the following:

(AREA [0.50-0.75])(HPOS [0.00-0.5])

is as easily expressed using several words as using one word: assuming we do not allow words to have an empty meaning, expressions with a length up to the number of meaning parts are possible.

The original discrimination game as used in the simple naming game was already designed to produce such composite distinctive feature sets. The way it does this is by exploring first the space of permutations it can make with the feature detectors it already has. If this does not produce a satisfactory feature set, new feature detectors will be created.

4.2.2 Predicate calculus

A commonly occurring way of discriminating a topic from other candidate objects in natural language is to relate the topic to other objects: "the book *on the table* is mine." This requires the semantics to not only consider the features of the topic, but also relations with other objects. The discrimination game is however not capable of expressing facts about more than one object.

In order to cope with this fundamental limitation of the discrimination game, a new semantic formalism was developed. It is a limited form of predicate calculus, lacking quantifiers ($\forall, \exists$) and using only $\wedge$ (and) as the connecting operator. With these limitations, these meanings function as filters on the input, like distinctive feature sets.

While it may seem as if we are discarding everything that is useful about predicate calculus by not using quantifiers and other operators, the key is that we gain the freedom of implementing the predicates any way we like, instead of being limited to a pre-implemented discrimination algorithm. For example, we could have predicates that can discriminate PUSH-events from the input.

The use of variables permits us also to signal that the arguments of different predicates in an expression are the same. In fact, we define that they *have to* be different when the variables are different.

The limitations of the current version of the semantics may be overcome in more elaborate versions of it, to further increase its expressive power.

Meaning Construction

Generating new meanings is done through a search process that starts from an existing "base" *operation*. Given a set of predicates, the search process systematically tries to add new statements (predicates + variables) to the operation, each time checking if the expanded meaning is distinctive for the topic. If so, the new meaning is returned; if not, another predicate is added, etc., until the meaning is distinctive or until no more predicates can be added. The search is breadth-first search that tries to add all (relevant) predicates in turn first, before utterly expanding the meaning. This way, the meanings are kept relatively shorter.

Figure 4.1 gives an example of a search tree as it is generated during the search process. The root node is given; in this case it is one single statement, but as indicated above, any composite operation can be used as the starting point for expansion. Every level in the tree represents the addition of a new statement to the operation. The subtree of a node shows how different candidate statements are added in turn to the same operation.

The key of the search process is the function that adds a new statement to an operation. This presents a number of potential problems, which all relate to a form of *correctness* that the operations must have before they can be meaningfully evaluated by the engine that interprets meanings:

Variables to use: a composite operation contains a number of variables that connect the parameters of the different composing predicates with each other. When a new statement is added, the variables have to be reused in a sensible way. For example, a statement must not be an "orphan:" at least one of the variables in the argument list of the statement must appear in one of the other statements in the operation. (This implies also that new variables can only be introduced by predicates with at least two parameters.)

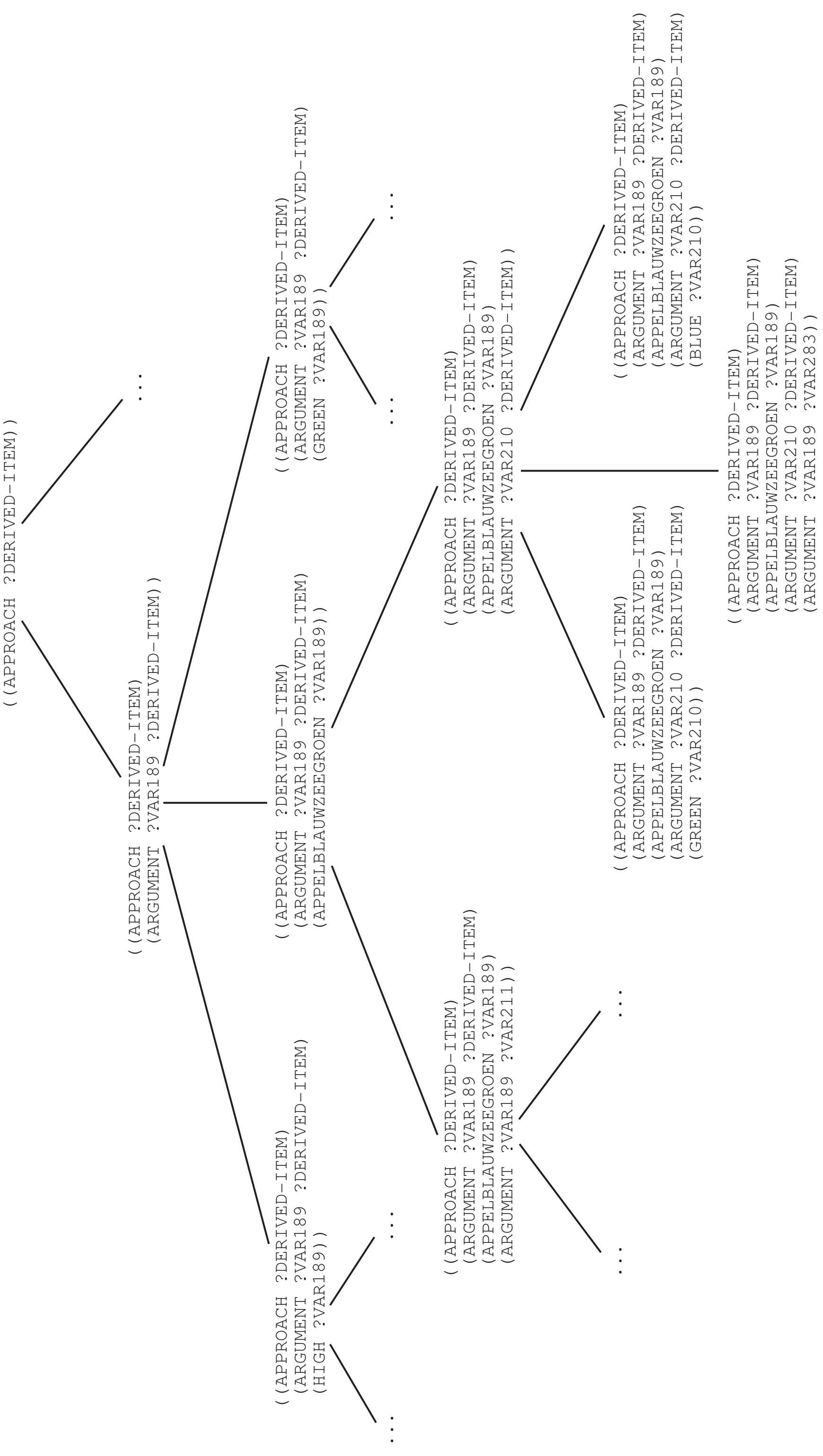

((APPROACH ?DERIVED-ITEM))
...
((APPROACH ?DERIVED-ITEM) (ARGUMENT ?VAR189 ?DERIVED-ITEM))
((APPROACH ?DERIVED-ITEM) (ARGUMENT ?VAR189 ?DERIVED-ITEM) (GREEN ?VAR189))
...
...
((APPROACH ?DERIVED-ITEM) (ARGUMENT ?VAR189 ?DERIVED-ITEM) (APPELBLAUWZEEGROEN ?VAR189))
((APPROACH ?DERIVED-ITEM) (ARGUMENT ?VAR189 ?DERIVED-ITEM) (HIGH ?VAR189))
...
...
((APPROACH ?DERIVED-ITEM) (ARGUMENT ?VAR189 ?DERIVED-ITEM) (APPELBLAUWZEEGROEN ?VAR189) (ARGUMENT ?VAR210 ?DERIVED-ITEM))
((APPROACH ?DERIVED-ITEM) (ARGUMENT ?VAR189 ?DERIVED-ITEM) (APPELBLAUWZEEGROEN ?VAR189) (ARGUMENT ?VAR189 ?VAR211))
...
...
((APPROACH ?DERIVED-ITEM) (ARGUMENT ?VAR189 ?DERIVED-ITEM) (APPELBLAUWZEEGROEN ?VAR189) (ARGUMENT ?VAR210 ?DERIVED-ITEM) (BLUE ?VAR210))
((APPROACH ?DERIVED-ITEM) (ARGUMENT ?VAR189 ?DERIVED-ITEM) (APPELBLAUWZEEGROEN ?VAR189) (ARGUMENT ?VAR210 ?DERIVED-ITEM) (GREEN ?VAR210))
((APPROACH ?DERIVED-ITEM) (ARGUMENT ?VAR189 ?DERIVED-ITEM) (APPELBLAUWZEEGROEN ?VAR189) (ARGUMENT ?VAR210 ?DERIVED-ITEM) (ARGUMENT ?VAR189 ?VAR283))

Figure 4.1: An example of a generation tree. Note that the operations in the tree use the actual LISP representation.

Predicates to add: in principle it is of course possible to add any statement to an existing operation, but in most cases the addition will not be meaningful, for exampe because it uses variables that do not occur elsewhere in the operation, or because a variable that is supposed to contain an object has to contain an event in the new statement.

The solution to avoid confusion about the variables' contents is to implement a *type system*, similarly to type systems in programming languages. Variables are of a specific type (in this implementation there are *objects* and *events*). This limits the statements that can be added to those that use variables that make sense in every argument position of the predicate.

Several identical solutions: in a breadth-first search process, it is possible that the same solution is found several times. This has to be avoided; operations have to be considered *equivalent* when the names of their variables differ, but when the variables are used in the same positions. To detect such operations, a standard pattern-matching algorithm is used.

Duplicate statements: statements that are already part of an operation, must not be added a second time.

Figure 4.2 shows the pseudo-code of the generation algorithm (Van Looveren, 2002). The bulk of this algorithm is made up by the algorithm that adds a statement to a pre-existing operation, and which ensures the correctness of the new operation. It works, briefly, as follows:

- The algorithm is supplied with the predicate for which a statement is to be added, and will, based on the variables already in the original operation, generate the possible argument lists.

  Take for example the following pre-existing operation:

  $$\text{APPROACH}(\mathit{item}) \wedge \text{AGENT}(x,\mathit{item}) \wedge \text{PATIENT}(y,\mathit{item})$$

  This operation has variables x, y and *item*, where x and y are of type object, while *item* is of the type event.

  Suppose we want to add to this operation a statement containing RED.

- We do this by collecting, for each formal parameter of the new predicate, the existing variables of that type.

  The predicate RED has one parameter, namely the object that is to be tested. Possible candidates for this argument are x and y. The variable *item* is not a candidate because it is of type event.

- To introduce variation and innovation, it is also necessary to introduce new variables. This is done by adding, to the list of variables for each parameter, a new variable.

  This means that in the example there are three candidate variables: x, y and a new variabele *new*.

- The possible parameter lists can then be generated by making combinations of the lists of variables, such that all variables appear with all other variables as parameters.

  The possible new statements are then:

  $\text{RED}(x)$
  $\text{RED}(y)$
  $\text{RED}(new)$

  The last of these statements cannot be added to the operation, because *new* does not appear in any of the other statements of the operation. (Remember that RED only has one parameter, and at least two parameters are required for introducing new variables.)

  The new statements can then be added to a copy of the pre-existing operation, so that there can potentially be many new operations. A possible strategy to limit the number of new operations could be by generating (e.g. randomly) only one new statement instead of all possibilities.

In practice it turns out that the newly generated operations are relatively small, and that generation is fast. The program has the capability to generate several solutions; however, when this option is used, it can indeed take a long time before all the requested solutions are found.

4.3 Form

Having more than one word in an utterance poses a subtle problem. At the symbolic level, it does not immediately surface: an utterance is merely a set of forms. This is however a simplification of the real situation. When humans talk, they do not talk in individual words. Speech is a continuous stream of sounds, which are separated into individual words by the hearer. Simulating this requires merging the forms into one large form, with no separation marks between the constituent subforms. The hearer has to split the utterance again according to the knowledge it has of the language.

The bulk of the experiments that have been done with the Multiple-Word Naming Game have been done using the simpler scheme, where forms are kept separate and hearers thus know what the constituing forms are of an utterance. However, as a preparation for the hybrid model, in which the agents have the option to use either the Single-Word or the Multi-Word strategy for each utterance, section 4.3.2 will describe specific issues that arise when agents have to split utterances themselves.

4.3.1 Explicit Subform Boundaries

Figure 4.3 shows the substeps involved in producing and interpreting a multiword utterance. As can be seen from the figure, these substeps are almost identical in the two cases. The only difference is that in the interpretation case there

```
generate(partialop)
  solutions ← null
  partialops ← list(partialop)
  repeat
    partialoperation ← dequeue(partialops)
    forall operations op
      newops ← add-statement(partialop,op)
      forall operation newop in newops
        result ← run(newop)
        if result == successful
          enqueue(partialops,newop)
        else if result == valid
          enqueue(partialops,newop)
        endif
  until partialops is empty

add-statement(partialop,op)
  solutions ← null
  possibleargs ← null
  forall arguments a of op
    possibleargs ←
          possibleargs + variables-of-type(partialop ,type-of(a))
  possiblealists ← permute(possibleargs)
  forall argument lists arglist in possiblealists
    newstatement ← append(op,arglist)
    if not duplicate-statement(partialop,newstatement) or
           isolated-statement(partialop,newstatement)
      newop ← partialop + newstatement
      enqueue(solutions,newop)
    endif
  return(solutions)
```

Figure 4.2: Pseudo-code for the generator.

are different constraints posed upon the hypotheses that are being generated. In the lexicalisation case, the constraint is the meaning while the choice of words is free, while in the interpretation case the constraint is the words, while the meaning is free (i.e. all possible meanings are candidates for discriminating the topic).

From the figure one can also see that there is only one subtask that is significantly different between single word naming games and multiple-word naming games. Reflection about and comparison of the existing single word naming games and the multiple word naming games revealed that a lot of the issues were similar, and that by making the division in subtasks clearer, it would be possible to partition them into subtasks that are identical and those that are different. It turned out that the set of different subtasks contained only one element: the composition of the hypotheses. All other steps in both the production and interpretation process are the same. This means, at least technically, that there is no real *fundamental* difference between the standard single word naming games and the multiple word naming games. Only one step of the process has become more complex; actually it seems that the "find hypotheses" step of the single word naming game is a special case of the "find hypotheses" step of the multiple word naming game. This means that when the "find hypotheses" module in the multiple word case is limited to producing hypotheses that contain only one association, one gets effectively a single word naming game.

Alternatively, the MWNG is a generalisation of the SNG, where some of the mechanisms work in a more general way.

Production. The production process proceeds in a way roughly similar to the Simple Naming Game, except that in this case the meaning does not have to match completely with the lexicon entries' meanings. Lexicon entries of which the meanings are subsets of the whole meaning are also looked up. The lexicon lookup phase is then followed by a combination phase in which lexicon entries are combined to yield complex meanings. Each combination gets a score based on the score of its components, and the best one is then selected for realisation (see section 5.1.2 for algorithm details).

Interpretation. The interpretation process also works in a way similar to production: first a lexicon lookup phase, and then a phase in which the lookup results are combined into hypotheses. This time the lexicon lookup phase is based on the words in the speaker's utterance, and making combinations is slightly more straightforward compared to the production phase.

Because words may have several meanings, many hypotheses may result. They each have to be evaluated to find their interpretations (the values of their variables). Only those hypotheses that allow a single interpretation can contain the topic, since it is a requirement that the topic's description is unique.

Figure 4.4 shows the pseudo-code that is used for both production and interpretation. The main function, `find-hypotheses`, takes two arguments, *Word* and *Meaning*. By setting either of them to `null`, one can choose between production

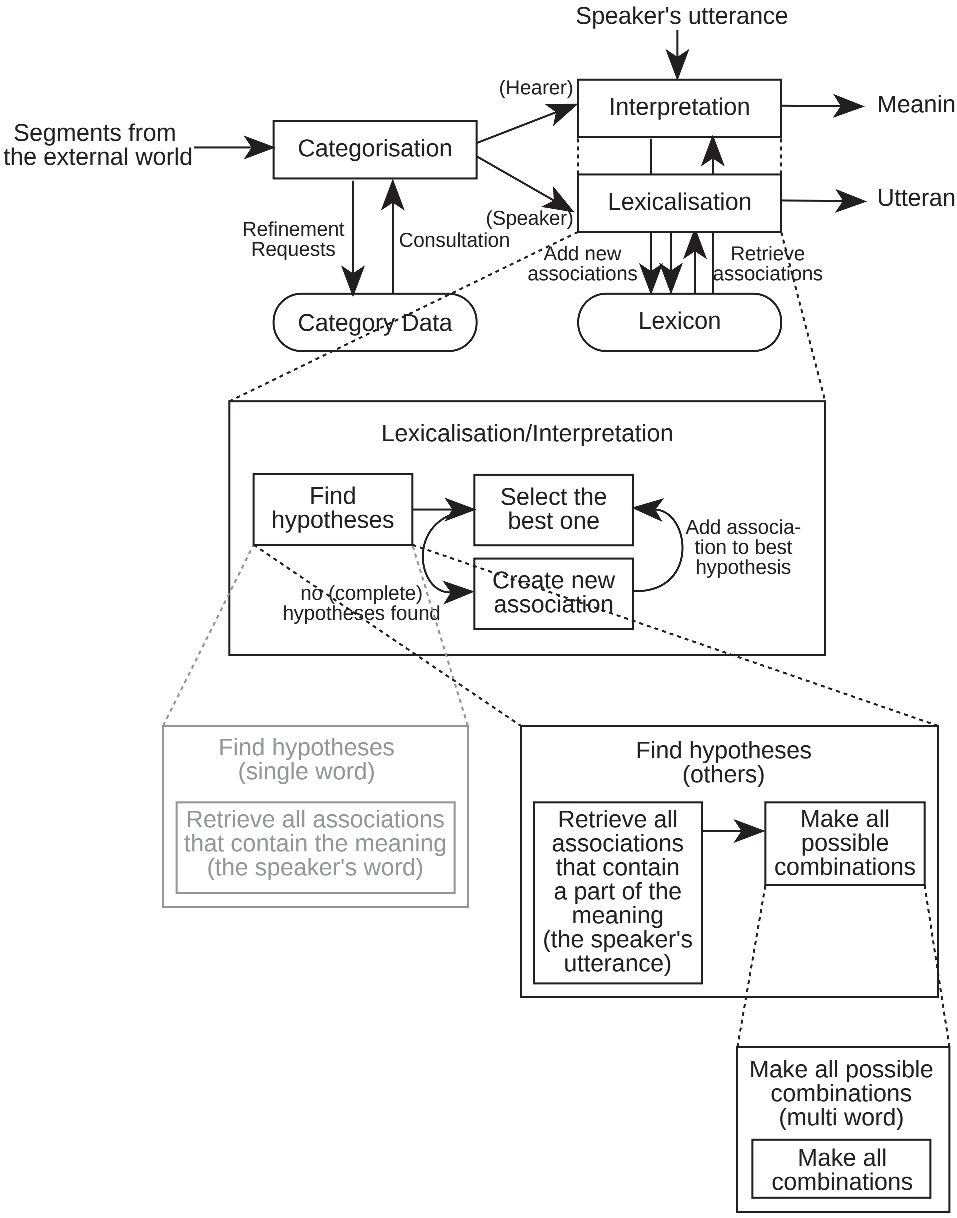

Speaker's utterance
(Hearer)
Interpretation
Meaning
Segments from the external world
Categorisation
Lexicalisation
Utterance
(Speaker)
Refinement Requests
Consultation
Add new associations
Retrieve associations
Category Data
Lexicon
Lexicalisation/Interpretation
Find hypotheses
Select the best one
Add association to best hypothesis
Create new association
no (complete) hypotheses found
Find hypotheses (single word)
Retrieve all associations that contain the meaning (the speaker's word)
Find hypotheses (others)
Retrieve all associations that contain a part of the meaning (the speaker's utterance)
Make all possible combinations
Make all possible combinations (multi word)
Make all combinations

Figure 4.3: "Pipeline" of the multi-word utterance system

```
retrieve-partial-covers(Words,Meanings)
  associations ← agents-words-to-meanings-table()
  groupedPartialCovers ← null
  for each group ∈ associations
    thisGroup ← null
    if ((word(first-element(group)) ∈ Words)
        or
        Words = null)
      for each association ∈ group
        if ((meaning(association) ⊂ Meaning)
            or
            Meaning = null)
         thisGroup ← append(thisGroup,association)
      groupedPartialCovers ← append(groupedPartialCovers,theGroup)
  return groupedPartialCovers

find-hypotheses(Word,Meaning)
  associationGroups ←
  retrieve-partial-covers(Word,Meaning)
  hypotheses ← list(emptyHypothesis)
  for each group ∈ associationGroups
    newHypotheses ← null
    for each hypothesis ∈ hypotheses
      for each assoication ∈ group
        if not(word(association) ∈ words(hypothesis)
               or
               meaning(association)∈ meaning(hypothesis))
          newHypotheses ← append(newHypotheses,
                                 add-association(
                                     copy-hypothesis(hypothesis),
                                     association
                                     )
                                )
    hypotheses ← newHypotheses
  return hypotheses
```

Figure 4.4: Pseudo-code for interpretation and production

and interpretation mode: for production, the meaning is known and the words are not, for interpretation, the words are known, but the meaning is not.
The function `retrieve-partial-covers` takes care of this bit. It assumes a lexicon in which associations are grouped by word, i.e. all associations that contain the same word are grouped together. When *Words* is `null`, it collects associations from all groups, if not it collects associations from those groups for which the word is contained in *Words*. Similarly, within each group, either all associations are collected (*Meaning* is `null`) or only those associations are collected for which the meaning is a part of *Meaning*. In any case, the result is again a list of lists of associations, grouped by word.
The `find-hypotheses` function will then combine the associations such that as much as possible of either the words or the meaning is covered.

Learning. When no suitable hypothesis/interpretation pair can be found, the hearer has to repair its lexicon. In that case, the speaker points out the topic so that the hearer can derive a meaning for it. Using this meaning, the speaker's words and its own lexicon, the hearer can try to establish the correct word-meaning mappings. By default, it only tries to fix cases in which the meaning of one word was hypothesised incorrectly. With more words there are too many possible word-meaning assignments so that the probability of the chosen assignment being the corect one is too low.

4.3.2 No Explicit Subform Boundaries

Splitting utterances is not necessarily complicated. It requires a relatively simple extra step while parsing the utterance. The hearer should match the utterance against the words it has in its own lexicon, to see if any of the words is a part of the utterance. If it is, the word should be isolated, and the rest should be checked against the lexicon. This continues until either the utterance is fully split, or until there is one part left over that the hearer could not identify.
A subtlety in the current implementation is that words may be prefixes of other words, e.g. an agent may have both a word "me" and another word "meli" with a different meaning. When it parses an utterance "meligi," both "me ligi" and "meli gi" could be valid parses. This introduces extra combinatoriality in the system. However, it does not undermine its general functioning.
A problem that sometimes arises and that may compromise the goal of reaching a communication system is the following. Imagine an interaction in which the speaker has the following (partial) lexicon:

Word	Meaning	Strength
...	...	...
wabaku	(RED [0.5-1])	0.7
tepi	(HPOS [0-0.25[)	1.0
...	...	...

and wants to construct an utterance for the meaning

(RED [0.5-1])(HPOS [0-0.25[)

The result would be "wabakutepi." If a hearer does not know either "wabaku" or "tepi", it will conclude that the utterance consists of a single symbol, and learn it as such.
Later on, when the same agent plays the role of speaker, it may itself use the word "wabakutepi" as a part of a multi-word utterance. This can lead to a vicious circle, in which words get longer and are continuously added to the agents' lexicons.
Other computational models of language emergence do not have this problem. For example, in the Iterated Learning Model (ILM) (Kirby, 1998; Kirby and Hurford, 2002) this problem was never an issue. The agents in the ILM use a context-free parsing and induction algorithm for its syntax component, in which basic symbols can be combined using rules, such that the lexicon is actually a part of the grammar. Essentially, in these models, splitting utterances comes free as a side-effect of the parsing algorithm.
The way in which grammar induction works in this algorithm, is by storing utterances and trying to generalise on a regular basis. For example, if the utterance "wabakutepi" was already stored as a unit meaning

(RED [0.5-1])(HPOS [0-0.25[)

and the algorithm encounters "wabakugipa" with the meaning

(BLUE [0.5-0.75[)(HPOS [0-0.25[)

it will be able to generalise that "wabaku" means (HPOS [0-0.25[).
A similar mechanism can be found in Neubauer's model for compositional colour categories, but there it is used for generalising meanings (Neubauer, 2002). In this model an agent stores colour samples, and compares new colour samples to the ones it already stored. If any parts of the new sample matches with any parts of a known sample, then the agent can generalise the matching part, using wildcards for the different parts.

4.4 Evolutionary Transition

In their critique of Jackendoff's (2002) book, Zuidema and de Boer (2003) state that while Jackendoff's proposal for milestones in the incremental development of language (see fig. 1.3) is a big step towards recognizing that no dramatic scenarios are needed to explain the existence of modern language, Jackendoff's argumentation is too verbal and unspecific to actually provide a reasonable explanation. They argue that good evolutionary explanations should state the assumptions they make about genetic and phenotypic variation, and the selection pressures that act upon the system to cause the stated evolutionary shift.
As explained in section 1.2.2, we do not want to go into the biological details of language evolution in this thesis. We do present in this section a model that attempts to address the problem of which selection pressures can come into

play when trying to explain the transition from the single-word naming game to the multi-word naming game.

An argument that has often been used to motivate a large innate, biologically evolved component for language is the so-called *poverty of stimulus* argument. The gist of this argument is that children only receive limited input during their language learning period, so that strong innate biases and algorithms need to be present in the child's brain to correctly guide the acquisition of language. The Iterated Learning Model has been used to operationalise the poverty of stimulus argument (Kirby, 2002; Brighton, 2002) and test whether imposing a learning bottleneck may have an influence on the emergence of structure in a language. Their results showed that indeed, when the amount of linguistic input for learning is limited, a grammar induction algorithm will succeed in turning initially coincidental regularities into a grammar. In the experimental results, this shows up as a transition from a holistic language to a compositional, syntactically structured language. However, the ILM model overlooks a number of issues that may have an influence as well. For example, in the ILM the student is passive in the sense that it only learns, and does not produce linguistic utterances itself. Also, the discovery of syntax depends on the presence of chance regularities in the student's input data; the issue of whether or not the language is actually successful in use is not considered (Steels, 2002). The ILM also does not address the issue of incremental evolution; its agents are already equipped with a grammar induction algorithm that initially simply performs below par.

The way in which we would like to proceed in our model is by testing whether the mechanisms for distributed negotiation, that we have implemented and studied in the previous two models, can also work for whole communication strategies. We assume that no biological changes are necessary to change from using one strategy to the other; only a different way of using available cognitive resources. Additionally, we would like to use the measurements used for the SNG and MWNG models, as inspiration for the internal pressure(s) that work(s) on the communication strategies.

The model presented here is equivalent to the previous models, except that the agents choose, in every interaction, the strategy with which they will produce or interpret the utterance. Thus, an agent can use the single-word strategy in one game, and the multi-word strategy in another game. The speaker and the hearer in an interaction do not know which strategy their antagonist is using, because utterances are always transmitted as single forms (see section 4.3.2). Hence, a single-word speaker can interact with a single-word or multi-word hearer, and a multi-word speaker can interact with a multi-word or single-word hearer.

In every interaction, the agents individually choose which strategy they use to produce or interpret an utterance. The actual strategy an agent chooses is governed by how well each strategy performed in previous interactions, and how well it manages the resources internal to the agent, such as the lexicon. To keep track of their performance, strategies have strengths, just like the meaning-word associations in the lexicon. After every interaction, the result (together with other parameters such as lexicon size) is used to update the strength of the strategy that was used, according to an evaluation criterion. The question we try to answer with this model is precisely what the effect is of different evaluation criteria on the strengths of the different modules.

Because the agents do not know which strategy the other agents in the population prefer to use, agents can use only their own internal data to evaluate the performance of the strategies. This means that it can be based only on data that the agent itself can collect, without examining other agents' data or using a global measure that can sense the direction in which the population evolves. Also, agents do not change strategies abruptly. Rather, when a strategy increases in strength, it will gradually be used more often by the agent, until it becomes the dominant one.

4.4.1 Game Result

An obvious measure (and pressure) is the result of each game: whether it is successful or not.[1] If one strategy is more successful than the other, it is clear that an agent whose task it is to communicate as successfully as possible, will gravitate towards the more successful strategy.

It is also not difficult for an agent to keep a record of its own successes and failures using each strategy; in fact, this is the most direct feedback an agent is likely to get. While the lexicon is shared by all strategies, the way in which utterances are produced and the way in which utterances are interpreted will be different. Hence, whenever a strategy is used, its result can be remembered and used to update that strategy's strength.

4.4.2 Lexicon Size

Another way to judge the performance of a strategy is the size of the lexicon. Assuming that an agent's memory capacity is not infinite, this means that it will not be possible for an agent's lexicon to keep expanding. More specifically, for this criterion we assume, following de Jong (2000), that an ideally efficient communication system would have a one-to-one mapping between the meanings and forms in its lexicon. This is what he measures with his *specificity* and *parsimony* measures; these measures are calculated at the level of the population however. Disregarding other sources of noise such as transmission from speaker to hearer, such a lexicon would allow perfect communication, because every word used would map unambiguously to one meaning.

The way in which this criterion measures lexicon quality is simply by dividing the number of words by the number of meanings. For "perfect" communication in the sense of De Jong, this number should be one: one word for every meaning, and no superfluous meanings or words. The number of words and the number of meanings each form one dimension of a two-dimensional matrix.

[1] We use the term *game result* here instead of communicative success to avoid confusion with the global measure, even if they are obviously closely related.

In the "ideal" case, only the diagonal will be filled in. Our simple division does not take this into account, but it gives a good approximation of the quality of the lexicon.

Of course, natural language does not have a one-to-one mapping between words and meanings. Natural language does contain different words that have the same meaning, or different meanings that are lexicalised using the same words. (Often, however, subtle differences in meaning can be found nevertheless, so that real homonymy and synonymy actually are rarer than one would think.)

4.4.3 Lexicon Expansion

Instead of considering the size of the whole lexicon, it might be interesting to look at the rate of expansion of the lexicon. Here again, the goal is to keep the lexicon as small as possible, or at least put a damper on continuous growth of the lexicon.

Rather, the pressure is on strategies to refrain from adding associations to the lexicon. Whenever a strategy adds an association to the lexicon, its score will be decreased.

Two different experiments have been done using this pressure: one experiment in which there is both positive and negative feedback (positive feedback when no new association is added, negative feedback when a new association is added), and an experiment in which there is only negative feedback. In both cases, there is lateral inhibition, i.e. when one strategy receives positive or negative feedback, the other one will receive negative or positive feedback, respectively.

4.5 Summary

This chapter describes a multi-agent model in which the agents can use composite utterances to describe meanings. The compositionality is not compositionality in the usual sense, which implies the presence of syntax, but nevertheless utterances can be composed of several parts. The constraint to which the agents implicitly conform, is that all parts of the utterance refer to the same external referent.

We have detailed the semantic and syntactic mechanisms, and described how they are implemented. We also looked briefly at the problem faced by agents using long utterances: a realistically modeled perception cannot show the "pauses" between symbols. In real language, an utterance is a stream of sounds with no separation in between apart from explicit pauses, e.g. for breathing.

We also described a model that we want to use to study the transition between different ways of creating and interpreting utterances. In this model, agents can choose whether they want to use the single-word syntactic strategy or multi-word strategy to construct utterances. The choice between the two strategies is not voluntary, but guided by selection pressures that are internal to the agent.

Different possible selection pressures were described: based on communicative success, lexicon size or the rate of lexicon expansion.

Multi-Word Naming Game: Experiments

THE PREVIOUS chapter described the Multi-Word Naming Game model. In this model, the agents can use utterances with lengths greater than one, although they are still limited to one referent. In this chapter we describe the results of the experiments that have been done with this model. We also look at the experiments done with the hybrid SNG/MWNG model. Section 5.1 shows the results of the basic experiments. Section 5.2 looks in more detail at the competition that arises between different words for the same meaning and different meanings associated to the same words. Section 5.3 looks at a specific problem for multi-word utterances, namely whether the hearer can perceive the boundaries between the forms or not. Finally, section 5.4 describes the experiments done with the hybrid model, and the different pressures that have been proposed to look at the transition between different stages of language.

5.1 Basic Experiments

5.1.1 Discrimination Game Algorithm Details

See section 3.2.1.

5.1.2 Predicate Semantics Algorithm Details

Merging predicate calculus expressions is done using a logical connector: $\vee$ (disjunction, "or") or $\wedge$ (conjunction, "and"). More sophisticated types of logics have other connectors as well, related e.g. to moments or intervals in time where the parts of the expressions are located (before, after,...).

The implementation used here uses always and only the $\wedge$ connector. This means that both parts of the expression must be true for the whole expression to be true. The fact of adding meaning to an existing meaning thus effectively makes the expression a stricter filter for the perceptual input.

5.1.3 Multi Word Naming Game Algorithm Details

Context Size

In the MWNG there is a difference between the number of objects (or segments, after perception) in the environment n_{obj}, and the context size for each interaction n_c. The parameter n_{obj} denotes the total number of objects that exists in the agents' environment, while the parameter n_c denotes the number of objects that will be perceived by the two agents participating in a specific interaction. The set of objects perceived by the speaker and hearer of an interaction is the same, and is always a subset of the global set of objects.

For the experiments reported, n_{obj} is 10, and the context size n_c is 7, unless stated differently.

Word and Meaning Combinations

The strength of a combination of associations is calculated from the strength of the individual associations that make up the combination. The formula is slightly different for the speaker and the hearer. For a combination of associations $a_1 ... a_n$, the strength is calculated as follows for the speaker:

$$\textit{strength} = \frac{\sum_{i=1}^{n} \text{strength}(a_i)}{i} * \# \left(\bigcup_{i=1}^{n} \text{meaning}(a_i) \right)$$

i.e., the average of the strengths of the associations, multiplied by the number of components of the meaning of the combination. For the hearer, the strength of the combination is simply the average of the individual association strengths, i.e. the first half of the above formula.

5.1.4 Communicative Success

Both the small-scale and larger-scale experiments with the Multi-Word Naming Game in which communicative success is measured (figs. 5.1 and 5.2) show similar patterns; the former for a very basic 1000-game two-agent experiment, and the latter for a series of 10-agent experiments over 20000 games. As in the SNG, communicative success gradually rises to a very high value, in this case approximately 90%–95%. This indicates that the MWNG agents are very capable of building a reliable communication system.

Figures 3.4 (a) and (b) (p. 47) show Simple Naming Game experiments done in similar circumstances to the experiments shown here. Comparing the results for communicative success, we see that the MWNG agents seem to perform better than the SNG agents, reaching 90% success as compared to 80% in the SNG case.

The lower success in the SNG case was attributed to the fact that in those experiments, agents need to create new words for previously unlexicalised meanings.

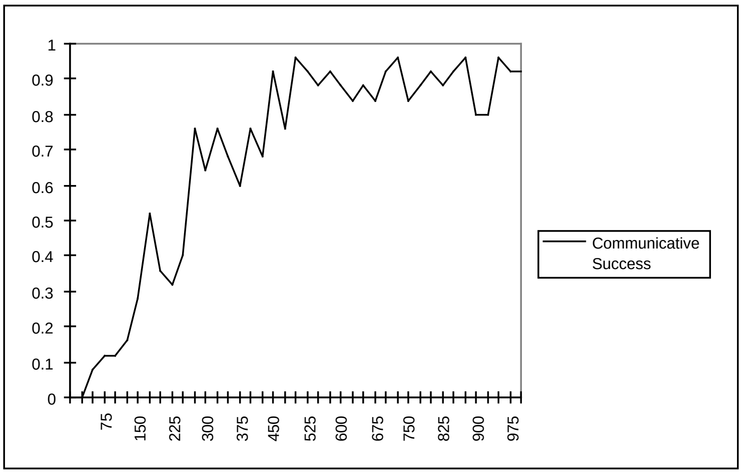


Figure 5.1: Rate of communicative success measured over 1000 language games played by two multiple word agents.

MWNG agents (presumably) have to do this less, being able to compose utterances for unknown meanings on the fly, provided they already know forms for the different parts of the meaning.

5.1.5 Coherence

In terms of coherence, on the other hand, the SNG seems to be doing better than the MWNG (50%–60% in the SNG, fig. 3.4 p. 47, as opposed to 45% for the MWNG). This is caused by the same phenomenon that caused the coherence in the Talking Heads experiment to register low: meanings that are used often and have high coherence, indiscriminately combined with meanings that are used less often and have low coherence. For good measure, experiments should be done with a coherence measure that is weighted, weighing heavily used meanings more than little used meanings.

5.1.6 Lexicon Size

The big problem with the Simple Naming Game agents in complex environments, as was pointed out already several times, is that for every new situation they want to describe, the have to create a new form-meaning association.

One of the motivations for developing a version of the naming game in which agents can *reuse* form-meaning associations that are already in their lexicons, is precisely to avoid having to create new associations all the time.

The most straightforward way of measuring whether the intended goal is met, is simply by monitoring the actual lexicon sizes, and compare them directly. Of course, this assumes that the experimental settings are similar enough to permit direct comparisons.

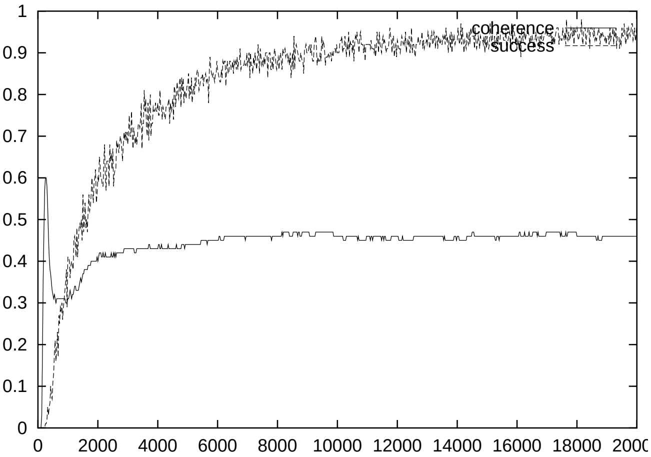


Figure 5.2: Success and coherence in a 10-agent experiment (20,000 games; averaged over 10 runs

Figure 5.3 shows a graph depicting lexicon sizes for both the Simple and Multi-Word naming game agents. In the very beginning of the experiment, the curves run more or less together, but already very soon the curves diverge, with the SNG curve continuing along its path, and the MWNG curve becoming much flatter.

It is clear that the SNG agents keep creating associations, and in this case, the curve shows no tendency to flatten out at some point. For the MWNG agents, the curve does not flatten out completely either, indicating that even very late in the experiments, it is still necessary from time to time to lexicalise new meanings. After about one third of the experiment, however, such events have become very rare.

5.1.7 Efficiency

Figures 5.4 and 5.5 show the same trends, but measured in a different way. Instead of looking only at the numbers of words in the lexicons, we look at the number of actual meanings that the agents have to express. Comparing this with the number of different meanings in the lexicon, we can measure how *efficient* the agents' lexicons are at lexicalising the meanings the agents encounter in their world. The *meanings in lexicon* curve shows the average number of different meanings that agents have in their lexicons. The number of *expressed meanings* is the number of meanings that the agents have had to lexicalise as speakers in a language game.

Two measures have been developed: the first one simply tracks the number of different meanings in the lexicon of an agent, and the other one keeps track of the number of meanings that the agent it monitors tries to express.

In the single word case the agent has to be able to retrieve every meaning it

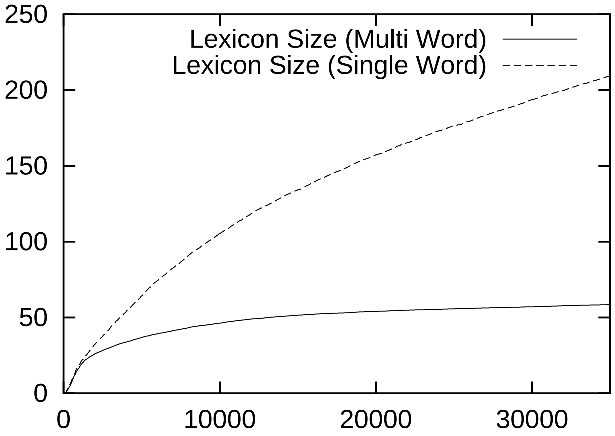


Figure 5.3: Lexicon Sizes for Single-Word and Multi-Word experiments (both 10 agents, 7-segment meanings, 35,000 games; averaged over 10 runs)

wants to express in the lexicon, otherwise it has to invent a new word and add it to the lexicon. This means that in this case the two curves coincide, because the agent can express a new meaning only at the expense of lexicon expansion. In the multiple word case on the other hand, even meanings that are not explicitly in the lexicon may be expressed without the need for a new word, provided they can be divided into meanings for which there are already associations in the lexicon. This means that the lexicon size curve should be lower than the expressed meanings-curve.

Figure 5.4 shows the two curves for a multi-word agent in a simple experiment with a population of two agents; the figure for the other agent shows a similar picture. The scale on the left shows the (absolute) number of meanings.

During the first 75 games the curves are the same, which indicates that the agent uses only single word expressions to express its meanings, and that for every meaning it has to create a new word. After that, the "expressed meanings" curve rises a lot faster than the "meanings in lexicon" curve, which indicates that from that point on the agent indeed reuses the words it already knows to describe the meanings it wants to express.

Figure 5.5 shows the same two curves for a single-word agent, also in a population of two agents. As you can see, the curves coincide exactly, which means that the lexicon of the single word agent indeed grows much faster than that of the multiple word agent.

The graphs also show that even though the multiple-word agent starts using multiple-word expressions very early on, the "meanings in lexicon" curve still rises at about the same rate as for the single-word agent until after 225 games. This is because the multiple-word agent still learns new meanings (mostly when it is the hearer) until it has a sufficiently large basic repertoire of meanings.

Figures 5.6 and 5.7 show the efficiency measures applied to a series of more

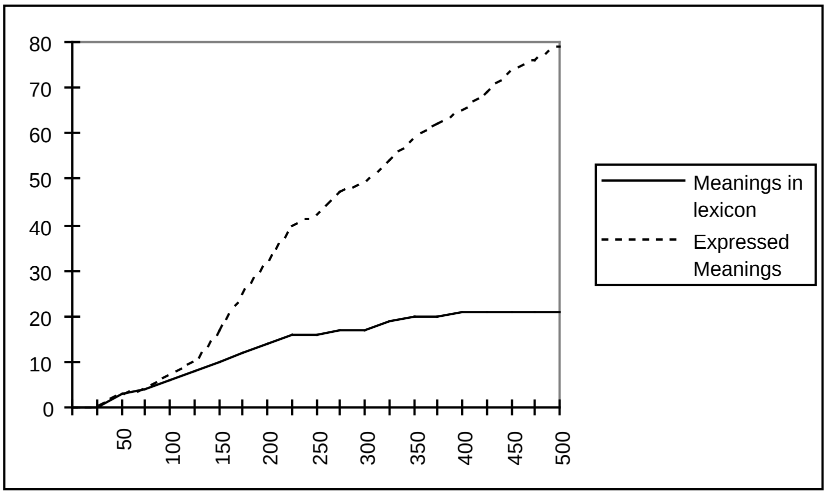


Figure 5.4: Efficiency in a multi-word agent.

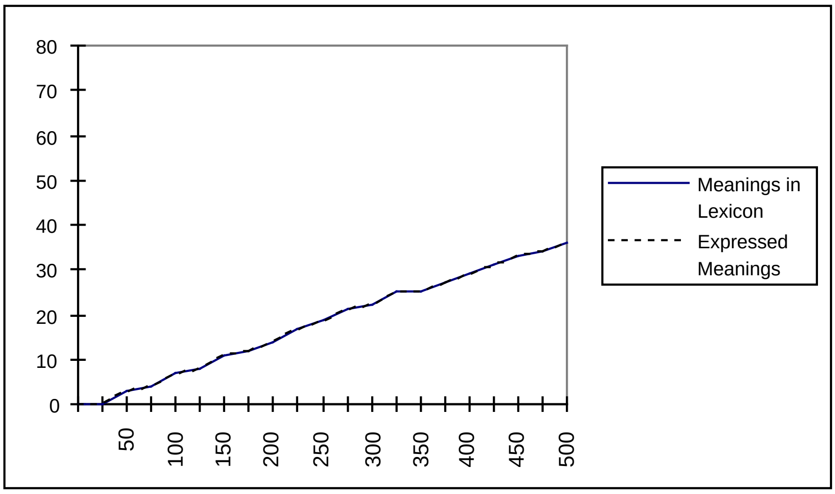


Figure 5.5: Efficiency in a single word agent. Note that both curves coincide in this graph.

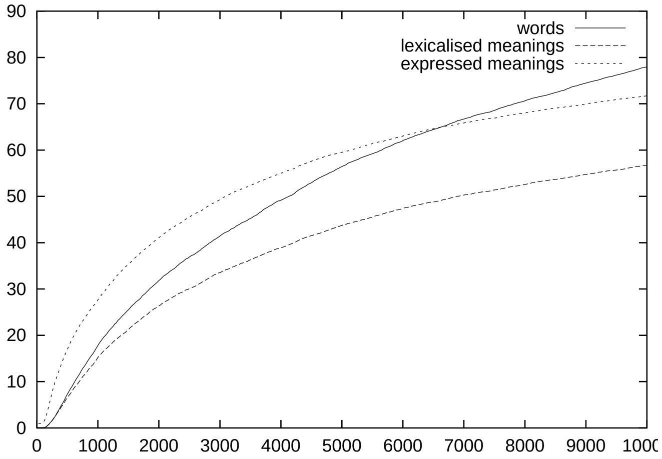


Figure 5.6: Number of meanings and words in a single-word naming game with 10 agents (10,000 games; averaged over 10 runs).

complex experiments. Additionally, the graphs contain a curve that shows the absolute sizes of the lexicons. The first figure shows the data for the single-word naming game, and the second figure shows it for the multi-word naming game. On the Y-axis, both graphs show the absolute numbers of words and meanings.

For the multi-word game (fig. 5.7), the curves measuring words and meanings in the lexicon are parallel from 4000 games onwards. In the beginning the number of words rises more rapidly than the number of meanings, suggesting a high level of synonymity (different words with the same meaning) in the agents' lexicons. Later on, the curves rise in parallel, which indicates that new words are only added when a meaning is unknown or not expressible. On the other hand, the number of expressed meanings is far larger than the number of meanings in the lexicon. This shows that the agents are effectively using multi-word expressions to lexicalise those meanings that they do not have in their lexicons. It is also worth noting that the number of words in the lexicon rises much more slowly in the multi-word naming games than in the single-word naming games.

In the single-word game (fig. 5.6), we see that it is the number of expressed meanings and lexicalised meanings that rise together, showing that whenever a new meaning has to be expressed, the agents need to add a new meaning to their lexicons. The number of words rises even faster than the number of meanings, indicating an increasing degree of synonymity.

It is clear that the multi-word agents have a much more efficient coding system for the meanings they have to lexicalise: they can lexicalise more meanings with smaller lexicons. Although there is no syntax involved (the order of the words in the utterances does not matter), expressivity is vastly improved.

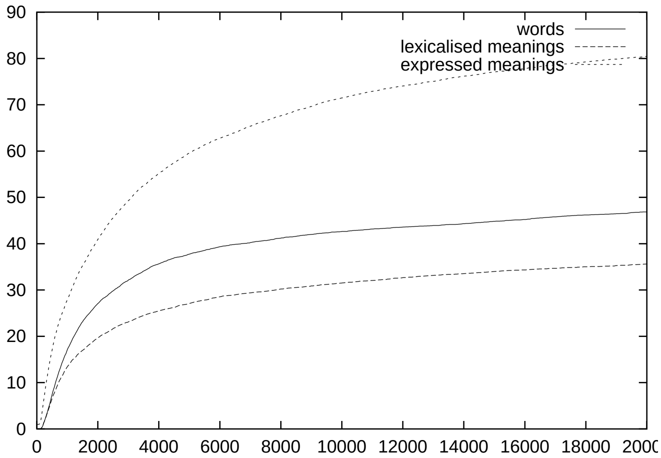


Figure 5.7: Number of meanings and words in a multi-word naming game with 10 agents (20,000 games; averaged over 10 runs).

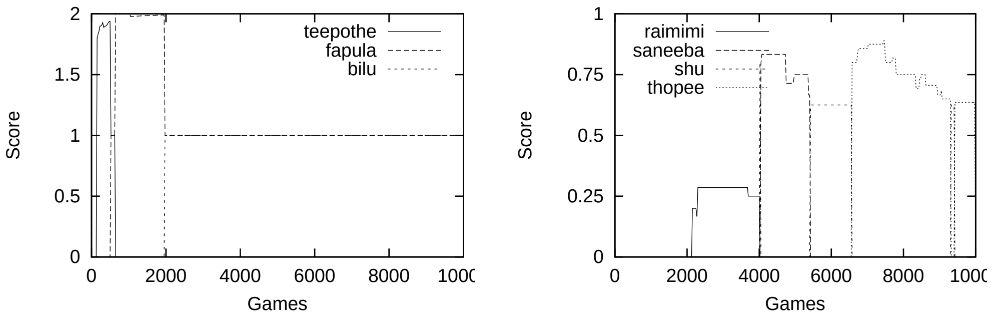
5.2 Competition between Words

Figure 5.8 (a) and (b) give examples of the competition that occurs between different words in the multi-word naming game.

In the two-agent experiment, the first word "teepothe" (which almost reaches full coherence in both agents) is replaced after a short time with "bilu" which also almost reaches full coherence in both agents. In the end, one of the agents creates a new word for WIDTH and continues to prefer that word. The situation from halfway in the experiment and onwards is thus that each agent prefers to use its own word, but both agents understand both words. Despite the low coherence (the meaning-word coherence for WIDTH is only 0.5) this is a stable state that persists from 2000 games until the end of the experiment at 10000 games. This shows again that a stable and predictable situation can emerge even when coherence is not very high.

In experiments with more agents, the dynamics become more complex. In the five-agent experiment, the agents do not reach a stable state in 10000 games. Several words in turn increase in strength but they do not become strong enough to settle the competition: the global strength of each word is hardly over 0.5.

The previous situation makes it clear that often it is not possible to explain the competition that goes on for a meaning on the basis of the graph of the competition alone. The same words are used for other meanings, and this can and will influence the competition going on for the particular meaning for which the graph was made.



(a)

(b)

Figure 5.8: Word competition in a 2-agent and a 5-agent experiment; meaning WIDTH in both cases.

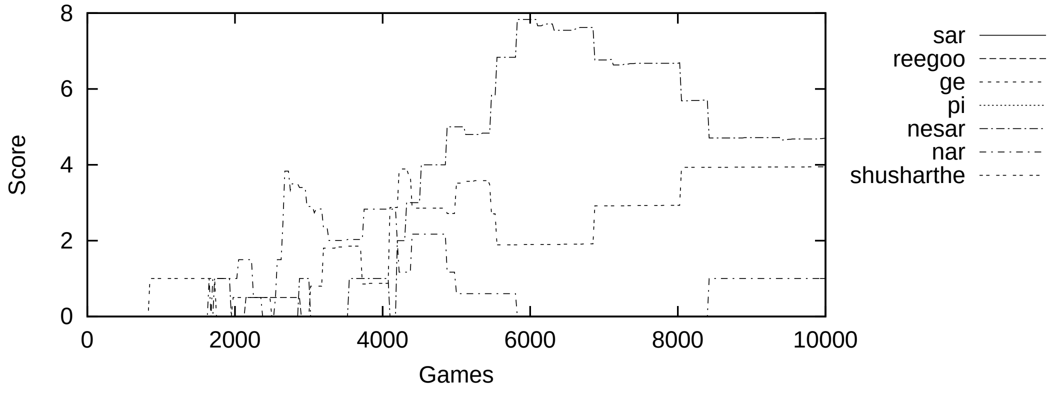


Figure 5.9: Word competition in a 10-agent experiment; meaning GRAY-LEVEL.

Population size	Number of words per meaning	Standard deviation
2 agents	1.70	0.74
5 agents	2.92	1.21
10 agents	4.32	3.04

Table 5.1: Average number of words per meaning (cfr. synonyms).

Population size	Number of meanings per word	Standard deviation
2 agents	3.73	2.00
5 agents	5.60	4.56
10 agents	5.36	4.68

Table 5.2: Average number of meanings per word (cfr. homonyms).

5.2.1 Synonyms and Homonyms

Results for more systematic experiments counting the numbers of words that are invented for a certain meaning are listed in table 5.1. The numbers were averaged over all the words for all the meanings that arose in a certain experiment. For every type of experiment (2, 5 and 10 agents), 10 experiments were done and the results were averaged again (experimental parameters other than population size were kept the same). There is an increase in the number of words that are created per meaning, but it seems to be much less drastic than the two previous graphs would indicate. This has probably again to do with the frequency with which meanings arise: frequent meanings are probably lexicalised again more often than infrequent meanings, giving a distorted picture for frequent meanings.

It should be noted that the numbers, calculated at the end of the experiment, include all forms for a meaning, created from the beginning of the experiment. This means that it is possible that some (or even all) of these synonyms may not be used actively any more.

The standard deviations are high in all cases, which means that there is a considerable variation from experiment to experiment: in some experiments there are very few synonyms, while in other experiments, there are many, even when all parameters of the system are the same.

The number of meanings per word (homonymy; table 5.2) shows a similar picture: here also, the standard deviations are high, indicating large variation between individual experiments.

5.3 No Explicit Subform Boundaries

In most of the experiments described in this chapter, the utterances exchanged between agents consist of several separate subforms (words). In reality, linguistic exchanges are a stream of sounds in which the boundaries of the subforms

Word	Meaning
deeshatheemeeparkartaiso	(BBY [0.0 0.5]) (BBY [0.75 1.0])(AREA [0.5 0.75])
deeshathee	(BBY [0.5 0.75]) (BBY [0.0 0.5]) (BBY [0.0 0.5])(AREA [0.0 0.5])
meeparkar	(BBY [0.5 0.75]) (BBY [0.75 1.0])(AREA [0.5 0.75]) (AREA [0.75 1.0]) (BBY [0.75 1.0]) (BBY [0.5 0.75])(AREA [0.75 1.0])
taiso	(BBY [0.5 1.0])(AREA [0.5 1.0]) (BBY [0.5 0.75])(AREA [0.75 1.0]) (BBY [0.0 0.5])(AREA [0.5 1.0]) (BBY [0.0 0.5])(AREA [0.5 0.75]) (AREA [0.5 0.75]) (BBY [0.75 1.0])(AREA [0.5 0.75]) (BBY [0.75 1.0])

Table 5.3: Partial lexicon of an agent in the single-symbol multi-word naming game.

are not perceptible.

Experiments where the model was changed to cope with single symbol utterances are often comparable to the experiments in which subform boundaries are explicit in the utterances. However, there are relatively many cases that exhibit "runaway" lexicon expansion. Figure 5.11 shows the evolution of the lexicon size in a series of multi-word naming game experiments in which utterances are always represented as a single form. The general trend is still the same as the traditional MWNG (see e.g. fig. 5.3): a flattening curve, which suggests that after a while the agents can reliably lexicalise most meanings, so that they need no longer expand their lexicons. However, the standard deviation is very high, which means that there is considerable variation between different experimental runs despite using the same experimental parameters.

Table 5.3 shows a partial lexicon from an agent in an experiment in which the lexicon kept growing. The table shows that the lexicon contains a number of words that are compositions of other words in the lexicon, while their meaning is the same or very close to that/those of the original words. The words each also have a number of different but very close meanings.

What happens here is that the speaker uses two of these words in a composite utterance. The hearer that is to interpret that utterance, can only do so on the basis of the words that it already knows and which are in its lexicon. When it does not know either of the words, it will necessarily interpret the whole utterance to be one word, and add it to its lexicon with the meaning for the composite utterance. In some cases this may result in a vicious circle, in which composite utterances are repeatedly interpreted as single words, such that the lexicon continues to grow.

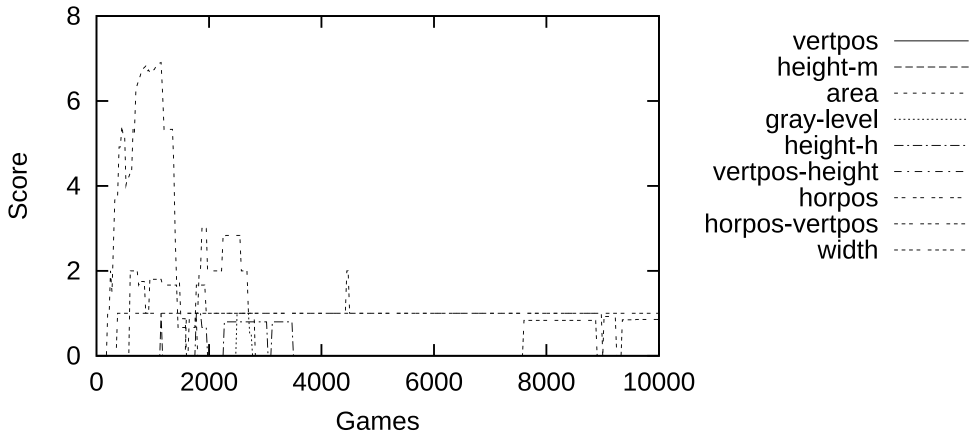


Figure 5.10: Meaning competition in a 10-agent experiment; word “lopoo”.

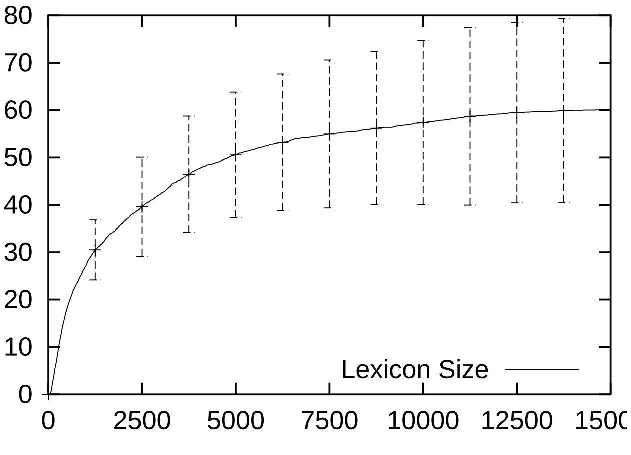


Figure 5.11: Lexicon size evolution in the single-form multi-word naming game.

5.4 Evolutionary Transition

This section presents the results from various simulations that were done using the model. This section will only show graphs directly relevant to the discussion; all graphs are given in appendix B.

The first thing to note is that generally, the hybrid model performs about as well as the models for each communication strategy individually. There is slight variation in the levels of communicative success reached. As an example, figure 5.12 shows communicative success with the game result pressure; communicative success for the other experiments can be found in appendix B.

The second thing to note is in all experiments, the lexicons become large on average, with a very large standard deviation, which indicates large differences in lexicon sizes between experiments. The reason for this is not so much the experiments with selection pressures; the model suffers from the same problem as the single-form MWNG. Again, figure 5.12 shows the lexicon size for the game result pressure.

It is possible to adjust the strength of a strategy in two ways: it can be recalculated every time (*absolute calculation*), or one can incrementally add or subtract a small amount to adjust the probability in a certain direction (*incremental adjustment*). Experiments of both types have been done.

All simulations reported here were done using a population of 5 agents, over 15000 games. The results are averaged over 10 experimental runs with the same settings. The context was fixed at 5 objects (chosen randomly from a larger set) in each game. Each time, 3 of the agents started the experiment preferring the single-word strategy, and 2 agents started the experiment preferring the multi-word strategy.

5.4.1 Dual Strategy Algorithm Details

Initial Strategy Strength and Distribution

In these experiments, the agents can use two strategies for constructing utterances. One issue is then what the agents' preferences are in the beginning of a series of interactions. This is governed by setting the initial strength of each strategy for all of the agents: part of the population can be set to prefer the Single Word strategy, while the others will prefer the Multi Word strategy. In our experiments, the populations consist always of 5 agents, where 3 agents initially prefer the SW strategy (with a strength of 0.9, and hence a strength of 0.1 for the MW strategy), and 2 agents prefer the MW strategy (with a strength of 0.9, with a strength of 0.1 for the SW strategy).

5.4.2 Results

Game Result

Absolute Calculation The results with absolute calculation for communicative success are shown in fig. 5.12. The new strength of each strategy is calcu-

lated on the basis of a record of the results of the x latest games played using that strategy.
The single-word strategy is used almost exclusively already after about a thousand games. Lexicon expansion seems to slow down considerably after 2500 games, although the standard deviation is very large, so that in some experiments the agents' lexicons have (almost) stopped growing, while in other experiments the agents continue to be very prolific at creating new associations.
Generally, we can say that using the game result as the pressure for strategy selection, the end result of the experiments corresponds effectively to a well-established single-word system on which all agents converge.

Incremental Adjustment Figure 5.13 shows essentially the same picture as for the previous experiment, except that the evolution is a little bit less pronounced as for the game result/absolute calculation experiment, in the sense that the variation between the experiments is much higher. Here too, the single-word strategy becomes dominant fairly quickly. Lexicon expansion is much faster than with absolute feedback, although there is again considerable variance from experiment to experiment. On average the lexicon seems to become two to three times as large as in the absolute feedback experiment.
The final result in these experiments is again that the single-word strategy becomes absolutely dominant.

Lexicon Size

Figure 5.14 shows the result the experiments in which lexicon size determines the strengths of the communication strategies; the strengths are calculated directly from the lexicon size. The graph shows that almost immediately the curve starts descending, towards more multi-word strategy use. Towards the end of the experiment single-word strategy use is less than 10%. Here then, the compositional strategy is the clear winner.
A problem with the lexicon size measure is that both strategies use the same lexicon. This means that the single-word strategy will expand the lexicon when it needs to accommodate for example multi-word utterances that it cannot analyse. This will automatically benefit the multi-word compositional strategy in the sense that every such new entry will push the lexicon away from the "ideal" that we as designers installed.

Lexicon Expansion

Figures 5.15 and 5.16 show the results of the experiments where the pressure on the strategies is lexical expansion. The first figure shows the experiment in which the strategies receive both positive and negative feedback; the second figure shows the experiment in which only negative feedback is given. In both experiments, the strengths are adjusted incrementally based on whether an association was added in the current game.

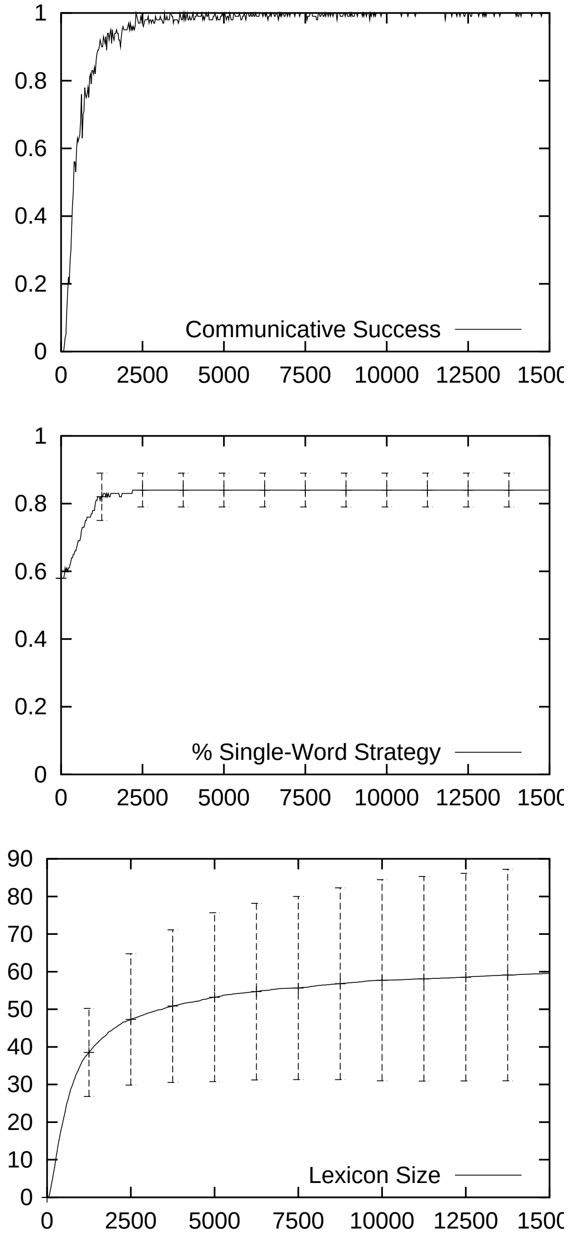


Figure 5.12: Game Result—Absolute Calculation.

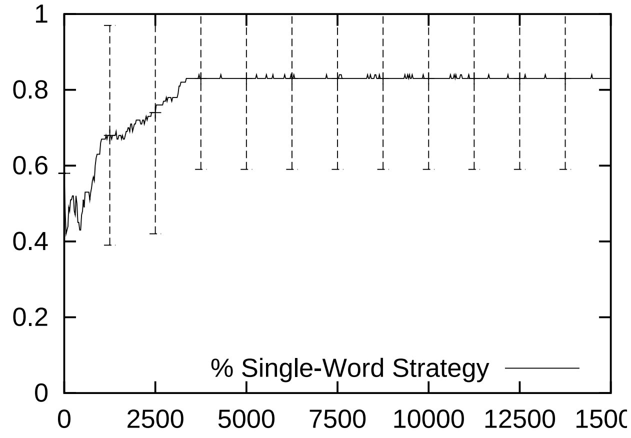


Figure 5.13: Game Result—Positive and Negative Feedback.

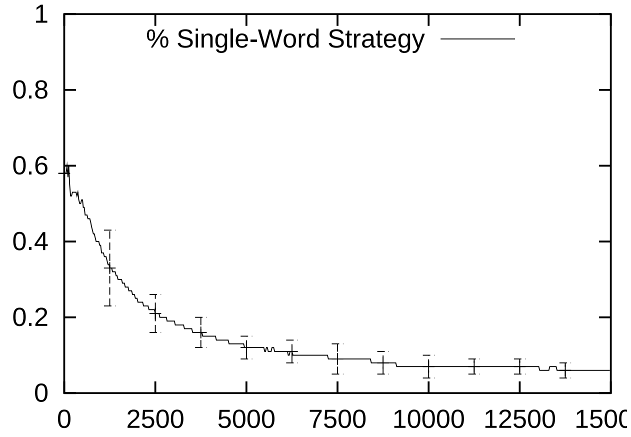


Figure 5.14: Lexicon Size—Absolute Calculation.

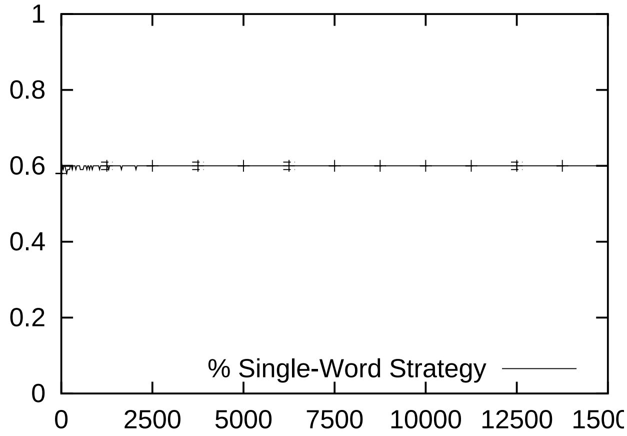


Figure 5.15: Lexicon Expansion—Positive and Negative Feedback.

While in the positive/negative feedback experiment the communicative success is certainly acceptable, it is clear that the agents are undecided on which strategy they should use. It seems that giving both negative and positive feedback really keeps the probability of each strategy being selected around 50%.

With only negative feedback, the picture changes quite drastically. The favoured strategy now consistently is the compositional strategy, but despite that the lexicon continues to grow very fast, even faster than in the other experiments.

5.4.3 Discussion

Deterministic vs. Probabilistic Choice

Experiments in which the "classic," deterministic mechanism for strategy choice is used, strangely have the exact opposite effect of what we would expect based on the assumption of gradually growing complexity in language evolution: where agents had decisively chosen for the single-word strategy before, now they chose for the multi-word strategy, and vice versa. The cause of this behaviour is in the combination of the nature of the language game experiments (initial negotiation phase with low but growing communicative success) and the choice of the pressures. The pressures we looked at until know are either calculated on the basis of communicative success directly, or on the basis of lexicon size or expansion, which are both closely correlated to communicative success.

This means that when the strategy is chosen deterministically, it is used consistently in the beginning of a series of language games. It will thus be punished for being used in the beginning, when many failures, and related lexicon expansions, occur. The affected strategy apparently never recovers from this, so that there is no real competition between strategies.

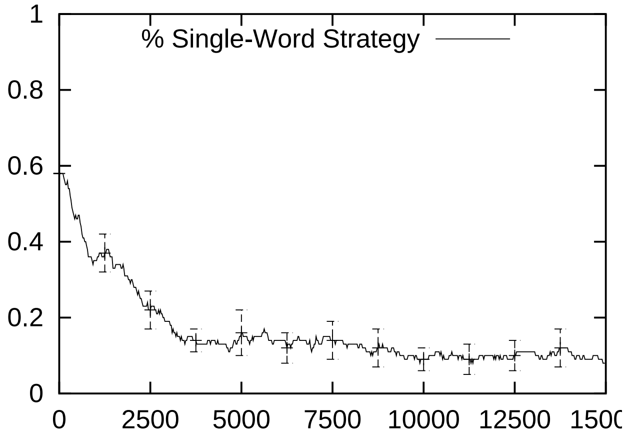


Figure 5.16: Lexicon Expansion—Negative Feedback Only.

Probabilistic selection works differently, in that all strategies always have a probability of being chosen. The probability is proportional to a strategy's strength, so that the probabilities shift during a series of interactions, but a strategy's strength usually does not become zero, which keeps open the possibility of that strategy being explored. As an example, fig. 5.12 above shows that, despite there being no upper or lower limit to the strengths of a strategy, the multi-word strategy remains between 80%–90% instead of going all the way to 100%.

Rather than conclude that the mechanism does not work in its basic form and that the probabilistic form is needed in selecting the strategy to use, we think it would be interesting to explore selection criteria that are less directly related to communicative success. An option would be for example to still use a lexicon size or expansion related criterion, but one that is non-linear, such that its influence is small in the beginning but increases as the lexicon becomes larger. This would simulate a situation in which it is easy to fill the lexicon, but there is a "soft" limit on the maximum size of the lexicon. The problem here would be to decide when the lexicon is "large enough," as the number of needed associations depends not only on agent-internal factors, but also on the complexity of the environment.

It may also be necessary to consider whether a generalisation algorithm such as described in section 4.3.2 is needed.

Pressures

There are two cases in which there seems to be "progress," in the sense that the agents tend towards the multi-word strategy. The pressures in these experiments are the lexicon size (direct calculation) and lexicon expansion with only negative feedback. The problem with the first pressure, is that the strengths

of the strategies are calculated directly from the results. In the original mechanisms, small increments or decrements are made to an initial strength, so that the strength is the only state the lexicon needs to remember about each association. A version in which the size of the lexicon influences the strengths of the strategies through small changes to the current strength has not been tested.
The second pressure that results in all agents using the multi-word strategy is lexicon expansion with only negative feedback. This is the most promising one up to now: it does not require that strengths be calculated directly, and the fact that the standard deviations are low indicates that it reliably causes the multi-word strategy to be chosen. However, it will be interesting to see if it would remain successful when the probabilistic method of strategy choice can be replaced by a deterministic one, similar to the mechanism used in the lexicon.

5.5 Summary

This chapter described experiments done with the MWNG model, and with the hybrid model.

- The basic experiments (communicative success, lexical coherence) perform as expected; i.e. they are at least no worse than the results for similar-sized single-word experiments. In fact, communicative success seems to be slightly higher, indicating that the agents can communicate slightly better.

- Experiments in which single and multi-word experiments are run side by side show that in multi-word experiments, the lexicon needed by the agents is noticeably smaller than the lexicon needed by single-word agents to express the meanings they need to express.

- An unresolved issue in the current multi-word model is that of how utterances should be formed. In the basic model, utterances are simply composed of several symbols. It would be more realistic for a speaker agent to consolidate the symbols in its utterance into one long symbol, since that is the way it works for real language. This poses an extra problem for the hearer, because it has to segment the utterance first on the basis of its lexicon. Usually this works, but in some cases the lexicon can degenerate when compositional utterances are repeatedly interpreted as a single-word utterance.

- The experiments with the hybrid model, in which the agents "choose" their communication strategies depending on one of a number of selection strategies, do not allow us to draw definitive conclusions yet. Two selective pressures push the agents towards using the multi-word strategy: lexicon size (where strategy strengths are calculated directly from the lexicon sizes) and rate of lexicon expansion (only with negative feedback), but too few experiments have been done to draw firm conclusions.

It shows however that the measures we used before to gauge the model at the population level can be implemented as pressures internal to each agent. Several pressures have been tried, based on communicative success, lexicon size and lexicon expansion rate.

A quirk of the model at this point is that the agents' strategy selection must be done probabilistically rather than simply selecting the strategy with the highest strength. In the latter case, because of the high number of failures in the beginning of an experiment, the strategy competing with the one that has the highest strength in the beginning will receive a high reward and take over.

Simple Syntactic Naming Game

"Baby Eat Cookie!"

EVEN THOUGH the above utterance, that could have been uttered by a typical three-year-old child, seems simple, it exhibits already a great deal of complexity. Notably, it refers to two different referents, the baby and the cookie, which are semantically related through the third word, *eat*. It is this word, the verb, that assigns semantic roles to the referents of the other words.

This is already at the limit of what the multi-word naming game can do, and if we consider that the child probably takes into account the proper word order of English (SVO[1]), it is beyond what the MWNG agents can do. The problem that the child can handle but the MWNG cannot was already mentioned in the Multi-Word Naming Game chapter: solve equalities of variables.

Essentially, when a hearer assembles the meaning of an utterance, it looks up the meanings of the words in its lexicon and constructs sets of the predicates it finds in its lexicon. Each piece of meaning uses its own variables; actually, the variables must be *made* distinct *explicitly*, because any coincidental occurrences of variables with the same names will cause there to be equalities that may not even be valid. These variables must then be bound sensibly in order to arrive at the correct interpretation.

In the Multi-Word Naming Game, it is implicitly assumed that all distinct meaning parts refer to the same referent, i.e. all variables are equal to one another. In this chapter, we drop this assumption. On the one hand, this introduces a lot of additional ambiguity, but on the other hand, it provides opportunities for the language to become much more spohisticated in terms of what it can convey. We will examine how language copes with the increased ambiguity, and try to implement computational equivalents to the techniques found in natural language.

In principle, the context in which an utterance is interpreted, and in which the variables are assigned to their referents will reveal which assignments are valid and which are not. From there it is possible to identify which variables are bound to the same values. However, using only the context to provide information about possible variable bindings creates a strong dependency on the context, and has other drawbacks as well. Computationally, exploring all possi-

[1] SVO = Subject-Verb-Object: this is the "usual" sentence pattern in English, when no elements in the sentence are being specifically stressed.

ble variable assignments that the context has to offer becomes rapidly infeasible as the number of candidate combinations increases.

Though the end-result may be the same, it makes a big difference whether the equalities are known beforehand or not. For example, relying on the context alone, the meaning

$$\text{BIG}(x) \wedge \text{BOX}(y)$$

may have from one to a huge number of possible bindings. The meaning allows x to be bound to any BIG object, while y can be bound to any BOX object.

If there would be a way to encode, in the utterance that encodes the above meaning, that $x = y$, the bindings would already beforehand be limited to those that satisfy both predicates at the same time. In natural languages, grammar is the tool that allows to incorporate such constraints on the meaning into the utterance.

Looking at Jackendoff's (2002) language evolution proposal again (see fig. 1.3 on p. 8), we see that resolving equalities corresponds more or less to what he calls "use of symbol position to convey basic semantic relations," and "grammatical categories." The model in this chapter does not (yet) have hierarchical phrase structure, or any of the other grammatical features that serve as milestones in Jackendoff's proposal. We could thus say that the grammatical system as it is implemented at the time of writing (September 2004) corresponds more or less to Bickerton's proposal for *protolanguage* (Bickerton, 1990).

This chapter will explore the impact that creating, using and maintaining a grammar has on the agents in a population. The next section gives an overview of related work. Section 6.2 first presents a number of calculations that illustrate the complexity of the problem of variable binding, once the assumption is dropped that all parts of the meaning refer to the same referent. This is followed by a detailed explanation of the semantic system, which is equivalent to the semantic system introduced with the MWNG, but is implemented differently. Finally, the syntactic system that deals with equalities will be explained.

6.1 Related Research

From a non-linguist's point of view, it might seem logical to approach the problem "language" from the bottom up: first look at words, and only later look at phrases and sentences. In linguistics, however, language is often equated with "grammar," because grammar seems at first sight to be the most striking and unique aspect of language. The lexicon is taken for granted as a simple list of word-meaning pairs.

One consequence of this attitude, and one that is relevant in this setting, is that in the past far more studies have been carried out about grammar than about the lexicon (such as in chapter 2) or about useful but more limited forms of grammar (such as in chapter 4).

Looking more specifically at computational approaches toward language, we see that many if not most of the efforts are targeted at understanding linguistic utterances. Hence, the whole field of Computational Linguistics is dedicated to extracting information from natural-language sources. Initially, the goal was to fully interpret texts, but more recently the focus is on less "deep" but faster techniques, such as shallow parsing, stemming or part-of-speech tagging, that can be used as input filters for other applications. See for example Dale et al. (2000) for an overview of current work in Computational Linguistics.

Despite this bias in much of the language research, several computational experiments have been done that were aimed at acquiring a more in-depth understanding of the phenomenon "language" itself.

A number of the experiments that have been done focused on syntax, without meaning. These models tend to be formal, and use implementations of context-free grammars. For example, Hashimoto and Ikegami (1996) describe such experiments. Their agents' communicative success is used as a fitness measure, and the grammar rules were the genotype. A genetic algorithm is used to evolve subsequent populations of coordinated grammar users.

The L_0 project (Feldman et al., 1996) was an effort to build a system that could learn to understand a language (in principle any language) by correlating visual input (scenes from an environment) along with descriptions of the scenes in the "target" language. However, the core of the system was still a grammar induction algorithm for *probabilistic context-free grammars* (Stolcke, 1994), and the goal was to learn the language, not to study the dynamics of language in a *population* of language users. The project as a whole was not successful, but it spawned several interesting related lines of research.

A project that resembles the L_0 project in many respects is the language understanding and learning project by Peter Dominey (2000). Here also the goal is to correlate visual and linguistic input. Dominey provides his system with knowledge about the difference between function words and open-class words, such that the problem of assigning semantic roles to participants in the visual scene amounts to connecting the order of the function words to the order of the semantic roles.

Another model that resembles our model in terms of its architecture is the Iterated Learning Model (Kirby, 1999). The object of study in this experiment was the learning bottleneck: the fact that a language learner only hears a finite, relatively small number of examples from the target language but still succeeds in learning it. In a model that included agents with grammar-learning capabilities (again using a conventional grammar-induction algorithm) and a given semantics, Kirby looked at vertical transmission of language. In the early stages, a holistic language appeared, which was remodeled into a composite language during a short transition period.

Kirby only studied vertical transmission: a student learns its language from a teacher, and after the teaching period the student becomes a teacher himself. The model does not contain a population of agents in the sense that they do not interact with each other except in the teacher/student sense.

Despite the superficial likeness between horizontal transmission models like the language game models and vertical transmission models, there are important differences in philosophy between the two models. The vertical transmission model is an explicit teacher-student system, where the student's task is to derive a compositional grammar from a series of input utterances generated by the teacher. The goal of the model is to study the effect of the "bottleneck" that arises as the amount of input for the hearer decreases, and the coverage of the language by the teacher's utterances decreases. In the Language Game model, where the agents alternate between the speaker and hearer roles, a speaker is always required to have the structures that a hearer will need to interpret its utterance. (In fact, the speaker must be able to correctly interpret the utterance it wants to utter before it may actually do so.) So in this model, not only the student creates new structure (lexical and grammatical), but the teacher does so as well while it is generating an utterance, if it needs to.

Batali (1999) describes a model in which, in the same way as done in our model, a population of agents interacts to create a language by exchanging series of characters. Like in the Iterated Learning Model, initially the exchanges between different agents are uncoordinated and incoherent, but after a while the agents develop coherence and it becomes possible to analyse their utterances in terms of compositionality. Contrary to our model however, the agents in Batali's model use meanings encoded in binary strings and recurrent neural networks to produce character strings for the meanings and vice versa. (Batali, 2002) discusses a model that is much closer in spirit to our model. The agents in this model use an inventory of *exemplars* to store their linguistic knowledge. Exemplars correspond more or less to partial parse trees. Batali also touches on the concept of variable equalities (see section 6.4).

A very recent model (Vogt, submitted) combines the ideas behind Steels's language games and Kirby's iterated learning model into a single, population-based generational model where the agents have a semantics that is not fixed beforehand (a variant of the discrimination game) and a grammar induction algorithm that generates semantically annotated rules in the style of context-free grammars.

Vogt's main focus in these experiments is the learning bottleneck that new language learners encounter when they are "born" into the population. In fact, the experiments he describes in this article always use a population of two agents: an adult (speaker) and a student (hearer), so that in effect the model boils down to an iterated learning model with a different, more open-ended semantics and a (slightly) different grammar induction algorithm. In contrast, our model emphasises intra-generation interactions, and serves to study the emergence of language and the grammatical functions that make it useful as a communication tool.

A computational model of a very different theory (the classical linguistic parameter setting theory to be precise) is the Structural Triggers Learner (Sakas and Fodor, 2001). It is an implementation of the classical linguistic theory of grammar learning: setting parameters in an inborn Language Acquisition Device, based on evidence gathered from the linguistic input. The authors recognise that setting n parameters (for 2^n possible grammars) requires a search process with a complexity of O(2^n), something that is often not acknowledged by people advocating the parameter setting theory. The cause of these problems is essentially that a linguistic utterance usually is not unambiguous with respect to the parameter settings that formed it. The authors refer to proposals that tackle this complexity, but do not implement them.

6.2 Decreasing Ambiguity: Illustration

Interpretation of utterances is a complex task. However, the speaker unconsciously helps the hearer by embedding clues for the interpretation of an utterance in the utterance itself. Language has two ways of "publishing" information about relations between different referents in an utterance: grammatical, but also lexical. A certain referent may be lexicalized by different words, that are not synonyms, but contain information about their relation to other concepts in the utterance. For instance, in English there are three words for self-reference ("I", "me", "my"), which code not only for the self-reference, but also its semantic function in the utterance (agent, patient, and possessor, respectively). For the sake of presentation, we will assume in the remainder of this section that words are "atomic", meaning that they contain no information about the relationship between the referent and other elements of the utterance.

6.2.1 Meaning of an Utterance

Using the notation already introduced in section 4.2.2, we will use a limited form of predicate calculus to represent meaning: meaning parts are represented using predicates, and their referents are represented using variables of different types. For example, "book" can be represented as BOOK(x), "red" as RED(x) and "approach" as APPROACH(x), where the type of x would be item, item, and event respectively. Relationships between meaning items are expressed using variables: predicates having the same variables in their arguments refer to the same referents.

More complex language elements such as quantifiers ("all," "some,"...) etc. can be modeled by increasing the complexity of the meaning representation, for example by adding quantifiers such as $\forall$ and $\exists$ to the representation.

With or without quantifiers however, the key to using these meanings it that ultimately the variables in the utterances have to be bound to (references to) elements of the world in order to determine the truth value of the meanings. (Or, alternatively, assuming the meaning is true, to determine the possible bindings in the current state of the world to make it true.) In short, we can consider meanings to be functions with arguments that can be bound in order to obtain a result.

6.2.2 Composite Meanings

The main obstacle to the interpreter of a compositional utterance is in combining the parts in the correct way. Every word in the utterance is associated with a part of the final, complete meaning. Every partial meaning has a number of "free variables" that need to be bound in the context. The interpreter is thus faced with the problem of determining which free variables in the different parts of the meaning are actually equal.
An example: suppose the context contains a blue book and a red pen. A composite utterance might be "red book," where each word corresponds with a predicate carrying the meaning RED and BOOK, respectively. Without regard to the context, the utterance allows for two distinct interpretations:

1. RED(x) $\wedge$ BOOK(x)
2. RED(x) $\wedge$ BOOK(y) (with $x \neq y$)

in other words: either it refers to one object, namely a red book, or to two distinct objects, namely a red one and a book. In the given context, only the second interpretation would be possible.
Of course, the more complex the composite meaning becomes, and the more referents it references (such as an event together with its agent and patient), the more possible interpretations will have to be considered. Table 6.1 shows the number of distinct partitions that can be formed using the given number of variables. This corresponds to the number of different assignments of referents to the variables, assuming that the size of the context equals the number of partitions.
This is all assuming no ambiguity in the lexicon, i.e. words that are associated with several meanings; if there is ambiguity in the lexicon, the numbers need to be multiplied accordingly. Additionally, the table does not take into account arbitrary context sizes.

6.2.3 Context

The numbers in table 6.1 do not refer to any context, they merely calculate the number of possible partitionings of the variables themselves. In order to find the number of possible variable assignments in a specific context, we need to know the number of subsets a partition has. Every referent in the context can be assigned to every subset in a partition. For each type of partition, every subset can be assigned to every referent. So, in order to obtain the number of possible interpretations for x subsets and y referents, we calculate the number of permutations of y referents and x subsets, $P(y, x)$. In cases where there are more subsets than referents, no valid assignment can be made. Table 6.2 shows the results of this calculation. As it happens, the table shows a simple regularity: for x variables and y referents, the number of possible interpretations is equal to y^x.

No. of variables	No. of partitions
1	1
2	2
3	5
4	15
5	52
6	203
7	877
8	4,140
9	21,147
10	115,975
⋮	⋮

Table 6.1: Number of possible partitions for x variables.

	1	2	3	4	···	(no. of referents)
1	1	2	3	4		
2	1	4	9	16		
3	1	8	27	64		
4	1	16	81	256		
⋮						
(no. of vars)						

Table 6.2: Number of possible interpretations for specific numbers of variables and referents.

"book red"	$\text{BOOK}(x) \wedge \text{RED}(x)$
	$\text{BOOK}(x) \wedge \text{RED}(y)$
"book-S red-S"	$\text{BOOK}(x) \wedge \text{RED}(x)$

Table 6.3: The way in which a suffix can reduce interpretational complexity.

It is clear from these calculations that the number of possible interpretations increases rapidly as the number of partial meanings (with one or more free variables each) increases, and as the size of the context increases. Consequently, it is in both the speaker's and the hearer's interest to reduce the number of possible interpretations as much as possible in order to reduce interpretation complexity and the possibility for misinterpretation.

6.2.4 Syntax

Consider the following sentence:

> "A red book is on the blue table."

Suppose that word order is irrelevant. How would you know which one of the book and the table is red? If word order is irrelevant, there is no way of knowing except from the context: you could check with the context whether the book is red or blue, and likewise for the table. This demands a lot of extra effort in interpretation though, and it would not yield a solution if there were both a red and a blue book, and a red and a blue table.
However, English provides word order to bring order in this chaos: since words that refer to the same referent are grouped together, you know that it is the book that is red and the table that is blue. Other languages may trade word order for alternative mechanisms such as agreement, where case sometimes is explicitly marked on every word that refers to the same referent.

Given the calculations in the previous sections, what would be the effect of syntax on the number of interpretations that a hearer has to sift through?
To understand this, one should realize how the function of the syntactic devices mentioned above is connected to the meaning. Suppose that words only refer to a predicate, and that for simplicity we only talk about predicates with one argument. When two words refer to the same referent, their arguments will be equal. By extension, when a grammar indicates, for example with common suffix, that two words refer to the same object, this actually means that the arguments to their associated meanings should be equal.
Table 6.3 gives an example of how a suffix might restrict the possible interpretations of a phrase and code for the variable equalities. Note that suffixes in natural language are much more complex than in the example. "Real" suffixes may agree with the words they modify in terms of grammatical gender, the

same suffix may have different meanings in different contexts, etc. So in reality, suffixes will still allow ambiguity; however, they make it possible to cut down on the number of possible interpretations using only sentence-internal information.

The calculations above are based on certain assumptions that make the results not really suited for application to actual language. For example, pragmatics has been entirely disregarded. The assumption that lexical items do not contain information about the relationship, leads to results that are more pessimistic than if lexical items could contain this information. On the other hand, the sizes of the context used in the calculations no doubt severely underestimate the complexity of actual contexts.

Also, the calculations above should give some idea of the *differences* in complexity between interpretation without information on relationships between words and interpretation using this information. It is obvious that including such information in the utterances is hugely beneficial for a hearer; any communication system that uses this type of encoding will have a communicative advantage over communicative systems that leave the full burden of interpretation to the hearer alone.

6.3 Semantics

The Simple Syntactic Naming Game uses the same semantic notation as the second type of the Multi-Word Naming Game (see section 4.2.2), but realizes semantic descriptions in a different way. The Multi-Word Naming Game essentially uses a simple search to find a distinctive meaning for the topic: it starts with a partial meaning and a collection of predicates, and extends the meaning by adding clauses with each of these predicates when this is syntactically possible.

For small meanings this is a viable strategy, but as the meanings become more complex, it does not scale up. In the Simple Syntactic Naming Game, this naive strategy has been superseded by a modular approach that performs better. This system uses meanings that are compartmentised in units, which may be merged and split up as needed. The actual meaning collection and discrimination is done through a set of *specialists*, that each specialize in a specific part of the perceptual input domain. In view of the previously described types of semantics, the system described here can be seen as consisting of two layers. At the specialist layer, we find "discrimination modules" that are specialized for different parts of the sensory input (objects, actions, time,...). At the combination level, the results of the individual specialists are represented using predicates which can be combined to create more complex meanings. The activation of the specialists and the combination of the individual meaning parts they produce are handled by a separate, coordinating process.

6.3.1 Specialists

In the simpler models presented in the previous chapters, the discrimination game is used as the meaning generator. In the discrimination algorithm (section 2.3), the input is divided into channels, which are represented along a single dimension (the interval [0–1]) Categories in these channels are then combined by a global mechanism into a complex meaning.

For most aspects of the input, it is not realistic (both cognitively and computationally) to represent them as a single dimension. For example, colour might need as much as three dimensions to be represented adequately, e.g. red, green and blue components. Of course, it is possible to use several channels to represent colour, but this makes it impossible to treat colour as a single aspect of the input, because the channels are treated independently of each other by the discrimination game algorithm. It is also conceivable that different aspects of the input have to be represented in different ways: while colours may adequately be represented using triples, this representation will probably not work for time-related aspects of the input.

These realisations have led to the design of a "semantic engine." For every aspect of the input, instead of one or more channels, a *specialist* categorises the data. A specialist can use arbitrary internal representations, that are invisible to the rest of the system, including the other specialists. Specialists construct predicates; these predicates represent (potentially very complex) categories over the input. It is these predicates that are used by the rest of the semantic system to sconstruct complex, discriminating meanings. (Incidentally, individual specialists might still use the discrimination algorithm as their internal categorisation mechanism.)

Time specialist. As a case in point, the Time Specialist (De Beule, 2004a) nicely illustrates this point. The input data that comes from the perception includes data about the time interval when certain events or objects "went on." (More precisely, the perception data is formulated in predicate calculus clauses, and with every predicate is included the interval during which the predicate with its arguments was observed to be true. The presence of a predicate in the perception data does not imply "eternal" truth; its truth value is only guaranteed within the interval supplied along with it.)

The time specialist has its own internal "language" for comparing points in time, including internal predicates called BEGIN and END, which refer to the begin point (in time) and the end point (in time) of the object, i.e. the points in time when it entered the agent's perception, and when it disappeared again from the agent's perception. These can be compared to each other using relational operations such as $<$ or $=$, and to predefined points in time such as YESTERDAY or NOW. (Of course, in due time we would like the agents to develop these reference points in time themselves, but for now, they are given.)

Using this internal language, the time specialist constructs the expressions that compare the time aspects of a referent event to those of another referent event or the internal reference points. Every complex expression forms a category that can be used, by filling in the arguments, to compare whichever referent to whichever other possible referent, and yield true of false. These expressions can be used as predicates by the agent, and evaluated during interpretation by calling back to the specialist with actual values.

Definiteness specialist. Definiteness is a linguistic concept that links a syntactic phenomenon (the absence or presence of "defining" elements in an utterance; in English and Dutch they are called *articles*) with several different semantic phenomena. Hawkins (1978) and Lyons (1999) name (among others) *identifiability* and *familiarity* as the semantic phenomena represented using definiteness. The Definiteness Specialist (Van Looveren, 2003) only implements a minimal subset of these phenomena: identifiability. Concretely, the specialist looks at the topic, and examines its agent's history of conversational topics. If the topic is found in the history, it succeeds and returns a meaning part. This meaning part simply consists of the predefined predicate KNOWN followed by the referent, which indicates that the referent has been referred to before. This specialist does not construct new predicates. The KNOWN predicate effectively narrows the search scope for the referent in question to the previous referents that are on the agent's conversation history. The other aspects of definiteness have not yet been implemented.

6.3.2 Combining specialist data

The modularisation of the treatment of perceptual data forces us to consider another issue: does language influence categorisation, does categorisation influence language, do they both influence each other, or neither?

This is a tricky question, for which a definitive answer has certainly not been given. It stems from the fact that modules deliver their meaning chunks nicely packaged in separate units. If we leave it that way, the lexicon lookup will be biased toward the meaning units as delivered by the specialists. If we merge the units, the language algorithms will split the meaning purely according to performance in communication.

In our system, we chose to combine all meaning chunks into a single unit. This permits us to examine how the linguistic algorithms split the meaning when it is not biased by an a-priori split of the meaning. This could be altered in a later stage if linguistic or psychological evidence would require so.

Another factor guiding this decision, is that the structure of the specialist system is designed by the programmer of the system. Keeping the unit structure with the different units supplied by the different specialists, could mean that the formation of the lexicon is biased by this predefined structure.

6.3.3 Processes

Apart from specialists, the system is composed of more generic *processes* that each perform a subtask of the global task of producing an utterance about the external world (De Beule, 2004b). As a matter of fact, specialists are processes that are specialized towards generating meaning.

Figure 6.1 shows the processes and their dependencies. All subtasks involved in the global process, from perceiving the external world to rendering the final utterance, are performed by processes. The arrows in the diagram show the direction in which the data flows between processes.

In fact, each processes is composed of three parts:

An *action* function The action function represents the "usual" way of processing data. It takes the input data, and performs the action on it that is expected of the process. For example, the action function of the lexicon production process will take a meaning as its input data, find the relevant lexicon entries, and add the syntactic parts of these entries (the words) to the syntactic structure.

Whenever the action function encounters data that it cannot process, it will generate a *problem description* and return that to the process engine. Usually, an action function will fail in one of a limited number of ways. The lexicon production process will issue a *missing-word-meaning* problem when it finds unlexicalised meaning parts.

A *fix proposal* function The fix proposal function will take a problem description, and translate it into a fix proposal. A *missing-word-meaning* problem will be translated into a *invent-new-word* request. Of course, along with the type of problem, the actual data are passed along with the fix proposal request, and this data is in turn attached to the fix proposal.

(Of course, the same problems are solved by applying the same fix, so essentially, the fix proposal function represents an unnecessary step and could be omitted altogether. However, it was introduced for conceptual reasons, and continues to be relevant in that context.)

A *fix* function The fix function will take a fix proposal and actually execute it. This means that this function will make alterations to the process' internal data structures. It will do so using the data that it receives in the fix request.

The architecture of the processes corresponds to the architecture of language game modules, as described in section 1.3: an internal data structure accompanied by a default algorithm, and a mechanism to apply fixes to the data structures when the default algorithm fails to deliver a solution to a particular input. As such, the new architecture can be seen as a finer-grained version of the previous architectures, where the tasks to be solved by each module (process) are smaller and more varied.

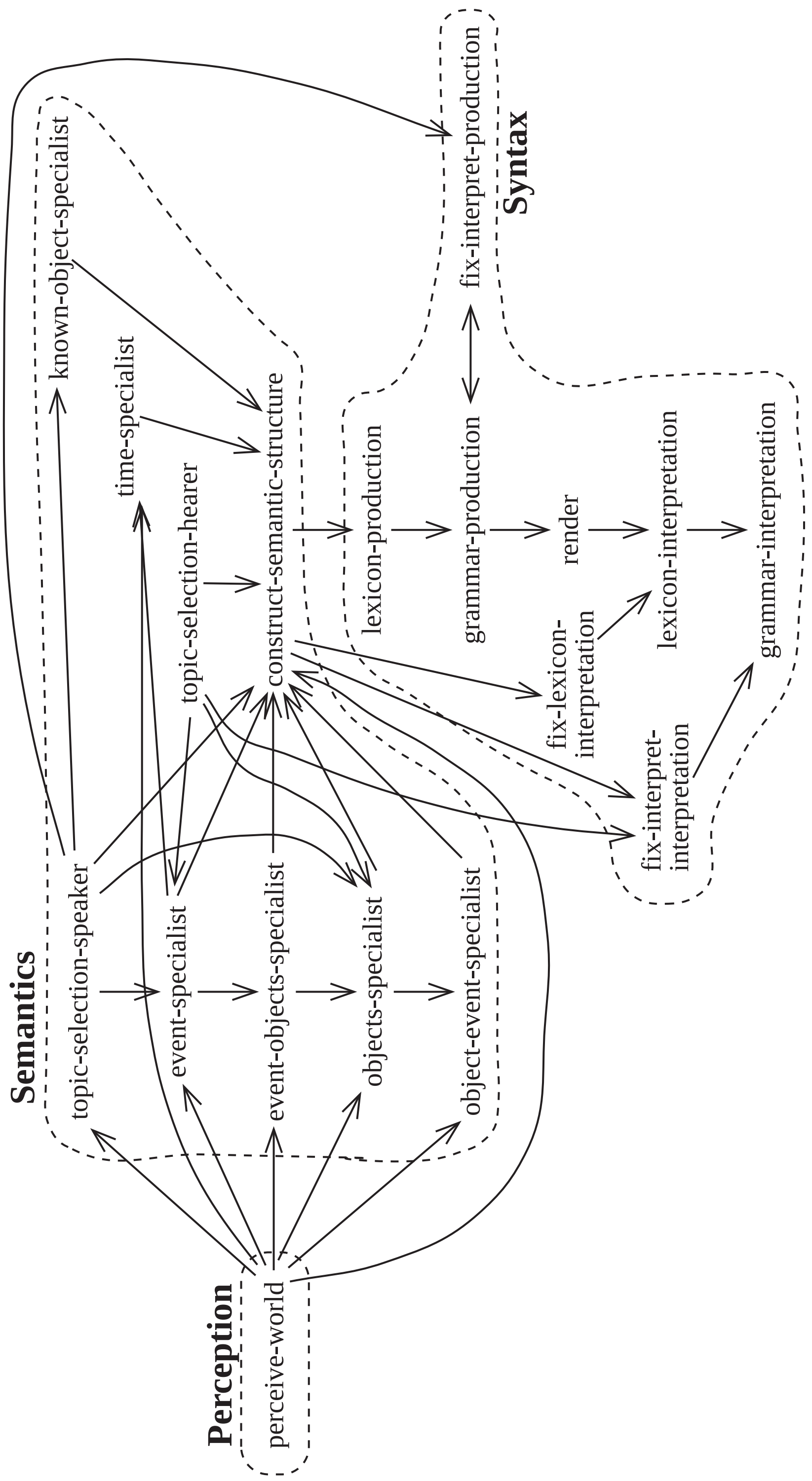


Figure 6.1: Schematic representation of process dependencies in the process engine.

```
run-cycle(Task)
  result ← null
  for each process ∈ process-states(Task)
     if (process-active(process) and
         not(process-reported-problems(process)))
       returns ← call(process-action-function(process))
       for each return ∈ returns
         newTask ← copy-task(Task)
         for each problem ∈ return
           report-problem(newTask,problem)
       result ← append(result,newTask)
  if result
    return result
  else
    return list(Task)

run-tasks(Queue)
  returnTask ← null
  while not(empty(Queue) or (returnTask = null))
    task ← pop(Queue)
    proposedFixes ← collect-fixes(task)
    fixResult ← false
    for each proposedFix ∈ proposedFixes
      fixResult ← fixResult
                    or
                    try-to-fix-process(task,proposedFix)
    task-changed(task) ← task-changed(task)
                            or
                            fixResult
    if (goal-achieved(task) = true)
      returnTask ← task
    else if (task-changed(task) = true)
      task-changed(task)  ← false
      newTasks ← run-cycle(task)
      for each newTask ∈ newTasks
        enqueue(Queue,newTask)
  return returnTask
```

Figure 6.2: Pseudo-code for the process engine.

6.4 Form

It is important in interpretation that the variables used in the different pieces of meaning retrieved from the lexicon be made different, in order to avoid chance equalities that could compromise interpretation. Nevertheless, as has been argued in section 6.2, interpretation would become a lot easier and faster and a lot of ambiguity would be resolved beforehand if a hearer would know in advance which variables would come to refer to the same objects. This would constrain interpretation considerably, because the predicates in the expression that are linked together with equalities act like a single big filter. Predicates with no equalities each act like a small filter, so that relationships between the arguments still need to be established.
Suppose that a speaker utters the expression "gobibi wabaku" to describe a blue, square object in the context. The hearer converts this utterance to the following expression based on its lexicon:

$$\text{BLUE}(x_1) \wedge \text{SQUARE}(x_2)$$

If the context contains for instance two blue objects and three square objects (one of which has both features, so four objects altogether), there are $2 * 3 = 6$ possible interpretations for the utterance, only one of which is the correct one. The context and/or the speaker's pointing may show the correct interpretation, and based on that information the hearer is able to infer that the two variables refer to the same object.
Imagine that the speaker and hearer somehow reach a consensus that in two-word utterances, the meanings of both words have a variable in common. In that case, the hearer's interpretation of the above utterance would have been:

$$\text{BLUE}(x_1) \wedge \text{SQUARE}(x_1)$$

which can in the given context only be interpreted in one way, yielding the correct result.

6.4.1 Rules & Constructions

Like in all other grammar formalisms, the formalism used here is based on a set of rules. The different types of rules are explained later, but all rules have to fulfill a number of criteria. These criteria set the formalism apart from other grammar formalisms.
First of all, the grammar must be useful both in production and in interpretation. The agents in our experiments must be able to do both, and it introduce extra complexity to have separate grammars for production and for interpretation. Translating this requirement into a concrete guideline, this means that all rules in the grammar's rule repository must be *reversible*. An entry in the lexicon should not only map a certain meaning to a word (useful in production), but the same rule must be used to map the word to the meaning (useful in interpretation).

Secondly, the grammar must be able to adapt itself and expand continuously. This follows the practice used in developing the lexical models of the preceding chapters: the agents in a population develop their own lexicon, and should therefore be able to respond to unknown situations (topics) by creating new words. Similarly, if an agent encounters a type of event that it has no syntactic construction for, it should be able to extend its grammar to cope with it. A hearer should be able to identify unknown grammatical structures and be able to add it to its own grammar, to the extent possible.

Thirdly, the grammar must not only be able to construct and validate utterances, it must deal with meaning as well as form. That is, in production it must produce an utterance on the base of a meaning, and in interpretation, it must produce a meaning based on the utterance.

The final shape of the grammar used in these experiments uses rules that feature a tight coupling between meaning and syntax. This is a defining feature of a type of linguistic theory called *Construction Grammar* (Goldberg, 1995). In Construction Grammar, all *constructions* have a form pole and a meaning pole, and all analyses of sentences contain a form *and* a meaning, as opposed to analyses based on (Chomskyan) context-free grammars or its derivatives.

There seems to be only one other version of construction grammar that was developed with the explicit goal of implementing it: Embodied Construction Grammar (Chang and Maia, 2001; Bergen and Chang, submitted). This implementation is aimed largely towards interpretation of language however. Thus the construction grammar formalism used in these experiments is quite unique in several respects. Steels (2004) dubbed it *Fluid Construction Grammar* (FCG) because of the fact that the grammar is not fixed: it can learn new constructions, and adapt to changing linguistic circumstances.

Category assignment: rules. Rules that describe syntactic constructions containing several other components, do not refer to these other components directly. This would make it more difficult to have constructions where different components can fill the same role, or in other words, it would become difficult to generalise rules. To solve this problem, both on the semantic and the syntactic side, individual items such as words or predicates are mapped onto more generic categories.

In Construction Grammar, the concept of categories seems to be absent. Instead, Construction Grammar has a hierarchical structure in which constructions can be specific cases of more general constructions, much like the class-subclass hierarchy in object-oriented programming languages.

The FCG formalism does at this point in time not have such a hierarchy; constructions cannot refer directly to other constructions to establish such a hierarchy. Nevertheless, the category system provides the possibility to have an indirect hierarchy.

Since constructions (usually) do not work on specific items directly, but on category identifiers added to the semantic and/or syntactic structures, constructions can selectively trigger other constructions by referring to the categories associated to these higher-level constructions. The category system even makes it possible to have not only hierarchical activations; in fact, any other construction can be activated in this way. Also, one category may trigger several constructions at once.

In computer science terms, one could compare this distinction between "direct" hierarchical structure and "indirect," explicit structure with the distinction between backtracking search and depth-first search. In backtracking search, the search algorithm makes use of the hierarchical, stack-driven nature of mainstream modern programming languages. This means backtracking is very efficient, and requires relatively little effort from the programmer, because a large part of the search mechanism is already in place. Depth-first search, on the other hand, is more complicated, because the programmer has to explicitly keep track of the stack of hypotheses that the algorithm still has to elaborate. The advantage however, is that the stack can be changed while the search progresses, or exchanged to any other data structure without modifying the rest of the program. Exchanging the data structure has the effect of modifying the behaviour the search program so that it exhibits other characteristics, that may be more suitable to a specific problem at hand. For example, using a first-in-first-out hypothesis queue instead of a last-in-first-out stack will turn the depth-first search algorithm into a breadth-first algorithm. Characteristic of a depth-first search is that it will explore each hypothesis until it is exhausted before moving to the next one, while breadth-first search will advance each hypothesis one single step, before going to the next one. The flexibility of the algorithm where the control structure is made explicit and put under the programmer's control is also what distinguishes FCG's category system from the strictly top-down hierarchy in general Construction Grammar theories.

Category assignment is effected by two types of rules:

sem-rules: semantic rules assign categories on the semantic side. By making a more abstract representation of (parts of) the current meaning, semantic rules create hooks to which constructions can link.

Here is an examples of a semantic rule. Whenever it encounters the predicate BLUE in a meaning, it assigns the semantic category SEM_1 to that part of the meaning. The reverse is more subtle, because in principle SEM_1 may be associated with several meaning fragments. In the current implementation this is not (yet) the case, so that reverse application is (still) straightforward.

Of course, the meaning part of a sem-rule can be as complex as needed.

$$\text{BLUE}(x) \leftarrow\!\!(0.5)\!\!\rightarrow \text{SEM}_1(x)$$

syn-rules: syntactic rules act exclusively on the syntactic part of a structure, and serve mainly to generalise parts of an utterance, so that the constructions can link to the syntactic pole of an utterance and build up more complex structures.

The following two rules exemplify the structure of syn-rules. In this case, both rules act on specific words that may appear in an utterance. However, they are able to act on any syntactic predicate of the utterance that is represented explicitly in a (partly) decoded utterance, such as word order.

$$\text{“tameri”} \leftarrow\!\!(0.5)\!\!\rightarrow \textsc{syn}_1$$
$$\text{“wabaku”} \leftarrow\!\!(0.6)\!\!\rightarrow \textsc{syn}_2$$

In both cases, the goal is the same: to abstract away specifics. Theoretically but not in practice at this moment, this may lead to several different words (or meaning parts) sharing the same category.

To implement this would require a strategy to reuse existing categories instead of inventing new ones every time a new piece of syntax is created. It is clear that this strategy would have a major impact on the grammar: it would decide which words could replace which other ones in a con-rule. Hence, additional research is needed to develop a good strategy for reusing categories.

Connecting semantics and syntax: constructions. The core of Construction Grammar is the concept that it is not possible to separate syntax and semantics, contrary to theories based on generative grammar, which are essentially devoid of semantics. In Fluid Construction Grammar, the connection between semantics and syntax is made using two types of rules: lexicon rules, which represent the lexicon, and construction rules, which represent more abstract constructions that link semantic patterns to syntactic patterns.

In Construction Grammar, there is no explicit division between the lexicon and grammar. It might seem that Fluid Construction Grammar does implement the two differently. This is not so: both “types” of rules are implemented using the same type of rule. Also, even if Construction Grammar does not explicitly separate the two, it does allow for both to have different features.

lex-rules: lexicon rules represent the lexicon, and simply associate words with their meanings. Additionally, lex-rules have a strength that indicates their success in communication. Whenever a word is used successfully by a speaker, the associated lex-rule's strength is increased, while the strengths of lex-rules with the same meaning (but a different word) will be decreased. In the hearer's case, the associated lex-rule's strength is increased as well, while the strength of lex-rules with the same word (but a different meaning) will be decreased.

$$\text{“tameri”} \leftarrow\!\!(0.3)\!\!\rightarrow \textsc{blue}(x)$$
$$\text{“wabaku”} \leftarrow\!\!(0.6)\!\!\rightarrow \textsc{square}(x)$$

con-rules: construction rules are triggered by matching pairs of syntactic and semantic categories, and cause equalities to be enforced in the meaning or the syntactic representation, depending on whether the rule is being used in production or in interpretation. An example of a con-rule is shown below. Note that for these rules, the order in which the syntactic categories appear in the utterances is relevant. (In the actual LISP implementation, this constraint is represented explicitly in the rule, but here we assume this is the case without introducing an explicit notation for it.)

$$\text{SYN}_1 \wedge \text{SYN}_2 \leftarrow\!(0.5)\!\rightarrow \text{SEM}_1(x) \wedge \text{SEM}_2(x)$$

6.4.2 Example

Recapitulating the example at the beginning of section 6.4, assume that an agent has the lexical, syn-, sem-, and con-rules displayed in the previous paragraphs. The following example shows how, using the different types of rules defined above, the mechanism works that implements equalities to reduce interpretational complexity. Processing of the utterance "tameri wabaku," which should culminate in the meaning given above, goes as follows.
First, the syn-rules are applied:

$$\text{tameri}_{\text{SYN}_1}\ \text{wabaku}_{\text{SYN}_2} \qquad (1)$$

At the same time, in the lexicon the possible meanings for these words are looked up:

$$\text{BLUE}(x_1) \wedge \text{SQUARE}(x_2)$$

To this meaning, the semantic rules are applied:

$$\text{BLUE}(x_1)_{\text{SEM}_1} \wedge \text{SQUARE}(x_2)_{\text{SEM}_2} \qquad (2)$$

Finally, combining (1) and (2) and applying the con-rule to them, equates the variables x_1 and x_2, because in the utterance, SYN_1 appears before SYN_2:

$$\text{BLUE}(x_1)_{\text{SEM}_1} \wedge \text{SQUARE}(x_1)_{\text{SEM}_2} \qquad (3)$$

Here, meaning (3) corresponds to the one given in the example at the beginning of section 6.4.

6.4.3 Implementation

The implementation contains a few other features, most importantly to deal with the combinatorial complexity that can result from processing, usually when interpreting an utterance.

Branching. At several points in the production or the interpretation of an utterance several possibilities may arise. For example, a certain word may have several possible meanings, each of which is a valid candidate for the final meaning of the utterance.

In these cases, each possible hypothesis should be computed until it either succeeds or turns out to be false. The process engine is capable of

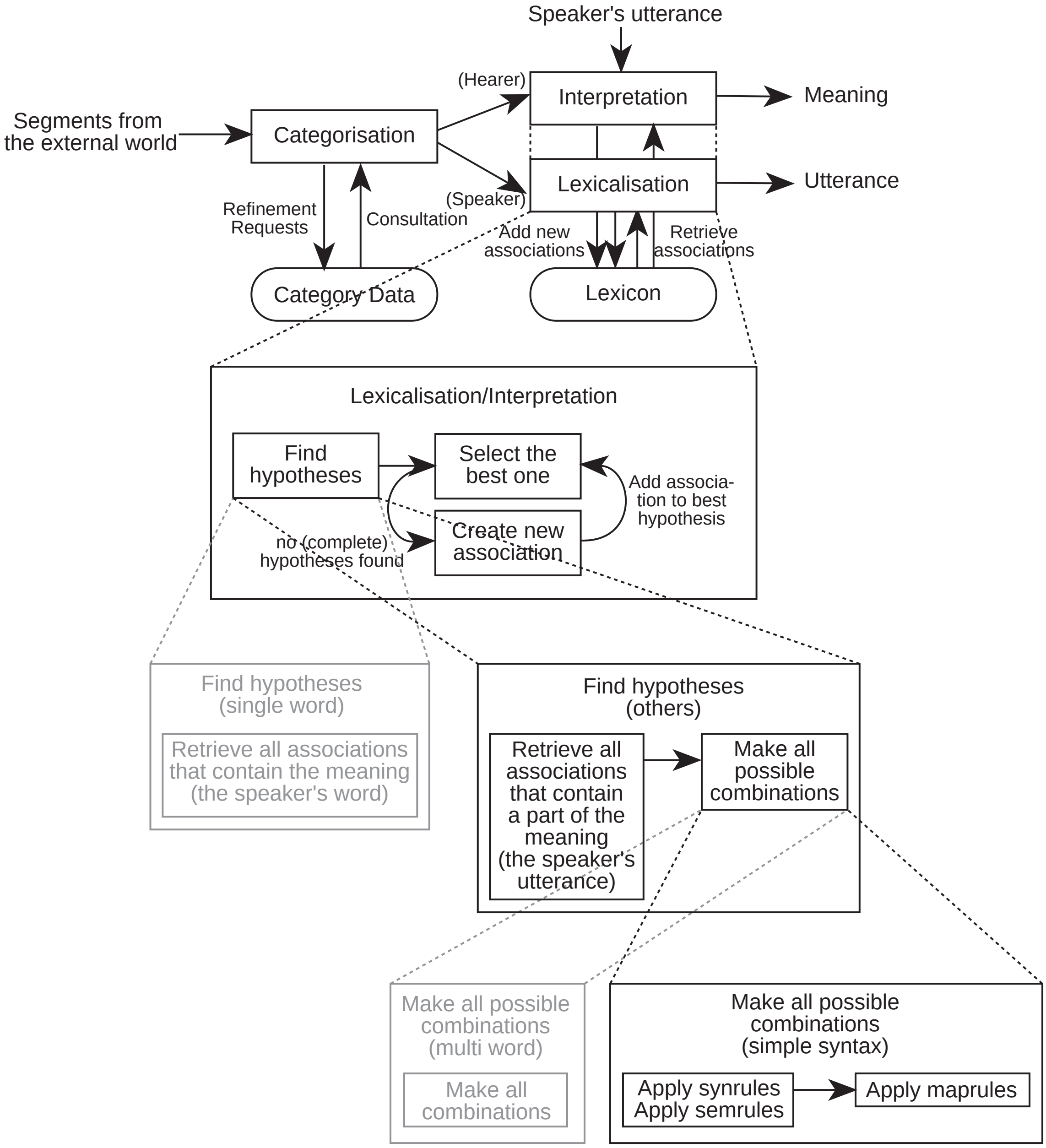

Speaker's utterance
(Hearer)
Interpretation
Meaning
Segments from the external world
Categorisation
Lexicalisation
Utterance
Refinement Requests
Consultation
(Speaker)
Add new associations
Retrieve associations
Category Data
Lexicon
Lexicalisation/Interpretation
Find hypotheses
Select the best one
Add association to best hypothesis
Create new association
no (complete) hypotheses found
Find hypotheses (single word)
Retrieve all associations that contain the meaning (the speaker's word)
Find hypotheses (others)
Retrieve all associations that contain a part of the meaning (the speaker's utterance)
Make all possible combinations
Make all possible combinations (multi word)
Make all combinations
Make all possible combinations (simple syntax)
Apply synrules
Apply semrules
Apply maprules

Figure 6.3: “Pipeline” of the simple syntactic system.

computing several hypothesis in a semi-parallel fashion, i.e. it will record different possibilities and keep them on hold until the current hypothesis has been computed fully. If the result was unsuccessful, the engine will continue computing the next hypothesis until one is found that works or there are no more hypotheses left.

Proxies. Several of the data structures used in processing an utterance are essentially global, such as the lexicon, and the grammar rules. However, the process engine may need to make modifications to these data structures in order for it to be able to continue processing. For example, when it discovers that a certain word is unknown, it will add it to the lexicon and retry interpretation of the words.

To make sure that the different branches that are being computed do not influence each other, each branch has local proxies that record all changes to the data structures and otherwise act as if they are the data structures themselves. Only when it is known which branch computed the final solution will the changes recorded in the proxies of that branch be committed to the actual "global" data structures of the agent.

6.5 Discussion

6.5.1 Categories

Two observations have provided the key to building a working syntactic system: variable equalities, and semantic and syntactic categories. The use of variable equalities has been extensively documented in this chapter: when separate meaning parts are chained together, the arguments that refer to identical referents have to be made equal to constrain the search space of referents to be bound.

What has not been talked about in depth before but is equally important, is the use of semantic and syntactic categories. The categories are used as an intermediate layer between the parts of meaning (form) and the constructions that combine them into larger units. Without categories, i.e. when the parts of meaning or form are plugged directly into the constructions, there is no scope for generalisation, and the grammar, even though it is a real grammar, will remain ad-hoc. For example, you can have a rule that specifies that $\text{BLUE}(x_1)$ and $\text{SQUARE}(x_2)$ can be taken together and solve the equality $x_1 = x_2$, so that the meaning will contain $\text{BLUE}(x_1) \wedge \text{SQUARE}(x_1)$. But it will not be possible to reuse this rule for other combinations of BLUE or SQUARE.

Categories provide a solution for this problem. Rules can refer to categories instead of directly to meaning fragments or form fragments, and categories can refer to several different meaning or form fragments that can appear in the same construction.

Initially, when a new construction is made, new categories will be assigned to the meaning and form fragments that prompted the construction of the new construction. Later, when a new but similar construction would be made in an ad-hoc grammar, the categories can be reused instead of actually having to create a new construction.

In the form of FCG employed in this thesis, reuse of categories is still missing. Although the technical support in is place (in the form of sem-rules and syn-rules) no strategy for reuse of categories has been implemented: whenever a new construction is made, new categories are made along with it without considering if reusing categories is possible.

Deciding when to reuse categories is not a trivial task. It is not difficult to see that deciding whiich meaning and form fragments can appear in the same place as others in an utterance, will have an enormous influence on the appearance of the final grammar. It will take quite some further research to arrive at a good stragtegy for reusing categories.

6.5.2 Syntactic Devices

Natural language has a tendency to, rather than signalling equalities in one specific way, to hint at the presence of equalities in several ways simultaneously. For example, in the sentence "I see the tree", "I" and "see" are members of an equality, namely the equality between the referent of "I", and the agent of the verb "see". This is signalled by the word order, but also by the fact that the form of "see" corresponds to first person singular, which "I" does too. However, what makes this last cue less reliable is that "see" also corresponds to other types of subjects, namely all of them *except* third person singular ("he"). So, instead of using one way to signal each equality, natural language uses several cues, that together increase the probability of an equality being present.

An important way in which equalities are signalled in modern languages, is to *reify* the semantic relationships within the phonetic (phonologic) domain that is otherwise reserved for lexical items only. By introducing markings and function words that represent semantic relationships, the possibilities for expressing different semantic relationships in one phrase or sentence increase enormously.

The grammar will thus have to allow other ways of signalling equalities apart from word order, such as morphology (affixes) and function words. On the other hand, signalling equalities often is not a present/absent affair, so that a probabilistic approach may be called for (at least for the hearer).

Of course, even in modern languages, word order remains an important principle for assigning semantic roles to referents, but (1) there is no *direct* link between (relative) word position and semantic role, and (2) in the vast majority of languages where word order is important, it is nevertheless complemented by an array of other syntactic devices that help in interpretation.

The grammar as it is used in these experiments does not allow to use other syntactic devices than word order. This will certainly be necessary if we want the grammars that arise in the experiments to be more natural-language-like.

6.6 Summary

This chapter introduced a naming game model in which the agents are capable of developing syntactic rules to structure their utterances even more that in the MWNG model.

We looked first at the theoretical complexity introduced by the possibility of referring to more than one referent in an utterance, and how this complexity can be reduced by introducing syntactic structure in utterances.

We looked at the implementation of the model itself: on the semantic level, where the model introduces a new way of structuring the generation and interpretation of semantic descriptions. On the syntactic level, the model uses a prototype of Fluid Construction Grammar. We describe the grammar is structured internally, and how it adds structure to utterances and decodes it again. We also discussed the absence of "real" categories.

Simple Syntactic Naming Game: Experiments

The Simple Syntactic Naming Game is the most recent model of language in the series of models reported on in this thesis. The addition of syntax, based on Fluid Construction Grammar, to the model went hand-in-hand with a redesign of the internal architecture of the agents that implement the agents' cognitive capabilities. This chapter will describe a number of experiments that have been done with this model. First, section 7.1 will describe a number of basic experiments: communicative success and coherence. It also looks briefly at the syntactic component: grammar coherence and grammar use. Section 7.2 looks at the competition between words.

The SSNG model and the Fluid Construction Grammar formalism it implements are still in the testing and development phase. While the SNG and the MWNG have been established and used for some time, many experiments done with the SSNG are still at the level of single interactions, to examine if the mechanisms perform well at the basic level. As a result, the experiments reported in this chapter should be considered as being preliminary. Both the multi-agent model used to produce the graphs and the Fluid Construction Grammar underlying it are constantly evolving, and the results will change (hopefully for the better) as the model changes and improves.

That being said, the results shown in this chapter shed a useful light on the current state of the model, and may provide ideas on the direction that further improvements should take.

7.1 Basic Experiments

7.1.1 Predicate Semantics Algorithm Details

Specialist Trigger Order

The predicate semantics as implemented in the SSNG uses several specialists to conceptualise different perceptual subdomains. However, not all specialists are used every time a meaning needs to be constructed. While a meaning is being constructed, it is continually evaluated to see if it is discriminative or not. As specialists provide submeanings, they are added to the "global" meaning. One parameter of the predicate system could thus be the order in which the different specialists are triggered. In the current system, the order is determined implicitly by the number of iterations of the system needed by each specialist to compute its result.

7.1.2 Simple Syntactic Naming Game Algorithm Details

Word Creation and Storage Probabilities

In the previous experiments, probabilities were associated with the creation of new words and learning a word from the speaker (section 3.1.1). In these experiments, there are no such probabilities. This means that a speaker will always create a new word when it cannot lexicalise a meaning, and a hearer always learns a new association (or associations) when it fails to understand the speaker.

Association Strength Updates

Contrary to the previous experiments (see section 3.1.1), word-meaning associations in the SSNG have a single number (between 0 and 1) that represents their strength. These numbers are increased or decreased depending on the outcome of an interaction. In case an interaction involved several worlds, one could imagine many strategies to update the strengths of all the associations, ranging from very simple linear increases or decreases, to complex schemes where the weight of a word in an interaction is taken into account. In the experiments reported here, the strengths of all associations involved in an interaction are increased or decreased with the same amount (0.1).

In these experiments, like in the earlier experiments, a lateral inhibition mechanism takes care of decreasing the strengths of associations competing with the ones used in the final utterance.

Word and Meaning Combinations

The strength of a combination of associations is calculated from the strength of the individual associations that make up the combination. For a combination of associations $a_1 ... a_n$, the strength is calculated as follows:

$$\mathit{strength} = \frac{\sum_{i=1}^{n} \text{strength}(a_i)}{i}$$

i.e., the average of the strengths of the associations. There is no difference in score calculation between speaker and hearer in the SSNG experiments, as there was in the MWNG experiments (section 5.1.3).

Grammar Rule Strength Updates

Grammar rule strengths are used and updated in the same way as association strengths, including lateral inhibition (see above).

7.1.3 Communicative Success

Figure 7.1 shows the success rate (averaged over 10 experiments) of a basic 2-agent syntactic naming game experiment. It is on average 10% lower than a similar multi-word naming game experiment (fig. 5.2, p. 84), and on the same level as the final, complex single word naming game (fig. 3.4, p. 47).

First of all, after a while, most if not all of the games in which a single word is used, are successful. Figure 7.2 shows the percentage of successful single-word games: after the initial period of confusion, almost all single-word games are successful. This shows that the basic structures of the naming game are still operational. This makes sense, since the basic lexicon mewchanism is the same as in the previous models (despite being implemented using the rules used in the grammar), and the grammar does not come into play for single-word utterances.

On the other hand, the grammar does come into play for *all* multi-word interactions. When a speaker wants to use more than one word, and does not have a grammatical construction in its repertoire that is applicable, it has to invent a new construction to accommodate the words and meaing it wants to combine. Quite a number of these games fail, and not only in the beginning of a series of interactions; the SSNG does words on multi-word utterances than the MWNG. The failures can be traced to several low-level causes. In some cases, the problem is the interactions between the processes when an agents tries to interpret an utterance. These are coding problems that will no doubt eventually be ironed out.

Systematic failures

Other problems are more systematic, and require a decision by the modeler to solve. For example, figure 7.3 shows the amount of "communicative failures"; this is $(1 - s)$ for every interval, where s denotes the number of communicative successes. The other curve shows the amount of multi-word games in which the function `invent-new-words` was responsible for the failure. The graphs show that the amount of failures of this type is fairly high compared to the total amount of failures.

An important way in which this function fails, and which is not a coding problem, is that it is not capable of assigning words to meanings when more than one of the word in a multi-word utterance is unknown (or is known, but with a meaning that is not a part of the correct meaning). This is a consequence of a conscious decision that has been made about the rest of the system in the design stage of the model. The idea behind the decision is that in such cases it is better to be conservative and trust that in future interactions, either or all words that

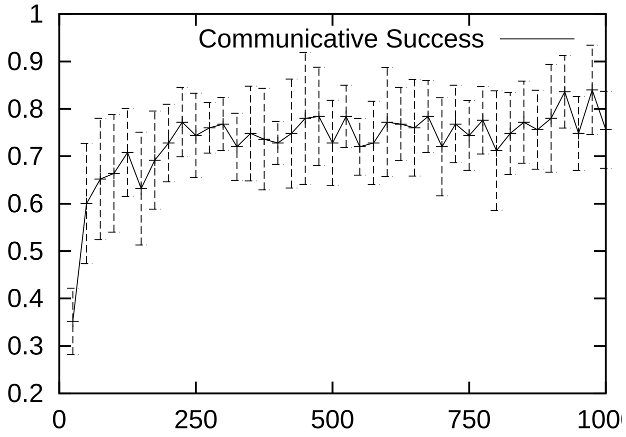


Figure 7.1: Communicative success in a Simple Syntactic Naming Game (averaged over 10 experiments).

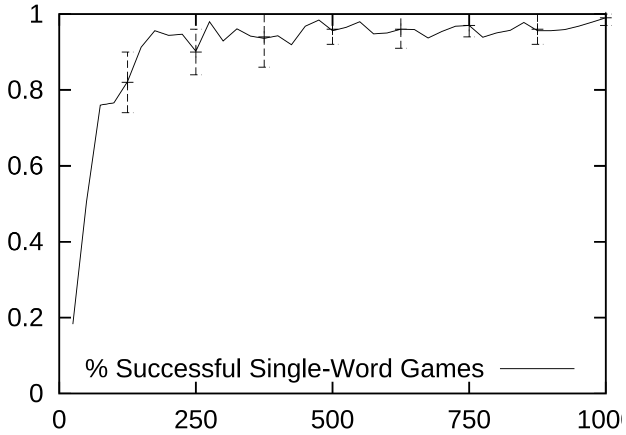


Figure 7.2: Percentage of single-word interactions that was successful (averaged over 10 experiments).

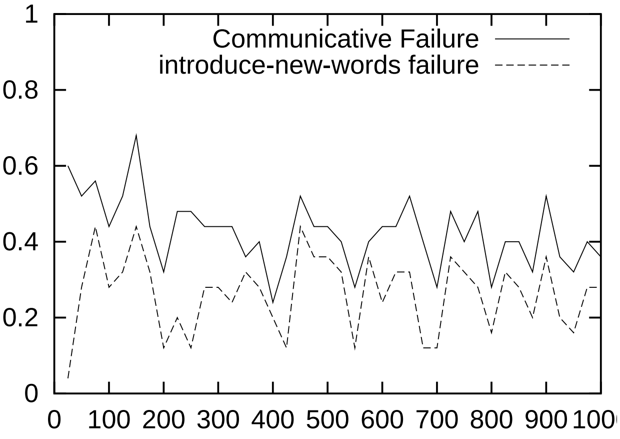


Figure 7.3: Fraction of failures in `introduce-new-words`.

are unknown at this point will be learned. In future interactions in which the same utterance occurs, the agent would then be able to interpret the utterance.
As described in section 6.3.2, the semantic system of an agent will collect all pieces of meaning in one large unit, in order to avoid biasing the language with the structure the designer puts in the specialists. The explicit action of consolidating the meaning in a single meaning block is what hampers the listener in this case: it has one block of meaning and several words it should attach to parts of that meaning. The system does not contain an algorithm yet that splits the meaning again, so that these failures could be used productively to learn the words. Also, in general the problem of assigning several words to several meanings without guidelines on what belongs together creates a combinatorial problem that could pollute the lexicon with large amounts of useless entries.
As can be seen from the graph, the number of games that fail in this way is substantial. This means of course, that learning will proceed much slower.

7.1.4 Coherence

Vocabulary Coherence

Figures 7.4 and 7.5 show the evolution of form-meaning coherence in a three-agent and a five-agent population over 1000 games. Form-meaning coherence is high in both populations. This means that that the basic mechanism of vocabulary organisation still work will underneath the complexity of the grammar mechanisms. For the word-meaning associations that have been learned, the agents agree largely on their meanings, so the agents seem to be capable of creating a basic lexicon that is coherent.
Comparing vocabulary coherence with the results of the MWNG (fig. 5.2, p. 84), shows that coherence in the grammar experiments is higher than in the multi-

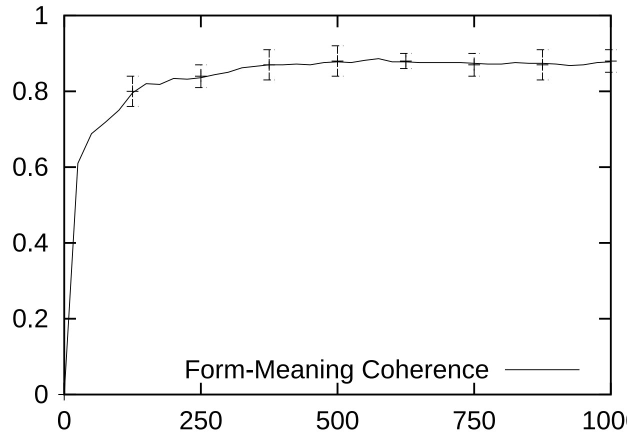


Figure 7.4: Form-meaning coherence in a 3-agent experiment.

word experiments. This would seem to be a good thing, however, because of the different scales of the experiments (in terms of population size and number of interactions played), we cannot be certain that the good result is not due to the smaller population or the shorter duration of the experiments. Nevertheless, it is a good sign that at least smaller populations reach high vocabulary coherence levels.

Grammar Coherence

A new way in which the communication systems of the agents in this model can and should be evaluated is checking whether their grammatical systems converge. Section A.2.2 explains a simple way of measuring grammar coherence: calculate all possible combinations of words that each agent's lexicon and grammar allow, and then calculate how similar these allowed combinations are between agents.

In a sense this measure does not really calculate a coherence measure, since what it calculates is each agent's possible utterances. The other coherence measures actually look inside the agents' data structures and compare the rules themselves. This is not straightforward with the current implementation of the grammar, because of the way in which the rules are implemented.

Figures 7.6 and 7.7 show the grammar coherence measure applied to a three-agent and a five-agent population. In both cases, grammar coherence is low. This may be a consequence of the fact that the negotiation about the lexicon is not settled sufficiently. The agents will not attempt to interpret an utterance grammatically when it cannot be interpreted fully lexically. Thus, interactions that fail on a lexical basis will not contribute to the grammar.

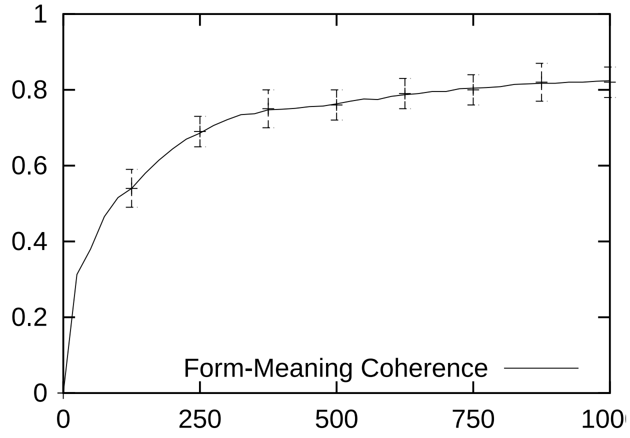


Figure 7.5: Form-meaning coherence in a 5-agent experiment.

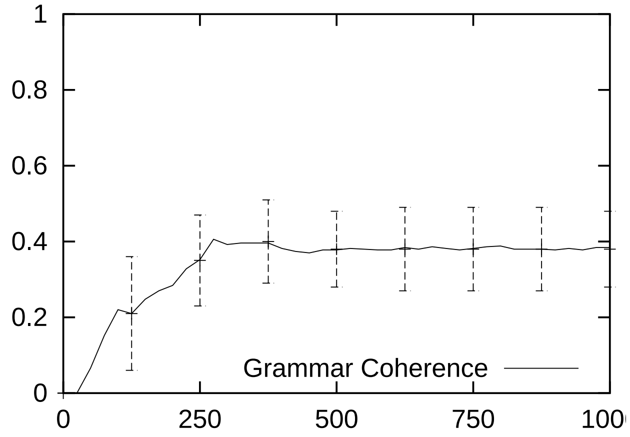


Figure 7.6: Grammar coherence in a 3-agent experiment.

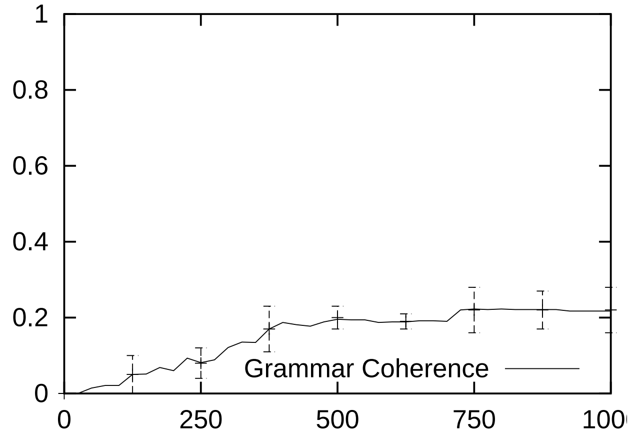


Figure 7.7: Grammar coherence in a 5-agent experiment.

7.1.5 Lexicon Size

Figure 7.8 shows the evolution of the size of the agents' lexicons over a period of 1000 interactions. A tentative comparison with fig. 5.4 (p. 86) shows relatively similar lexicon size numbers, although it would seem that the MWNG agent reaches a platform where the agent does not really need to create new words any more but can rely on its lexicon to utter most any meaning it encounters in its world.

The SSNG agents do not seem to reach a platform, but the curve becomes at least much flatter as that of a SNG agent (fig. 5.5 p. 86). An explanation of the continuing increase in size of the lexicon was already given above—and seems to be an issue of the model, in terms of the strategy used by the hearer to decide whether or not to absorb unknown words.

From fig. 5.4 and fig. 5.7 (p. 88) it would seem that even if an agent, measured individually, reaches a "mature" lexicon, if we take a broader look at more and longer experiments with larger populations, this is certainly not a given. Despite this, however, the trend to create much less new words is obvious.

7.1.6 Grammar Use

Figure 7.9 shows the evolution of the use of grammatical rules during an experiment. It measures, for games in which multi-word utterances are used (and hence grammar) the number of con-rules used. It disregards lex-rules, sem-rules and syn-rules, because these are applied in the single-word games as well, and do not really give an indivation of grammar usage (or, more precisely, the use of equality-resolving rules). Like for the success and coherence measures, this is measured over intervals of 25 games, except that only those games are taken into account in which grammar is actually used.

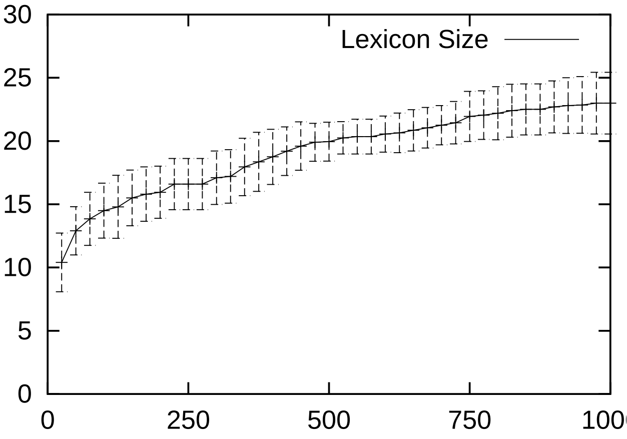


Figure 7.8: Evolution of lexicon size.

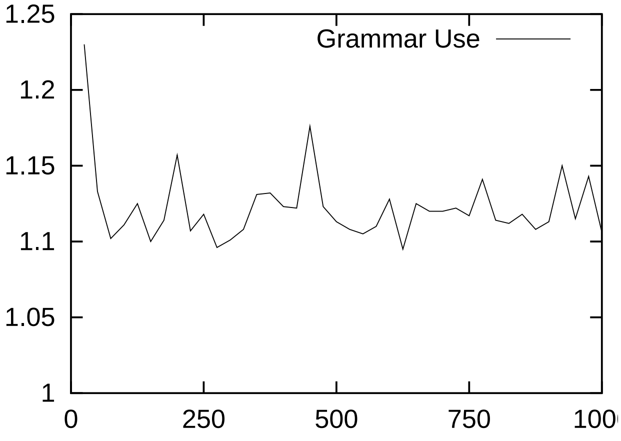


Figure 7.9: Evolution of grammar use.

The graph indicates an average grammar use of 1.1 to 1.15 rules per utterance that uses grammar, i.e. slightly more than one rule. This corresponds with the observation that most of the grammar games involve two words and one con-rule, with occasionally three-word and even four-word games.

The measure does not take into account the success or failure of grammar games. Observation of the running model indicates that most of the two-word games are successful, but three- or four-word games are successful only occasionally. Mostly the grammar has problems when three variables are equal (i.e. two con-rules are required where one part of each con-rule applies to the same part of the utterance). If the con-rules do not overlap, the grammar is capable of applying them to the same utterance. This would seem to be an implementation problem, as this is certainly not a design requirement.

7.2 Word Competition

The figures on page 135 give examples of how the "competition" between different words for the same meaning in a simple syntactic naming game proceeds. For each word, the value depicted on the graph is simply the sum of the stengths for that word in each agent's lexicon.

Figure 7.10 shows the typical form of competition in a 2-agent experiment. Essentially, there are two words for one meaning. At one point, one word is introduced by one of the agents, and increases to a score of 1. Some time later, another word appears, after a while also with a score of 1. What happens here is, that one agent uses a word consistently, and some time later the other agent introduces another word for the same meaning, and also uses it consistently. Hence, like in the other types of language games, there seems to be no "real" competition between the words in such a small population.

Figure 7.11 shows word competition in a population of 5 agents. Overall, only the word "madoxi" is used only very briefly. The other words are all used with some success. In the end "gikaze" and "kakuxi" settle on a score of 1, which implies that they each are used exclusively by one single agent that uses it consistently. The word "kaxufo" settles at a score of 2.6, which means the other 3 agents in the population prefer using it, but not with a perfect score. Towards the end of the experimental run, there is a still a small increase in score, which means that its coherence is still being strengthened.

Figure 7.12 also shows a case in which the population settles on more than one word. Already after approx. 600 games, the final situation is reached: "dazoze" is preferred by 6 out of 8 agents and "lelefa" is preferred by the other 2 agents.

Concluding about word competition, we can say that despite the fact that the lexical coherence reached by a population is often not optimal, they do converge to a stable state that can be used as a platform for bootstrapping more complex communicational mechanisms. Possibly other factors such as a different population dynamics, could allow coherence to become higher.

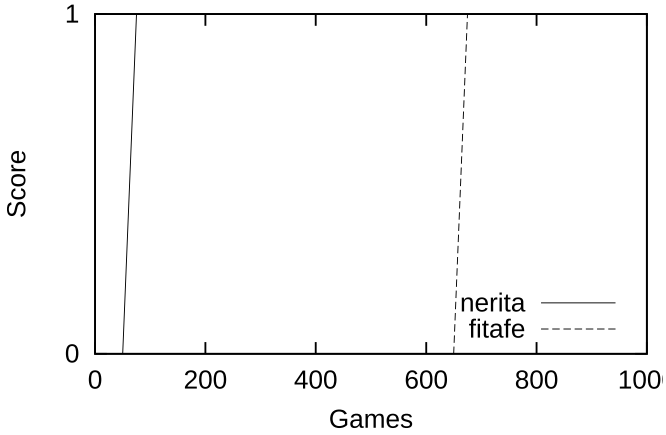


Figure 7.10: Competition between words for BOX in a 2-agent population.

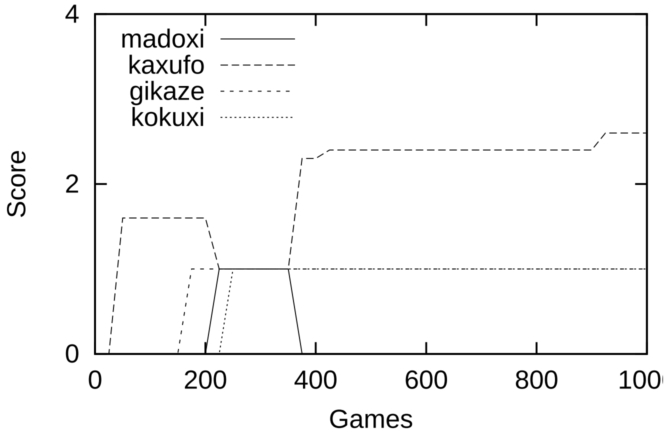


Figure 7.11: Competition between words for TINY in a 5-agent population.

Population Size	Number of words per meaning	Standard deviation
2 agents	1.21	0.43
5 agents	1.81	0.98
8 agents	2.33	1.53

Table 7.1: Average number of words per meaning (cfr. synonyms).

Population Size	Number of meanings per word	Standard deviation
2 agents	1.49	0.59
5 agents	1.44	0.65
8 agents	1.39	0.58

Table 7.2: Average number of meanings per word (cfr. homonyms).

7.2.1 Synonyms and Homonyms

Tables 7.1 and 7.2 show data about synonymy and homonymy in the SSNG. The data are averaged over ten experiments. As the population level increases, synonymy increases, and homonymy remains at about the same level. These trends corresponds to the trends seen in the SNG (tabs. 3.2, p. 49 and 3.3, p. 51) and the MWNG (tabs. 5.1 and 5.2, p. 90). In these previous experiments, there is so much variation between experiments (witness the high standard deviations) that it is impossible to see if they are actual trends. In these experiments the standard deviations are lower compared to the previous experiments, but still too high to draw actual conclusions.

7.3 Summary

This chapter presented experiments done with a model that enables its agents to use syntax to structure its utterances. Briefly, the following conclusions can be drawn on the basis of the experiments performed and described.

- In general, the model is still in its early stages of development. Also, it consumes a lot more computational resources than any of the other models. This is not really a problem when using the model to test the language model on a single-agent basis, but it poses problems when performing experiments with larger populations and environments.

- Communicative success is relatively high for the model, but an analysis of the communicative failures shows that there may be a fundamental issue underlying these failures. Most failures have the same cause: more than one word in the utterance is unknown to the hearer, and hearers are designed to defer such problems, in the hope that later interactions may resolve either or all of the unknown words in this interaction.

- The games that are successful, which is still the great majority of interactions, show that the basic mechanisms are still sound. The lexicons that the agents develop, despite failures, show relatively high coherence. At this point though, it is hard to say whether this is the result of the grammar, or because it is the result of the smaller scale of the experiment as compared to the multi-word model.

- Grammar coherence is low, which indicates that there is still work to do before the grammar will be able to self-organise efficiently. Also, the grammar coherence measure is relatively ad-hoc, and may not give an accurate assessment of the grammar as it is actually used by the agents during an experiment.

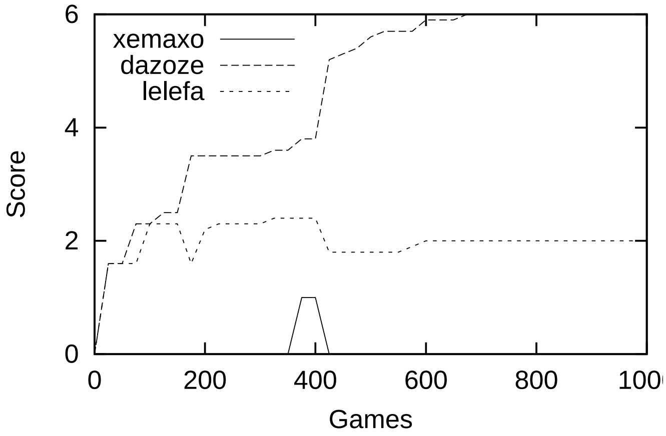


Figure 7.12: Competition between words for MOVE in an 8-agent population.

Conclusion

RESEARCH INTO the origins and evolution of language is difficult in many ways. There is a lack of factual evidence, because the earliest recordings of language are in written form, and these show that language had already matured by the time they were made. Another difficulty is the mere fact that language is extremely complicated, and this is only aggravated by the fact that linguistic utterances are generated by the brain, which in itself is extremely complicated and as yet poorly understood.

Nevertheless researchers have come up with many clever ways to approach the subject anyway. These range from dissecting language itself to understand it better, to relating language to other aspects of human evolution, to trying to reverse language change in order to go back in time, etcetera.

This thesis uses synthetic modeling to approach the evolution of language in yet another way. Assuming that language (in the sense of the group language of a population) evolved continuously, we try to reconstruct different stages of language. These models allow to compare both the languages and the individuals generating these languages in different respects. Populations of individuals are modeled using multi-agent models, in which the individuals (agents) interact in predefined ways about a simulated or real environment that they can perceive.

8.1 Models

Concretely we present three individual models, which can be viewed as snapshots of different stadia in the evolution of language:

Simple Naming Game: a model in which the agents are capable of producing and interpreting single-word utterances. Of this model there are two versions: one without a separate meaning layer, and one which uses the Discrimination Game as its meaning generator. Two variants of the SNG are also presented: a stochastic variant, where several aspects of the model are replaced with stochastic analogues, and the Talking Heads experiment.

Multi-Word Naming Game: a model in which the agents can produce and interpret utterances consisting of several words. Here too, there are two semantic variants: one which uses the Discrimination Game, and one which uses a more complex and expressive predicate-based notation.

Simple Syntactic Naming Game: a model in which the linguistic capabilities of the agents include a syntactic component that can structure utterances longer than one word. This model uses also the predicate-based semantics, but it is implemented differently. The new architecture is described in detail.

Evolution between these different stadia can only be possible if each stadium conveys a certain advantage to the agents compared to the previous stage. We look at this by evaluating each communication system according to several criteria: communicative success, coherence, and lexicon properties, and comparing them.
The fourth model is a hybrid model incorporating the communication strategies (and capabilities) of the first two models. The agents can use both strategies, and "decide" on which strategy to use based on selection pressures. We look at a number of these selective pressures, which are based on the measures that have been used to evaluate the other models.

8.2 Results

8.2.1 Individual Models

The Simple Naming Game (chapters 2 and 3) showed good communicative success and coherence. The variants on the basic model, which introduced a form of semantics, stochasticity, embodiment and large-scale experiments, also performed well. This shows that the basic mechanisms of lexicon organisation work well. The results obtained with these experiments also represented the base line against which the results of the other models are judged.
The Multi-Word Naming Game can be said to be at the same level as the Simple Naming Game in terms of communicative success and coherence. It is more efficient than the SNG in the sense that it reaches the same performance levels with smaller lexicons. A problem with the MWNG, and something which should be looked at further, is the fact that the basic model uses separate forms in utterances. This is not realistic, and the model should be adapted to cope with utterances in which the forms are not individually separated.
The Simple Syntactic Naming Game marked the first language game experiments with a syntactic component. Despite the fact that the model is still in early development, it works relatively well in terms of communicative success and lexicon coherence. The results are less clear where the syntax is concerned; a simple ad-hoc grammar coherence measure shows relatively low coherence. The fact that the experiments with the SSNG are of a much different (smaller) scale than the experiments with the previous two models, makes them difficult to compare.
The hybrid model shows that it is possible to use the external measures that were used for evaluating the other models as internal pressures on communication strategies. A caveat with the current model is that the agents must select their communication strategy probabilistically each time rather than deterministically. In the latter case, the numerous failures in the beginning of a series of

language games simply pushes each agent towards the strategy it is *not* using in the beginning.

The pressures that were tested in the experiments do not allow to draw definitive conclusions; two of them reliably caused the population to migrate towards using the multi-word communication strategy using the current experimental parameters, but it is not clear yet whether they are robust enough when experimental parameters change, such as e.g. the fraction of the initial population that already prefers the multi-word strategy, or the initial strength of the respective strategies. More experiments should be done using this model.

8.2.2 Global Interpretation

The results obtained from the four models described in the thesis are not always clear; especially between the two first models and the third model.

Also, in interpreting the results of simulation experiments, a measure of relativisation is necessary: it is impossible to state that the simulations recreate the situation as it was in reality, both because of a discrepancy in complexity, and because in our case it is simply impossible to know what the real situation used to be.

Nevertheless, on the basis of the experiments in this thesis it seems that it is indeed possible to conclude that:

1. communication systems of different levels of complexity can be very successful in their own right;

2. it is possible to state that more sophisticated communication systems are more efficient than less sophisticated communication systems, and that it is possible to measure these differences using relatively simple measures;

3. these measures can be reshaped as agent-internal pressures that can guide the evolution of language.

These conclusions support the hypothesis that language evolved continuously, as is also suggested for example by Jackendoff. The models in the thesis seem to intersect at different points with the milestones he proposed. However, some models incorporate several milestones at once, while other milestones are not treated here. This suggests that the milestones in Jackendoff's schema should not be seen as separate stages in the evolution of language, but rather as suggestions of capabilities that must have arised at some point.

How all this relates to the biological evolution of humans and the brain is hard to say. It is obvious that different models, and hence communication systems of different levels of sophistication, will need different mechanisms, but whether these mechanisms need a specific biological basis, and whether this basis would be unique for language or not, is impossible to say at this moment.

8.3 Future Research

There are some immediate possibilities for further research based on the models described in the thesis.

In the Multi-Word Naming Game it is necessary to look at the difference between the case where utterances can be composed of several symbols versus the case where "long" utterances are concatenated in one symbol. The results of the current experiments show that in some cases the agents become trapped in a vicious circle where composite utterances are repeatedly interpreted as single words, such that the rate of expansion of the lexicon remains high, instead of becoming progressively lower as in the current multi-word model. (This issue is also of concern for the hybrid model, in which the single-word strategy and the multi-word strategy must coexist, and cooperate.) One suggestion to approach this problem would be to use an explicit generalisation mechanism, such as those used in the Iterated Learning Model's grammar inducer or Neubauer's colour naming model.

In the Simple Syntactic Naming Game, the grammar learning and production algorithms need to be extended further. Since the experiments were done with the model described in the thesis, a lot of work has already been done on the grammar, but outside the context of the multi-agent games such as those described here (Steels, 2004). Concretely, in order to evolve towards a "real" grammar (of the type of model 5 in fig. 1.1 on p. 5) the most important extension would be to introduce "real" categories, as discussed in section 6.5.1. Other than that, there are still a number of other essential features of natural language grammar that are not (yet) supported by FCG: recursivity, subordinate clauses, function words, etc. If we wish to comment further on the evolution of grammar, all these features need to addressed at some point. Finally, the grammar model should also be able to cope with utterances in which the forms are not separated.

The hybrid model has not produced any clear results. This model needs to be worked on further in two areas. First of all there is the issue that also surfaced in the multi-word naming game model of using a single symbol to convey composite utterances. This problem needs to be solved for the two strategies to be able to coexist. Secondly, other selection pressures need to be looked at. The ones examined in this thesis are all linearly related to communicative success, i.e. every failure has the same negative influence, and every success has the same positive influence on the strategy used. There are however more failures in the beginning of a series of language games than later on. In a number of cases, this seems to lead to indecision, because the population does not move in a specific direction. In other cases, most notably when the agents deterministically select their communication strategy rather than probabilistically, this seems to kill off the agents' initial strategies and simply push them toward the other one. Two criteria were promising, and need also to be looked at further. Another possibility is to look at pressures that are not influenced as directly by communicative success.

The concept behind the hybrid model, giving agents the opportunity to use different communication strategies, extends to other communication strategies as well. The transition from the MWNG to the SSNG could be approached in a similar way, as well as the transitions between different types of semantics. There is potential for many interesting experiments here.

Measures

IN THE experiments with the different models described in this thesis, a limited number of measures are used for all experiments. Understanding these measures is fundamental to understanding the experiments that have been described in the previous chapters, and to understand how they can be used to validate the models. Therefore, this chapter will describe these measures in more detail, using mathematical notations where appropriate to eliminate ambiguity.

A.1 Communicative Success

Basic communicative success Communicative success is the most basic measure for a series of interactions. Over the course of an interval, it records the results (success or failure) of every language game. These results are then combined into a single number by dividing the number of successes by the size of the interval, to yield a number in the interval $[0, 1]$ that gives the percentage of successful games in the interval.

$$\begin{aligned}
\text{res}(j) &= \begin{cases} 1 & \text{(game } j \text{ is successful)} \\ 0 & \text{(game } j \text{ fails)} \end{cases} \\
\text{CS}_m^n &= \frac{1}{(n-m)} \sum_{j=m}^{n} \text{res}(j)
\end{aligned}$$

The function $\text{RES}(j)$ returns the result of game j. The measurements are done over subsequent, non-overlapping intervals. The size of the intervals is usually 25 games, which gives a resolution of 4%. Figure A.1 shows an example of the basic communicative success measure.

Irrelevant and "unplayed" games Depending on the complexity of the world in which the agents live and the perceptual and semantic capabilities the agents have, a speaker may not always be capable of producing an utterance. When this happens, the game the speaker is initiating will automatically be a failure, since the hearer will have nothing to interpret. Usually it is desirable to monitor the communicative success only when there has actually been communication, so it can be useful to disregard these games, when there are many of them.
As a rule of thumb, monitoring the communicative success of particular types of games is useful to determine the cause of a low global level of communicative success. When a particular type of interaction yields a low success, this

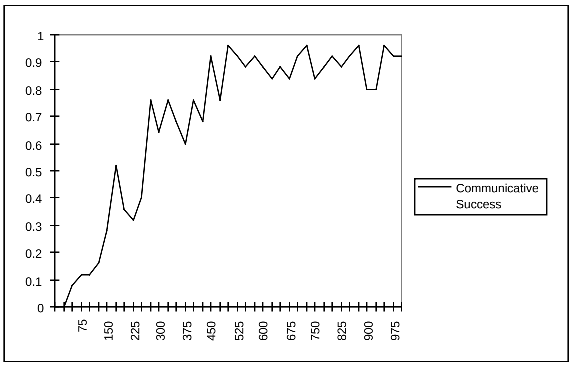
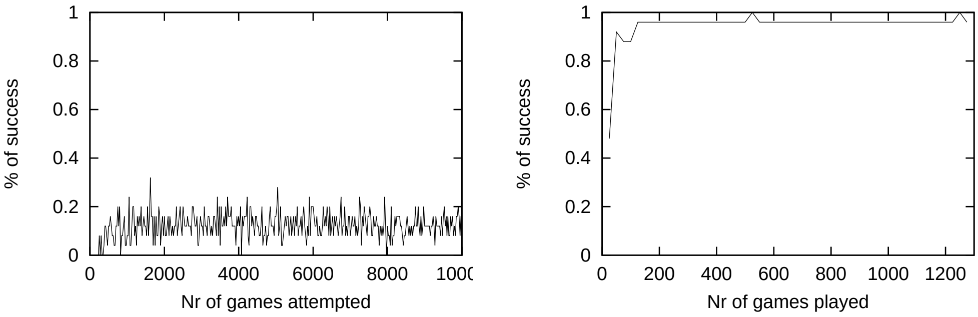


Figure A.1: Communicative Success.



(a) Success in all games

(b) Success in played games only

Figure A.2: Communicative success.

may indicate that the mechanisms responsible for this type of game is faulty or perform below par.

An example of this is shown in Van Looveren (2003). In this paper, a model is presented that aims to study definiteness. Initial tests with the model reveal an unusually low level of communicative success (see fig. A.2 (a)). It turned out that the low success rate was due to a high fraction of unplayed games—the world was often too complex for the speaker agent to conceptualize, precluding the possibility of successful communication. Temporarily disregarding these unplayed games revealed a normal success pattern: low at the beginning, but rising fast and consistently high after that (fig. A.2 (b)).

In the Multi-Word Naming Game, there are two types of interactions: interactions in which only one word is used, and interactions in which several words are used. Sometimes, we are only (or specifically) interested in the subset of games in which long utterances were used. Figure A.3 shows the data for communicative success both for all games and for multi-word games. In this case, the curve for multi-word games has been projected onto the actual games. To do this, whenever an interval of multi-word games was completed, the game in

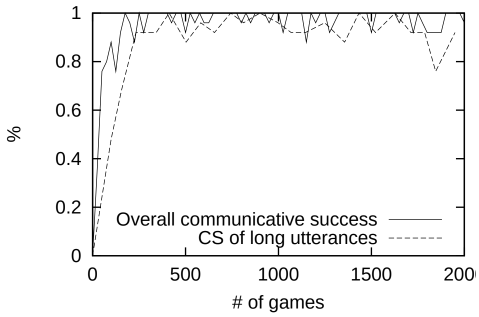


Figure A.3: Success in all games and multi-word games.

which this happened was recorded with the result, so that it became possible to "stretch" the data across the whole X-axis.

In terms of the formula that calculates the success over an interval, we are adding a condition to the sum:

$$\begin{aligned} \mathrm{res}(j) &= \begin{cases} 1 & \text{(game } j \text{ is successful)} \\ 0 & \text{(game } j \text{ fails)} \end{cases} \\ \mathrm{CS}_m^n &= \frac{1}{(n-m)} \sum_{\substack{j=m \\ \mathrm{utterance}(j)\neq \mathrm{null}}}^{n} \mathrm{res}(j) \end{aligned}$$

where the function UTTERANCE(j) returns the utterance that was generated by the speaker in game j, and RES(j) returns the result of game j.

A.2 Coherence

In the Simple Naming Game, coherence is a simple measure between two agents, which compares their lexicons. The goal is, as opposed to communicative success, to have an *internal* measure of how well the naming game converges. Communicative success looks at communication from an external point of view, whether a hearer is able to point out the correct topic based on the speaker's utterance.

While this can be an important way of assessing whether the communication works, it is also interesting to look at the (distributed) internal representation of the communication system itself in the different agents, and more specifically, in how far these internal representations resemble each other. By comparing the lexicons and taking into account the differences between different agents' lexicon entries, it is possible to measure the quality of the global communication system.

To this end, *coherence* measures have been introduced. Coherence is measured between two agents, and compares the agents' corresponding internal datastructures, such as their lexicons. Global coherence is then calculated by averaging over the size of the population:

$$\text{Coh(pop)} = \frac{1}{n^2} \sum_{j=1}^{n} \sum_{\substack{i=1 \\ i \neq j}}^{n} \text{Coh(pop}_i\text{,pop}_j\text{)}$$

This formula actually simply calculates the average of all elements of the agent pair matrix. Notice in this formula that we treat Coh(pop$_i$,pop$_j$) separately from its converse: Coh(pop$_j$,pop$_i$). The coherence measure used is not necessarily symmetrical, and while we could include this consideration into the Coh(x,y) measure itself and only average over one half of the combination matrix, this is equivalent to simply averaging over the whole matrix. (Unless one uses a different function for combining Coh(x,y) and Coh(y,x), which we do not.)
Notice also that we explicitly exclude the diagonal of the matrix in the calculation. An agent will always have a coherence of 1 when paired with itself. Additionally, since agents will never interact with themselves in a language game, there is no interest in including these pairs in the final measure.

A.2.1 Lexical Coherence

Coherence measures the extent to which all agents in the population use the same words for the same objects. It is measured by counting, for every object, how many agents prefer to use the same word for it.
Generally, coherence will not be total, because some agents will prefer different words than others. In this case, the word that most agents prefer will be considered the one that the population prefers. Averaging this over all meanings gives a measure for the quality of the language that the agents developed.

Basic lexical coherence In the Simple Naming Game, lexicon entries consist of a word, a referent and a score. The referent is an entity from the external world, such as an object or an agent. It is important to understand, in this most simple version of the naming game, that the "external" referent (in the simulated world) and the "internal" referent (which is associated with the word in the lexicon) are one and the same. In the Simple Naming Game, there is a one-to-one relationship between external referents and their representation in the lexicon.
There are two types of lexical coherence in this game: word-referent coherence and referent-word coherence. In the first case, what we want to know is in how far the agents refer to the same referents with each particular word. In the second case, we want to examine for each particular referent, in how far the agents prefer the same word.
In principle, since the lexicon is a symmetrical data structure, one would think that both would result in the same number. Ideally this would be the case, but this happens only when there is exactly one word for each referent. This may be the stable state of a small experiment after a series of language games,

but in intermediate stages, and in more complex experiments where there is no one-to-one relationship between internal and external referents any more, such "bijective" stable states are less likely to occur. Where necessary then, we should take into account this difference.

The formula below defines lexical coherence for the word-referent case. Suppose a_1 and a_2 are agents from a population, L_{a_i} denotes agent a_i's lexicon, and W_{a_i} denotes the set of words w appearing in a_i's lexicon. The function lookup(a_i, w) looks up the best association for word w in a_i's lexicon, "best" being defined by having the highest score:

$$\text{LexCoh}_{\text{SNG}}(a_1, a_2) = \frac{\sum_{w \in (W_{a_1} \bigcap W_{a_2})} (\text{lookup}(L_{a_1}, w) = \text{lookup}(L_{a_2}, w))}{\#(W_{a_1} \bigcap W_{a_2})}$$

The result of this formula is a number that gives the fraction of the words for which both agents prefer the same referent and the total number of (different) words in both lexicons.

Experiments with meaning The extension of the naming games to include the discrimination game decouples the lexicon from the referents, using meaning as an intermediate step. This also means that there is now a strict division between an agent's external and internal world: the agent's "interface" layer of senses is the only contact layer between the world and the agent's internal state.

This decoupling of the world from the lexicon means that it is no longer self-evident that lexicons are directly comparable. The discrimination step between perception and lexicalization may have different results in different agents, even in identical states of the world, because it also depends on the internal state of the agent.

For the Talking Heads and other experiments in which there is no direct, onto-one relation between meanings and referents, the notion of coherence has to be extended. Since the Talking Heads have to use their robotic bodies to perceive the environment, they have to use their own internal representation of each object instead of the object itself in their associations (as can be done in simulations). Additionally, their task is to find a unique description for the topic that is not applicable to other objects in their environment, which means that in different interactions, the same object may be represented by different meanings.

In this case, coherence can be calculated not only between words and objects, but also between words and meanings, and meanings and objects. Unfortunately, in the Talking Heads experiment calculation of meaning-object or word-object coherence is not possible, because there is not enough information in the database to reconstruct the referents of the interactions.

Different types of coherence This in turn gives rise to three different types of coherence: form-meaning coherence, referent-meaning coherence, and form-

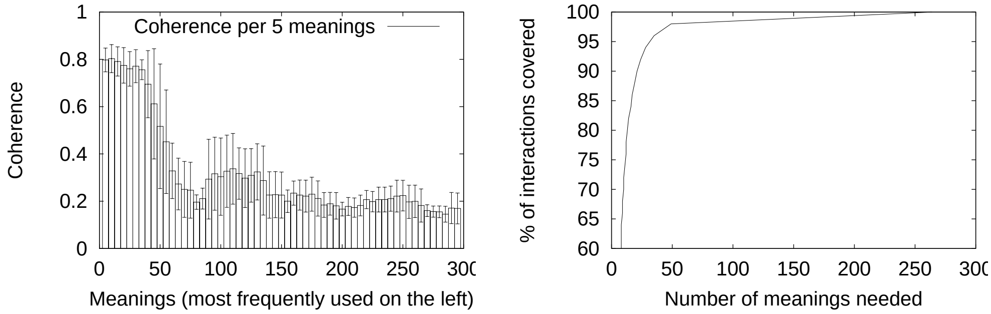


Figure A.4: Coherence data from the Talking Heads experiment.

referent coherence. The last one is again an external measure: disregarding the meaning step, look at what forms are being used for which referent.

In all experiments with meaning, form-meaning coherence has been used as "the" coherence measure. Essentially, we are still basically comparing the different agents' lexicons directly. This means that coherence includes the coherence between agents in terms of meaning. If meaning coherence can be shown to be high enough, it will become less important in the end result, and the coherence measure will be more representative of the actual lexicon coherence. (See e.g. fig. 3.12 for examples of meaning coherence in the SNG with Discrimination Game.)

The formula for calculation remains the same for different types of coherence, except that the function LOOKUP(x, y) and the source set $(W_{a_1} \bigcap W_{a_2})$ will be different depending on the type of coherence to calculate.

Non-uniform meaning distributions It is also important to take into account here how the meanings are distributed over an experiment: some meanings may occur very frequently, while others only occur very seldomly. Weighing every meaning the same does not give a balanced view of the actual coherence involving meanings: a meaning that has been used only once during an entire experiment would be counted in the same way as a meaning that was used over and over again. Thus, meanings should be weighed according to their frequency of use. A good example of the influence of meaning distribution on the coherence is described in (Van Looveren, 2001a). Briefly, consulting figure A.4, it is clear that in the Talking Heads case, only about 50 out of 300 meanings account for 95% of the interactions, and it is those 50 meanings that have high coherence. The other meanings have low coherence, but since they are hardly ever used (many of them have been used only once) they should not contribute as much to the total coherence figure as the oft-used meanings.

Lexicon		Syntactic rules	Semantic rules
Word	Meaning		
w_1	m_1	$w_1 \rightarrow \text{syn}_1$	$m_1 \rightarrow \text{sem}_1$
w_2	m_2	$w_2 \rightarrow \text{syn}_2$	$m_2 \rightarrow \text{sem}_2$
w_3	m_3	$w_3 \rightarrow \text{syn}_3$	$m_3 \rightarrow \text{sem}_3$

Table A.1: Example lexicon and grammatical rules (shared by the agents)

con-rules (agent 1)	con-rules (agent 2)
$\text{syn}_1 + \text{syn}_2 \rightarrow \text{sem}_1 + \text{sem}_2$	$\text{syn}_1 + \text{syn}_2 \rightarrow \text{sem}_1 + \text{sem}_2$
$\text{syn}_2 + \text{syn}_3 \rightarrow \text{sem}_2 + \text{sem}_3$	$\text{syn}_1 + \text{syn}_3 \rightarrow \text{sem}_1 + \text{sem}_3$

Table A.2: Example con-rules

A.2.2 Grammatical Coherence

The goal of grammatical coherence is similar to the goal of lexical coherence: provide a measure that indicates the similarity of the grammatical rules of two agents, and by extension the grammatical rules of a larger population of agents. The measure for grammatical coherence used in this thesis is ad-hoc, inspired on the measures for lexical coherence, and tailored to the fact that the grammar uses con-rules which always consist of two elements.

In measuring grammatical coherence, we encounter a problem similar to the introduction of meaning in the Simple Naming Game, where decoupling the referents and meanings introduces an indirection that is not one-to-one. In the case of our grammar system, due to the use of syntactic and semantic categories, it is not possible to simply compare con-rules (the principal components of the grammar) across agents, because the categories are internal to an agent, and may have different contents.

Therefore, the approach taken is more extensional: for every word in the lexicon, it is determined which words are allowed to appear to the right of it by finding out its syntactic categories, by finding corresponding con-rules in which that category appears on the left-hand side, and going back via syn-rules and sem-rules to determine which words are allowed.

The result of this is a set of words, with a set of permitted words for each word. These clusters of permitted word pairs can be compared across agents and in whole populations, in almost the same way as for lexical coherence.

Example Suppose an agent has the lexicon and semantic and syntactic rules in table A.1. For agent 1 (with the con-rules on the left in table A.2), the clusters are $w_1\,\{w_2\}$ and $w_2\,\{w_3\}$. For agent 2 (con-rules in table A.2 on the right), the cluster is $w_1\,\{w_2, w_3\}$. The words to which no con-rules (indirectly) apply are omitted from the clusters.

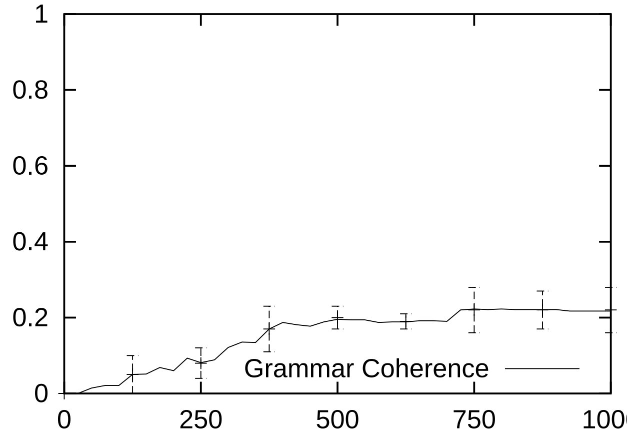


Figure A.5: Grammar coherence in a 5-agent experiment.

Both agents have the (partial) utterance w_1w_2 in common. Agent 1 can also produce w_2w_3 and agent 2 can produce w_1w_3. So out of three possible partial utterances, only 1 is shared. Therefore, the grammatical coherence between these hypothetical agents is $1/3$.

Figure A.5 shows the grammar coherence measure applied to a population of 5 agents for 1000 games. The graph shows that coherence as measured in this way is low, but this may be due to the simple nature of the measure. As the measure compares the set of *all possible* grammatical combinations of all agents, this disregards the grammatical combinations that are actually used in the games. As seen e.g. for coherence in the previous section and in chapter 3 with regard to meaning in the Talking Heads experiment, this can make a big difference in the result measured.

Selection Pressure: Experiments

This appendix shows the graphs of the experiments with selection pressures described in section 5.4.

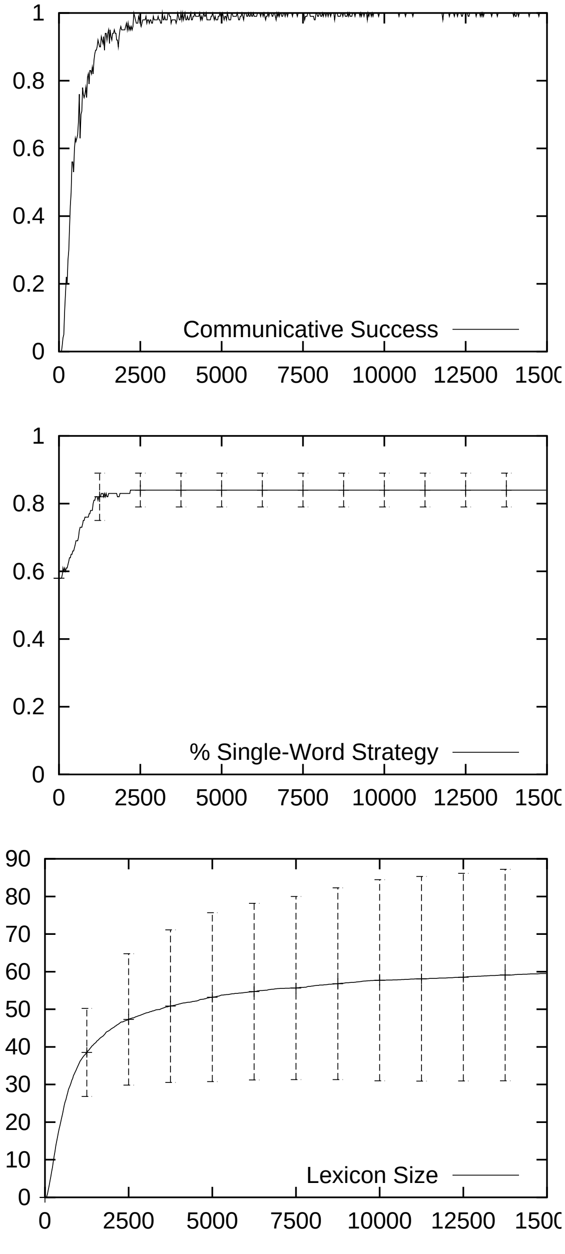


Figure B.1: Game Result—Absolute Calculation.

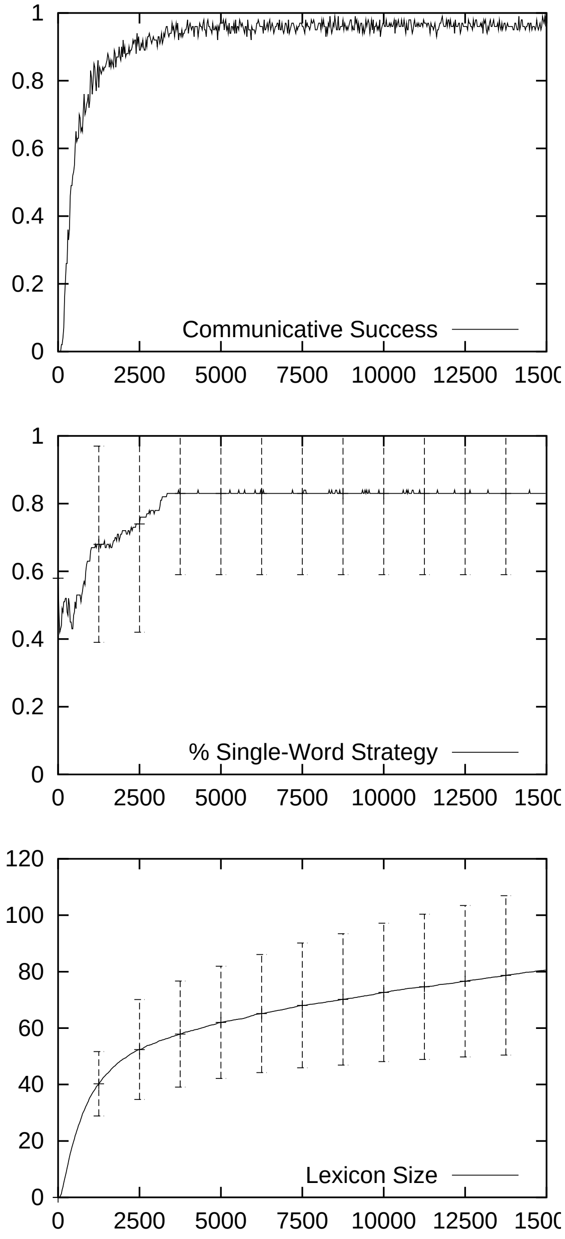


Figure B.2: Game Result—Positive and Negative Feedback.

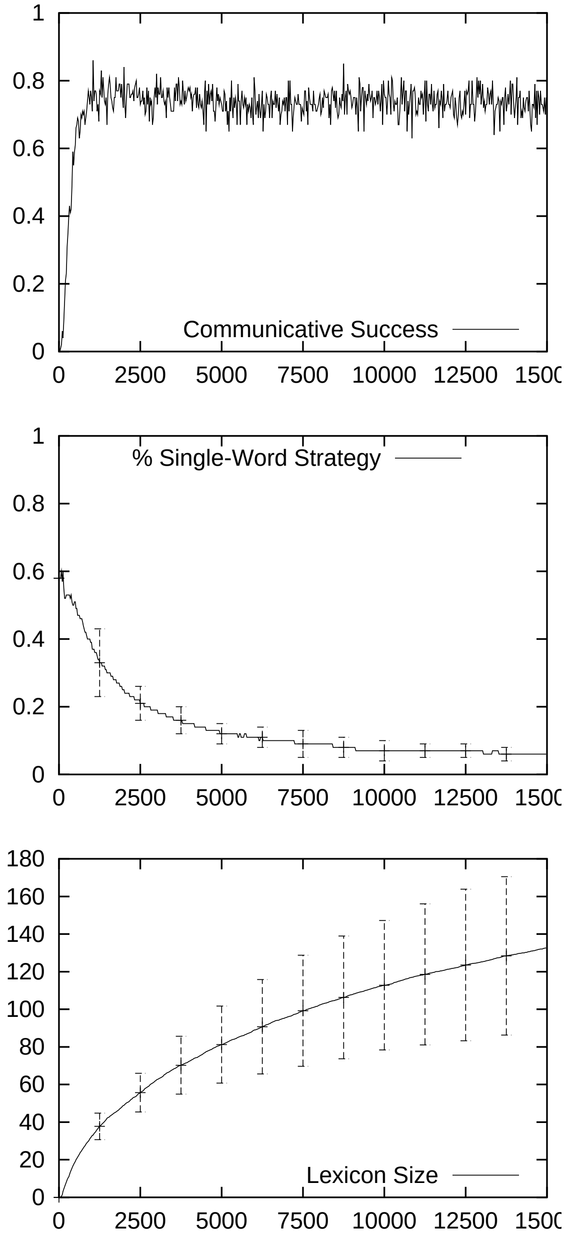


Figure B.3: Lexicon Size—Absolute Calculation.

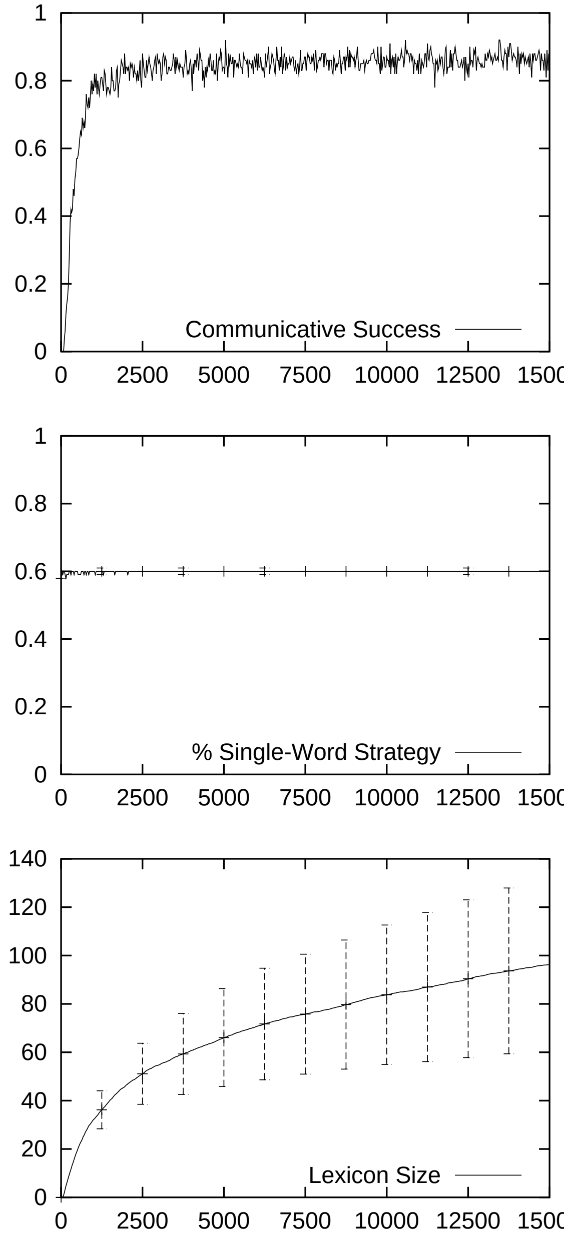


Figure B.4: Lexicon Expansion—Positive and Negative Feedback.

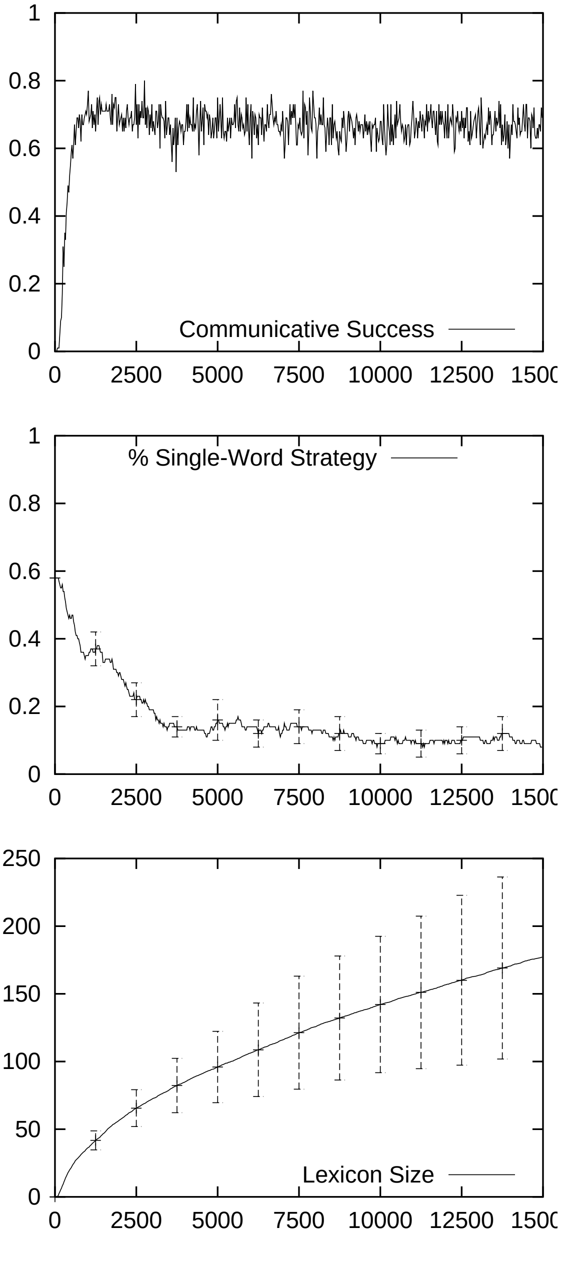


Figure B.5: Lexicon Expansion—Negative Feedback Only.

Bibliography

Avesani, P. and Agostini, A. (2003). A peer-to-peer advertising game. In Orlowksa, M., Papazoglou, M., Weerawarana, S., and Yang, J., editors, *First International Conference on Service Oriented Computing (ICSOC-03)*, Lecture Notes in Computer Science 2910, pages 28–42. Springer-Verlag, Berlin.

Axelrod, R. (1984). *The Evolution of Cooperation*. Basic Books, New York.

Batali, J. (1999). Computational simulations of the emergence of grammar. In Hurford, J. and Studdert-Kennedy, M., editors, *Approaches to the evolution of language: social and cognitive bases*. Cambridge University Press, Cambridge.

Batali, J. (2002). The negotiation and acquisition of recursive grammars as a result of competition among exemplars. In Briscoe, T., editor, *Linguistic evolution through language acquisition: formal and computational models*. Cambridge University Press, Cambridge, UK.

Belpaeme, T. (2002). *Factors influencing the origins of colour categories*. PhD thesis, Vrije Universiteit Brussel, Artificial Intelligence Laboratory.

Belpaeme, T., Steels, L., and Van Looveren, J. (1998). The construction and acquisition of visual categories. In Birk, A. and Demiris, J., editors, *Proceedings of the 6th European Workshop on Learning Robots*, Lecture Notes on Artificial Intelligence, Berlin. Springer Verlag.

Belpaeme, T. and Van Looveren, J. (te verschijnen). Klare taal: wat kunnen computermodellen ons leren over taalevolutie. In Gontier, N. and Mondt, K., editors, *Dynamisch Inter(en -trans)disciplinair Taal Onderzoek—De Nieuwe Taalwetenschappen*. VUBPress, Brussel.

Bergen, B. K. and Chang, N. (submitted). Embodied construction grammar in simulation-based language understanding. In Östman, J.-O. and Fried, M., editors, *Construction Grammar(s): Cognitive and Cross-Language Dimensions*. John Benjamins.

Bickerton, D. (1990). *Language and Species*. University of Chicago Press, Chicago.

Bickerton, D. (1998). Catastrophic evolution: The case for a single step from protolanguage to full human language. In Hurford, J. R., Studdert-Kennedy, M., and C., K., editors, *Approaches to the Evolution of Language: Social and Cognitive Bases*. Cambridge University Press, Cambridge.

Brighton, H. (2002). Compositional syntax from cultural transmission. *Artificial Life*, 8(1).

Buchholz, S. and Daelemans, W. (2001). Complex answers: a case study using a www question answering system. *Natural Language Engineering*, 7(4):301–323.

Cangelosi, A. (2001). Evolution of communication and language: using signals, symbols and words. *IEEE Transactions in Evolutionary Computation*, 5:93–101.

Cangelosi, A. and Parisi, D. (1996). The emergence of language in an evolving population of neural networks. In *Proceedings of the 18th Conference of the Cognitive Science Society*. San Diego.

Chang, N. and Maia, T. (2001). Learning grammatical constructions. In *Proceedings of the 23rd Annual Meeting of the Cognitive Science Society, Edinburgh, UK*, pages 176–181.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. The MIT Press, Cambridge, MA.

Crumpton, J. J. (1994). Evolution of two symbol signals by simulated organisms. Master's thesis, University of Tennessee, Knoxville.

Dale, R., Moisl, H., and Somers, H. (2000). *Handbook of Natural Language Processing*. Marcel Dekker, New York.

De Beule, J. (2004a). Creating temporal categories for an ontology of time. In Verbrugge, R., Taatgen, N., and Schomaker, L., editors, *Proceedings of the 16th Belgian-Dutch Conference on Artificial Intelligence*, pages 107–114. Groningen, the Netherlands.

De Beule, J. (2004b). Processes and the process engine. Technical report, AI-lab, Vrije Universiteit Brussel. AI-memo 04-06.

de Boer, B. (1997). Self organisation in vowel systems through imitation. In Coleman, J., editor, *Computational Phonology, Third Meeting of the ACL SIGPHON*, pages 19–25.

de Boer, B. (2001). *The origins of vowel systems*. Oxford University Press, Oxford, UK.

de Jong, E. and Vogt, P. (1998). How should a robot discriminate between objects? In *Proceedings of the Fifth International Conference of the Society for Adaptive Behavior (SAB'98)*, Cambridge, MA. The MIT Press.

de Jong, E. D. (1998). The development of a lexicon based on behavior. In La Poutré, H. and Van den Herik, J., editors, *Proceedings of the Tenth Netherlands/Belgium Conference on Artificial Intelligence (BNAIC'98)*, pages 27–36. CWI, Amsterdam.

de Jong, E. D. (2000). *Autonomous Formation of Concepts and Communication*. PhD thesis, AI-Lab, Vrije Universiteit Brussel.

de Jong, E. D. and Steels, L. (2003). A distributed learning algorithm for communication development. *Complex Systems*, 14:315–334.

Dominey, P. F. (2000). Conceptual grounding in simulation studies of language acquisition. *Evolution of Communication*, 4(1):57–85.

Dunbar, R. (1998). *Grooming, Gossip, and the Evolution of Language*. Harvard University Press.

Eldredge, N. and Gould, S. J. (1972). Punctuated equilibria: an alternative to phyletic gradualism. In Schopf, T., editor, *Models in Paleobiology*, pages 82–115. Freeman, Cooper and Co., San Francisco.

Feldman, J., Lakoff, G., Bailey, D., Narayanan, S., Regier, T., and Stolcke, A. (1996). L_0: The first five years. Technical report, International Computer Science Institute, University of California, Berkeley.

Goldberg, A. E. (1995). *Constructions: a Construction Grammar Approach to Argument Structure*. The University of Chicago Press, Chicago.

Goss, S., Aron, S., Deneubourg, J., and Pasteels, J. (1989). Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76:579–581.

Harnad, S. (1990). The symbol grounding problem. *Physica D*, 42:335–346.

Hashimoto, T. and Ikegami, T. (1996). The emergence of a net-grammar in communicating agents. *BioSystems*, 38:1–14.

Hawkins, J. A. (1978). *Definiteness and indefiniteness: a study in reference and grammaticality prediction*. Croom Helm, London, UK.

Hurford, J. R. (1989). Biological evolution of the Saussurean sign as a component of the language acquisition device. *Lingua*, 77(2):187–222.

Hurvich, L. M. and Jameson, D. (1957). An opponent-process theory of color vision. *Psychol Rev*, 64:384–404.

Jackendoff, R. (2002). *Foundations of Language: Brain, Meaning, Grammar, Evolution*. Oxford University Press.

Kaplan, F. (2000). *L'émergence d'un lexique dans une population d'agents autonomes*. PhD thesis, Université Paris 6, Sony CSL-Paris.

Kirby, S. (1998). Language evolution without natural selection: From vocabulary to syntax in a population of learners. Technical report, Language Evolution and Computation Research Unit, University of Edinburgh.

Kirby, S. (1999). *Function, selection and innateness: the emergence of language universals*. Oxford University Press, Oxford.

Kirby, S. (2002). Learning, bottlenecks and the evolution of recursive syntax. In Briscoe, T., editor, *Linguistic evolution through language acquisition: formal and computational models*. Cambridge University Press.

Kirby, S. and Hurford, J. (2002). The emergence of linguistic structure: An overview of the iterated learning model. In Cangelosi, A. and Parisi, D., editors, *Simulating the Evolution of Language*, chapter 6, pages 121–148. Springer Verlag, London.

Lehmann, W. P., editor (1967). *A Reader in Nineteenth Century Historical Indo-European Linguistics*. Indiana University Press.

Liljencrants, J. and Lindblom, B. (1972). Numerical simulations of vowel quality systems: the role of perceptual contrast. *Language*, 48:839–862.

Lyons, C. (1999). *Definiteness*. Cambridge University Press, UK.

MacLennan, B. (1990). Evolution of communication in a population of simple machines. Technical Report CS-90-99, University of Tennessee, Knoxville, Department of Computer Science.

Maynard-Smith, J. and Szathmáry, E. (1995). *The major transitions in evolution*. Morgan-Freeman.

McIntyre, A. (1998). Babel: A testbed for research in origins of language. In *Proceedings of Coling-ACL '98*. Montréal, Canada.

Mufwene, S. S. (2002). Competition and selection in language evolution. *Selection*, 3(1):45–56.

Neubauer, N. (2002). Emergence in a multi-agent simulation of communicative behaviour. Bachelor's Thesis, University of Osnabrück.

Ogden, C. K. and Richards, I. A. (1969). *The meaning of meaning: a study of the influence of language upon thought and of the science of symbolism*. Routledge and Kegan, London, tenth edition.

Oudeyer, P.-Y. (2003). *L'auto-organisation de la parole*. PhD thesis, University Paris VI.

Quinn, M. (2001). Evolving communication without dedicated communication channels. In Kelemen, J. and Sosik, P., editors, *Advances in Artificial Life: Sixth European Conference on Artificial Life: ECAL2001*, pages 357–366, Prague, Czech Republic. Springer.

Ruhlen, M. (1996). *The Origin of Language: Tracing the Evolution of the Mother Tongue*. Wiley.

Sakas, W. and Fodor, J. (2001). The structural triggers learner. In Bertolo, S., editor, *Parametric Linguistics and Learnability: A Self-contained Tutorial for Linguists*. Cambridge University Press, Cambridge, UK.

Savage-Rumbaugh, S., Shanker, S., and Taylor, T. (2001). *Apes, Language and the Human Mind*. Oxford University Press.

Schank, R. (1984). *The Cognitive Computer*. Addison-Wesley Publishing Co. Inc.

Smith, K. (2002). The cultural evolution of communication in a population of neural networks. *Connection Science*, 14(1):65–84.

Sonka, M., Hlavac, V., and Boyle, R. (1996). *Image Processing, Analysis and Machine Vision*. International Thomson Computer Press.

Steels, L. (1996a). Emergent adaptive lexicons. In Maes, P., editor, *Proceedings of the Simulation of Adaptive Behaviour Conference*. The MIT Press, Cambridge, Ma.

Steels, L. (1996b). Perceptually grounded meaning creation. In Tokoro, M., editor, *Proceedings of the International Conference on Multiagent Systems (ICMAS-96)*, pages 338–344, Menlo Park, CA. AAAI Press.

Steels, L. (1996c). Self-organizing vocabularies. In Langton, C., editor, *Proceedings of the Conference on Artificial Life V (Alife V) (Nara, Japan)*.

Steels, L. (1997). The synthetic modeling of language origins. *Evolution of Communication*, 1(1):1–34.

Steels, L. (1998a). Synthesising the origins of language and meaning. In Hurford, J., Studdert-Kennedy, M., and Knight, C., editors, *Approaches to the evolution of language*. Cambridge University Press.

Steels, L. (1998b). Synthesising the origins of language and meaning using co-evolution, self-organisation and level formation. In Hurford, J. R., Studdert-Kennedy, M., and Knight, C., editors, *Approaches to the Evolution of Language: social and cognitive bases*. Cambridge University Press, Cambridge, UK.

Steels, L. (1999). The talking heads experiment. Available from the VUB Artificial Intelligence Laboratory, Brussels, Belgium.

Steels, L. (2000). Waarom evolueert taal? In Gillis, S., Nuyts, J., and Taeldeman, J., editors, *Met Taal om de Tuin Geleid. Opstellen voor Georges De Schutter*, pages 297–310. Universitaire Instelling Antwerpen.

Steels, L. (2002). Iterated learning versus language games. two models for cultural language evolution. In Hemelrijk, C., editor, *Proceedings of the International Workshop of Self-Organization and Evolution of Social Behaviour*. University of Zurich, Switzerland.

Steels, L. (2004). Constructivist development of grounded construction grammars. In *Proceedings of the 42nd Assoc. for Comp. Linguistics Conference*. Barcelona.

Steels, L. and Belpaeme, T. (submitted). Computational simulations of colour categorisation and colour naming. *Behavioral and Brain Sciences*.

Steels, L. and Kaplan, F. (1998). Stochasticity as a source of innovation in language games. In Adami, G., Belew, R., Kitano, H., and Taylor, C., editors, *Proceedings of the Conference on Artificial Life VI (Alife VI) (Los Angeles, California)*, Cambridge, MA. The MIT Press.

Steels, L. and Kaplan, F. (1999). Collective learning and semiotic dynamics. In *Advances in Artificial Life: Proceedings of the 5th European Conference on Artificial Life (ECAL'99)*, volume 1674 of *Lecture Notes in Computer Science*, pages 679–688. Springer-Verlag, Berlin.

Steels, L. and Kaplan, F. (2002). AIBO's first words: The social learning of language and meaning. *Evolution of Communication*, 4(1):3–32.

Steels, L., Kaplan, F., McIntyre, A., and Van Looveren, J. (2002). Crucial factors in the origins of word-meaning. In Wray, A., editor, *The Transition to Language*. Oxford University Press, Oxford, UK.

Steels, L. and Vogt, P. (1997). Grounding adaptive language games in robotic agents. In Husbands, P. and Harvey, I., editors, *Proceedings of the Fourth European Conference on Artificial Life (ECAL'97), Complex Adaptive Systems*, Cambridge, MA. The MIT Press.

Stolcke, A. (1994). *Bayesian Learning of Probabilistic Language Models*. PhD thesis, University of California, Berkeley.

Terrace, H. S. (1987). *Nim: a Chimpanzee Who Learned Sign Language*. Columbia University Press.

Van Looveren, J. (1999). Multiple word naming games. In Postma, E. and Gyssens, M., editors, *Proceedings of the 11th Belgium-Netherlands Conference on Artificial Intelligence*. Universiteit Maastricht, Maastricht, the Netherlands.

Van Looveren, J. (2000). An analysis of multiple-word naming games. In Van den Bosch, A. and Wiegand, H., editors, *Proceedings of the 12th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC00), Kaatsheuvel, the Netherlands*.

Van Looveren, J. (2001a). Robotic experiments on the emergence of a lexicon. In Kröse, B. e. a., editor, *Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC01), Amsterdam, the Netherlands*.

Van Looveren, J. (2001b). Self-organisation of a lexicon in embodied agents. In *Proceedings of the Workshop on Developmental Embodied Cognition (DECO-2001)*. Edinburgh, UK.

Van Looveren, J. (2002). Technical report: Semantic engine. Technical report, AI-lab, Vrije Universiteit Brussel. AI-memo 02-01.

Van Looveren, J. (2003). Artificial agents and natural determiners. In Banzhaf, W., Christaller, T., Dittrich, P., Kim, J. T., and Ziegler, J., editors, *Advances in Artificial Life (Proc. of ECAL 2003)*, pages 472–481. Springer-Verlag, Germany.

Vogt, P. (2000). *Lexicon grounding on mobile robots*. PhD thesis, Artificial Intelligence Lab, Vrije Universiteit Brussel, Belgium.

Vogt, P. (2002). The physical symbol grounding problem. *Cognitive Systems Research*, 3(3):429–457.

Vogt, P. (submitted). The emergence of compositional structures in perceptually grounded language games. *Artificial Intelligence*. ISSN 0004-3702.

Vogt, P. and Coumans, H. (2003). Investigating social interaction strategies for bootstrapping lexicon development. *Journal of Artificial Societies and Social Simulation*, 6(1).

Vogt, P., de Boer, B., and Van Looveren, J. (2000). Luisterende robots en het ontstaan van taal. *Natuur & Techniek*, februari 2000.

Werner, G. and Dyer, M. (1991). Evolution of communication in artificial organisms. In Langton, C., Taylor, C., and Farmer, J., editors, *Artificial Life II, Vol. X of SFI Studies in the Sciences of Complexity*. Addison-Wesley Pub. Co., Redwood City, Ca.

Winograd, T. (1976). *Understanding Natural Language*. Academic Press.

Woods, W. A. (1968). Procedural semantics for a question-answering machine. In *AFIPS Conference Proceedings, Fall Joint Computer Conference, Montuole, NJ*, pages 457–471.

Yanco, H. and Stein, L. (1993). An adaptive communication protocol for cooperating mobile robots. In Meyer, J., Roitblat, H., and Wilson, S., editors, *From Animals to Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pages 478–485, Cambridge, MA. The MIT Press.

Zuidema, W. and de Boer, B. (2003). How did we get from there to here in the evolution of language? *Behavioral and Brain Sciences*, 26(6).

Zuidema, W. and Westermann, G. (2003). Evolution of an optimal lexicon under constraints from embodiment. *Artificial Life*, 9(4):387–402.