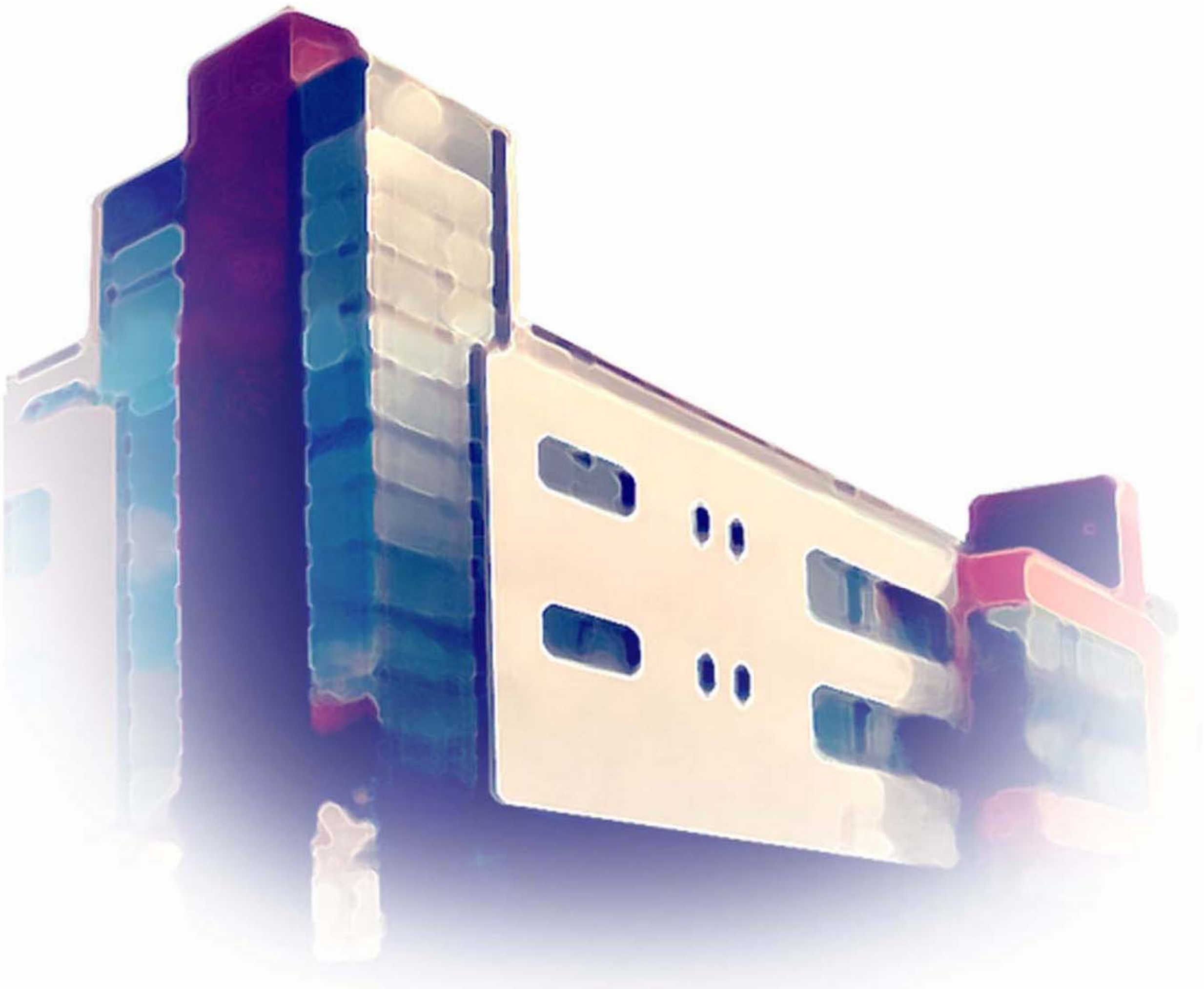
Nicolas Neubauer

Emergence in a Multiagent Simulation of Communicative Behaviour



PICS

Publications of the Institute of Cognitive Science

Volume 11-2004

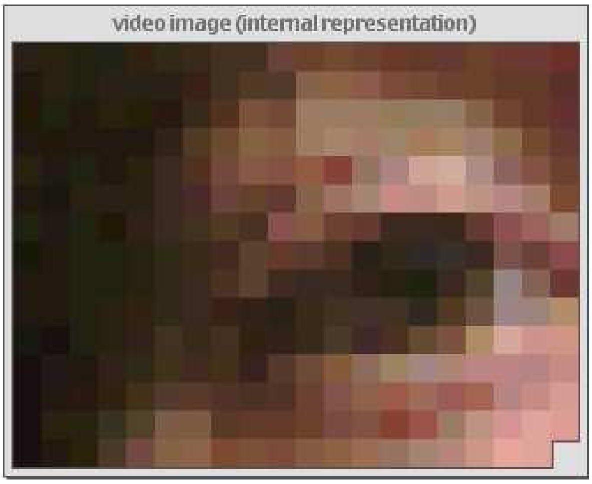
ISSN:	1610-5389
Series title:	PICS Publications of the Institute of Cognitive Science
Volume:	11-2004
Place of publication:	Osnabrück, Germany
Date:	November 2004
Editors:	Kai-Uwe Kühnberger Peter König Petra Ludewig
Cover design:	Thorsten Hinrichs

UNIVERSITÄT OSNABRÜCK

Emergence in a Multiagent Simulation of Communicative Behaviour

Nicolas Neubauer
nic@nicolasneubauer.net

October 5, 2002



Contents

1 Introduction

How does a group of individuals that do not initially share a language agree upon a set of signals with which to communicate? The mechanisms used by humans may be regarded of as highly influental on the form languages have taken on today.

This thesis describes a system simulating a population of agents repeatedly engaging in interactions called "language games", trying to maximize their success in communicating about real-word input. This system has been the software operating an installation by Olafur Eliasson and Luc Steels called "Look Into the Box!" that was exposed during Eliasson's exhibition "Chaque matin, je me sens different. Chaque soir, je me sens le même" in the Musée d'Art Moderne from March to Mai 2002.

The system shows how a shared and potentially compositional lexicon is reached in a population as a result of a series of interactions between two randomly chosen agents, without assuming previous, "innate" knowledge about the lexicon. This is also a demonstration of how a globally coherent behaviour may be achieved without central coordination or preprogrammed knowledge. That means that techniques from Artificial Life are used to approach questions from linguistics. A key issue in Artificial Life as well as in the concrete example is "emergence", the genuine novelty (either temporally or regarding the level of complexity something occurs the first time) of properties or behaviour. In the course of this thesis, this concept of emergence will be examined for how it can be applied on the simulation.

First, the history and the theoretical background that the system is embedded in will be outlined. It is following the "Language Game Approach" (LGA) whose goal is to model the origins of various aspects of language stressing the importance of functional pressures during actual communication. This implies that individuals adapt language norms during communication to increase their performance. This "interactionist" point of view will be set out against the opposing, "innatist" position stressing the importance of genetic rather than cultural evolution. Also, the difference between LGA and another interactionist model, the Iterated Learning Approach (ILA), will be examined.

Then, the exact design and implementation of the system will be explained.

In a next section, results of actually running of the system in different modes will be presented, showing how different linguistic phenomena can be created and examined with it.

The following section will introduce the concept of emergence in detail.

Finally, the observations made using the system and the definitions found for emergence will be merged to see whether certain phenomena observed in the system can be called emergent and in which way this might be of use.

2 Language Games

2.1 Evolution of language

The goal of the system described in this thesis is to demonstrate basic mechanisms that may be responsible for the creation and propagation of shared communication norms among numerous agents without central control. This is relevant for the investigation of the evolution of language:

- How can individuals come to share a common language without having a way to communicate in the first place?
- How do individuals that are born into a community of speakers already speaking a language learn this language?
- How can the occurrence of certain properties of languages be explained (for example, the shift from holistic to compositional language?), especially in relation to outside pressures?
- How do changes in language come into existence, and how are they propagated?

These are some of the questions that appear as "subtasks" when looking for an answer on how language may have come into existence at all and how it has come to be the way it is today. Building systems that allow a population of simulated linguistic agents to master the tasks mentioned above on a very limited scale may be a good step on the way to understanding the origins of real language.

The system examines phenomena on a certain complexity level: the invention and propagation of a lexicon, and the mechanisms and influences needed to shift from a holistic to compositional lexicon. One might go back to more fundamental questions like the sources of communication in the first place, or examine features further increasing complexity of language, like grammar. Neither of this is done in this thesis.

2.2 Artificial Life

The idea of approaching nature by trying to reconstruct it in computer simulations has recently become a discipline of its own called Artifical Life (AL) (this is the so-called "weak" definition of AL; there is a stronger position that actual life is being produced in this process – I will not go further into this point as it is not relevant for the problem at hand). See [Langton, 1995] for a broad overview of topics dealt with in this field. An important point of AL is the assumption that complexity in nature does not necessarily go back to complexity in individual behaviour; the interactions among the agents as well as their coping with the environment may be even more important factors, leading to what seems like a global behaviour of the whole system. The process by which such higher level behaviour comes into existence without central control is called *self-organisation*. Numerous examples from nature where complex phenomena like ant paths could be reproduced in a self-organising way can be found in [Camazine et al., 2001].

An interesting point in this context is that human reasoning seems to prefer explanations including a central control, even in the absence of any evidence for it. [Resnick, 1995] claims this as a result of his teaching experience and explains why he sees that as a pedagogic problem; he also invented a program called STARLOGO that allows to create AL simulations without lots of programming expertise to bring concepts of self-organisation to students early.

It is always the question how much of what appears to be system-level behaviour or properties really *is* a system-level behaviour or property. Maybe the properties could have been deduced from the individuals' properties? For example, whether the behaviour of chemical substances is a system-level property or whether it can be expected from or explained by the atoms' properties heavily depends on the state of physical knowledge we have about the atoms. So what seemed to be a system property before may suddenly appear as an individual property as soon as more is known about the consequences of the individual's properties. But maybe there are properties that cannot be reduced for principle reasons? This concept of novelty that is tried to be defined here is called "emergence". That is a term used in a wide range of meanings, but still, what seems to be common to all its connotations is that it is used to describe something coming into existence, often, but not necessarily with a flavour of novelty, unexpectedness or irreducibility. So it is tightly related to the way AL research tends to look at things (as [Resnick, 1995] cites Christopher Langton: "The 'key' concept in AL is emergent behaviour"). Towards the end of this thesis, various definitions of emergence will be tested for their applicability to the concrete system described here.

Anyway, the way global coherence is supposed to be reached by local interactions in the system described here make it an instance of an AL simulation; there are simulated agents dealing with an environment (that is, talking about it) and performing iterated interactions using only their own knowledge and their perception; these interactions are called Language Games.

2.3 The Language Game Approach

Language games are formalized social interactions suitable to be implemented in a computer program to simulate language learning and construction ([Steels, 1999]).

Each agent tries to optimise his personal communicative success by adapting his personal language using information gained through his game.

Note that the term "language" is to be taken with a grain of salt when refering to the agents' behaviour; one might argue whether a set of word/meaning associations actually qualifies for being called a language. For this reason, "communicative behaviour" was preferred over "linguistic behaviour" in the title; in the same manner, "communicative system" should be prefered over "language". "Language" is used liberally, however, throughout the whole text, for reasons of readability and convenience. This does not imply a strong claim that the system really implements a language according to any more demanding definition of language requiring, for example, grammar; one might as well read all related occurences of "language" in quotes.

The History of the Language Games

There are various types of language games: First, there were games to simulate the creation of lexical coherence. Then, the coevolution of language and meanings was examined: Each agent would first create the concepts, then name them. So the agents would have to cope with other agents who do not only use different words, but also perceive reality differently than themselves ([Steels, 1997]). Currently, efforts are made to create systems with agents developing grammar.

Parallely, it has been tried to get systems to cope with reality; so games have been played by real robotic agents coping with inputs coming from sensors sensing the real world ([Steels and Vogt, 1997]). Like this, the mechanisms used are proven to be resistent to noise; in addition, there is a structure in real world data (real shapes, real colours and so on) that may influence the language creation process. This is important as a source of complexity and structure in the language created, because as mentioned above as one of the assumptions of AL, complexity in behaviour may well result from simple algorithms dealing with a complex environment.

Simulated deaths and births to examine the impact of new language learners on the language system, as well as spatial models of the agents' world ([Steels and McIntyre, 1999]) to simulate several language communities[1] have also been tried out. Other domains of language development have been studied as well, for example the origins of vowel systems ([Oudeyer, 2001]).

The most complex system following the Language Game Approach (LGA) so far is the Talking Heads system ([Steels, 1999], [Kaplan, 2001]), where agents play a language game coevolving their concepts and their language, while moving to different geographic sites via internet and using real hardware.

The system described focusses on lexical properties, although there also is a currently disabled function that allows the agents to build their own conceptualisations.

[1] [Dixon, 1997] gives an overview about the long-term effects of geographical/political changes on real languages

Nature vs. Nurture

It should have become clear that the LGA claims that language is being transmitted culturally. This is not so clear in general. There are two opposing views on how language is transmitted: the innatist view, claiming that most of human language capabilities are determined genetically, and the interactionist view claiming the opposite, that language is learnt by social interaction, transmitted culturally, without innate knowledge. Since the agents start without any knowledge of language and learn everything by communication with others, the LGA takes a clear interactionist point of view.

Since, for the innatist, language is determined by birth, functional issues occuring during the use of the language should not have any influence on the form language takes. Grammar is seen as an arbitrary construct ("autonomy of syntax"). The LGA disagrees on this point as well; the agents have the pressure to optimise their communicative success and adapt their language in order to do so. In linguistics, new approaches like cognitive or functional grammar have been pursued ([Tomasello, 1998]) to account for the social and cognitive aspects of grammar as disregarded by the autonomy-of-syntax paradigm.

It may seem odd to state that these things happen "without innate knowledge" when looking at the machinery used "inside" the agents that is described below. But what's being denied is not the innateness of everything, but the innateness of specialised language capabilities and that language structure is genetically determined as supposed by the assumption of an innate language aquisition device:

> "[Cognitive linguistics] contrasts with formalist approaches by viewing language as an integral facet of cognition (not as a separate "module" or "mental faculty")" – [Langacker, 1998, 1]

Although the functions the agents use are in fact specialized in the sense that the agents do nothing but language processing, the underlying mechanisms are nevertheless very general purpose ones like associative learning, lateral inhibition, pattern recognition, so that despite the specialized programming of the agents, one could claim that no mechanisms are used that would not also have to be implemented in an agent with other or more cognitive abilities.[2]. Still, it is true that the general form of the grammar produced does depend on the programming on the agents (the "innate" functionality). One might have to admit that the difference between interactionist and innatist points of view is not as binary as it seems at first, but rather a gradual one; it is on *how much* of language is predetermined, *how many* functions are hardwired in the brain in order to get humans to talk. The differences, however, are so strong that, as mentioned above, there are completely different theories on grammar depending on which point is taken because different aspects of language are regarded as important.

Other AL approaches to evolution of language

The LGA is not the only attempt to approach evolution of language using artificial life simulations. For example, the iterated learning approach (ILA, [Kirby, 2002]) pursued by Simon Kirby, Henry Brighton and others, tries to explain certain features of language (mostly compositionality, e.g. [Brighton, 2002]) by focussing on the "learning bottleneck" that takes place when a new generation of individuals has to learn the language of their "parents" with limited examples. A crucial difference between LGA and ILA is that in the ILA, language is again not seen as serving any function but as something passive which has to be transmitted. So compositionality is supposed to arise out of transmission problems and not because of communicative needs. While this may be true, the decision not to actually *use* the language seems like ignoring important parts of the problem (and, in fact, fails to explain phenomena like emerging coherence). See [Steels, 2002] for a detailed comparison between ILA and LGA.

The Guessing Game

Now, two different types of language games will be introduced: The "Guessing Game" as an example

[2] This would be nicely underlined if it was possible to show how the learning strategies can be simulated by simple neural networks. I am pretty sure this would work but it would probably take some time and is somehow off the point in this paper

for a very complex game, and the "Naming Game", the one that is played in the system described.

A language game that seems very close to everyday scenes is the "Guessing Game" as used in the Talking Heads project. Two agents observe a scene (each from its own visual perspective), then one chooses an object and tries to communicate this object to the second agent, the hearer. The hearer hears the speaker's utterance and tries to find out which object could be intended by the speaker, given the shared context.

The game is successful if the speaker correctly guesses the choice of the hearer. To check this without having to use language, the agents can "point" to their intended choice.

This type of interaction resembles verbal interactions we perform in our everyday life. When we order vegetables in a shop, we pick an object, express our wish, taking the part of the speaker. The communicative success can be seen without further language needed as we either get what we want, or something different.

Note that there are three different processes:

conceptualisation abstracting from the picture the properties of the objects ("red", "green", "pepper", "cucumber") - this is a very complex part in itself in the Talking Heads experiment that is beyond the current context.

discrimination choosing which properties to express in order to distinguish the chosen object from the others - "red peppers" if there are non-red peppers (otherwise we could just say "the peppers") as well as non-pepper red objects (otherwise we might just say "the red ones" – which would not appear very natural for other reasons but do the job of discrimination).

verbalization choosing words to express the chosen properties.

The hearer has to do about the same, partially reversed: he too has to conceptualise the objects' properties. Then he has to interpret the words he hears back to properties and check for which object these properties might apply.

Learning takes place as both hearer and speaker adapt the rules they used in the game positively or negatively, depending on the success of the game.

The Naming Game

For the system described, a much simpler language game is played. The payoff for the simplicity is that the parallels to everyday behaviour are not as obvious as in the guessing game. Nevertheless, important processes of language use are covered and, looking hard, it is in fact possible to find this kind of interaction in real life.

The meaning (colour values taken from pictures) is known to both the hearer and the speaker from the beginning. The speaker then verbalises this meaning, and the hearer tries to find a set of rules that allow to map the meaning he recognizes onto the words he perceives from the speaker.

Communicative success is then the degree to which the hearer agrees with the meaning-to-word mapping performed by the speaker. There is no intended reaction by the speaker, he just repeats the words, either with an asking intonation (if they were new, generally or in this context) or an asserting one (if there was a sensible interpretation).

Coming back to the vegetables, an example of this kind of game being played would include the vendor and a, perhaps foreign, client who have agreed on a vegetable type, say, by pointing. They now have agreed on the meaning, and the vendor, if interested in his client's linguistic capabilities, might pronounce "cucumber"[3]. The client would then either remember an association between his representation of a cucumber and the word "cucumber", or create a new association, if he never heard of it before (maybe saying, respectively "(oh yes, right, that's a) cucumber!" or "(so you call that) cucumber?").

Learning takes place in the speaker as the rules that were used to create the chosen utterance are promoted, while rules that were used to create potential alternative, unused utterances are inhibited In the hearer, the rules contributing to the chosen interpretation are rewarded as well. But there are now two different kinds of "losing" interpretations: first, those ones that were correct but just not chosen. These interpretations' scores are decreased just like the concurrent verbalizations in the speaker (lateral inhibition). Beyond that, there are those interpretations that actually predicted a wrong meaning. That means that rules existed that

[3]This has actually happened to me while writing this thesis in France.

mapped the words onto meanings that weren't existent in reality. These rules are additionally diminished.

Looking at the difference between the guessing and the naming game, it turns out what is missing or impoverished in the naming game is basically those parts that go beyond the actual lexicon generation and learning algorithm, to the more *pragmatic* aspects of language use (namely conceptualization and discrimination). While this takes away a bit the striking effect of seeing two machines engaging in real communication, what stays is an environment in which to observe the dynamics of an emerging lexicon.

3 Implementation

The program parts will be described in order of increasing complexity: First, I will introduce the matching and unification toolkit fundamental to the agent's linguistic capabilities. Second, I will introduce the agents themselves by describing what each one does during one round (although one "round" means one language game, I use the word "round" to keep the term "game" to describe all the rounds played with a given population). This is the main part, but the agents themselves are hosted by the game environment which provides them with input, chooses speaker and hearer, keeps track of global statistics, interfaces the hardware and handles some other administrative issues. This may be thought of as the environment the agents "live" in. Whether pictures are grabbed from the camera or load from disc, whether statistical data is simply written to disc as a list, transferred to a remote web server or kept in a file with database commands for later execution as well as how many agents exist is decided by this game environment. Finally, there is a website which displays the pictures and the data produced by the installation.

The system was, except for the website programming, written in ANSI Common LISP as described in [Steele, 1990], on a Macintosh platform using Mac Common LISP. The high level program code was completely written by me, while I was lucky enough to be able to acccess several libraries developed by Angus McIntyre mainly for interfacing issues like camera, network and graphics access.

3.1 The Matching and Unification Toolkit

A fundamental part of the agents' program is a matching and unification toolkit (MAUT). It defines a format for linguistic structures and rules to manipulate these structures, and provides functions to actually perform the manipulations, thus implementing the agents' basic language representation and processing capabilities.

The task of keeping structured linguistic information and performing matching or merging operations on those structures is not a new one. The PATR-II formalism (refer to [Shieber, 1986]), for a detailed description) describes a commonly used way to represent those structures and introduces the unification operator which returns what could be called the "union" of the information contained in two argument structures if possible (that is, if the information is not contradictory).

MAUT is inspired by the idea of PATR-II, borrowing the two key concepts of feature structures and unification, but realising them differently. MAUT may seem a bit too complex for what it is doing inside the current system, and in fact it is. It was programmed during my internship at Sony CSL to be used for further experiments on the origins of language, namely the emergence of grammar. It is supposed to be used in ongoing development on the origins of case grammars whose rules are expressed in this format. By the time this thesis is being completed, the formalism has been and still is developed further and renamed Fluid Construction Grammar (FCG). So for the storage of rather simple meaning/word combinations, one could have thought of more lean engine. We used it anyway because the code was, although doing somewhat too much, well working for the task, and because it keeps the system open for more complex rules later on (besides that, it was a good opportunity to use the toolkit in a real program for the first time).

The MAUT grammar

The elements used in the MAUT are defined as follows:

```
<rule> :=
  <feature-structure>
  <--->
```

```
  <feature-structure>

<feature-structure> :=
  (<unit-1> ... <unit-n>)

<unit> :=
  (<unit-name>
    <feature-1> ... <feature-n>)

<unit-name> :=
  symbol | variable

<feature> :=
  (<attribute> <value>)

<attribute> :=
  referent | meaning | ...

<value> :=
  symbol | predicate | list |
  variable | ...

<predicate> :=
  ([symbol | variable] [symbol | variable] ...)
  ordered

<list> :=
  ([symbol | predicate | ...]
   [symbol | predicate | ...]
   ...)
  unordered
```

The structures as defined by this grammar are valid LISP lists and are in fact processed like this by the program code. For displaying, a more convenient representation will be used that resembles more the way feature structures as typcially used in linguistics are printed. The transition from LISP representation to the convenient one is shown by an example in Figure 1.

Feature structures are structures that can hold attribute/value pairs. The major difference between PATR-II and MAUT is that in PATR-II, a value can either be an atom or a pointer to another feature structure. So complexity is dealt with via recursion, with structures being nested arbitrarily deep and potentially cyclic. With MAUT, there is no recursion, which makes processing its structures a lot easier. Since feature structures cannot be recursive, all potential complexity must be handled

Figure 1: sample feature structure in LISP and in "feature struture-style" representation

```
((UNIT-1
   (FEATURE SYMBOL-1)
   (LIST-F
     (?VAR-1
      SYMBOL-2
      (PRED-1 ARG-1 ARG-2))))
 (UNIT-2
   (FEATURE ?VAR-1)))
```

$$\Updownarrow$$

$$\begin{bmatrix} \texttt{UNIT-1} & \begin{bmatrix} \texttt{FEATURE} & \texttt{SYMBOL-1} \\ \texttt{LIST-F} & \begin{pmatrix} \texttt{?VAR-1} & \\ \texttt{SYMBOL-2} & \\ \texttt{(PRED-1} & \texttt{ARG-1} \\ & \texttt{ARG-2)} \end{pmatrix} \end{bmatrix} \\ \texttt{UNIT-2} & \begin{bmatrix} \texttt{FEATURE} & \texttt{?VAR-1} \end{bmatrix} \end{bmatrix}$$

inside a single structure, which is the reason why the definition of a MAUT feature structure is a bit more complicated:

- feature structure do not directly contain attribute/value pairs but are organised into named substructures called units which then contain the pairs
- relations between units are not achieved by a direct link since recursion is not allowed, but by naming the unit:

$$\begin{bmatrix} \texttt{TOP-UNIT} & \begin{bmatrix} \texttt{SUBUNITS} & \begin{pmatrix} \texttt{UNIT-1} \end{pmatrix} \end{bmatrix} \\ \texttt{UNIT-1} & \begin{bmatrix} \texttt{FEATURE-1} & \texttt{VALUE-1} \end{bmatrix} \end{bmatrix}$$

- values may be more complex than just atoms and be composed of lists (an unordered set of values) or predicates (an ordered list of atoms)
- for matching lists, there is an additional operator == ("include") which can be used in the pattern to indicate that the following list mentions items that must be present in the source list, but that the source list may well contain other items beyond those

In the following, the two basic operations on structures are described, matching and unification. The third operation mentioned, rule application, is a combination of both.

Matching

Match(pattern, source, state) checks whether source satisfies the constraints posed by pattern and given the current state (see below). This means that for each unit in pattern, there must be a matching unit in source, and for two units to match, each feature in the pattern unit, a feature of the same name must be present in the source unit and the values must be compatible.

Obviously, this is a yes/no question - as long as there are no variables. Taking into account variables, the question must be reposed from "Does *source* satisfy *pattern*" to "For which variable bindings does *source* satisfy *pattern*?". So what match truly returns is not a boolean value, but a set of states describing valid variable bindings, and an emtpy set if source does not match at all.

Variable bindings may need to be kept, and so the result state of a match operation may be passed on to the next match or unification operation. For example, unit names in rules are typically left variable. When a unit is found that meets the criteria for the unit in the pattern, the variable is instantiated with that unit's name. By saving the variable bindings, further operations can be done on the same structure while the information which unit is meant by the variable is preserved. This is crucial for rule application.

Unification

Unify(pattern, source, state) checks whether *pattern* and *source* do not contain contradictory information and returns the union of the information contained in both. In comparison to *match*, where *source* had to contain everything *pattern* did *and* all the values had to be compatible, it is now only compatibility that is checked for and the result is the *source* that is augmented by any information it did not contain but *pattern* did.

This is basically what the unification operator in PATR-II does – check for congruence and construct the union of two structures. It turns out that having a feature in unification only reduces the number of possible structure it can be unifyable with; if two feature structures A and B are equal except for one additional feature in A, B will be unifyable with all the structures A is unifyable with; *plus* the ones that don't unify with A because they have a feature of the same name with a different value and are hence incongruent. This means that B is more general than A, or "B subsumes A".

So what's really different about MAUT compared to PATR-II on a functional level (that is, beyond representation or efficiency issues) is the way features are seen. In PATR-II they serve as mere constraints. B can "do" everything A can, and potentially more. But what if we want to *require* the existence of a feature such that a specialized A matches while a more general B does not? It seemed to me that such a kind of criterium, as it will be required in rule application, is impossible to pose in PATR-II without "hacking" the formalism. In MAUT, this is done by the *match* operator as described above.

Again, what *unify* returns is not a boolean value but a set of states describing valid variable bindings. I will not go into this a lot deeper as it is not required for the rest of this thesis.

Rule application

Having the two operators *match* and *unify* now, rules can be constructed and applied of the type

"if a structure $Source_{Match}$ satisfies constraints as given by a structure $Pattern_{Match}$ (the *if-part* of the rule),

then apply changes as described by a structure $Pattern_{Unify}$ (the *then-part* of the rule) to structure $Source_{Unify}$".

$Source_{Match}$ and $Source_{Unify}$ can be the same structure, but they do not have to be. In the current program, two different structures are kept, one with syntactic, one with semantic information. This seems more efficient because it keeps $Source_{Match}$ constant even when rules are applied so rules do not have to be checked several times with changed $Source_{Match}$ structures.

In such a setup, a rule stating that a unit whose goal is reference should be equipped with a marker 'er' on the syntax side, would be expressed as $Rule_1$ (see Figure 2).

Figure 2: $Rule_1$

$$\begin{bmatrix} \left[\ \texttt{?UNIT-1}\ \left[\ \texttt{GOAL}\ \ \texttt{REFERENCE}\ \right]\ \right] \\ \Updownarrow \\ \left[\ \texttt{?UNIT-1}\ \left[\ \texttt{MARKER}\ \ \texttt{==}\left(\ \texttt{ER}\ \right)\ \right]\ \right] \end{bmatrix}$$

Note how the variable *?Unit-1* is used in both structures, forcing that the change is applied to the same unit in which 'er' was found. Considering the application of $Rule_1$ to a sample semantic structure Sem_1 (see Figure 3), *match(if-part(*$Rule_1$*),* Sem_1*,[empty state])* would succeed, returning one possible set of variable bindings, $state_1$, where *?Unit-1* is bound to top-unit. Assuming that the syntactic counterpart for Sem_1, Syn_1, is still empty, *unify(then-part(*$Rule_1$*),* Syn_1*,*$state_1$*)* would return $state_1$ – since no variable bindings have been changed – and result in a structure Syn_2 as shown in Figure 4). Finally (this happens only when all other operations have been finished in order not to remove any information too early), the variables are replaced as determined by $state_1$, resulting in a final syntactic structure Syn_3 as shown in Figure 5, the result of the rule application.

Figure 3: Sem_1

$$\begin{bmatrix} \texttt{TOP-UNIT} & \begin{bmatrix} \texttt{GOAL} & \texttt{REFERENCE} \\ \texttt{REFERENT} & \texttt{?REFERENT-1} \end{bmatrix} \end{bmatrix}$$

Figure 4: Syn_2

$$\left[\ \texttt{?UNIT-1}\ \left[\ \texttt{MARKER}\ \ \left(\ \texttt{ER}\ \right)\ \right]\ \right]$$

An important point is that rules can be used in both directions. When creating an utterance for a given meaning, the rules are used in the way described above. When parsing a given meaning, if- and then-part are reversed.

Figure 5: Syn_3

$$\left[\ \texttt{TOP-UNIT}\ \left[\ \texttt{MARKER}\ \ \left(\ \texttt{ER}\ \right)\ \right]\ \right]$$

Simplifications

It turns out that feature structures to represent, for example, three colour vectors already look quite complex (see Figure 8 on page 12). A reason to keep this complexity is the expressive power of the rules that can be created by manipulating those structures. For example, one might formulate a rule for the case that the L components of all three colours contained have the same value, just by inserting one variable at each of the L value places.

Having shown this once (which is important to understand the inner workings of the system as well as the wide range of potential uses for the future), a much shorter description will now be introduced that will be used to talk about rules quickly without considering the real implementation; as mentioned above, the rules used in the current application are much simpler than what can be expressed. In fact, the only type of rules used maps components of a single colour onto a word (and vice-versa). This may be written very compactly as

```
((L x) (U y) (V z)) <---> WORD
```

This maps structures with a feature 'L' of value x, a 'U' feature of value y and a 'V' feature of value z, with x, y and z being integers, onto the word WORD[4].

Not each component has to be present in a rule:

```
((U 1)) <---> BA
```

is a complete rule. See Figure 6 and 7 for the implementation of those two exemplary rules.

Embedding MAUT in the agents

Both semantic and syntactic information is expressed in feature structures, and the whole linguistic knowledge of an agent lies in its set of rules (each augmented with a certainty score) about how to process such structures.

[4]see the section on conceptualisation (3.2) for the introduction of LUV

Figure 6: internal representation of rule `((L x) (U y) (V z)) <---> WORD`

$$\begin{bmatrix} \begin{bmatrix} \texttt{?UNIT-1} & \begin{bmatrix} \texttt{GOAL} & \texttt{REFERENCE} \\ \texttt{REFERENT} & \texttt{?REFERENT-1} \\ \texttt{MEANING} & \begin{pmatrix} \texttt{(U} & \texttt{X)} \\ \texttt{(V} & \texttt{Y)} \\ \texttt{(L} & \texttt{Z)} \end{pmatrix} \end{bmatrix} \end{bmatrix} \\ \Updownarrow \\ \begin{bmatrix} \texttt{?UNIT-1} & \begin{bmatrix} \texttt{FORM} & == \begin{pmatrix} \texttt{WORD} \end{pmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 7: internal representation of rule `((L 1)) <---> BA`

$$\begin{bmatrix} \begin{bmatrix} \texttt{?UNIT-1} & \begin{bmatrix} \texttt{GOAL} & \texttt{REFERENCE} \\ \texttt{REFERENT} & \texttt{?REFERENT-1} \\ \texttt{MEANING} & \begin{pmatrix} \texttt{(U} & \texttt{1)} \end{pmatrix} \end{bmatrix} \end{bmatrix} \\ \Updownarrow \\ \begin{bmatrix} \texttt{?UNIT-1} & \begin{bmatrix} \texttt{FORM} & == \begin{pmatrix} \texttt{BA} \end{pmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The basic processes of the agents' behaviour can then already be described at this level:

Verbalisation is the process of applying rules to a semantic structure in order to create a valid syntactic structure, while

Interpretation is the same in reverse direction.

Ambiguity in the rules shows up as multiple possible structures as the outcome of applying all known rules to the input structure.

Learning means adapting the rule set by creating new rules, adapting existing rules' scores, or pruning old ones.

Where input structures come from, where output structures go to, how possible rule applications are found and, most importantly, how the learning of the rules works in detail, will be managed by the coming program layers.

3.2 The Speaker

The speaker's task is to create an utterance describing a scence in the outside world. This involves three different tasks:

Perception Get input from the sensors.

Conceptualisation Transform the perceived input into a semantic structure.

Verbalisation Transform the semantic structure into a syntactic structure.

Note that there is a difference between what is called here the agent's task and what is actually in the agent's code. Perception and conceptualisation are important parts of a linguistic agent's capabilities, should as such be looked upon as parts of the simulated agents' behaviour and are therefore mentioned here, but in the program, perception and parts of the conceptualisation code are kept in the game environment for reasons of efficiency, as for the time being, these parts are non-adaptive and hence return the same results for both agents; by keeping the corresponding code outside the agents, the resulting information can just be fed on to both agents at the same time.

Perception

The input consists of a bitmap with 320x240 pixels that is grabbed from the camera or load from disc. A description of the eye recognition software will be given in the section dealing with the game environment (3.4).

Conceptualisation

Colours are initially represented as points in a three-dimensional space spanned by the red, green and blue (RGB) axis. These points are projected into LUV space, an alternative representation based on a "luminance" (perceived brightness) value L and two coulour coordinates U and V. This colour space comes closer to human colour perception: not only does the difference between a luminance and two color dimensions better model the human visual system – a property which is nice but not of immediate use for our system –, also transformation rules have been adjusted such that the euclidic distance between two points is proportional to the average *perceived* difference between the corresponding colors. This is not given in RGB space where, for example, a change on the green axis results in a small subjective colour change while the same change on the red axis is already well perceivable.

These points are normalized and discretized to be integer numbers from 0 to *grainsize* which is a global parameter and usually set to 8 or 16, resulting in 512 or 4096 different perceivable colors.

Now, a histogram of the colour distribution is created by counting which colour appears in how many pixels in the image. The point of using LUV space is that now, each of the "bins" (that is, each of the discrete colour values that are counted) appears equally different to its neighbours as any other bin and the agents' colour discrimination is likely to appear more realistic. After each pixel has been counted, the three colours that appear most frequently are picked. This results in three vectors out of $\{0..grainsize\}^3$ which are then transformed into a semantic structure as shown in Figure 8.

Figure 8: sample semantic structure produced by feature vector

$$\left\langle \begin{pmatrix} L_1 \\ U_1 \\ V_1 \end{pmatrix}, \begin{pmatrix} L_2 \\ U_2 \\ V_2 \end{pmatrix}, \begin{pmatrix} L_3 \\ U_3 \\ V_3 \end{pmatrix} \right\rangle$$

$$\Downarrow$$

$$\begin{bmatrix}
\texttt{?TOP-UNIT} & \begin{bmatrix} \texttt{SUBUNITS} & \begin{pmatrix} \texttt{TOP-1-COLOR} \\ \texttt{TOP-2-COLOR} \\ \texttt{TOP-3-COLOR} \end{pmatrix} \end{bmatrix} \\
\texttt{TOP-1-COLOR} & \begin{bmatrix} \texttt{GOAL} & \texttt{REFERENCE} \\ \texttt{REFERENT} & \texttt{TOP-1-COLOR} \\ \texttt{MEANING} & \begin{pmatrix} (\texttt{U} \quad U_1) \\ (\texttt{V} \quad V_1) \\ (\texttt{L} \quad L_1) \end{pmatrix} \end{bmatrix} \\
\texttt{TOP-2-COLOR} & \begin{bmatrix} \texttt{GOAL} & \texttt{REFERENCE} \\ \texttt{REFERENT} & \texttt{TOP-2-COLOR} \\ \texttt{MEANING} & \begin{pmatrix} (\texttt{U} \quad U_2) \\ (\texttt{V} \quad V_2) \\ (\texttt{L} \quad L_2) \end{pmatrix} \end{bmatrix} \\
\texttt{TOP-3-COLOR} & \begin{bmatrix} \texttt{GOAL} & \texttt{REFERENCE} \\ \texttt{REFERENT} & \texttt{TOP-3-COLOR} \\ \texttt{MEANING} & \begin{pmatrix} (\texttt{U} \quad U_3) \\ (\texttt{V} \quad V_3) \\ (\texttt{L} \quad L_3) \end{pmatrix} \end{bmatrix}
\end{bmatrix}$$

Verbalisation

Verbalisation is the major part for the language dynamics on the speaker's side. During this process, the agents' language is created, used and adapted.

The goal is to map the semantic structure – containing the colour values – that was created during the conceptualisation process onto a syntactic structure containing the words for these colour values.

Decomposal As shown in Figure 8, each color is kept as a subunit in the main structure, but as mentioned above (Section 3.1), the rules that are used do not operate on such complex structures with component colour units but only on structure containing a single colour unit.

So instead of dealing with one big structure, the program will map each of the three colour units in a seperate feature structure. Finally, the three resulting syntactic structures will be put back into one final syntactic structure that is structured like the semantic one. Each of the following steps but the final one ('Getting the words') describe the processing of those single colour vectors.

Using existing rules The semantic structure containing the single colour vector is matched against all the if-parts (which are the semantic parts during speaking) of all the rules in the agent's grammar. Each of the possible rule applications (called parses) gets a score based on

correctness – how well the structure satisfies the if-part, the ratio between the numbers of features in the if-part and the number of features that are matched by the structure,

coverage – how much of the information present in the structure is used by the if-part, the ratio between the number of features in the structure and the number of matching features in the if-part,

certainty – the rule's score

and

generalised – the number of rules that had to be generalized which will be further explained by the next step.

For example, matching the structure

$$S_1 : \texttt{((L 1)(U 3)(V 2))}$$

with the if-part (now the semantic, left part) of the rule

$$Rule_2 : \texttt{((L 1) (U 2)) <---> BA}$$

results in a correctness of .5 and a coverage of $\frac{1}{3}$.

Rules may be chained: If the coverage score of one parse is below 1, not all information in the original structure was verbalised, a rule handling less than all three components was used. In this case, all rules are tested again for this parse (unless it has a correctness score of 0, which would mean that the rule just did not match at all), checking if they can increase coverage of the whole parse, that is, if they handle components that were not covered by the first rule. This may happen again and again, but in practice, since there are only three components to be handled, there is a maximum of three rules to be used serially. When this is done,

correctness of all the single rule applications is multiplied,

coverage is added such that three rules covering each one of three components finally lead to full coverage,

certainty is again multiplied because, e.g., using three rules of certainty .5 in series should result in an overall certainty of .125

generalised is just a number and hence added.

Finally, there is a set of possible parses, each with a score. One is chosen randomly, with the probabilites weighted by the score. So if there are two parse, one with a score of .5 and one with a score of 1, the first one is chosen with a probability of $\frac{.5}{1+.5} = \frac{1}{3}$, the second one with a probability of $\frac{1}{1+.5} = \frac{2}{3}$.

A crucial parameter for adjusting the agents' behaviour is the function that computes the final score given the four arguments listed above. Each time a search for parses is started, the agent has to provide such a function (called convert-evaluator-function). Like this, it would be possible that different agents have different judging mechanisms or adapt their judgement over time. The function currently used is the following:

$$\mathrm{evaluate}(\mathrm{cor}, \mathrm{cov}, \mathrm{cert}, \mathrm{gen}) = \left[\begin{array}{ll} 0 & \text{if cor} < 1 \text{ or cov} < 1 \\ \mathrm{cert} * \mathrm{g}^{\mathrm{gen}} & \text{else} \end{array} \right]$$

So only full parses that do not contain any rule inflictions are allowed. g is an agent-dependent parameter between 0 and 1 determining the agent's willingness to generalize rules.

New rules: Generalisation So what is generalisation? It is the key to compositionality, one of the phenomena that are supposed to be examined by the system. It occurs when the agent recognizes similarities, but not complete equality between the current input and a rule. If, e.g., there is a rule

```
((L 1) (U 1) (V 1)) <---> X
```

and the current input is

```
((L 1) (U 1) (V 2)),
```

`((L 1) (U 1))` appears to be a reoccuring combination of values and a new rule is created to map `((L 1) (U 1))` onto a new word.

So, compositionality in language occurs where compositionality in the world becomes evident to the agents. The parameter *generaliserate* mentioned above influences how much a parse's score is diminished if such a generalisation has taken place. Whenever it is below 1, a parse with generalisation will get a worse score than one without. However, a holistic rule may not be at hand and even when it is, the generalisation parse may be chosen in spite of its lower score due to the way parses are picked. If such a parse is chosen, the generated rules are added to the agent's grammar and ready for later reuse, the score of the new rule is the product of the original rule's score and the ratio of the features reused in the new rule.

New rules: from scratch When there is no matching rule, for example in the beginning, when there is no rule at all, a new word is invented for the input, so a new rule is created and instantly used that maps the input vector onto a newly created word. The initial score of these rules is again a fix parameter, currently .1.

The creation of the word consists of assembling a random number of syllabels, with each syllable being composed of either a consonant followed by a vocal, or a consonant, a vocal, and another consonant.

Updating the rules' scores Finally, there should be a successful parse and eventually some non-successful ones (the ones that were not chosen). The score of the rule or the rules that were used in the successful parse are increased by a given factor each, and the rules that were used by the non-successful ones are decreased.

There is an option that would weight the score changes by the parses score; so rules that seemed to fit very well but were not chosen are decreased stronger than ones that did not fit well anyway. This was supposed to keep ambiguity down by decreasing strong concurrent rules while trying not to damage rules that are not in directly in conflict. It turned in later experiments, however, that turning this off leads to faster positive results.

Getting the words When all three colours have been translated into a syntactic structure, one big syntactic structure is built out of them. Then the form elements are retrieved and the elements of each subunit are put out as a "sentence". If one colour has been translated to (ba), one to (buh xam) and one to (fil), the speaker says

`Ba. Buh Xam. Fil.`

If speech output is available, the agent will "speak" those three sentences. This is in practice done by calling the speech output software that comes with Mac OS.

3.3 The Hearer

The hearer's task is to understand the sentence in such a way that its interpretation becomes congruent with its perceptions. This involves again three different tasks:

Perception Get input from the sensors.

Conceptualisation Transform the perceived input into a semantic structure.

Interpretation Transform the utterance heard from the speaker into a semantic structure best resembling the semantic structure that was created during conceptualisation.

Perception

Perception for the hearer works the same as for the speaker.

Conceptualisation

Conceptualisation also works the same for the hearer as for the speaker. This does not only mean that the program is the same; in fact, the data structure passed on to the hearer's interpretation task is the same as the one passed on the the speaker's verbalisation. This is a major simplification, especially for the conceptualisation part as one of the paradigms of the LGA is the coevolution of language and meanings. There is a now unused option for the agents to create their own colour prototypes. It is unused because it further complicates communication and it seemed better to first try the simpler model; the behavioural changes that such an individual conceptualisation would cause might be one of the points still to be investigated.

Interpretation

Creation of a syntactic structure As the hearer hears the three "sentences" uttered by the speaker, it builds one syntactic structure with three subunits, each containing all the words of one sentence (potentially just one). The task is then to use the existing rules in reverse order (compared to speaking) to map back the syntactic structure onto a semantic one.

Decomposal In analogy to the speaker, the big syntactic structure is split into smaller ones, each dealing with one sentence.

Using existing rules Again, the structure is matched against all the if-parts of the existing rules (that are now those parts of the rules that served as then-parts in the speaker's verbalisation), with potentially several rules being applied in series. Also the parses are judged based on

correctness – how many of the word forms in the rule are found in the source structure; since currently, there are only rules with one word, this is either 0 or 1

coverage – how many of the words in the source structure have been used for a rule application. This is 1 when all words have been translated

certainty – the product of all the certainties (that is, scores) of all the rules used

generalised – remains 0 for the whole interpretation process because no new generalisations are made at this stage. The generalisations made by the speaker appear to the hearer just as multiple word sentences. If these are indeed new to the hearer, it does not find rules so the reaction to the speaker's generalisations takes place not here but in the routine that starts when no rules are found.

With this "rule score", the congruence of the resulting interpretations with the rule based is judged. What is still missing is a "reality score" how well the interpretation matches reality. So the resulting semantic structure is matched against the perceived semantic structure: The three features (L, U and V) of both structures are matched against each other. These can either be equal or not, so the reality score is 0, 1, 2 or 3 thirds. The two scores, grammar and reality score, are multiplied resulting in the final score that is used for the probability distribution for the selection of the final parse.

New rules: from scratch Again, it is possible that there is no satisfying rule application. In this case, a new rule or several new rules are created:

one word if there is one word, it can just be mapped onto the whole colour vector that was perceived:

- `(L l) (U u) (V v) <---> BA`

two words if there are two words BA and BU which have to be mapped onto the LUV values, there are six possibilities to assign words to meanings, resulting in twelve new rules:

- `((L l) (U u)) <---> BA,`
  `((V v)) <---> BU`
- `((L l) (V v)) <---> BA,`
  `((U u)) <---> BU`
- ...
- `((L l) (U u)) <---> BU,`
  `((V v)) <---> BA`
- ...

three words with three words, BA, BU, BO, each word may mean any component, resulting in nine new rules:

- `((L l)) <---> BA,`
  `((U u)) <---> BO, ((V v)) <---> BU`
- `((L l)) <---> BO, ...`
- ...

Each of the possible rules is inserted into the agent's rule base with a low score; when the words appear in later runs, the right rules should make possible a correct interpretation while the wrong ones create a wrong one, so eventually the right rules will become stronger and stronger while the wrong ones die out (there is a mechanism that prevents the hearer from using any of the newly induced rules before it has heard further evidence for one interpretation or the other).

Still, the hearer may take the role of the speaker in another run and use one of the wrong rules which might strengthen them – so also the wrongly induced rules may spread and eventually become a valid part of the language).

Updating the rules' scores Once a final parse is chosen, the rules of the chosen parse are promoted just like in the hearer. Also, the rules from the parses that were not chosen will have their scores decreased, again weighted with the score the parse. But there is an additional updating feature here: Besides those rules that produced a valid parse but were just not chosen, there is a possibility for a parse to create an interpretation that gets a good score for its congruence with the rule base while producing a wrong interpretation. This is clearly worse than just providing an alternative path to the right interpretation, so there is an extra decrease for parses that have a positive 'rule score' but a zero "reality score". The rules used in such a parse are decreased again by another parameter. So rules that strongly claim to be right while producing wrong interpretations are strongly decreased.

Communicative success The round is seen as successful if the hearer could understand the utterance without creating new rules. More precisely, the overall parse score of each final parse of the three substructures is taken as the score for the game. If new rules had to be created in the processing of one substructure, the understanding score for that particular structure is 0. The overall score of the round is the sum of these three scores divided by three.

Uttering words The speaker as well utters words. It utters the same words as the speaker, each subunit as one sentence composed of on to three words, but it may either utter them "understanding" (as a statement), when there is a score greater than zero for that sentence, or "asking", when the score is zero. Taking the example from the speaker, the hearer would reply, assuming it has understood only the first sentence:

`Ba. Buh Xam? Fil?`

Sending this to the speech output directly results in the last two sentencs pronounced with an asking intonation.

3.4 The Game Environment

The game environment is responsible for doing all game- or population-level operations as well as for administrative issues:

- hosting the agent population
- providing input to the agents
- measure global behaviour
- store and retrieve data

These issues will be discussed in detail below.

Hosting the agents

There is not much to be done here. A list of agents is initialised at the beginning of a game, and then a speaker and a hearer are randomly chosen each round. At this place, however, spatial factors or deaths and births could be introduced.

Providing input

Unless the program is running from disc with random images from a directory being loaded continuously, the game environment must guard the camera input and react when it recognizes an eye. Only then, the image is passed on to the agents and a round begins. So when running normally, the program is in a loop of getting pictures from the camera and classifying them as containing an eye or not.

Classifying the pictures Each picture is classified using a nearest-neighbour algorithm in histogram space. When starting, the system is given positive and negative examples of images. It extracts the color frequencies in those images and saves the resulting array (histogram) together with the classification. When classifying a new image, the χ^2-distance of this image's histogram to each of the other histograms is computed and the k nearest neighbor's classifications are averaged (weighted by difference), providing a value that says whether this image should either be regarded as a positive or negative example.

Parameters to this procedure are

k how many of the "nearest neighbours" to ask,

G the number of bins in the histograms (in each of the three color dimensions)

and

F the function computing a neighbour's final vote depending on its own score S and its distance D.

While F remained F = S/D all the time, some combinations of k and G were tested regarding their performance (taking into account speed and recognition success) on a test set after having been taught a training set of pictures, and the parameters that showed optimal performance were k=3 and G=40.

Measuring the behaviour of the population

In order to get useful information out of the experiments with the system, several variables indicating the system's current state are stored after each run. The game environment takes care of getting and administrating this data. Which functions are applied in detail will be introduced in section 4.1.

Storing and retrieving data

The data that is produced by the image grabbing and the evaluation functions must be stored. There are several types of game environments to handle this: a very simple one that just stores the values as lists in memory, forgets about the pictures and regularly dumps the lists into a text file. The more sophisticated one stores all data to a remote web server, whose inner working will be described in the next section.

Besides the need to store data, there also is the less obvious need to get data back to the program: unique numbers are required to make databases consistent. Asking the database each time such a number is required turned out to be too slow, a more flexible solution was found that would work for a small number of programs running in parallel: Each game environment is initialised with the currently highest unique number (which is the only time the database is asked for a unique number), and then counts on its own, multiplying by 3 (which makes three possible sites) and adding n<3, with n being specific to a certain site. So the program in the museum with, e.g., n=0 uses the unique numbers 0, 3, 6, 9, . . . , while my local version with n=1 takes 1, 4, 7, ...

3.5 The Web Interface

The system is designed to make its results publically available on a web site[5] that is automatically updated as new games are played.[6] This is realised on an Apache[7] web server running a MySQL database[8] to store the data and PHP4.0[9] to create dynamic web pages.

Online storage

One part of the online programming is required to get the data produced by the program onto the remote server. This is accomplished on the side of the game environment by calling a website with parameters containing the data to be stored. For example, calling the script store-statistics.php4 (see Figure 9) in the server directory /dir with the parameters game = 1, run = 100, slot = 4 and value = 1, which means that in run number 100 in game 1,

[5] www.look-into-the-box.de

[6] in fact, this whole construction was pretty fragile and crashed three days after I had left Paris without ever really recovering. This was already after having recovered from a hacker's attack and moving to a more secure server which would not provide the library functions needed for graphical visualisation of the statistics. So this whole section has a somewhat theoretical flavour; nevertheless, the software does exist in its basic form described here and the website could be viewed up to a .

[7] www.apache.org

[8] www.mysql.com

[9] www.php.net

success (slot 4) had a value of 1, would be achieved by a HTTP GET request equivalent to opening

```
http://www.servername.com/dir/
  store-statistics.php4?
  game=1&run=100&slot=4&value=1
```

with a common browser.

On the server side, the script store-statistics.php4 translates this request into the database command

```
insert into Statistics
       (game, run, slot, value)
       values (1, 100, 4, 1)
```

creating a table entry

game	run	slot	value
1	100	4	1

Figure 9: sample script: store-statistics.php4

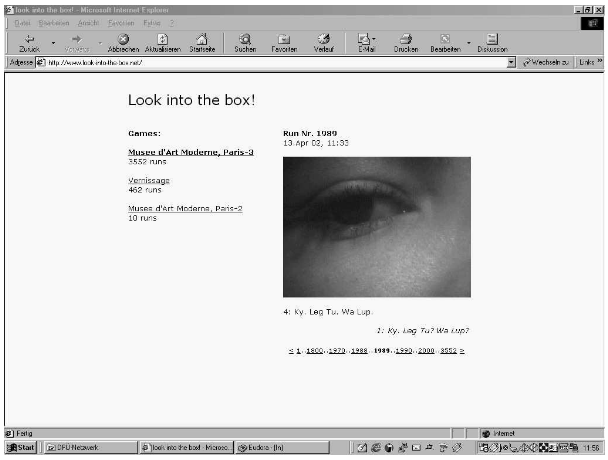
```
<?PHP
# code for database access etc
include ("../init.php");

# create a new database connection and
# send query
$db = new DB_Sql(
  sprintf("insert into Statistics
 (game, run, slot, value)
 values (%s, %s, %s, \"%s\")",
 $HTTP_GET_VARS["game"],
 $HTTP_GET_VARS["run"],
 $HTTP_GET_VARS["slot"],
 $HTTP_GET_VARS["value"]));

# affected rows is 1 if 1 record
# was correctly inserted
# client can search the output for
# "SUCCESS" or "FAILURE"
if ($db->affected_rows()==1)
{ print "store-statistics: SUCCESS"; }
else
{ print "store-statistics: FAILURE"; }
?><
```

There are several scripts to store various data on a server. Although some additional code is needed for the upload of whole images, the basic way those scripts work is always the same.

Online retrieval

Once the data is uploaded to the server, the server can create a webpage displaying the games played, see figure 10.

Figure 10: www.look-into-the-box.net



4 Running the System

In this section, the system will be run with various inputs and parameters and the behaviour will be tested. Several kinds of situations and phenomena will be examined:

- The emergence of a coherent lexicon without compositionality will be examined.
- Once lexical coherence has been achieved, the experiment will be re-run with compositionality enabled.
- For both cases, the influence of the environment on the agents' behaviour will be tested: Two simulations will be run with similar parameters; in the first run, random data will be given to the agents. In the second run, the data produced by the eyes will be used, resulting in a non-uniform probability distribution of the input vectors. It will be examined whether the created languages differ, whether structure existing in the environment can be found in the language as well.

How can we judge what is happening with the population? In fact, there are various ways to extract data describing the current state of a game. Before describing the simulations, these ways to measure the behaviour of the system will be introduced.

4.1 Measuring the System's Behaviour

There are two ways of measuring what is going on inside the system. First, the system's behaviour can be regarded. This is what could be seen by any curious spectator looking at each language game's result: The words used and the success and understanding values. Opposed to these *performance*-based values, there is the possibility of looking into the agents' brains from a godlike perspective, judging their *competence*. The game environment keeps track of both kinds of values, it both observes the values returning from the individual games and computes functions of the whole system by taking into account, for example, all the agents' lexica. In fact, most of these functions are computed by doing operations on the lexicon, there are only a few yet-to-be-implemented ones that would additionally keep track of other things like past lexica or seen items.

It turned out when looking at the results of games that the population functions do give a good idea of the overall direction the population is heading to. For example, it can be monitored how lexical coherence and average understanding increase parallely, and how, during this process, the average sum of rule scores given by an agent becomes stable while the number of rules typically decreases. When less rules together get the same amount of score points as mroe rules did before, this indicates some rules getting very strong while other, weaker ones are pruned. This is what we expect given the fact that lexical coherence increases.

Still, these are descriptions that work on a rather high level of abstraction. What is really happening with the lexica remains pretty unclear. Looking at the rules directly may on the other hand be a pretty tedious enterprise. This is why a more comfortable form of lexicon inspection is introduced in the second part of this section.

Population Functions

Some of these functions can be computed for one agent, others between two agents. However, what is supposed to be measured is the behaviour of the whole population. To get population coherence, for example, which is defined for two agents, each agent would have to be compared to each other agent each round which takes a lot of time to compute (growing quadratically with the number of agents). Often, values like these are approximated for the whole population by just taking the values of speaker and hearer. As each combination of agents should occur equally often, this should in the long run amount to the same.

Understanding *How well was the speaker's utterance understood by the hearer?*

$$\text{understanding}: \quad \textit{game} \rightarrow [0..1]$$

As mentioned in section 3.3, this is the average of the three parses' scores that were created by the hearer.

Success *Was there any understanding at all?*

$$\text{success}: \quad \textit{game} \rightarrow \{0, \frac{1}{3}, \frac{2}{3}, 1\}$$

Similar to understanding, but making a binary distinction between understood and not understood; each of the three subparts counts either as 0 (if ununderstood) or as 1 if it has a positive score.

Average number of rules *How big is the individuals' lexicon, in average?*

$$\text{avnum}: \quad \textit{population} \rightarrow \mathbf{R}_\mathbf{0}^+$$

Average sum of rule scores *What's the sum of an individual's rules' scores, in average?*

$$\text{sum}: \quad \textit{population} \rightarrow \mathbf{R}_\mathbf{0}^+$$

Coherence *How much do the lexica of the different agents equal each other?*

$$\text{coherence}: \quad \textit{agent} \times \textit{agent} \rightarrow [0..1]$$

Coherence is measured by computing the coherence between each rule of the first agent with each rule of the second one. Computing coherence is somewhat hard to define because two rules may be not only congruent or contradictory, they might also just be unrelated to each other. So the coherence function for two rules returns two values: Both the if-parts and the then-parts of the rules are compared. Each pair may be congruent or not. What is returned by the function is both the greater congruence value (which indicated how much the rules should have in common as one of their parts matches) and the smaller congruence value (how much do they in fact agree?). So if neither if- nor then-parts match, the function returns 0/0. But if one part matches completely and the other one does not, it would return 1/0. And finally, if two rules do absolutely agree, the result would be 1/1. In the agent-rating function, these values are added, weighted by the product of the two sum's certainty scores, and the ratio between real and potential coherence is the result.

Compositionality *How strongly are compositional rules preferred over holistic ones?*

$$\text{compositionality}: \quad agent \rightarrow [0..1]$$

Each rule gets a compositionality score; 0 if it is holistic (that is, requires three vector components to match), .5 if it takes two components and 1 if the semantic parts consists of only one component. The compositionality measure of the agent is the average compositionality of its scores (weighted by the rules' score).

Outlook: More Population Functions

There are more things that could be measured but aren't yet. These functions are just proposed to show how more abstract properties of the created language could also be tracked, like the ability to deal with unseen input ("induction") or a general measure of usefulness. This might be especially interesting when going further into the compositionality matter and comparing compositional languages to holistic ones; the latter might look better on the more basic aspects just because they're simpler to learn. The functions listed below are likely to account for a compositional language's advanced capabilities.

Expressivity *How many meanings can be expressed by a language?*

$$\text{expressivity}: \quad agent \rightarrow [0..1]$$

$$\text{expressivity(agent)} = \frac{|\mathcal{M}_e(agent)|}{|\mathcal{M}|}$$

$\mathcal{M}_e(agent)$ the set of meanings the agent can express

$\mathcal{M}$ the set of all possible meanings:

$$\mathcal{M} = \{0..grainsize\} \times \{0..grainsize\} \times \{0..grainsize\}$$

Two related properties can be thought of that might turn out to be interesting in judging how a language deals with pressures that occur in real life, like limited memory and poverty of stimulus:

Recall *How many of the meanings seen can still be expressed?*

$$\text{recall}: \quad agent \rightarrow [0..1]$$

$$\text{recall(agent)} = \frac{|\mathcal{M}_{expressable}(agent)|}{|\mathcal{M}_{seen}(agent)|}$$

$$(\mathcal{M}_{seen}(agent) \neq \emptyset)$$

Be it due to a limited memory or due to misleading communication acts, it is possible for an agent to forget signals for meanings that he already knew. The number of meanings an agent has been able to express at least once during the game is equal to the number of meanings it has seen (since after a game is played, the agent will have a way of expressing the vectors that were seen).

Induction *How many of the unseen features can be expressed?*

$$\text{induction}: \quad agent \rightarrow [0..1]$$

$$\text{induction(agent)} = \frac{|\mathcal{M}_{expressable,unseen}(agent)|}{|\mathcal{M}_{unseen}(agent)|}$$

$$(\mathcal{M}_{unseen}(agent) \neq \emptyset)$$

As long as there are more meanings than those that were mentioned in a speech act (as it is extremely the case in natural language and (although with a comparably tiny meaning space) reconstructed in the experiment), a language may have the capability to talk about unseen meanings. This is a measure to express this capability. Here, the advantage of compositional languages comes out clearly: There cannot be unused meanings being able to be expressed in a holistic language, as there is no structure that the agents could use to derive an uttering for a new meaning from similar, known meanings.

Language Score *"How good is the language doing?"*

$$\text{score}: \quad run \rightarrow [0..1]$$

$$\text{score(run)} = expressivity(run) * coherence(run)$$

In search for a unique indicator for the development of a language towards a useful communication tool, the first idea was to look at coherence alone. The closer the individual agents' dictionaries would resemble each other, the more successful the process of building a language would seem. Seeing that, for complex cases, coherence often drops from the beginning and never again reaches initial values, it became clear that trivial languages concerned with only very few meanings would get very high scores, while a language that expands to express new meanings while self-organising would not make any visible progresses regarding coherence.

So, it is hoped (although not sure because expressivity is not yet implemented) that this function would return a value that gives a sensible information whether success is to be expected to increase or decrease.

Lexicon Overview

As mentioned above, an inspection utility was needed to give more in-depth information than the numeric measures, while still abstracting enough to make it more comfortable than browsing the original rules. This is what the *lexicon overview* is intended to do.

Each rule describes the agent's believe in the mapping of an LUV vector and a word. The idea is now to put all LUV/word mappings on one axis, and all the agents on the other axis. Then, each rule can be displayed by entering its score into the field crossing the mapping and the agent.

The mappings are grouped by meanings. So what appears as a group in the table are all the concurring word forms for one meaning. So situations like "all agents are pretty sure that (1 1 1) should be called ba, but there are two agents that also tend to call it bu" can be directly perceived (see figure 11).

Figure 11: sample lexicon overview

luv	**h**	**Form**	%	**s**	**Agents**		
					0	**1**	**2**
111	0.811	`BA`	0.75	3.0	1.0	1.0	1.0
		`BU`	0.25	1.0	0.4		0.6
avg.h	0.811						

There are also some calculations done to further increase the overview's expressiveness:

s - the score The sum of the scores of all the rules that express this mapping, or "how strong this mapping is in the population". The different word forms are sorted by this value.

% - the relative score This is the score of the particular mapping divided by the sum of all the concurring mappings' scores.

h **- lexicon entropy** This is a value that is computed for each meaning as a function of all the concurring word forms for that meaning. It is defined as

$$H(X) = \sum_{x \in X} P(x) log \frac{1}{P(x)}$$

and describes the amount of information (in bits) contained in the choice of one element x out of a set of choices (a random variable) X with different probabilities $P(x_i)$ associated to each element x_i. For example, the information which of two equally probable choices was made is "worth" exactly one bit. If one choice was known to be more probable before, the information content is a bit less because some of the information is already contained in the probability: as the probability of the two choices approach 1 and 0, the information approaches 0. Similarly, the information content of choices containing more than two elements can be computed.

When this measure was originally invented by Shannon ([Shannon, 1948]), the goal was to express the informational properties of a signal as opposed to its properties as a mere piece of data. The value of information may be seen as the degree of uncertainty in absence of it, and this degree of uncertainty is what is of interest here: uncertainty which word to use for a given meaning means ambiguity in the rules, and so this measure gives a numeric information of how much beliefs about the correct word for a given meaning still vary. If there is only one word left, entropy is zero.

avg h - average entropy This is the average of the entropy values of all meanings. This is again a population-level variable that can be tracked by the game environment and be graphed along with the other variables defined in the previous section. As the population develops towards a shared lexicon with one word for each meaning, it approaches zero.

This should correlate heavily with the inverse of the coherence function. Still, the coherence function regards ambiguity in both directions (also in mapping back from words to meanings) which avg h does not. For this, another table would be required showing the concurring meanings for a single form (this is an issue only when using compositionality).

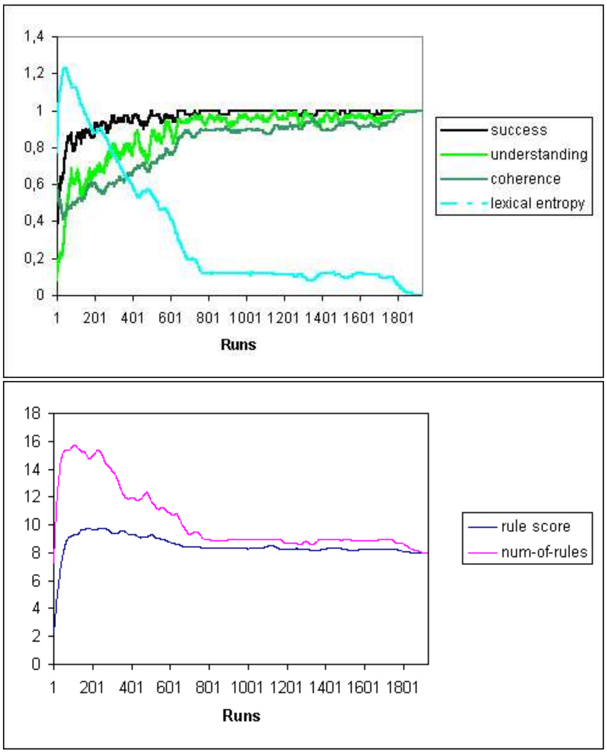


Figure 12: Game 'lexical coherence': Population functions

4.2 Lexical Coherence

The first game played is about lexical coherence only. It is played with random binary three-dimensional input vectors, (0 0 0), (0 0 1), ..., (1 1 1). The global outcome can be examined in figure 12

So what can be read from these graphs?

- The performance-based values success and understanding tell us that the population reaches 100% success, which would indicate the existence of a lexicon shared by all the individuals in the population.

- Looking at the lower graph, we can see (now looking at competence-based data gathered from a "divine" perspective) that each agents ends up having in average 8 rules, with an overall score of 8. This amounts to 8 rules with 100% certainty. What could not be charted because it cluttered the graph is that the ratio between the two in fact develops almost exactly like the understanding curve.

- The two measures that analyze the lexicon more deeply indicate the same: 0 entropy and a coherence value of 1 only leave one interpretation: All the agents have the same rules.

Whether this conclusion is correct can now easily be checked by looking at the final lexicon overview – see Figure 24 on page 50. Is there more to read from the graphs? So far, only the final outcome has been analyzed. There are also several things to be found out about the process of getting to this point:

- After about fifty runs, there is a peak in entropy and in the average number of rules and a minimum in coherence. This indicates a maximum of variety in the rules, a minimum of structure to be found in the lexica. Interestingly, success already reaches over .8 at about the same time. So entropy is not so bad? In fact, high entropy implies that agents understand a lot of words for one meaning. This is what brings up success here: While there is widespread dissent which word is the best one to use in a given context, the agents do somehow understand even words not favoured by them. Of course, those rules' scores are not very high and this is why understanding (which takes into account the score each parse has achieved, in contrast to the binary success measure) stays down longer until coherence increases.
- At about run 800, most of the actions seems to be over. As success and understanding slowly approach 1, there seems to be some ambiguity left, because entropy stays at about .15 for a very long time (about 1000 runs).
- At about run 1800, whatever the problem was is suddenly resolved and total coherence is reached.

In the following, the assumptions made by looking at the graphs will be justified by looking at the lexicon overview for given points in time. Before the spots mentioned above are examined, the first few runs will be presented in order to demonstrate the basic principles of the game at work. See Appendix A for the tables.

Run Nr. 1: The first rules Both agents see (0 1 0), (0 0 1), and (0 1 0) again. Agent 1 takes the role of the speaker, creating the rule `(L 0) (U 1) (V 0) <---> GILCY` with an initial score of .3 (all the absolute numbers mentioned here are parameters to the game). The hearer, agent 0, induces the same rule and initializes it with .1 (induced rules can have another score than self-invented ones). The same happens for the rule `(L 0) (U 0) (V 1) <---> DA`. Then, the rule invented in the first place is used again by agent 1 as (0 1 0) has to be verbalised again. The rule get a .2 increase for being reused. The hearer can also reuse the rule successfully and also promotes it by .2.

Run Nr. 2: More rules In this run, more rules are invented by agent 3 talking to agent 5 trying to communicate the vectors (0 0 0), (1 0 0) and (1 1 0): `POWI. FIP. POGIDLUS.` See Figure 17 for the state of the lexica after this run.

Run Nr. 3: Ambiguity in the world In the third run, agent 5 tries to talk to agent 2 about the vectors (0 0 1), (0 1 1) and (1 0 0). It can correctly reuse the rule `(L 1) (U 0) (V 0) <---> FIP` and it invents a rule `(L 0) (U 1) (V 1) <---> VOV` without anything exciting happening. The interesting thing happens when (0 0 1) is about to be verbalised: there is a word for this ("DA"), but the rule for this is only known to agents 0 and 1. So, a new rule is invented by agent 5 because it does not know that there exists a word for this in the population (see Figure 18).

Run 4: Ambiguity within an agent While the ambiguity created in the last run is only visible when comparing different agents, now a new type of ambiguity will come into play: concurring rules inside one agent. Agents 3 and 5 had negotiated the rule `(L 1) (U 1) (V 0) <---> POGIDLUS` in run 2, but when agent 4 tries to communicate (1 1 0) to agent 5, he cannot know this so he makes up another rule mapping that vector on "DU". This leaves agent 5 who correctly induces the new rule with two possible words for the meaning (1 1 0), "DU" and "POGIDLUS" (see Figure 19).

Run 5: Damping ambiguity This description of the first few steps will close with a demonstration of how ambiguities can be damped again. Agent 5 has to communicate (1 1 0) to agent 1. “DU” and “POGDILUS” are both equally probable, but “DU” wins randomly. This gives a .2 reward to “DU” and a .04 decrease for “POGDILUS” (the total score of the parse, .1, multiplied by the parameter for lateral inhibition, .4 – which may in fact be too little and cause the long time it takes to fully eliminate all ambiguities). Figure 20 shows the updated scores in agent 5.

Run 51: Maximum entropy When examining the population functions, there was a peak in entropy and rule number at about run 50. See figure 21 to get an idea of the chaos in the lexicon at that point in time. Note how up to 4 words for one meaning have been created at that point in time. At the same time, however, also note that most agents understand at least two words for a given meaning. So the lexicon has evolved to a not very elegant, but roughly useful form allowing basic communication, and from now on it will evolve towards efficiency (that is, removal of ambiguity).

Run 400: First favourites At run 400, the language has already had some time to be negotiated by the agents. There is a first commonly accepted rule: `(L 0) (U 1) (V 0) <---> GILCY` is the only rule for (0 1 0) and shared by all agents with a 1.0 score. As for the other meanings, mostly two concurring rules are left, sometimes with a winner already to be expected, others, like “KETADI” and “LUCGAGER” at about equal scores.

Run 800: A mostly stable lexicon After 400 other runs, as we had expected from the graphs, most rules are already fixed. There is a soon-to-be-resolved ambiguity for meaning (1 0 0), and besides that, only the two concurring mappings that could already be seen in run 400 remain unstable. One can see that concerning those two mappings, not much has changed in the last 400 runs. As we already know from the graphs, not much will happen for the next 1000 runs, and now that this will be due to those two mappings going back and forth in the agents. With two mapping being about equally well scored and distributed, there is not much reason to choose the one over the other, so the only way to resolve this problem is to wait for one part to get stronger by a random series of promotions; once one mapping is considerably stronger, it will be chosen considerably more often, thus amplifying the effect.

Run 1952: The final lexicon At run 1952, the simulation was stopped, shortly after a completely coherent lexicon was reached. The struggling between “KETADI” and “LUCGAGER” was finally decided in favour of “KETADI” .

Conclusions The assumptions based on interpretations of the graph could be justified. Two distinct phases could be identified: a quick spread of new “ideas” after the start, and then a longer period of “cooling down”. Already, it might be interesting to do further experiments: Where do those phases come from? The parameters influencing lateral inhibition might to be too weak. Another run with some parameters changed might reveal whether the split into two distinct phases was due to implementation details or if the reasons for this kind of behaviour lie inside the system dynamics.

In fact, looking at the graphs and the lexicon overviews again a few days after I first looked at the experiment’s results, it occured to me there are good reasons to suppose that the two phases come out of the system dynamics: Once an agent has seen each meaning once, he has at least one word for it. When each agent has seen all meanings, no more words will be invented because the only occasion to invent new words (in absence of compositional functionality) is when a new meaning is encountered. So after this point, no new meaning/word-mappings are invented and only the existing ones coexist, with some eventually dying out.

This kind of behaviour is actually predictable from the individuals’ program, but still it did not become clear to me in advance and even when looking at the game, I did not immediately recognize the mechanism at work. So although the system’s behaviour is neither irreducible nor principally unpredictable, running the system has uncovered some knowledge about its components’ behaviour that would have probably remained completely unknown to me otherwise. Is this a case of emergence? In section 6, questions like these will be treated.

4.3 Environmental Influence

The above experiment has been made with idealized data in such a way that there were only few possible input vectors which would appear with equal frequency. This was a comfortable way to demonstrate the basic mechanism at work without having the sheer size of the lexicon obstruct the understanding of what is happening.

Also, prototypical behaviour could be observed well because the environmental influence was basically turned off. The "creative explosion" and "cooling down" phases behaviour, for example, could be observed so well because all input vectors would in average appear equally often. This is because each input vector has its own creative explosion phase (until shortly after each agent has seen it at least once), and a cooling down phase (the rest of the time). These phases, however, only show up on the graphs when they taking place synchronously for all input vectors (which is the case when they appear equally often).

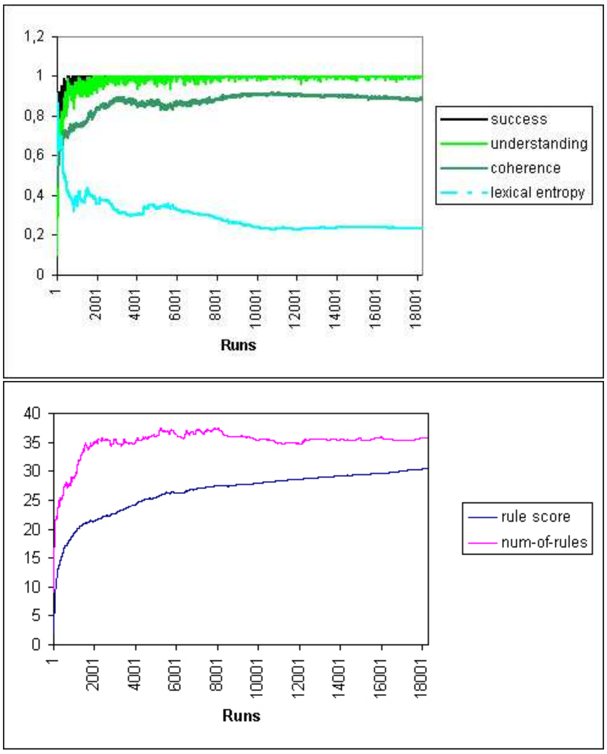
So the "sandbox" environment is good for documentation purposes, but to see whether the model makes sense in a larger context, it must be checked how it performs with real-world input.

The 'Real World'

The 'real world' for the system consists of input taken from the pictures of eyes taken by the camera during the course of the exhibition (I used 3000 out of about 13000 taken pictures). As decribed above, each picture is processed into three vectors, each of these vectors consisting of three components (L, U and V) ranging from 0 to $grainsize - 1$, that is 7 for the simulations described here. This means that there are 512 potential input vectors; in fact, however, only 31 of them actually occur. Refer to table 1 for a list of the vectors that do appear and their relative probabilities.

Visualization Issues I have tried to create a diagram of this distribution. This is not quite simple because the data is in fact 4-dimensional as there are three components describing each data point's position and another one being the actual variable to be plotted. There is no fully satisfying solution to this problem that I know of (especially when the graphs are to be printed – otherwise, one of the dimensions can be put on the time scale). For a mere impression of the way the colours are distributed, however, I have tried to graph the data by plotting each input vecor as a sphere situated in three-dimensional space; the diameter of the spheres is then proportional to the square root of the relative frequency of the vector. I took the square root as a compromise because it is not sure whether the subjectively perceived "size" of a sphere depends on its volume, surface or diameter. Also, very small but non-zero values were increased to a minimum. Refer to the first figure on page 31 for such a visualization of the input space, made with the raytracing program Povray[10].

Figure 13: Game 'environmental influence': Population functions



[10]www.povray.org

Table 1: Distribution of Elements in Input Space (the “real world”, as seen by my system)

l	**u**	**v**	**relative frequency**
5	3	5	0.27965805
5	4	5	0.15999997
5	3	4	0.109059796
6	3	4	0.099487156
4	3	4	0.05675212
5	3	6	0.052991435
6	3	6	0.049230754
6	4	5	0.036923066
6	3	5	0.032478623
4	3	5	0.027692301
7	3	4	0.026666658
4	2	5	0.0259829
3	2	5	0.010940168
6	4	6	0.008205126
7	2	4	0.0075213653
6	2	4	0.0034188023
5	2	4	0.0020512815
3	3	4	0.0020512815
5	4	6	0.0017094011
3	3	5	0.0017094011
4	3	6	0.001367521
7	4	5	0.0006837605
5	2	5	0.0006837605
7	3	6	0.0003418802
7	3	5	0.0003418802
6	4	4	0.0003418802
6	2	6	0.0003418802
6	2	5	0.0003418802
5	4	4	0.0003418802
4	4	5	0.0003418802
4	2	4	0.0003418802

Results

Figure 13 shows the graphs for a game having run 18000 rounds, taking inputs from the space described above. One can see that success and understanding tend to 1 very early. The other measures do show the signs for a converging lexicon (rule scores approaching the number of rules, convergence approaching 1, entropy approaching 0), but even after these 18000 games played, they are still away from perfect convergence.

I will argue that the behaviour is consistent with the one found with artificial inputs anyway. It can also be found in the previous game that the performance-based measures success and understanding virtually approach 1 long before the competence-based ones. This is because a perfectly coherent lexicon is not needed for successful communication.

The first experiment has shown that it may take 1800 games played before each vector has been assigned its final word. This was for a vector appearing with a probability of $\frac{1}{8}$. Taking into account there are three vectors involved in each game, there was a $\frac{3}{8}$ probability for each vector to appear in a game. So in the 1800 games played, each vector should have appeared in average 675 times. Looking at the distribution in table 1, it turns out that there are vectors that have a probability of appearing as low as .0003 (meaning that they appeared only three times in the sample containing 9000 input vectors from 3000 pictures). This translates into 5.4 appearances in average during 18000 runs. I would judge, therefore, that the missing convergence in this game is due to some vectors not appearing often enough. If one let the game run around 100 times longer, one could expect it to behave in the same way like the first one.

Again, looking at the lexicon overview (figure 25 on page 51) can give further evidence. As there are 31 possible inputs, I made the overview a bit smaller, deleting all rules that had a score of 1.0. Like this, the rules displayed are those responsible for the lacking congruence. It turns out that these are the rules concerned with the following input vectors: 335, 424, 445, 525, 544, 625, 626, 644, 735, 736. These are in fact 10 out of the 12 least frequent vectors. The two vectors from the twelve least frequent ones that are not among these, 745 and 436, are indeed described by 1.0-scored rules, however, these rules are shared by three agents only (this is only visible in the full lexicon overview not printed here).

Refer to page 31, the picture in the middle, to see the distribution of rules in input space. The spheres' size corresponds to the sum of the rules describing this particular vector. One can see how nearly all the vectors that appear at all in the input space have become equally big spheres. There are also a few left that are smaller, these correspond to very small spheres in the input space as well: Those are the ones that keep coherence from reaching 1.

Concluding, the remaining incongruence can really be traced back to some input vectors appearing just too rare.

4.4 Compositionality

A whole new set of experiments can be made when the behaviour of the agents is changed as to allow compositional rules. What this means in detail has been explained in the Implementation section. In short, compositional rules are just rules that do not cover all three components of the input vectors, but only one or two. Then, those rules can be chained (like one expressing a certain L value and another one that expresses a combination of a U and a V value). The interesting point about this is the way these compositional rules come into existence. They are not created randomly but by generalization (see page 14). It means that agents are able to detect similarities between the vectors they are currently processing and those that are expressed by their existing rule base. So it is only when an agent has, for example, a rule for expressing ((L 1) (U 1) (V 1)) and it sees a vector ((L 1) (U 1) (V 3)) that there is a chance compositionality comes into play: It then won't create a new rule for the first vector, but generalize the second one so that a new word is created for ((L 1) (U 1)) and then another one for ((V 3)).

Compositionality in Real Language

Compositionality contributes heavily to the complexity found in real language, from the way the alphabet is used to form words[11] over morphology to syntax (not even mentioning the semantic aspects).

Of course, what is called compositionality in the context of the system is a miniature version of real languages' compositionality. But the basic feature, the use of reusable components that combine to new wholes remains the same.

The following simulations might propose mechanisms how compositionality might come into existence and spread. There are other simulations (e.g., [Brighton, 2002]) that focus on compositionality. As shortly mentioned already in the beginning, they focus on the fact that language has to be taught from one generation to the next and that compositionality can compress the number of rules that have to be learnt (compare, for example, the 8x8x8 holistic rules as opposed to 8+8+8 compositional ones needed in an optimal case to span the input space described above). But focussing on compression alone might neglect important other properties: Most of all, language is there not to be taught but to be used, to refer to meanings "in the world". So the difference in my system is that the rules are actively used all the time and being shaped by the interactions between the agents and the influence of the input space.

Experiments with Compositionality

Due to the design of the system, coherence is hard to achieve when compositionality is switched on – whenever a compositional rule describing a certain part of an input vector has died out, the rule creation mechanism will eventually propose the creation of a new rule of the same content. So there is a permanent deleting and creating of rules. Because of that, the graphs describing the holistic experiments do not say as much in this context – they show very bad coherence and entropy values because statistically, the chaos of the created and deleted rules shows up a lot stronger than the few consistent rules (only the success and understanding graphs indicate that there must be something going right after all).

Also, the number of rules in use by an agent is a lot bigger and it is hard to see anything when examining an unedited lexicon overview.

Still, one should not conclude from the cluttered

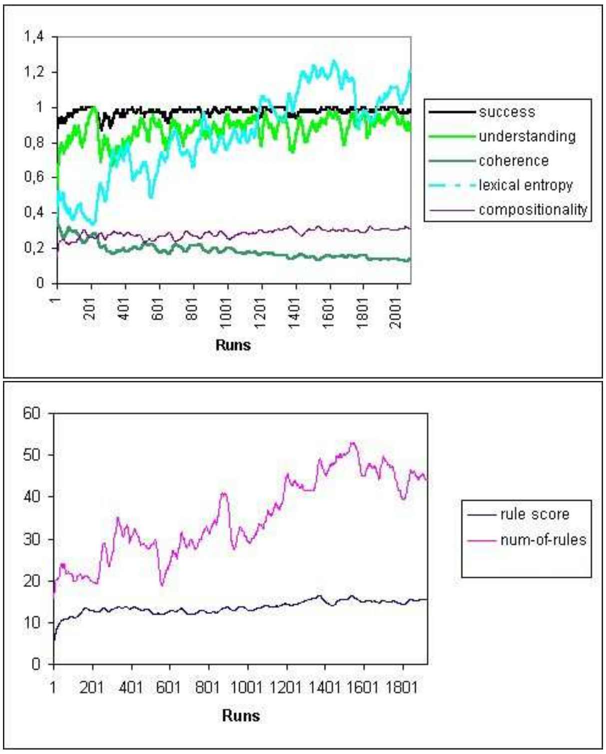
[11] At least in western societies; Asian iconic alphabets give interesting evidence that compositionality is not required to develop on this level, while it may occur on different ones, like the systematicity existing between these icons.

graphs of the population functions that there is nothing to observe but noise. To get an idea of what has happened in the population, I will have a look at only those rules that all agents have a score of .6 or above for.

There are two experiments which have been made with compositional processing turned on in the agents: One with the "idealized" environment of eight equally distributed input vectors and one with the input space as provided by the image data. Both games have been played 2100 rounds with two agents only (it turned out that even with two agents, things tend to become complex enough in this case).

The question is whether a compositional language structure arises in any of the two.

Figure 14: Game 'compositionality without environmental influence': Population functions



Compositionality without Structure

Looking at the trends in figure 14 after 2100 games, things do not seem to go very well. Entropy seems to be inreasing, as does the average number of rules per agent. Still, the language does work somehow: Success is around 1 and understanding is shaky but remains at least around .7 (note that the graphs are smoothened; what is plotted is the moving average of the original values with a width of 50 values – so it cannot be said in a literal sense that understanding never drops under .7).

The lexicon overview (see figure 26 on page 52) explains what is happening. As mentioned above, this overview is pruned to only show the strong rules (> .6). There a seven rules that are agreed upon by both agents with a score greater than .6, in fact, they have a score of 1.0. These are holistic rules expressing seven out of the eight possible input vectors. One input vector (0 0 0) seems to be expressed compositionally as a corresponding rule is missing, but no canonical way has been found to express it.

So all the chaos in the system is caused by the agents inventing new compositional rules that never make it because there is no advantage in using them: The way for a compositional rule to get stronger than its holistic concurrents is to be used more often than they are. This happens only when the rule is used for several vectors, especially ones that are not expressed by any holistic rule either. In this case, however, the holistic rules are created very early (looking at the internal numbering shows that the surviving rules are among the first ones that have been created by the agents ever), soon reach 1.0 confidence and it becomes unlikely (although not impossible, as the example of (0 0 0) shows) that later created compositional rules can ever catch up with them.

Seeing seven out of eight vectors expressed in a holistic fashion may be enough evidence to state that in the absence of a structure, the communication system that evolves also remains unstructured.

Compositionality with Structure

Again, the graphs of the population (figure 15) looks little promising. But again, the success values indicate the language works nevertheless. Also, entropy does not seem to keep on rising and the

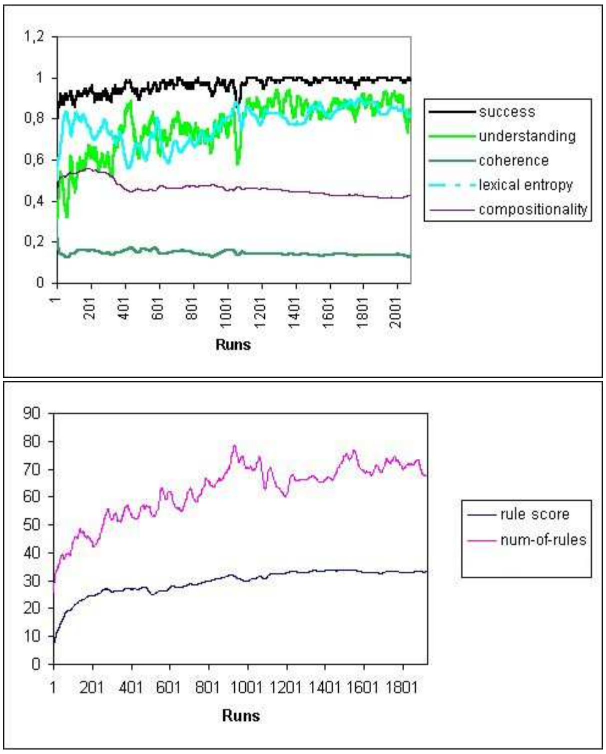


Figure 15: Game 'compositionality with environmental influence': Population Functions

number of rules does not to increase as strongly as above. An interesting value is compositionality: it seems to settle at about .4. While it is hard to judge what this means exactly, looking back to the previous example, where it remained at about .3, one might take this as evidence that compositionality is at least stronger than there.

Looking at the lexicon overview of the strong rules after 2100 runs, there are in fact stable holistic and compositional rules. The ten most frequent colours have been assigned holistic rules (except the second strongest one, 545). The strong holistic rules deal accordingly with many of the more frequent colours that were not assigned holistic rules (including 545). The only rule that has two free components ((L ?) (U 2) (V ?)) has successfully found the largest regularity in input space (when not taking into account the nine rules that are covered otherwise) – that 9 out of the remaining 22 vectors have a U value of 2.

Refer to the picture on the bottom on page 31 for a visualization of the rules. It shows the distribution of rules in the input space with the score of a rule, e.g. 1.0, being distributed over all vectors a rule covers, so ((L ?) (U 1) (V 1)) counts as a 0.125 scored rule for 011, 0.125 for 111, and so on. It can be seen how many vectors can be expressed that are not represented in the input space. One can also see how the highly frequent input vectors are bigger then other vectors right next to them, which means that there a holistic rules increasing their score only, but not the one of their neighbours. This basically fits the detailed analysis given above.

Concluding, it can be said that in this case, compositionality did arise.

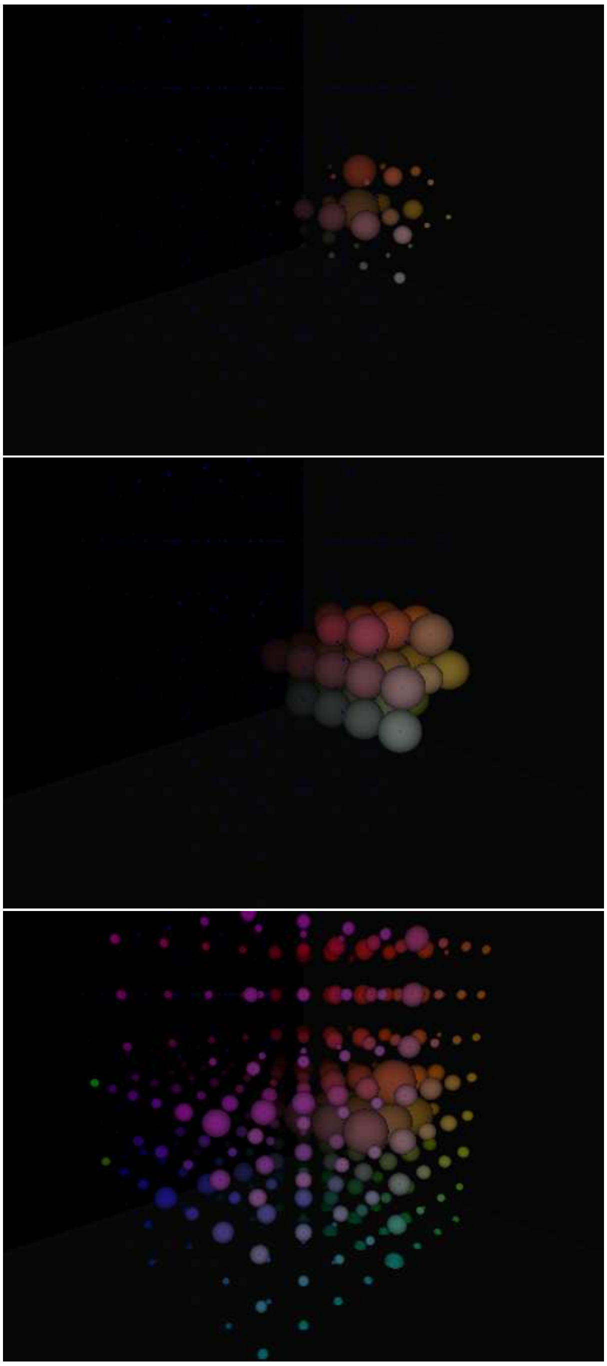
Results

It has been shown how compositionality arises, given a suitable environment. The language adapts to best match the meaning space it is used to talk about. Similarities in the "world" are found and reflected in the way they are expressed.

Of course, the results have to be seen relative to the parameter settings. It would be possible to boost compositional rules so strong that they won in an unstructured environment, or to make it so hard for them that they cannot succeed in the structure example. So I cannot claim that compositionality arises under any circumstances as long as structure is to be found in the environment. What makes the results interesting in spite of their dependance on parameters is that those parameters remained the same for the two experiments. So, all other things being equal, presence or absence of compositionality in language can be said to depend on the amount of structure in the environment.

Despite its simplicity, the model already shows some behaviour that is known from actual language. High-frequency verbs (like “to go” and “to be” in English) tend to behave irregularly; regular behaviour however resembles compositional behaviour on a morphological level, such that one might argue that frequently used rules tend to resist being split up and integrated into a compositional system just like they do in the simulation.

Figure 16: Distribution of Input Vectors, Holistic Rules and Compositional Rules



5 Emergence

Regarding the program described so far, we can see how certain effects take place that can only be observed when looking at the whole system. Although it has been shown how the global behaviour of the system is rooted in local interactions, the formation of a shared lexicon is a behaviour exhibited by the population, not by single agents.

If we examine charts displaying values related to this behaviour, they describe the behaviour of the whole group of agents, as if observing the group as a single individuum, on a higher level of description. While some values are just averages over the contained agents' values (which does not make them particularly strong examples of group-level properties), other values like the lexicon entropy are unique to the group itself and would not make sense in a single-agent context.

Further abstraction from such collective values can lead to qualitative statements about the group's behaviour, like the 'creative explosion' and 'cooling down' phases described above, or the fact that a succesful communication system is established at all.

The rest of this thesis will focus on questions regarding this relation between the simple and the complex properties of systems.

5.1 Explaining Complexity

Understanding how complex behaviour is formed by simpler mechanisms is fundamental to science. As complexity grows, new theoretical levels may be formed. Only when, for example, more atoms are regarded at once, chemical properties like temperature begin to have a meaning. For them to be scientifically grounded, however, it is important that they can be explained by terms of a lower-level theory like, in this example, physics. Often, however, it is the more complex behaviours that we become aware of first. Temperature was known long before the existence of atoms, to stay in the example. It is then the task of science to reduce phenomena to underlying theories, or even to find the underlying theories in the first place.

Although this is the way modern science has worked for a long time now, it is not proven that this approach leads to success with every possible phenomenon. There have been (and there still are) cases where it seems that properties cannot be explained like this.

Facing a failure to apply the traditional approach, two contradicting positions were traditionally held. The physical monist point of view would insist that the phenomenon in question is reducible in principle and that a failure to do so can only be explained by missing knowledge about the basic theories (or that there is something wrong with the explanandum itself, like in eliminative materialism claiming that the whole mind/body-problem comes from sticking to what, for example, [Churchland, 1988] calls "folk psychology").

One might argue, however, that, for example, conscious experience ("Qualia"), the genuine feeling of, e.g., seeing the color red, has something to it that cannot be grasped by even the most sophisticated neurological analysis. The way "life" comes into existence from biochemical interactions is another example of a field that led to raise doubts about the completeness of the reductionist method. It seems that in both cases, something qualitatively new (as opposed to just continuously rising complexity) seems be innate to these phenomena. And so, an opposing, dualist point of view would claim that these properties belong to another domain that exists beyond that one where the laws of natural sciences apply. For example, a supernatural "res cogitans" has been postulated by Descartes to be responsible for mental states.

Both approaches have their problems: neglecting certain phenomena for the sake of a consistent scientific model may be regarded as more acceptable as leaving the domain of the sciences at all, but it is not an ideal solution anyway.

The emergentists then tried to find a sort of compromise between the two approaches, with the first fundamental books on this topic, according to [Stephan, 1999], being published in the beginning of the 1920s: [Alexander, 1920], [Sellars, 1922], [Morgan, 1923] and [Broad, 1925].

First of all, everything was supposed to remain under the causally closed domain of the physical world (no dualism). Still, the existence and relevance of system-level properties was accepted. Properties could show up only from a certain level of complexity onward. These would then be called emergent. Attempts to further narrow down the concept and to point out its consequences were the made in the works of the emergentists.

5.2 Weak Emergence

Basis for all theories of emergence is the so-called *weak emergence*. It outlines the basic character of emergence already mentioned and can be defined by three axioms, taken directly from [Stephan, 2002]:

1. **Physical Monism** Entities existing or coming into being in the universe consist solely of physical components. Likewise, properties, dispositions, behaviours, or structures classified as emergent are instantiated by systems consisting exclusively of physical entitities.

2. **Systemic Properties** Emergent properties are systemic properties. A property is a systemic property if and only if a system possesses it, but no part of the system possesses it.

3. **Synchronic Determination** A system's properties and dispositions to behave depend nomologically on its micro-structure, that is to say, on its parts' properties and their arrangement. There can be no difference in the systemic properties without there being some differences in the properties of the system's parts or their arrangement.

So the basic nature of emergence is decidedly non-esoteric. The first axiom rules out any supernatural entities involved and the third one makes sure that in the relation between those entities, there are no supernatural influences either. The definition itself, in the second axiom, does not make any strong claims either; nothing is said that would preclude reductionism.

5.3 Stronger Definitions of Emergence

To solve the philosophical problems mentioned above, the weak definition of emergence does not help much. The two major questions, how properties might arise that have new qualities in such a way that these qualities cannot be explained to lower-level properties only, and how properties may arise that are genuinely new, are dealt with by two branches of emergentist theories labelled synchronic and diachronic emergence by Stephan (ibid.).

Synchronic Emergence

Synchronic emergence is what weak emergence is turned into when we add the condition that the emergent properties be *irreducible*. While practical difficulties in explaining system properties are well compatible with weak emergentism, synchronic emergentism aims for the *principal* irreducibility. If such properties could be proven to exist within a scientific frame, mental properties irreducible to their physical implementation, for example, could be shown to be consistent with a scientific worldview.

Stephan (ibid.) lists the necessary conditions for a property to be reducible:

1. The property to be reduced must be functionally construable or reconstruable, respectively;

2. It must be shown that the specified functional role is filled by the system's parts and their mutual interactions;

   and

3. The behaviour of the system's parts must follow from the behaviour they show in isolation or in simpler systems than the system in question.

So irreducibility would require at least one of the above conditions to be violated. Stephan continues showing that violation of any of the latter two conditions would imply new types of causal influences going back from the system to its components parts ("downward causation"), which would violate fundamental assumptions as the causal closure of the physical domain. This is the very kind of conflict emergence was supposed to solve – so these two cannot work.

For the first condition to be broken, however, it would suffice that the systemic property not be functionally construable. I cannot go into this in very much detail, but this finally amounts to saying that for a property to be synchronically emergent, it needs to be epiphenomenal, that is, it must not have any causal role.

Diachronic Emergence

Diachronic emergence emphasizes the notions of *novelty* and *unpredictabilty*. Citing two criteria from Stephan (ibid.):

1. **Novelty** In the course of evolution exemplifications of genuine novelties occur again and again. Already existing building blocks develop new constellations; new structures are formed that constitute new entities with new properties and behaviours.

2. **Unpredictabilty** The rise of a new structure is unpredictable in principle, if its formation is governed by the laws of deterministic chaos. Likewise, any property that is instantiated by the novel structure is unpredictable in principle.

Note that the second point does not exclude other reasons for unpredictabilty. Randomness or irreducibility (which implies unpredictability because if we cannot know how the base properties cause the emergent property, we cannot use them to make any prediction either, so irreducibility implies unpredictability) would be two other candidates, but unpredictability caused by randomness is trivial and irreducibility has been dealt with above.

The Fate of Emergentism

Just to quickly continue the story of classic Emergentism – we know that reductionist approaches dominate natural sciences today, and emergentist theories in the stronger sense are not a part of scientific explanation (although the mind/body problem still resists scientific analysis with usual methods).

Emergentism could not introduce a solution to the mind/body problem or other questions of similar structure. The seeming contradiction between a deterministic universe and genuine novelty could be resolved by refering to deterministic chaos. Synchronic emergence however, which could have been the bridge between physical monism and dualism, a foundation for a "non-reductionist naturalism" – turned out, as shown above, to be logically inconsistent with science as long as the properties in question are supposed to have a causal function. It is currently debated whether this is the case for qualia (see, for example, [Chalmers, 1996]), but in general, this does not seem to be a very satisfying solution. Still, emergence has recently had a revival as it has become a key concept in fields concerned with self-organization and bottom-up effects. The notion of emergence used in this context will be dealt with in the following.

In addition, the reductionist program is not as unchallenged as one could assume it, especially in the natural scienes, to be: [Prigogine and Stengers, 1984] give a detailed account of how nature came to be thought of as deterministic, mechanic and dead (ranging back to Newton's discovery of gravity which was thought to be the one force to reduce everything else back to), and how this view neglects other theories (namely thermodynamics, going back to Fourier's observations on the diffusion of heat) stressing the random, irreversible and complex sides of nature. Prigogine describes how, by this, a distance has been driven between nature and humans by viewing nature as mechanic and ourselves as special, animated, non-deterministic organisms.

While this does not exactly go into the same direction as the Emergentism debate, both discussions agree in their critic of a purely reductionist approach and by stressing the importance of paying attention to higher-level phenomena. As Prigogine puts it, the method of reductionism should have been to disassemble parts, look at how they work in isolation, and put them back together to understand the whole. He blames today's scientific practice of "forgetting to put the parts together again".

5.4 Weak Emergence Revisited

In the emergentists' enterprise, the goal was to find a theoretical framework in which properties could exist that would escape standard analysis without leaving the natural scienes' domain. It has been shortly sketched how these efforts lead to problems, especially in the case of synchronic emergence. In the presence of these metaphysical questions, weak emergence seemed rather unspectacular and did not promise much advance.

In the context of Artificial Life, however, one of the fields where emergence has experienced a renaissance these days, it seems that the only type of emergence that would be plausible to talk about is the weak one. Not only are cases hard to think of where stronger kinds of emergence might take place in such a context – they would actually even be undesirable: Yes, new properties, new behaviours are among the key interests of AL research, but in such a way that the very mechanisms of how they come into existence are the target of the researchers' efforts. Diachronic emergence, the instantiation of previously inpredictable properties, would just not contribute to answering these question (although it would, of course, be an impressive phenomenon – maybe the adherents of Strong AL, striving to create "real" new life, would in fact take a more positive stance towards something like this). And yes, properties, behaviours coming into existence out of lower-level ones are the subject, but again, it is the mechanisms, the interactions that cause them that are of interest – strong synchronic emergence would in fact mean that the search for these mechanisms on lower levels is principally futile.

So it turns out that the whole motivation of using a concept like emergence is different between the emergentists' approach and today's studies of complex behaviour. Stephan (2002) mentions that he only included weak emergence as a theory on its own in his systematic account of emergentist theories because it is held as one today, not because it was held during the emergentists' period. In Stephan (1999), modern uses of the word emergence (like in connectionism or studies of self-organization) are examined and in fact, they are all "weak". This may seem disappointing, but it should have become clear that the demands for the concept are just different; weak emergence then is just the link between two theoretical frameworks.

Weakly Emergent as Special Systemic Properties

If today's meaning of emergence is identical with the one defined by weak emergence, for a property to be emergent would mean that it is a systemic property, i.e., that it exists in a system, but not in its parts. This provokes the question, however, why such a property should not just be called systemic in the first place.

The answer is that systemic and emergent properties are not the same after all. There is something special required for a systemic property to be emergent that seems pretty clear on an intuitive level (as, once familiar with the word, one tends to have a strong, instant opinion on whether something should be called emergent or not), but which is very hard to grip on a formal level. We have seen that the hard criteria of irreducability and unpredictability do no get us further in this context, and yet, there is something about novelty on both a systematic and a temporal scale, about things happening "from the bottom up" that remains associated with emergence.

It turns out that the intuitive approach to emergence contains a serious danger: one may easily fall for a notion of emergence that contains subjective judgement.

A typical example of emergence is the ant behaviour that leads to the characteristic ant paths – the question is how the ants manage to find the shortest way between a source of food or material and their home. It turned out that ants permanently pour out pheromones, and that they also follow the pheromone gradient (produced by themselves or other ants before) in their environment. So if a food source is found, an ant is likely to go back to the nest just to go to the same point again to fetch further portions of food. Thereby, it enforces the pheromone trail, increasing the probability of other ants to go into the same direction. Those ones that find a shorter path will cross this path more often, resulting in the pheromone trail being reinforced even more often. So, the behaviour that makes the society of ants seem like an individuum being able to find an efficient path is caused by the interactions of its component parts, the single ants.

Another example of parts working together to create a new property is a car. There is no speed in a wheel, neither is there speed in a motor (in a sense that the motor alone would change its location). Still, when all the parts are put together correctly, the whole has the property of moving around in space with a certain speed. However, one does not find claims that a car's speed be regarded as an emergent property. And one would not expect it to be. But what exactly is then the difference between those two systemic properties?

[Ronald et al., 1999] propose a definition of emergence that contains "surprise" as a criterion. This is then further specified as requiring a "non-obvious" relation between the potentially emergent and the basis properties. So one subjective term is replaced by another – one can easily imagine how one and the same phenomenon may be obvious to one person and absolutely mysterious to another. Emergence would then somehow describe a psychological phenomenon inside the observer, and occur or not occur depending on the person trying to understand a system. This cannot be the solution –

at least it is not a useful one. It basically equates emergent properties with (at least spontaneously) ununderstood properties.

Still, this approach gives a written account of the intuitions underlying emergence. The question is whether there actually is a formally valid, sensible account of this that could help us understand certain mechanisms once they're classified as emergent, or whether things get logically consistent as soon as we try to get more than intuition out of it.

There is a notion of emergence using the terms of uncontrolled and invisible variables ([Steels, 1995]) that [Clark, 1998] uses to define what he calls an "emergent explanation" of a complex system. This emergent explanation stands as alternative to what he calls "toss- and throw-explanation" (which is not relevant in this case) and "componential explanation" which is what could typically be called a reductive explanation. There is a reason, however, why Clark uses this term instead of mentioning reduction: He wants to make sure that the third, emergent explanation is not understood as an anti-reductionist approach. This shows he holds a weak emergentist position; at the same time, he provides two alternative approaches to understanding complex systems. So it seems that he does not automatically equate collective with emergent properties (which is the kind of weak emergence I have been looking for above).

A controlled variable (in this context) is a variable that an agent or a system can influence directly. A visible variable is one that it can, if not influence, at least observe or read. Clark's definition of emergence is that a phenomenon should be considered emergent, if it is best described by describing variables that are uncontrolled or (the stronger claim) even invisible to the system. The global pheromone gradient causing the ant path to take a certain form, for example, is not available to the single ant. Neither can it change it as a whole, nor can it see all of it. So the ants are guided by a structure that is beyond their scope (they can access it locally, but that is not a contradiction – there has to be some way of interacting with it). A car, however, has dedicated parts to measure and influence its speed (to put things very simple).

Because of this, the car can be explained well by refering to its components and their function; the ant society, on the other side, shows behaviour that is best described by refering to values its components cannot be related to. Following Clark's approach, this is the difference between componential and emergent explanation; phenomena suggesting an emergent explanation would then be called emergent.

As Clark remarks, there is still a certain subjective taste in this by refering to what is supposed to be a "good" explanation. This seems a viable compromise, however. Saying something like "Emergence is when we have to go to a higher level of description to understand what is happening" is a lot more valuable than saying "Emergence is when we are surprised". Whether a given theory succeeds or fails to explain a certain phenomenon should be a rather objective criterion as well, or at least less dependent on individual factors (trying to explain where a single ant's path comes from without refering to the greater context seems to be an objectively worse explanation than trying to explain where a car's speed comes from without refering to the greater context of the whole car system). This concept is of course never as strong as the proof that a new level of complexity has been reached that makes all reductive explanation futile. But it gives a rather stable criterion about which systems have to be regarded as being more complex than others, at least when trying to explain them.

A Last View on Strong Emergence

It is a difference whether we talk about things that really happen in the world or whether we use them as a way we see it. There is always a bias in the way we perceive reality so there may not be a binary cut, but in the case of emergence, there clearly is a difference: Whether there are really situations in nature where genuinely new things arise, and things happen that do not happen in lower levels, or whether it is just theoretically convincing not to reduce them. This difference could be seen as another way to distinguish weak from strong emergence.

In fact, the dispute whether certain concepts refer to reality or serve as theoretical tools in explaining it, is common to other "weak/strong" debates. Is Artificial Life about creating life or about understanding mechanisms of life? Is Artificial Intelligence about creating real intelligence or about simulating it? The question whether emergence refers, e.g., to something happening out there in a brain, for instance, or whether it is an approach to understanding what is happening in the brain seems to fit into this family of questions.

6 Emergence in the System

Now that various definitions of the emergence have been introduced, I will look at the actual system to check whether any of the potential phenomena observed can be called emergent. The purpose is to, on one hand, see how the system's behaviour can be further categorized, thus enabling, for example, comparisons with other work dealing with collective phenomena. On the other hand, the notions of emergence themselves are also examined with regard to their applicability in the current context.

To do so, I will set up three working hypotheses about emergence in the system analogous to common uses of "emergence" in related situations. I will then continue to check each one in detail.

1. "As a series of games is played, a communication system emerges."

Here, emergence is used to refer to the process of something coming to existence, a lot like in everyday language, or, more specially, to the process of linguistic structures coming into existence, like in [MacWhinney, 1999].

2. "The communicative behaviour of a single agent can be said to be emergent."

This is a use of emergence refering to the relation between properties of a robotic (or simulated robotic) agent and the properties of its functional components, as described in [Steels, 1995].

3. "The population shows emergent behaviour."

Now, the agent ist the part and the whole system regarded is the population of all agents. This is the typical way emergence is used to describe behaviour in self-organizing systems (see, for example, [Camazine et al., 2001]).

6.1 Emergence of a Communication System

The agents start without any communication system at all. Not only do they not share any vocabulary, the single agents do not even have words on their own to name the inputs they get. As games are played, agents make up words to express what they see, and as they share those words with their communication partners, the word/meaning-pairs spread over the whole population, resulting in a successful communication system. Although ambiguities remained in the simulations with real world input, mainly due to some input vectors appearing just too rarely to guarantee full convergence, it has been shown that the mechanisms at work can principally produce a coherent lexicon shared by all agents. Even with those slight variations among the individual lexica, the communication system can be said to be successful (these variations exist in human language as well) as understanding approaches 1. There is a major part of the vocabulary that all the agents agree upon, and an agent may understand a word even if it does not favour it.

So it is surely justified to speak of the existence of a "global" lexicon as a linguist might create it after intensely studying the communicative behaviour of the agents. Or, as a new agent would learn it (or, more likely, again a small variation of it) when it enters the population (this could actually be tried out but hasn't been yet; one can imagine, however, how the new agent's new words will have a hard time concurring with the mostly stabilized existing ones, resulting in the new agent taking over the established rules while changing them at most a little bit).

Emergence as a Process

So something is created and one might be willing to call this emergence – but does it make sense to ask about the emergence of a lexicon using the definitions introduced?

If we say that something emerges, in the sense of something coming into existence over time, emergence is used to describe a *process*. In the definitions above, however, there is no temporal aspect. Properties can be emergent with respect to others, but this refers to properties being instantiated at the same time. In fact, there is some temporal aspect in diachronic emergence, but this is about properties being instantiated at a given point in time that have never been instantiated before. There is no reference to a process in this either.

Also, another point why above defintions are not applicable: A structure (like a lexicon) is not a property. To take a very simple example: My hand is not a property. Having a hand is a property of mine, and having grown a hand years ago is a property of my organism, and it will be shown later how the properties of creating something and the thing itself hang together when talking about the emergence of either of them; but for emergence as introduced so far, it just does not make sense to ask if the lexicon is emergent because this translates to "it the lexicon is an emergent property" which is an invalid in itself because the lexicon is not even an unemergent property.

So it seems there is even another type of emergence which I will call the "process-type". I would claim, however, that this is more like the "everyday" use of the word, in the absence of explicit definitions (cf. "emerging markets", for example). As an example, "Emergence of Language" ([MacWhinney, 1999]) does not give a definition of what emergence might be at all – it does, however, state that emergence can take place on a variety of time scales. So process-type emergence seems to be meant when no other definitions are given.

Emergence of Language

An interesting point is "emergence of language".

When I first heard about the work on emergence of language, I thought it was about the relation-type: How does the abiltity of language processing and creation emerge from lower-level routines like associative learning, pattern matching and so on? To claim that language is emergent in this particular way could be useful in the innatist / interactionist debate: If there is no such thing like a universal language device, one might express this by saying that language emerges (somewhat as a side-effect) over the more basic intellectual capabilities of humans, wheras the existence of such a specialized device would surely forbid calling the very capability it is designed for emergent.

It turned out that research on emergence of language was, at least partially, meant in a different way: What are the processes involved in the creation of language, or of languages[12]. In this context, language is viewed not as a property, but as a structure that's being built and modified.

Of course, the two questions of language capability and language formation are intertwined, but I think one can clearly distinguish the two uses of the word emergence in that case. The current context is about language formation; the other two points are about language capabilities (language always to be seen in slight quotation marks).

Result

So can we say that during the course of the game, a communication system emerges or not?

We could just say it because there is no definition attached to it and it seems like common use of the word in everyday language.

However, the kind of activity or behaviour leading to the creation of a structure might be the relevant point whether to call its coming into existence emergent or not, with a stronger meaning. Below, it will be checked whether the behaviours that cause the communication system to be created are emergent or not. It will turn out that (according to Clark's definition) there is an instance of emergent behaviour involved in the creation of the communication system, which at least supports the definition of process-type emergence I would like to propose just a final note on this: "The instantiation of a structure can be called emergent if the behaviours contributing to its creation can be called emergent." (according to which the answer would still be yes).

6.2 Emergence of Communicative Capabilities inside a Single Agent

It might be interesting whether the capability of language production and comprehension results from interaction of other, independent systems or whether it is "hard-coded".

For humans, this might amount to the question whether or not there is a specialized language device. Talking about computer programs, the corresponding task would be to find out if the behaviour in question is explicitly written down in the program code or whether it takes place as a side effect. [Cangelosi and Parisi, 1998], for example, show a model of how a simple, binary signalling system may come into existence in a population of neural

[12] As I have recently learnt, French is more precise in this distinction, calling *a* language *language*, but the phenomenom language as a whole *langue*

networks. Although there is a selectional pressure for understanding signals, there is none for producing (there is no reward for altruistic behaviour or anything like this either). Still, the networks arrive at producing coherent output. It turns out that the production of coherent signals in their nets develops in parallel with the ability to correctly classify input. The coupling between the classification and the output parts of the net then takes place, as pointed out by Barbara Hammer (personally), as a result of dynamics inherent to the network architecture ("genetic drift").

In the agents I programmed, however, the communicative behaviour consists of pattern matching and associative learning – both of which are hard-coded. There is no behaviour in the agents' program which goes beyond the parts' functionality.

Proving the absence of emergence

How can this be used to formally show that these properties are not emergent? It turns out that claiming emergence and failing to prove it is easier than actually proving the non-existence of emergence: In order to really prove the absence of emergence, it had to be shown that there cannot be any uncontrolled or invisible variables using which the behaviour of the system could be described better than by describing the behaviour of components it is made of and their interactions. First, this condition sounding a bit strangely shows that the definition found is still not as formal and observer-independend as one could wish. Second, disregarding the difficulties of "principal absence of a better explanation", it might be hard to prove that it is impossible there are system-level variables unknown to me. So I can only argue that there is but one that would occur to me.

The only candidate I can think of is what could be called a "single lexicon emergence": one could measure the average amount of entropy in the agent's own lexicon, and claim that the agent's behaviour can be described as striving to minimise this entropy, which is a variable invisible to all of the agent's system. Although this looks tempting, entropy in fact does not really exist in an isolated agent. It is only with other agents that multiple forms for a meaning enter the agent's lexicon – and only then, entropy starts to rise above 0. So I would suggest counting this behaviour as belonging to the population dynamics. Furthermore, one may argue whether the agent's behaviour is entropy-reducing in the first place. As it turned out in the first simulation described, two meanings with about equal score may survive in parallel for a long time. When finally one of them wins, it is due to random choices which rules and also which agent to use. So describing the agent's behaviour using its internal lexicon entropy might be futile anyway.

So the individual agent's communicative capabilities cannot said to be emergent.

6.3 Emergence of Communicative Capabilities inside the Population

In the first examination of emergence, it has been shown that there is a communication system and that the process of its instantiation may be called emergent – mainly due to the lack of a strict definition. As mentioned above, the disposition to build such a communication system, that is, agreeing on a shared lexicon within a population, is a property to which the assumed for criteria for emergence may apply.

So the question is whether there is a description of the system and the way it creates a coherent lexicon that refers to system-level variables uncontrolled or invisible to the simple agents. If one could then judge that this description gives a better acount than a componential one, the observed phenomena can be called emergent.

The componential description already exists: See chapter 3 on the system's implementation. The fact that letting the system actually run may produce interesting results not obvious from the system's setup – as shown in chapter 4 – may already be seen as a first hint that the system's dynamics are not fully described by the single agents' dynamics.

I will only look at a description for the coherence-establishing behaviour in the population and not to deeper into an analysis of the behaviour that can lead to compositionality. Also, I will not make up a complete system-level theory of how the system works. I will just list the two sentences that I use to describe the system's behaviour in a quick, understandable way, and then check for each one whether it is an emergent or a componential description.

1. "The population creates a shared lexi-

con."

It turns out that even this most superficial description of what is happening already is an emergent one: To say that a shared lexicon is created refers to the relation between the single agents' lexica – to which the single agent does not have access, as it can only see its own one. The coherence measure defined for two lexica is hence an invisible variable to the agent.

Also, the shared lexicon in itself is a very inaccessible kind of system-level variable. In the end when the agents match perfectly, one can just take one of the agent's lexica to get hold of it. As long as there is incoherence, however, one might actually say that there is a good portion of a lexicon that already is shared – but it is nowhere to be found as a data structure in the program. One might only try to access it by looking at the commonalities of all the agents' lexica (a bit like what I have done in the lexicon overviews for the compositional experiments).

The final, strongest argument is that in order to predict the system's behaviour – to predict, for example, which of the form/meaning pairs concurring in the population will eventually win –, local information does not help at all. What is needd is the global picture of all the scores in all the lexica.

2. "The system's behaviour can be split into two phases."

Back in section 4.2, I was astonished to find the behaviour of the system to be (in the idealized case) split into two phases which I labelled the "creative explosion" and the "cooling down" phase. I then posed the question whether this was a case of emergence. Now, enough fundament has been laid to answer that question. The answer is, however, yes and no. "No" because the fact that I was surprised of course does not make anything emergent.

"Yes" because these properties are nevertheless emergent, but for another reason than surprise: Because these are phenomena observed in systemwide variables uncontrollable und invisible to the single agents. The transition from the first phase to the second shows up as the entropy stops increasing and decreases, approaching 0, while coherence (as a somewhat complementary measure) turns to increase and starts to approach 1. These variables are, as has been argued before, system-level variables which cannot be seen by the agents as they capture the relation between two lexica. Still, the transition can also be seen in the number of rules per agent starting to decrease – which is although not immediately controllable at least visible to the agent. So one might argue that coherence and entropy are just derived values based on local properties. Although coherence and entropy do not just go back to rule numbers anyway (as one might imagine each agent having a minimum number of rules without anyone understanding the other), this may still serve as a motivation to find a better clue for the emergent character of this phenomenon.

Looking for the actual causes of the phase transisition may be of help. As mentioned above, it lies in the fact that for each input vector i, there is a point t_i in time when each agent has seen it at least once. From that point on, no more words for this vector can be invented. The agents may continue a while learning new words from other agents, but the set of word/meaning-pairs in the population does not get larger anymore. The only thing that can happen is that complete word/meaning-pairs die out because in all agents, the corresponding rules have dropped to 0. But this means that the behaviour of the whole system depends on the points t_i at which all agents have seen input vector i once. These are, much clearer than coherence and entropy, real system-level variables depending on the number of agents and the structure of the input space (plus randomness, although not decidingly). There is now way in which the t_i can be seen from within the scope of single agents.

This explanation seems like a better argument for emergence than just pointing out to the importance of coherence and entropy; not only because the t_i are clearer of systemic character; also because the property is not only described, but actually explained by means of a systemic variable.

Both ways of going into the details of the described behaviour, however, are emergent ones; it is the question whether there exists a componential explanation of it at all. At least, in the specification of the agents and the way they are made to interact, I did not find evidence for such behaviour. If there is a componential explanation or even only a description, it is well-hidden. This nicely prevents me from having to compare the emergent explanation to a componential one; I just claim that a plausible, available description is a better one than one which is well-hidden and may actually be non-existant. So the emergentist explanation is the one of choice – which would classify this behaviour, according to Clark's defintion, as emergent.

The cooling down phase alone, by the way, is analog to one of the paradigmatic examples of emergent behaviour: simulated piling behaviour of termites ([Resnick, 1995]). As pieces of material are randomly shifted from one pile to another (like scores being reduced and increased many times in row), there is a possibility of one termite taking the last piece from a pile (all agents reducing the score of a rule to 0), but there is – because pieces are only dropped onto existing piles – no possibility for new piles to be created (no new rules are created because this only happens in the absence of any rule for a meaning, which is per definition not the case in the cooling down phase).

To come back to the infamous surprise element of emergence, there actually is a relation between my astonishment and the emergence, but it is different. It is not my surprise that causes emergence. It is also not the case that emergence has to cause surprise. But my surprise and the emergent character of my system have the same cause, invisible variables. As I have been writing code for single agents, not for the population, some systemic properties are for me in a way as potentially uncontrollable as for my agents. So although this surprise is not necessary (it could have been my goal to achieve exactly this behaviour and in fact, much of the emergent behaviour was really planned to take place), it is at least plausible that system-level uncontrolled or invisible variables are instantiated outside the designer's attention (as they do not show up as variables in the program code). This explains the frequent cooccurence of emergence and surprise. It also explains the importance of emergence for AL . Emergent properties of Artificial Life systems are those that are implicit consequences of the written code. As stated above, they do not have to be irreducible (in fact, we wouldn't want them to be); nor do they have to be unknown first. But the reason why it makes sense to use multiagent systems as exploratory tools at all is to find those of the emergent properties that were not obvious in the first place, to the programmer who has built the agent, or to the biologist being able to describe an organism's structure and its behaviour in isolation, but unable to deduce its situated behaviour from that.

7 Results & Outlook

7.1 Emergence

There are two directions in which the results concerning emergence can be interpreted.

Taken the definition of emergence finally used, it could be shown that those parts of the system that intuitively seemed to behave in an emergent manner can really be called emergent based on formal criteria.

If, on the other hand, the definition is not taken for granted and its use in the given context is to be examined, the results suppose that this definition correctly identifies those behaviours that work in a non-centrally controlled, bottom-up fashion. One can imagine situations in which a system is more complicated and it is less obvious whether its behaviour can be called emergent or not. Which might, for example, affect the decision if a given, 'high-level' explanation can be seen to be satisfying or if an explanation that relies more on the individual components would be pursued. Then, showing that such a system can be called emergent with respect to a formal definition could help argument for a system-level explanation.

Of course, there is a certain circularity in this argumentation: Assuming that A (definition of emergence) is true to prove that B (sytem shows emergent properties), and then assuming that B to show A makes a poor proof. The point, however, is not really to prove anything; so this may be excusable. The point of departure for me was rather that I saw the term emergence appear frequently in (especially evolutionary -) linguistics and AL literature, that I also knew it from philosophy, and that there seemed to be a conflict between the two (or more..) uses of the word. This has now been resolved so far that as it has been shown how a definition used in the AL context can be systematically connected to its emergentist roots.

Many interesting aspects in this context were not covered here and might be further examined: The relation between "self-organization" and "emergence", for example. It seems to me that these terms are often used synonymously. I think however that although self-organization might actually imply emergence, there are probably emergent phenomena without self-organization. Seriously dealing with this question would have required going deeper into the definition of self-organization, which would have been nice but could not be done here due to length and time restrictions.

7.2 Evolution of Language

Computational simulations of language evolution is still a very young field. There still are many phenomena in natural language that have never been simulated. There will probably much forthcoming work, especially as traditional linguists gradually become convinced of the method. A major advantage of these simulations is that factors of social interaction and context can be integrated that are hard to capture in former formal descriptions of language processes. This is paralleled by developments in functional and cognitive lingustics:

> "The recognition of [the foundational status of the functions of language (semiological and interactive function)] is the primary feature distinguishing *functionalist* approaches to language from the *formalist* tradition (notably generative grammar)."
> – [Langacker, 1998].

One of the key arguments in favor of an innate system for language aquisition is still that language be too hard to learn without genetic predisposition considering its complexity and the number of examples children typically observe before they have learnt a language. Further work stressing the aid social and contextual information gives might eventually lead to new models of language complexity and especially humans' potential to learn them.

Compositionality

Mechanism have been that described that produce compositionality depending on the environment's structure.

The work on compositionality could be further pursued with some changes on the system to make compositionality more obvious. For example, the rule creation might be somewhat dimmed so there not so many low-scored rules cluttering the lexica all the time. Also, one might take more performance-based measure (measuring language use, not the lexica) such that these mainly unused rules just do not show up; the behaviour in the example where no compositionality did arise was wholly holistic, but the measures used failed to clearly express it.

Evolution of Grammar

The next step after dealing with lexical issues is the evolution of grammar. As this thesis is being finished, a first public demonstration will be given of a system by Luc Steels in which case grammar evolves (using the FCG formalism described above).

8 Acknowledgements

The major part of this thesis and basically all of the implementation was written at the Sony Computer Science Lab in Paris, providing a highly motivating environment. I thank Angus McIntyre for helping me out whenever LISP just seemed to fool me, with neverending patience and humor.

I also thank everyone who helped with useful comments and discussions regarding this thesis, notably Anna Krusch, Jan Peters and my mother, and Matthias Peter and Jennifer Sahm for not hesitating to drive for hours to provide me with hardware in the hour of need.

References

[Alexander, 1920] Alexander, S. (1920). *Space, Time, and Deity.* Macmillan, London.

[Brighton, 2002] Brighton, H. (2002). Compositional syntax from cultural transmission. *Artificial Life.*

[Broad, 1925] Broad, C. D. (1925). *The Mind and its Place in Nature.* Kegan Paul, London.

[Camazine et al., 2001] Camazine, S., Deneubourg, J., and Franks, N. (2001). *Self-organisation in biological systems.* Princeton university press.

[Cangelosi and Parisi, 1998] Cangelosi, A. and Parisi, D. (1998). The emergence of a 'language' in an evolving population of neural networks. *Connection Science*, 10(2):83–97.

[Chalmers, 1996] Chalmers, D. J. (1996). *The Conscious Mind.* Oxford University Press, Oxford.

[Churchland, 1988] Churchland, P. S. (1988). *Neurophilosophy: Toward a Unified Science of the Mind/Brain.* MIT Press, Cambridge, MA.

[Clark, 1998] Clark, A. (1998). *Being There : Putting Brain, Body, and World Together Again.* The MIT Press.

[Dixon, 1997] Dixon, R. (1997). *The Rise and Fall of Languages.* Cambridge University Press, Cambridge, U.K.

[Kaplan, 2001] Kaplan, F. (2001). *La naissance dúne langue chez les robots.* Hermes Science.

[Kirby, 2002] Kirby, S. (2002). Natural language from artificial life.

[Langacker, 1998] Langacker, R. W. (1998). Conceptualization, symbolization, and grammar. In Tomasello, M., editor, *The New Psychology of Language: Cognitive and Functional Approaches to Language Structure*, chapter 1. Lawrence Elbaum Associates, Mahwah, New Jersey.

[Langton, 1995] Langton, C. J., editor (1995). *Artificial Life: an overview.* The MIT Press, Cambridge, Massachusettes.

[MacWhinney, 1999] MacWhinney, B., editor (1999). *The Emergence of Language.* Lawrence Elbaum Associates, Mahwah, New Jersey.

[Morgan, 1923] Morgan, C. L. (1923). *Emergent Evolution.* Williams and Norgate Ltd., New York, London.

[Oudeyer, 2001] Oudeyer, P.-y. (2001). Coupled neural maps for the origins of vowel systems. In *Proceedings of ICANN 2001, International Conference on Artificial Neural Networks, Vienna, Austria, LNCS, springer verlag*, Lectures Notes in Computer Science. Springer Verlag.

[Prigogine and Stengers, 1984] Prigogine, I. and Stengers, I. (1984). *Order Out of Chaos: Man's new dialogue with nature.* HarperCollins, London, U.K.

[Resnick, 1995] Resnick, M. (1995). Learning about life. In Langton, C. J., editor, *Artificial Life: an overview.* The MIT Press, Cambridge, Massachusettes.

[Ronald et al., 1999] Ronald, E., Sipper, M., and Capcarrère, M. (1999). Testing for emergence in artifical life. In Floreano, D., Nicoud, J.-D., and Mondada, F., editors, *Advances in Artificial Life - 5th European Conference on Artificial Life.* Springer Verlag, Berlin, Germany.

[Sellars, 1922] Sellars, R. W. (1922). *Evolutionary Naturalism.* Open Court Publ. Co., Chicago.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communiation.

[Shieber, 1986] Shieber, S. M. (1986). *An Introduction to Unification-Based Approaches to Grammar.* CSLI, Stanford University, Stanford, California.

[Steele, 1990] Steele, G. L. (1990). *Common LISP: The Language.* Digital Press, Bedford, MA, second edition.

[Steels, 1995] Steels, L. (1995). The Artificial Life roots of Artificial Intelligence. In Langton, C. J., editor, *Artificial Life: an overview.* The MIT Press, Cambridge, Massachusettes.

[Steels, 1997] Steels, L. (1997). Synthesising the origins of language and meaning using coevolution. In Hurford, J., editor, *Evolution of Human Language.* Edinburgh Univ. Press, Edinburgh.

[Steels, 1999] Steels, L. (1999). The talking heads experiment. volume 1. words and meanings. Special pre-edition for LABORATORIUM.

[Steels, 2002] Steels, L. (2002). Iterated Learning versus Language Games. two models for cultural language evolution.

[Steels and McIntyre, 1999] Steels, L. and McIntyre, A. (1999). Spatially distributed naming games. *Advances in complex systems*, 1(4).

[Steels and Vogt, 1997] Steels, L. and Vogt, P. (1997). Grounding adaptive language games in robotic agents. In I. Harvey, C. H., editor, *Proceedings of the Fourth European Conference on*

Artificial Life. The MIT Press., Cambridge MA and London, 1997.

[Stephan, 1999] Stephan, A. (1999). *Emergenz - Von der Unvorhersagbarkeit zur Selbstorganisation.* Dresden University Press.

[Stephan, 2002] Stephan, A. (2002). Emergence. In *Encyclopedia of Cognitive Science*. Nature Publishing Group.

[Tomasello, 1998] Tomasello, M., editor (1998). *The New Psychology of Language: Cognitive and Functional Approaches to Language Structure*. Lawrence Elbaum Associates, Mahwah, New Jersey.

A Lexicon Overview Tables

Figure 17: Game 'lexical coherence': first rules at run nr. 2

Meaning	h	Form	%	s	Agents 0	1	2	3	4	5
000	0.0	`POWI`	1.0	0.4				0.3		0.1
001	0.0	`DA`	1.0	0.4	0.1	0.3				
010	0.0	`GILCY`	1.0	0.8	0.3	0.5				
100	0.0	`FIP`	1.0	0.4				0.3		0.1
110	0.0	`POGIDLUS`	1.0	0.4				0.3		0.1
avg.h	0.0									

Figure 18: Game 'lexical coherence': the sources of ambiguity at run nr. 3

Meaning	**h**	**Form**	%	**s**	**Agents** 0	1	2	3	4	5
000	0.0	POWI	1.0	0.4				0.3		0.1
001	1.0	ZOREP	0.5	0.4			**0.1**			**0.3**
		DA	0.5	0.4	**0.1**	**0.3**				
010	0.0	GILCY	1.0	0.8	0.3	0.5				
011	0.0	VOV	1.0	0.4			0.1			0.3
100	0.0	FIP	1.0	0.7			0.1	0.3		0.3
110	0.0	POGIDLUS	1.0	0.4				0.3		0.1
avg.h	0.16									

Figure 19: first lexical ambiguity inside an agent at run nr. 4

Meaning	**h**	**Form**	%	**s**	**Agents** 0	1	2	3	4	5
000	0.0	POWI	1.0	0.4				0.3		0.1
001	1.0	ZOREP	0.5	0.4			0.1			0.3
		DA	0.5	0.4	0.1	0.3				
010	0.0	GILCY	1.0	0.8	0.3	0.5				
011	0.0	VOV	1.0	0.4			0.1			0.3
100	0.0	FIP	1.0	0.7			0.1	0.3		0.3
101	0.0	PIP	1.0	0.8					0.5	0.3
110	1.0	DU	0.5	0.4					0.3	**0.1**
		POGIDLUS	0.5	0.4				0.3		**0.1**
avg.h	0.28									

Figure 20: first damping of an ambiguity at run nr. 5

Meaning	**h**	**Form**	%	**s**	**Agents 0**	**1**	**2**	**3**	**4**	**5**
000	0.0	`POWI`	1.0	0.4				0.3		0.1
001	1.0	`ZOREP`	0.5	0.4			0.1			0.3
		`DA`	0.5	0.4	0.1	0.3				
010	0.0	`GILCY`	1.0	0.8	0.3	0.5				
011	0.0	`VOV`	1.0	0.4			0.1			0.3
100	0.0	`FIP`	1.0	1.4	0.3		0.1	0.3		0.7
101	0.0	`PIP`	1.0	0.8					0.5	0.3
110	0.92	`DU`	0.66	0.7	0.1				0.3	**0.3**
		`POGIDLUS`	0.33	0.36				0.3		**0.06**
avg.h	0.27									

Figure 21: Game 'lexical coherence': maximum entropy at run nr. 51

Meaning	**h**	**Form**	%	**s**	**Agents**					
					0	**1**	**2**	**3**	**4**	**5**
000	1.25	POWI	0.56	3.41	0.1	0.3	0.7	1.0	1.0	0.3
		ZIWUKE	0.36	2.18		1.0			0.18	1.0
		FANE	0.06	0.4	0.3		0.1			
001	1.77	DA	0.37	2.29	0.9	1.0	0.09			0.3
		LUCGAGER	0.32	1.98			0.9	0.58		0.5
		KETADI	0.25	1.56	0.06	0.3		0.1	1.0	0.1
		ZOREP	0.04	0.3						0.3
010	0.91	NA	0.67	3.98	0.38	0.1	1.0	1.0	1.0	0.5
		GILCY	0.32	1.95	0.7	0.7		0.06		0.5
011	1.23	DYNUPY	0.67	3.22	0.9	0.45	0.26	0.9		0.7
		VOV	0.16	0.8		0.1	0.3		0.3	0.1
		HE	0.15	0.76		0.5	0.26			
100	1.28	FIP	0.65	3.97	0.53	0.3	1.0	1.0	0.13	1.0
		LA	0.19	1.2		0.3			0.9	
		HORIMI	0.15	0.94	0.26	0.5	0.18			
101	1.48	PIP	0.47	2.56	0.18	0.42	0.06		1.0	0.9
		NOG	0.33	1.8	0.5	0.5		0.5		0.3
		KAVSA	0.18	1.0			0.7			0.3
110	0.77	DU	0.77	3.76	1.0	0.7	0.06		1.0	1.0
		POGIDLUS	0.22	1.1			0.3	0.7		0.1
111	1.47	GEBE	0.43	2.05	0.3		0.5	1.0		0.25
		PE	0.41	1.95	0.3	0.7	0.1		0.15	0.7
		KAFEC	0.15	0.76	0.1		0.1	0.06	0.5	
avg.h	1.27									

Figure 22: Game 'lexical coherence': favourites coming up at run nr. 400

Meaning	**h**	**Form**	%	**s**	**Agents**					
					0	**1**	**2**	**3**	**4**	**5**
000	0.8	`POWI`	0.75	5.1	0.1	1.0	1.0	1.0	1.0	1.0
		`FANE`	0.24	1.67	1.0		0.49		0.18	
001	0.99	`KETADI`	0.54	4.16		1.0	1.0	0.85	1.0	0.3
		`LUCGAGER`	0.45	3.42	1.0		0.12	1.0	0.3	1.0
010	0.0	`GILCY`	1.0	6.0	1.0	1.0	1.0	1.0	1.0	1.0
011	0.38	`DYNUPY`	0.92	6.0	1.0	1.0	1.0	1.0	1.0	1.0
		`VOV`	0.07	0.48			0.48			
100	0.95	`FIP`	0.62	4.58	0.6	0.1	1.0	1.0	1.0	0.88
		`HORIMI`	0.37	2.76	0.62	1.0	0.6	0.17		0.36
101	0.89	`PIP`	0.69	4.92	0.32	1.0	0.6	1.0	1.0	1.0
		`NOG`	0.3	2.2	0.56	0.1	0.67	0.8	0.07	
110	0.0	`DU`	1.0	6.0	1.0	1.0	1.0	1.0	1.0	1.0
111	1.06	`KAFEC`	0.52	4.45	1.0	0.6	1.0	1.0	0.67	0.17
		`PE`	0.46	4.01		0.58	1.0	0.43	1.0	1.0
		`GEBE`	0.0	0.07			0.07			
avg.h	0.63									

Figure 23: Game 'lexical coherence': first stability at run nr. 800

Meaning	**h**	**Form**	%	**s**	**Agents**					
					0	**1**	**2**	**3**	**4**	**5**
000	0.0	`POWI`	1.0	6.0	1.0	1.0	1.0	1.0	1.0	1.0
001	0.97	`LUCGAGER`	0.59	4.81	0.96	0.25	1.0	1.0	1.0	0.6
		`KETADI`	0.4	3.28	0.21	1.0	0.78		0.67	0.6
010	0.0	`GILCY`	1.0	6.0	1.0	1.0	1.0	1.0	1.0	1.0
011	0.0	`DYNUPY`	1.0	6.0	1.0	1.0	1.0	1.0	1.0	1.0
100	0.07	`FIP`	0.99	6.0	1.0	1.0	1.0	1.0	1.0	1.0
		`HORIMI`	0.0	0.05				0.05		
101	0.0	`PIP`	1.0	6.0	1.0	1.0	1.0	1.0	1.0	1.0
110	0.0	`DU`	1.0	6.0	1.0	1.0	1.0	1.0	1.0	1.0
111	0.0	`KAFEC`	1.0	6.0	1.0	1.0	1.0	1.0	1.0	1.0
avg.h	0.13									

Figure 24: Game 'lexical coherence': the final lexicon at run nr. 1952

Meaning	**h**	**Form**	%	**s**	**Agents**					
					0	**1**	**2**	**3**	**4**	**5**
000	0.0	`POWI`	1.0	6.0	1.0	1.0	1.0	1.0	1.0	1.0
001	0.0	`KETADI`	1.0	6.0	1.0	1.0	1.0	1.0	1.0	1.0
010	0.0	`GILCY`	1.0	6.0	1.0	1.0	1.0	1.0	1.0	1.0
011	0.0	`DYNUPY`	1.0	6.0	1.0	1.0	1.0	1.0	1.0	1.0
100	0.0	`FIP`	1.0	6.0	1.0	1.0	1.0	1.0	1.0	1.0
101	0.0	`PIP`	1.0	6.0	1.0	1.0	1.0	1.0	1.0	1.0
110	0.0	`DU`	1.0	6.0	1.0	1.0	1.0	1.0	1.0	1.0
111	0.0	`KAFEC`	1.0	6.0	1.0	1.0	1.0	1.0	1.0	1.0
avg.h	0.0									

Figure 25: Game 'environmental influence': unstable rules after around 18000 runs

Meaning	h	Form	%	s	Agents 0	1	2	3
335	0.89	BU	0.692	0.9	0.6	0.2		0.1
		ZY	0.307	0.4			0.4	
424	0.979	SAD	0.583	0.7	0.3			0.4
		RE	0.416	0.5				0.5
445	0.954	MU	0.625	2.0	0.5	0.9	0.5	0.1
		BIRO	0.375	1.2	0.2	0.3	0.1	0.6
525	0.0	SOVA	1.0	0.9	0.9			
544	0.0	SY	1.0	2.699	0.5	0.7	0.9	0.6
625	0.965	RECPAZCIT	0.608	1.4	0.1	0.4		0.9
		GU	0.391	0.9			0.8	0.1
626	1.509	WI	0.428	0.9		0.1	0.8	
		LUWEZ	0.38	0.8		0.8		
		HAGGE	0.19	0.4		0.4		
644	0.0	GUF	1.0	0.8			0.8	
735	1.553	WAS	0.38	0.8	0.5		0.3	
		MOW	0.38	0.8	0.1		0.3	0.4
		TUK	0.238	0.5		0.5		
736	0.65	VIL	0.833	2.5		0.8	0.8	0.9
		WEGY	0.166	0.5		0.1	0.4	

Figure 26: Game 'Compositionality without environmental influence': strong rules after 2100 runs

Meaning	**h**	**Form**	%	**s**	**Agents** **0**	**1**
001	0.0	`hyflo`	1.0	2.0	1.0	1.0
010	0.0	`nuca`	1.0	2.0	1.0	1.0
011	0.0	`syb`	1.0	2.0	1.0	1.0
100	0.0	`ki`	1.0	2.0	1.0	1.0
101	0.0	`gi`	1.0	2.0	1.0	1.0
110	0.0	`zapy`	1.0	2.0	1.0	1.0
111	0.0	`mevdycduf`	1.0	2.0	1.0	1.0

Figure 27: Game 'Compositionality with environmental influence': strong rules after 2100 runs

Meaning	**h**	**Form**	%	**s**	**Agents 0**	**Agents 1**
$3x5$	0.0	`ry`	1.0	2.0	1.0	1.0
434	0.0	`ga`	1.0	2.0	1.0	1.0
435	0.0	`hozu`	1.0	1.9	1.0	0.9
534	0.0	`pyzvoz`	1.0	2.0	1.0	1.0
535	0.0	`buprec`	1.0	2.0	1.0	1.0
536	0.0	`kemi`	1.0	2.0	1.0	1.0
$5x5$	0.918	`vi`	0.666	2.0	1.0	1.0
		`ru`	0.333	1.0	1.0	
634	0.0	`manam`	1.0	2.0	1.0	1.0
635	0.0	`demif`	1.0	2.0	1.0	1.0
636	0.0	`rapeg`	1.0	1.8	1.0	0.8
645	0.0	`cug`	1.0	1.9	1.0	0.9
$x2x$	0.924	`cu`	0.66	1.75	1.0	0.75
		`ruc`	0.339	0.9		0.9