



Autotelic Principle: the role of intrinsic motivation in the emergence and development of artificial language

Miquel Cornudella Gaya

► To cite this version:

Miquel Cornudella Gaya. Autotelic Principle: the role of intrinsic motivation in the emergence and development of artificial language. Linguistics. Université de recherche Paris Sciences et Lettres, 2017. English. NNT : . tel-01765234v1

HAL Id: tel-01765234

<https://hal.science/tel-01765234v1>

Submitted on 12 Apr 2018 (v1), last revised 7 Mar 2019 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT

de l'Université de recherche Paris Sciences et Lettres
PSL Research University

Préparée à l'École Normale Supérieure

Autotelic Principle: the role of intrinsic motivation in the emergence and development of artificial language

Autotelic Principle : le rôle de la motivation dans l'apparition et le développement d'un langage artificiel

École doctorale n°540

ECOLE DOCTORALE TRANSDISCIPLINAIRE LETTRES/SCIENCES

Spécialité SCIENCES DU LANGAGE

COMPOSITION DU JURY :

M. OUDEYER Pierre-Yves
DR INRIA, Président du Jury

M. DOMINEY Peter Ford
DR CNRS, Rapporteur

M. DE BOER Bart
Professeur, Vrije Universiteit
Brussel, Rapporteur

Mme. BOUVERET Myriam
Maître de conférences HDR
Université de Rouen, Examinatrice

M. VAN TRIJP Remi
Chercheur, Sony Computer Science
Laboratories Paris, Encadrant

M. POIBEAU Thierry
DR LATTICE-CNRS, Directeur

Soutenue par
Miquel Cornudella Gaya
le 19 Décembre 2017

Dirigée par **Thierry Poibeau**



Abstract

This thesis studies the role of intrinsic motivation in the emergence and development of communicative systems in populations of artificial agents. To be more specific, our goal is to explore how populations of agents can use a particular motivation system called autotelic principle to regulate their language development and the resulting dynamics at the population level.

To achieve this, we first propose a concrete implementation of the autotelic principle. The core of this system is based on the balance between challenges, tasks to be done to achieve a goal, and skills, the abilities the system can employ to accomplish the different tasks. The relation between the two elements is not steady but regularly becomes destabilised when new skills are learned, which allows the system to attempt challenges of increasing complexity. Then, we test the usefulness of the autotelic principle in a series of language evolution experiments. In the first set of experiments, a population of artificial agents should develop a language to refer to objects with discrete values. These experiments focus on how unambiguous communicative systems can emerge when the autotelic principle is employed to scaffold language development into stages of increasing difficulty. In the second set of experiments, agents should agree on a language to communicate with about colour samples. In this part, we explore how the motivation system can regulate the linguistic complexity of interactions for a continuous domain and examine the value of the autotelic principle as a mechanism to control several language strategies simultaneously.

To summarise, we have shown through our work that the autotelic principle can be used as a general mechanism to regulate complexity in language emergence in an autonomous way for discrete and continuous domains.

Résumé

Dans cette thèse nous étudions le rôle de la motivation intrinsèque dans l'émergence et le développement des systèmes communicationnels. Notre objectif est d'explorer comment des populations d'agents artificiels peuvent utiliser un système de motivation computationnel particulier, appelé l'*autotelic principle*, pour réguler leur développement linguistique et les dynamiques qui en résultent au niveau de la population.

Nous proposons d'abord une mise en œuvre concrète de l'autotelic principe. Le noyau de ce système repose sur l'équilibre des défis, des tâches à accomplir afin d'atteindre un objectif, et des compétences, les capacités que le système peut utiliser pour accomplir les différentes tâches. La relation entre les deux éléments n'est pas stable mais se déstabilise régulièrement lorsque de nouvelles compétences sont acquises, ce qui permet au système de tenter des défis de plus grande complexité. Ensuite, nous testons l'utilité de ce système de motivation dans une série d'expériences sur l'évolution du langage. Dans le premier ensemble d'expériences, une population d'agents artificiels doit développer une langue pour se référer à des objets ayant des caractéristiques discrètes. Ces expériences se concentrent sur la façon dont les systèmes communicatifs non ambigus peuvent émerger lorsque l'autotelic principe est utilisé pour réguler le développement du langage en étapes de difficulté croissante. Dans le deuxième ensemble d'expériences, les agents doivent créer un langage artificiel pour communiquer sur des couleurs. Dans cette partie, on explore comment le système de motivation peut contrôler la complexité linguistique des interactions pour un domaine continu et on examine aussi la validité de l'autotelic principe en tant que mécanisme permettant de réguler simultanément plusieurs stratégies linguistiques de difficulté similaire.

En résumé, nous avons démontré à travers de notre travail que l'autotelic principe peut être utilisé comme un mécanisme général pour réguler la complexité du langage développé de manière autonome en domaines discrets et continus.

A la meva mare

Acknowledgements

First of all, I want to thank Luc Steels, Thierry Poibeau and Remi van Trijp for the opportunity they gave me. For the last years I could work in a privileged environment and meet outstanding researchers during my time as a member of the language team at Sony CSL Paris.

I want to thank my thesis director, Thierry, for his continuous support and guidance during the writing of this manuscript. I'm particularly thankful to Luc who, using one of his analogies, passed me the ball of the autotelic principle and let me play with it. None of what is written here would have been possible without his ground breaking work. Special thanks go to Pieter Wellens and Joris Bleys, whose previous research on the autotelic principle and the colour domain have been an inspiration and a basis for this thesis.

I also want to thank the members of my PhD jury, Prof. Dr. Peter Ford Dominey, Prof. Dr. Bart de Boer, Prof. Dr. Pierre-Yves Oudeyer, Prof. Dr. Myriam Bouveret, Dr. Remi van Trijp and Prof. Dr. Thierry Poibeau as their comments and suggestions improved the quality of this manuscript significantly.

I'm undoubtedly grateful to all the people that worked in the Babel, the framework used in the experiments in this thesis, in the past. As Isaac Newton famously said: "If I have seen further, it is by standing on the shoulders of giants". Thanks a lot to all my colleagues in Barcelona, Brussels, Paris and Tokyo: Emília, Andrea, Maria, Jorge, Katrien, Pieter, Yana, Aurélien, Michael, Paul, Remi and Mica. Among them, I had the chance to share the office for nearly three years with an extremely talented researcher: Paul, it has been a pleasure! I also want to thank all the people at Sony CSL, in particular the former director François Pachet and Sophie Boucher, who has always been there ready to help me.

A special thanks to my friends in Paris: Dani, Timotée, Marco, Alba, Emily, all the people from Cerise and to my parisian wife, Emmanuel. I am extremely lucky to have met Nicolas, Jeanne, Élise, Françoise and Pierre, who have become part of my family.

Voldria acabar amb unes paraules d'agraïment per a la gent que ha viscut aquesta etapa des de la distància. Tinc la sort d'estar envoltat de gent m'ajuda i recolza a cada pas que faig. A la meua família i amics¹, moltes gràcies!

The research reported in this thesis has been financially supported by Sony Computer Science Laboratories Paris and by a CIFRE grant (agreement no. 2013/0730).

¹Menció especial pel Chris per ajudar-me a corregir l'anglès macarrònic.

Contents

1	Introduction	1
1.1	Situating this work: language evolution	2
1.2	Selectionist theory of language evolution	4
1.2.1	Language as a Complex Adaptive System	4
1.2.2	Language systems and language strategies	5
1.2.3	Linguistic selection	5
1.2.4	Self-organisation of language systems	6
1.2.5	Semiotic cycle: the importance of context	7
1.3	Managing complexity	8
1.3.1	Motivation as a force	9
1.3.2	Computational approaches to motivation	9
1.3.3	Linguistic complexity	10
1.4	Main hypothesis	11
1.5	Objectives	11
1.6	Structure of the thesis	11
I	Technical Background	15
2	Language games and technical background	17
2.1	Language games	17
2.2	Incremental Recruitment Language	19
2.2.1	Meaning as a network	19
2.2.2	Execution of an IRL-network	20
2.2.3	Building meaning	21
2.3	Fluid Construction Grammar	21
2.3.1	Representing linguistic structures	21
2.3.2	Transient Structures	23
2.3.3	Constructions	23
2.3.4	Construction application process	25
2.3.5	An example of how FCG works	27
2.4	Summary	29

II	The Autotelic Principle	31
3	History of Motivation: psychological and AI approaches	33
3.1	Psychological theories on motivation	33
3.1.1	Drive-reduction theory	34
3.1.2	Operant conditioning	35
3.1.3	Reconsidering motivation	36
3.1.4	Motivation and balance	37
3.1.5	Personal causation, or the opposition of intrinsic and extrinsic motivation	39
3.1.6	Flow Theory	39
3.1.7	Self-determination theory	41
3.2	AI approaches to motivation	43
3.2.1	Computation Reinforcement Learning	43
3.2.2	Classifying computational motivation systems	45
3.2.3	From Predictor Error to Compression Driven Progress	46
3.2.4	Intrinsically Motivated Reinforcement Learning	47
3.2.5	Temporal Difference Competence-Based Intrinsic Motivation	48
3.2.6	Intelligent Adaptive Curiosity	48
3.2.7	Self-Adaptive Goal Generation R-IAC	49
3.2.8	The Autotelic Principle	50
3.3	Summary	51
4	The Autotelic Principle	53
4.1	The notion of challenge	54
4.2	Architecture of the autotelic principle	55
4.2.1	Agent as a set of components	56
4.3	Development of skills	56
4.4	Deciding what to do next	57
4.4.1	Generating new challenges	58
4.5	Updating the internal state	58
4.6	Initialisation of the system	62
4.6.1	Deciding the current challenge	63
4.7	Summary	64
III	Experiments on Language Emergence	65
5	Experiments on the discrete domain	67
5.1	Description of the experiment	68
5.1.1	World and context	68
5.1.2	Multi-word guessing game	69
5.1.3	Diagnostics and repairs	70
5.1.4	Alignment	73
5.2	Baseline experiment	75
5.2.1	Experimental results	76
5.3	Addition of the autotelic principle	77

CONTENTS

5.3.1	Experimental results	79
5.4	Incremental learning of skills	81
5.5	Analysing the impact of internal evaluation	83
5.6	Increasing the challenge dimensions	86
5.6.1	Experimental results	87
5.7	Contribution to previous experiments	90
5.8	Summary	90
6	Experiments on a continuous domain: a case study on colour	93
6.1	The domain of colour	93
6.2	Language strategies for the domain of colour	95
6.2.1	Basic colour strategy	95
6.2.2	Graded membership strategy	95
6.2.3	Colour combination strategy	96
6.2.4	Lightness strategy	97
6.2.5	Chromaticity strategy	97
6.2.6	Colour strategies as a network of meaning predicates	98
6.3	Description of the experiments	99
6.3.1	World and context	99
6.3.2	Multi-word guessing game	100
6.3.3	Diagnostics and repairs	100
6.3.4	Alignment	104
6.4	Experiment 1: regulating linguistic complexity	105
6.4.1	Context	107
6.4.2	Experimental results	108
6.5	Experiment 2: Managing multiple challenges	110
6.5.1	Context	112
6.5.2	Misestimation of the speaker's challenge	113
6.5.3	Experimental results	114
6.6	Experiment 3: Simulating an embodied experiment	118
6.6.1	Perceptual deviation	118
6.6.2	Experimental results	119
6.7	Summary	122
IV	Conclusion and Future Directions	125
7	Conclusion	127
7.1	Overview of publications	128
8	Future Directions	131
8.1	Open-Ended Learning Environments	131
8.1.1	Multi dimensional environments	132
8.1.2	Dynamic challenge configuration	133
8.1.3	Generation of new language strategies	135
8.2	Other applications	136
8.2.1	Intelligent tutoring systems	137

8.2.2	Developmental robotics	137
8.2.3	Character modelling in video games	138
8.3	Summary	139
A	Example of execution of an IRL-network	141
B	Parametrization of the autotelic principle	147
	Résumé long de la thèse en français	159
	Introduction	159
	Jeux de langage (Language Games)	160
	Implémentation du cycle sémiotique	162
	Autotelic principle	164
	Définition des notions de défi et de performance	166
	Développement des compétences	166
	Phases du système	167
	Génération des nouveaux défis	167
	Actualisation de l'état interne	168
	Initialisation	170
	Déterminer le défi actuel	170
	Expériences	171
	Expériences dans le domaine discret	172
	Diagnostics, réparations et alignement	173
	Expérience de base	173
	Intégration du principe d'auto-motivation dans l'architecture	176
	Acquisition progressive de compétences	179
	Expériences dans un domaine continu : les couleurs	181
	Stratégies linguistiques pour le domaine de la couleur	182
	Stratégies linguistiques en tant que réseaux de contraintes sémantiques	184
	Alignement des prototypes	186
	Régulation de la complexité linguistique	187
	Gestion de défis multiples	189
	Conclusion	194
	Bibliography	195

Chapter 1

Introduction

This thesis studies the role of intrinsic motivation in the emergence and development of communicative systems in populations of artificial agents. To be more specific, it studies how populations of agents can use a motivation system to regulate their language development and the resulting dynamics at the population level. The core of the thesis consists on linking two different fields of study: evolutionary linguistics and computational approaches to motivation.

Evolutionary linguistics tries to explain the evolution of language as a process of cultural negotiation. Researchers in this field study the emergence and change of human-like communicative systems in a controlled environment through recurrent peer-to-peer interactions in a population of artificial agents [116, 129]. The different communicative systems presented in this thesis follow the language game paradigm [125, 128], based on the notion of language games first introduced by Wittgenstein [160], and assume that language is a complex adaptative system.

Researchers in evolutionary linguistics have used language games to tackle different language phenomena. It was first applied to the study of self organisation of vocabularies [125, 126, 127, 108], but during the last two decades it has shed light on a variety of domains such as colour [15, 24], space [124], quantifiers [92], flexible word meaning [157, 156], case systems [154], agreement [19, 94], aspect [48] or syntax [37, 46]. Despite the fact that this approach has successfully provided insights for these domains, in most of these experiments the control of the complexity relies on the experimenter, who usually explores a certain domain in different experimental setups.

How could agents manage the complexity of experiments themselves? Researchers in artificial intelligence (AI) and developmental robotics, inspired by psychological studies on motivation, have proposed different models that allow agents to deal with the complexity of their actions autonomously. The idea behind it consists of providing agents with mechanisms that evaluate the outcome of an action and use this feedback to decide on the next action to execute. These mechanisms let agents evaluate how well an action is performed, for example by comparing the expected and the actual result, and use this information to predict the behaviour of actions, decide on next actions to perform, explore the space of possible actions or improve their performance on a specific one.

This work contributes to the understanding of the role of intrinsic motivation

in the emergence and development of language in two ways. Firstly, it presents a detailed description of a computational model of motivation called the Autotelic Principle. Secondly, it tests its role in language development by applying this model to language games on different language domains. The thesis argues that complexity in language games can be managed by the artificial agents that take part in them and demonstrates that the autotelic principle can be used as a general mechanism in language games.

The work presented in this thesis is the result of a CIFRE fellowship (agreement no. 2013/0730) and has been developed with the authorization and agreement of Sony Computer Science Laboratory Paris. Sony wanted to re-investigate the notion of the autotelic principle as this mechanism has important relevant potential applications for the company, although the goal of the thesis was not to investigate these applications directly. Previous work by Luc Steels and colleagues [131, 145] were thus important but not fully satisfactory as (a) they did not explore the problem of dealing with several challenges simultaneously, (b) they did not integrate it in a continuous domain, which resembles better the kind of input a robotic agent may face, and (c) their implementation was not operational anymore. Part of my research objectives were to build on this initial work to make it really operational and take into account a number of features that were not addressed.

This chapter continues with an introduction to language evolution (Section 1.1) and to the specific approach used in this thesis: the *selectionist theory to language evolution* (Section 1.2). It then follows with a brief introduction on complexity management (Section 1.3). The chapter ends explaining the main hypothesis (Section 1.4), objectives (Section 1.5) and the structure of the thesis (Section 1.6).

1.1 Situating this work: language evolution

The origins and evolution of human language are difficult questions to address from a scientific point of view. The main reason for this is that there is no direct evidence of when it started or what it looked like. The lack of direct evidence has been a serious obstacle for this topic and led it to an ostracism in linguistics for more than a century. For instance, *la Société de Linguistique de Paris* banned the study of language origins in the second article of its statutes from 1866: “ART. 2. - La Société n’admet aucune communication concernant, soit l’origine du langage soit la création d’une langue universelle.”¹.

The question of the origins of human languages was raised again during the 1950s when ideas such as mass comparison [55] or universal grammar [29] appeared, but it is not until the late 1980s that it was established as a subfield of psycholinguistics. Nowadays, research on the origins and evolution of language is a highly interdisciplinary field that during the last 25 years has progressively gained attention from linguists, biologists, anthropologists and psychologists.

¹ART. 2. - The Society doesn’t admit any communication regarding the origins of language or the creation of a universal language.

As the interdisciplinary field that it is, a theory of language evolution should provide an explanation for the biological, social and cultural perspectives involved in human language evolution [137]. Researchers that take the biological perspective try to understand the underlying biological capacities required for human language. Evolutionary biologists work to answer which neurobiological structures are needed for language, what are its genetic basis and when and how these bases appeared. They tackle these research questions employing the Darwinian approach to the human language.

Language cannot be disconnected from its *social* component. Language is a social phenomenon, a mechanism of symbolic communication that is learned and used in a context of social interaction and cooperation. Furthermore, its usage varies depending on the social setting of the interaction. What are the ecological changes in earlier human societies that entailed the necessity of symbolic communication? Which other mechanisms, such as joint attention or trust, were necessary for this to happen? These questions can be addressed from an anthropological approach, which may consist in studying hunter-gatherer societies or comparing the social structures and social cooperation between humans and our closest species.

Still, human languages are not fixed but in a state of continuous change. Languages exhibit cross-cultural variation at different levels, from morphemes to syntactic and semantic structures. These changes stimulate the emergence of new linguistic items and the loss of others. The cultural approach to the evolution of language tries to identify and analyse the mechanisms that trigger these changes. Once these cultural processes have been identified researchers can explain certain phenomena recognised in languages, either observed or ongoing changes, and use the same cultural mechanisms to understand the origins of language.

These different evolutionary forces are strongly interconnected. For instance, language was crucial for the origin of human society [65] as much as social groups were necessary for the emergence of language. Also, these forces get involved into what Steels [137] identifies as *spiral forces*: an increase in complexity of one force requires a complexity raise in the other forces. For instance, an increase of social complexity requires an increase in both language complexity and brain capacities.

The time span of these changes, though, differs between the different forces. We know that Germanic languages, which include German, English, Dutch, Danish, Icelandic or Yiddish, are all descendants from Proto-Germanic, which researchers situate around 500 BC [73]. Another example can be found in Romance languages, which evolved from Vulgar Latin in a shorter time span [2]. Changes in languages occur every day. Day after day new words are incorporated into human languages, due to the rise of new concepts and fields. For instance, computer-related terms did not exist a century ago. Another example is the emergence of a communicating system of a particular social group. A good example is *Louchébem*, an argot used by butchers in Lyon and Paris that consist of camouflaging existing words according to a set of rules [109]: in *Louchébem* the word *couteau* (French for *knife*) becomes *louteaucé*, or the name of the argot (*Louchébem*) comes from applying the same rules to the word *boucher* (French for *butcher*).

The experiments presented as case studies for this thesis focus on the cultural mechanisms that can lead to the emergence of a shared language, not in its biological or social basis. It follows the *selectionist theory of language evolution* [137], by means of applying the same principles of *natural selection* used in biology to explain evolution into the language domain.

1.2 Selectionist theory of language evolution

Natural Selection states that living organisms exist in environments in continuous change. These organisms have a set of traits that allow them to prevail in the environment. But by the changeable conditions of the environment, a very advantageous characteristic in a specific context can become useless or even harmful if the environmental conditions change. Evolution solves this problem by introducing a small quantity of variants and *selecting* those variants that provide high survival in the environment. If a new variant that increases the chances of survival can be passed to the offspring, it will be *replicated* among the population in future generations.

Selectionism, therefore, consists of (a) the generation of variants, generally from an already existing variant, and (b) a process of choice among these variants according to selection criteria. As it happens continuously, a selected variant can be used to generate new variants, which will be chosen again according to the same selection criteria, entering into a self-enforcing loop. The selectionist theory of language evolution maps this selectionist process to the linguistic domain in order to study human language evolution from a cultural perspective. What can the variants be and how can they be introduced into a language? How are these variants selected? In order to answer these questions, it is important first to clarify what language is.

1.2.1 Language as a Complex Adaptive System

The view that language is a complex adaptive system has gained support during the last two decades [26, 128, 14, 42, 5, 138]. *Complex* as the set of different cognitive processes that allow humans to learn, use and adapt language are dependent on one another and *adaptive* because individuals shape their language based on these cognitive processes but also on their past and current interactions. A shared language in a population is, therefore, the result of the interactions of all its individuals.

This cognitive-functional approach to language is at the core of *cognitive linguistics* [34], which studies language investigating the relation between semantics, pragmatics, conceptualisation, syntax and language use. The principal approach to grammar development in cognitive linguistics is *Construction grammar* [44, 49, 70]. In construction grammar, all knowledge of a language is made of a collection of linguistic conventions that pair form and meaning, called *constructions*. Constructions range from specific units such as words or morphemes to more general ones, as phrasal patterns. The learning and creation of constructions are driven by cognitive mechanisms and language use, which is known as

the *usage-based theory* of language learning [61].

1.2.2 Language systems and language strategies

As there is no central control on language, an individual can only discover what the best way to communicate is by interacting with other individuals of the population. An individual doesn't know the way to communicate about new objects or conceptualisations, or if a hearer will be able to understand the message she is conveying. In other words, there is no individual with a complete view of the communicative system of the population, which is built in a bottom-up fashion.

Through interactions, individuals learn *language systems*, which capture the regularities and systematicity of language domains. Language systems are specific realisations of *language strategies*, which can be described as a particular procedure to express one subarea of meaning. A language strategy can lead to different language systems. For example, if we consider the colour domain we can appreciate that a common language strategy consists in dividing the colour space into different basic colour terms. Although this language strategy is used by all languages, its particular realisation into language systems differs among them. For instance, Russian has two different terms for the English term *blue*, *sinij* and *goluboj*, which can be translated as "dark blue" and "light blue" [101].

A language system is divided into the *conceptual system*, the semantic distinctions of a language system used for conceptualisation, and the *linguistic system*, the lexical or grammatical structures used to convert a specific conceptualisation into an utterance. Taking the previous example on colour, in a language system of basic colours terms (a) the conceptual system corresponds to the semantic distinctions that separate a colour space into categories (e.g. eleven for English) and (b) the linguistic system consists of the terms used to express these categories.

1.2.3 Linguistic selection

Assuming that language use in the form of interactions between individuals of a population is part of the core for language learning points us to the selectionist force in language: *communicative success*. Communicative success occurs when the speaker realises his or her communicative goal. For instance, if the speaker wants to draw the attention of the hearer to a certain object in the context, the interaction will be a communicative success only when the hearer has identified the object and showed to the speaker that he or she understood the message (for instance, by pointing to the intended object).

Several features influence communicative success, acting as selection criteria for language systems and language strategies. For instance, one of these features is *expressive adequacy*, which is determined by the set of conceptual distinctions and the linguistic items to refer to these conceptual differences that a language system has. Another feature is *cognitive effort*: how hard it is to come up with a satisfactory conceptualisation, construct a sentence or comprehend it has an impact on the success of a language system.

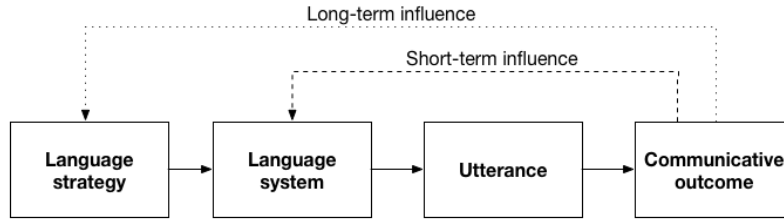


Figure 1.1: Visual representation of the relation between language strategies and language systems (taken from [137]). Language systems are specific realisations of language strategies, which are particular procedures to express one subarea of meaning. The communicative outcome has an impact on the language system used, and in the language strategies in the long-term, selecting language variants with a better outcome.

Variants, in the form of new words, categories or grammar, are introduced by language users when a language system does not allow them to convey a certain meaning. Among these language variants, those that lead to a higher communicative success have a higher probability to spread in the population. As linguistic selectionism occurs in every interaction, language enters into a self-enforcing loop, similar to the one observed in biology. The same linguistic selection forces that have spread a language variant in the population will again select new variants generated from that variant, therefore continuously affecting and shaping the language system and its associated language strategy in the long-term (Figure 1.1).

1.2.4 Self-organisation of language systems

So far the theory accounts for how variants are introduced, namely for expressing a meaning that the current language system of an individual cannot express, and for how these variants are selected using communicative success as a selectionist force. We still need to understand how these variants are spread over the population.

In the previous subsection it is mentioned that individuals can *create* new variants in order to express a meaning when their current language system cannot refer to that meaning. In order to spread these new words and concepts in the population, individuals can also *learn* them. A new variant, introduced by an individual, will be more or less spread to the population depending on how useful it is (if it allows to reference to a specific meaning successfully) and, in consequence, increase the communicative success.

With these cognitive mechanisms, each individual keeps a different inventory of conceptualisations and constructions for a language system, sometimes containing competing constructions: either having different ways to express the same meaning or having constructions that refer to multiple meanings. How can a population create cohesion in its language system? This process is done with another cognitive mechanism named *alignment*. Alignment consists of bringing

the language system and language strategies of both speaker and hearer after each interaction closer. These dynamics not only affect interlocutors in a short-term period, as psychological studies have shown for the alignment of linguistic conventions [162, 47], but it also has an effect on the long-term, shaping language systems and their related language strategies.

This results in the *self-organisation* of language systems, as a language system is a distributed system formed by multiple elements that behave autonomously. These elements are influenced by (a) external factors, communicative success and its selection criteria, and (b) the behaviour of the other elements in the system, in particular variants introduced by individuals.

Alignment not only provokes that the more a particular variant is used by an individual, the more likely other individuals in the population will employ it, but also means that the more a particular variant is used, the fewer its competing variants will be employed. As a consequence, a self-enforcing loop is produced in the language system, as the more language users are aligned, the more the resulting language system is coherent. In the same way, the more consistent a language system is, the stronger disposition towards alignment it has.

1.2.5 Semiotic cycle: the importance of context

There is a last element that plays a role in language emergence: the external ecological conditions of the interaction, that is, the *context* in which interactions between two or more individuals take place. Spoken language occurs within a physical context, and the configuration of this context influences which language strategies are used, as the expressivity of different language strategies depends on the context in which interactions occur, shaping the language system that will emerge. Therefore, a theory of language evolution should also take into account the ecological conditions in which interactions happen.

This can be nicely illustrated with examples from spatial language, as there is variation in how absolute frames of reference are used across languages [75]. For example, some languages in Nepal and Meso America, such as *Tzeltal*, due to the nature of their ecological situation, employ concepts such as 'uphill' and 'downhill' as landmarks to specify directions. Similarly, other languages use the river drainage as a landmark, making use of concepts like 'upstream' and 'downstream', as has been observed in languages from Arnhem Land and Alaska.

The existing relationship between the speaker, the hearer and the context of an interaction is captured in the *semiotic cycle* (Figure 1.2). Three different levels of processing are involved: sensorimotor, conceptual and linguistic. Both speaker and hearer perceive the context through their sensorimotor systems, building a world model. The speaker uses this model and her communicative goal to *conceptualise* a meaning or semantic structure. This meaning representation acts as the input of the linguistic system that *formulates* it into an utterance.

The hearer *comprehends* the utterance and reconstructs a semantic structure. This structure, together with her world model, is used to *interpret* the input utterance, that is grounding the reconstructed meaning representation to the actual context. Depending on the interpretation, the hearer can perform an

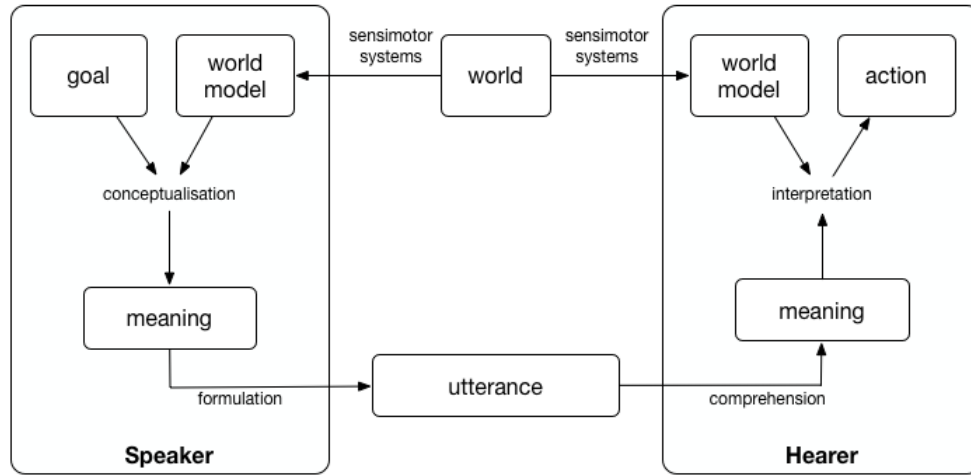


Figure 1.2: The semiotic cycle illustrates the relation between the speaker (left) and hearer (right) of an interaction and the context in which it takes place (adapted from [130]). It involves the sensorimotor systems and the conceptual and linguistic processing levels.

action, which allows the speaker to detect if her communicative goal has been achieved.

Notice that the semiotic cycle requires (a) *joint attention* to the interlocutors, as if it will not be the case their world models would differ and consequently the conceptualisation and interpretation processes would unlikely result in a successful communication, and that (b) the relation between communicative goals, meaning structures and formulated utterances is not unique. Even in interactions with simple contexts, the same communicative goal can be conceptualised into different meaning representations. Similarly, a particular semantic structure can be expressed using different utterances, or an utterance can lead to multiple interpretations when the language system of an individual is not completely aligned. Some conceptualisations and utterances have a higher chance of achieving the communicative goal, depending on the particular language systems of the interlocutors and the context in which the interaction takes place.

This is a particularly interesting problem in language emergence, as individuals want to maximise the odds of being understood while they continue to develop and align their language systems. How can individuals manage the dichotomy between maximising communicative success while continuously improving their expressivity? In other words, how do humans manage language complexity?

1.3 Managing complexity

Researchers have been looking at different mechanisms that humans adopt in order to deal with complexity. Developmental psychologists have studied the *scaffolding* in child development and how this affects their acquisition of abilities. They have identified two key features in developmental scaffolding: Firstly,

incremental learning usually occurs in stages, that is, points of stability (see [20]) or “fitness barriers” in developmental performance. Secondly, the reduction of points of stability makes available the development of new skills that, in turn, become self-scaffolding of later development performances [159]. In other words, the development of skills or capacities at a certain stage builds on the skills developed by the child in earlier stages: *id est*, the development of more sophisticated skills depends on skills learned before.

In the case of child development sometimes it is an external person, for instance the mother, who restricts and simplifies the context of the interaction to facilitate the development of certain abilities. For example, she can set up a safer context in which to develop the ability to walk to incite this skill. But sometimes the scaffolding is made by the child herself: some psychologists have shown that children choose to focus their attention on particular features which are determined by the context when the learning of new words for objects occurs [117]. Other researchers point to the fact that infants maintain intermediate rates of information attention when paying attention to their environment, avoiding too complicated or too simple stimulus [72].

1.3.1 Motivation as a force

Scaffolded learning can, therefore, respond to external or internal forces, but this does not provide any answer as to what induces humans to learn skills or abilities. Psychologists have extensively studied the role of *motivation* since the 1930s [62, 115, 158, 52], which can be defined as “to be moved to do something” [99]. Researchers have focused not only on differences in motivation levels between individuals, *why* some people are more motivated than others, but also on different types of motivation.

Regarding the goals to perform an action, that is, *what* moves us to do something, it is commonly accepted that motivation can be subdivided into two classes. On the one hand *extrinsic motivation*, when external forces influence the motivated behaviour, and on the other hand *intrinsic motivation*, when an activity is performed because it is inherently enjoyable or interesting. There are several reasons why an activity may result in interest to an individual, such as an intrinsic search for novelties or challenges, a willing to explore or discover new abilities or simply to exercise her capacities.

1.3.2 Computational approaches to motivation

AI researchers and developmental roboticists have shown a growing interest in motivation and, in particular, intrinsic motivation [3, 13, 91, 79]. They use insights from psychology to come up with mechanisms to manage active learning and spontaneous exploration, proposing different models to allow artificial agents to autonomously develop their skills while carefully managing the complexity of their actions (see [7] for an overview). How do these models work?

The main idea is to use the outcome of actions as feedback for the agents, providing them with mechanisms to perceive the result of their actions. This information, together with their experience (built from all previous actions),

permits agents to judge how well an action has been executed and decide on which actions they should perform next. Agents can employ this information to predict the behaviour of actions, explore possible actions or focus on a particular action in order to ameliorate their performance. The latter is extremely relevant for this thesis, as computational models of motivation are related to autonomous skill development. It not only allows agents to discover abilities to be acquired but also provides mechanisms to help agents decide which ability to train next. Moreover, these models often rely on a cumulative acquisition of a repertoire of skills, similar to what has been observed in developmental scaffolding.

In this thesis, a particular computational model of intrinsic motivation, called *Autotelic Principle*, is studied. This model is inspired by the theory of Flow proposed by Csikszentmihályi [36], who studied intrinsic motivation and found an existing relationship between *challenge*, a particular activity to be done, and *skills*, the abilities that an individual has to confront that task. He observed that unequal configurations of these two elements lead to mental states such as boredom or anxiety. Interestingly, the balance between both challenge and skills, which he identifies as the optimal state of experience or *flow*, provides the best conditions to learn and develop new skills. People try to stay in flow state but the fact that it is in continuous change, as skills evolve over time, triggers them to become self-motivated. The term *autotelic principle* was first used in the context of learning by More and Anderson [85].

The autotelic principle was conceived by Steels [131, 133] and applies the central idea of the theory of Flow (i.e., the relationship between skills and challenges) to allow artificial agents to autonomously manage the complexity of their actions in order to regulate their development. An earlier implementation of this system was previously used in a language evolution experiment by Steels and Wellens [145]. This thesis extends their work by providing a detailed implementation of the autotelic principle redesigned for language interactions and present a collection of experiments to further analyse its impact in the development of communication systems.

1.3.3 Linguistic complexity

This model will be used by a population of agents to regulate the complexity in the emergence and development of language. It is, therefore, necessary to define what complexity means in language development. Different researchers have addressed this question and have proposed models identifying different steps or stages in language evolution [66, 132]. All these models share the assumption that language started in a stage without no grammar and progressively gained complexity up to the grammar systems that human languages exhibit today.

The increase of linguistic complexity adopted in this thesis corresponds to the three stages presented in [121]. Spranger identified relevant linguistic stages for language evolution experiments in the model proposed by [132]: (a) *single-word utterances*, when all the conceptual information is conveyed using one word, (b) *multiple-word utterances*, when different words are used to express information of several semantic categories or referents in the context, and (c) *grammatical utter-*

ances, when grammar is used to disambiguate different semantic interpretations of an utterance.

1.4 Main hypothesis

This thesis explores the role of intrinsic motivation in the emergence and development of artificial language systems. The hypothesis of this thesis is that artificial agents are able to successfully manage the development of successive stages of a language system by regulating the complexity of their interactions with a particular computational model of motivation called the autotelic principle. This hypothesis is then validated in computational experiments where a population of agents collaborate to create language systems for both discrete and continuous domains.

1.5 Objectives

The objectives of this thesis are twofold:

1. Firstly, to present a complete description of a computational model of motivation called the autotelic principle redesigned for language evolution experiments, explaining its different components and providing detailed information about how it has been implemented.
2. Secondly, to demonstrate that the autotelic principle can be used as a general mechanism to regulate complexity in language emergence by testing it in experiments on different language domains.

The work presented in this thesis is inspired on Steels and Wellens [145] and aims to extend their work in order to explore in detail the impact the autotelic system has as a mechanism to regulate the linguistic complexity in language evolution experiments. Section 5.7 enumerates the contributions made in the discrete domain in comparison to the work by Steels and Wellens.

In order to provide evidence for the second objective of the thesis, the set of experiments carried out (a) study how agents can use the motivation system to successfully regulate the complexity of linguistic interactions in several stages in an autonomous way, (b) test the autotelic principle in multiple challenge dimensions on discrete and continuous domains and (c) integrate the creation and learning of several conceptual prototypes to these experiments and explore the impact of perceptual deviation in the resulting communicative system.

1.6 Structure of the thesis

The thesis has been organised into four parts. The first part (Chapter 2) focuses on the approach taken to study the emergence and development of language in a population of artificial agents and how it has been implemented. Among the different methodologies proposed to perform computer simulations of language

evolution I have chosen to use *language games*, as it takes into account both the agents that take part in a linguistic interaction (speaker and hearer) and also the context in which the interaction takes place.

Language games have been implemented using Babel, a framework written in Common Lisp specially designed for language-game experiments. This framework contains two especially important core systems: Incremental Recruitment Language, a computational system that allows agents to build complex meaning representations, and Fluid Construction Grammar, a bidirectional constructional grammar formalism that is used to formulate and comprehend utterances.

The second part of the thesis is devoted to the motivational system and comprises two chapters. The first of them (Chapter 3) presents an overview of the different psychological theories on motivation and the computational models they have inspired. The chapter is structured around the distinction between *extrinsic* and *intrinsic* motivation. The first type of motivation occurs when the motivated behaviour is caused by external forces, while the second type happens when the activity is inherently interesting. After a review of the psychological literature the chapter focuses on different computational models of motivation, first explaining *reinforcement learning*, the most common technique used in this field, and then continues with an overview of the most influential computer motivation systems.

The following chapter (Chapter 4) describes the autotelic principle, the motivational system implemented in this thesis. This system is used by artificial agents language evolution experiments as a mechanism to manage the complexity of their linguistic interactions, regulating their development in an autonomous way. Inspired by the psychological theory of Flow, the core of the autotelic principle consists of the balance between challenges and skills. The main idea is that this balance is systematically destabilised, as agents improve their skills while attempting challenges, and, in consequence, agents periodically have to adjust their challenges in order to match their skill level. Artificial agents are able to set their own challenges and use monitors to identify the evolution of their performance for those tasks. This information is used to decide when they should change the challenges in order to further develop their skills.

The third part of the thesis presents the different experiments conducted to test the role of intrinsic motivation in language emergence. In these experiments, a population of artificial agents engaged in language games is provided with the autotelic principle. It is composed of two chapters, one devoted to experiments on a discrete domain and the other to experiments on a continuous domain. The first one (Chapter 5) describes the first set of agent-based experiments in which a population of artificial agents has to develop a language to communicate about a set of objects in the scene. These objects have discrete properties, that is, fixed values for certain physical characteristics. Agents start using single word utterances but eventually their communicative skills progress. By the end of the simulations they are able to produce multi-word utterances, which reduces the ambiguity in communication.

The second chapter of this part (Chapter 6) explains the second set of experiments, in which agents become engaged in linguistic interactions in order to self-organise a vocabulary to refer to colour samples. This set of experiments

extends the previous ones in that agents not only have to agree on the words associated with properties or their order but also on the values associated to those properties. It starts explaining fundamental concepts of the domain, such as colour prototypes and colour categories, and how different conceptualisations of colour samples can be modelled using IRL as a network of cognitive operations. The second part of this chapter describes the three experiments conducted. In the first one the autotelic principle is tested as a mechanism to regulate the complexity of linguistic interactions. In the second experiment agents should manage several challenges of equal complexity, which requires them to develop a vocabulary for multiple conceptualisations. Finally, a third experiment studies the impact of perceptual deviation in the communicative system resulting from the second experiment.

Finally, the last part of the thesis presents the conclusions (Chapter 7) and future research paths. The last chapter (Chapter 8) enumerates possible continuations for the research on the autotelic principle. Its first part argues that moving to embodied, multi-dimensional environments would test the motivation system in situations where several language domains are relevant in order to communicate information about the context. Moreover, it will be interesting to explore, in such experimental conditions, the possibility of letting the agents expand the vector used for challenge configuration autonomously if they encounter relevant features in the context which are not present in the challenge dimensions. The second part of the chapter mentions three potential uses of the autotelic principle outside language evolution experiments: Intelligent Tutoring Systems, developmental robotic experiments and to model non-player controlled characters in video games.

Part I

Technical Background

Chapter 2

Language games and technical background

The linguistic experiments of emergence and development of language carried out in this thesis are based on computational simulations. More specifically, these experiments consist of recurrent situated interactions between individuals in a population of artificial agents with a communicative task to accomplish. Among the different existing approaches, this work takes the perspective of *language games* (Section 2.1), as it considers the agents that take part of the linguistic interaction (speaker and hearer) but also the context in which the interaction takes place.

The chapter continues introducing Babel, a language-game framework written in Common Lisp. It focuses on two of its core systems: Incremental Recruitment Language (Section 2.2), a computational system used to represent and build complex meaning structures, and Fluid Construction Grammar (Section 2.3), a fully operational constructional grammar formalism used to formulate and comprehend utterances.

2.1 Language games

The approach taken in this work to study the emergence and development of language in a population is that of computational simulations of linguistic interactions. The basic idea is to implement a set of functions to create, learn and align words and concepts and provide them to artificial agents in order to simulate the emergence of language systems [137].

These simulations have been implemented using the *language game* approach¹ [125, 126, 138], based on the notion of language games first introduced by Wittgenstein [160]. There are other approaches in the study the evolution of language using computational modelling articulated in terms of (a) adaptive behaviour [1], (b) *iterated learning* [116], focused on cultural transmission between generations,

¹See [136] for a review of how this approach has been applied to study different language domains.

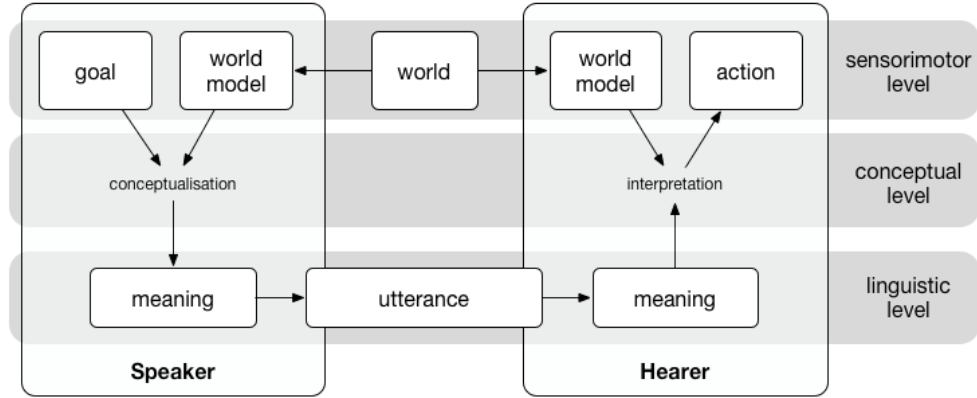


Figure 2.1: The semiotic cycle represents an interaction between a speaker and a hearer within a context (Figure adapted from [156]). It involves three processing levels: sensorimotor, used to perceive and interact with the context, conceptual, in charge of the mapping between perceived objects and their meaning representations, and linguistic, responsible of the pairing between meaning representations and utterances.

or (c) grammatical acquisition as a combination of Generalised Categorical Grammar and the echo framework [27, 60], among others (see for [88] for a review).

Language games are recurrent, simplified interactions between two agents of a population with a communicative goal [137, p. 71]:

A language game is embedded in a cooperative activity in which communication is useful. It attempts to model situated dialogue in contrast to the isolated sentences that are commonly used today in formal linguistics. Consequently, language games introduce a population of individuals (instead of an idealized speaker), a context and a communicative purpose, so that pragmatics is part of the modelling effort from the start.

These interactions between a speaker and a hearer within a context are represented using the *semiotic cycle* (Figure 2.1). It involves three levels of processing: (a) *sensorimotor*, as both agents perceive the world and interact with it through its sensorimotor system, (b) *conceptual*, responsible of the mapping between the elements in the context and their meaning representations and (c) *linguistic*, in charge of pairing meaning representations into utterances and vice versa.

There are different existing implementations of language games. In order of increasing complexity: (a) *naming game* [144], when the meaning is certain but there is competition at form level, (b) *guessing game* [38], when there is competition at both meaning and form levels, (c) *description games* [153], when the speaker formulates a description of an event in a scene and the hearer may or may not agree with it, or (d) *syntax games* [142], an extension to the naming game that allows meanings with multiple categories and relations between objects which is used to study the emergence of a shared grammar.

In all language games, though, an interaction follows a standard design: at the beginning, both the hearer and the speaker build their world models using their sensorimotor systems. This information is utilised by the speaker to *conceptualise* its goal into a meaning representation, which in turn is employed to *formulate* an utterance. The hearer *comprehends* the utterance to reconstruct the interpreted meaning representation by the speaker. It then uses the world model to *interpret* that meaning representation and performs an action. When the hearer's action corresponds to the goal of the speaker the interaction is considered a success. In all other cases, the interaction is not successful and therefore it is labelled as a failure. Agents are provided with different *learning mechanisms* that they use at the end of interactions to align their vocabularies.

The different language games presented in this work have been implemented using Babel [77, 143], an open-source, multi-agent language game experiment framework written in Common Lisp². Importantly, this framework allows researchers to design and implement the whole semiotic cycle in computational experiments of language emergence. The rest of the chapter introduces the two core computational systems of Babel, used to model the conceptual and linguistic levels of processing.

2.2 Incremental Recruitment Language

Incremental Recruitment Language (IRL) [141, 152, 122, 123] is a computational system designed to work with grounded data that allows agents to create discriminating meaning representations of objects in their context or identify objects in their scene given a particular meaning representation.

2.2.1 Meaning as a network

IRL represents meanings in *semantic constraint networks* called *IRL-networks*. An IRL-network is built combining two different meaning predicates: (a) *cognitive operations*, methods that represent some cognitive activity that the agent has to perform (such as categorising, filtering or selecting), and (b) *semantic entities*, predicates that refer to an entity (for instance, prototypes, concepts or categories).

How do these components interact in order to represent a meaning structure? On the one hand, cognitive operations are represented as predicates with a set of arguments. They are identified by their name (for instance, **filter-set**) and introduce a set of variables (noted as elements starting with a ?). Semantic entities, on the other hand, introduce only one variable that is used as an argument by cognitive operators. They are introduced by a special operator called **bind**. An IRL-network, therefore, consists of a set of meaning predicates that are linked using variables.

An example of an IRL-network for the Catalan utterance "la taula" (the table) is shown in Figure 2.2. This meaning network contains two semantic entities:

²The Babel framework is released under an Apache Licence and is freely downloadable from www.emergent-languages.org/Babel2/.

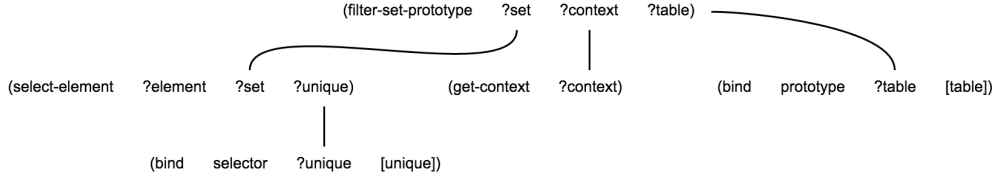


Figure 2.2: Example of an IRL-network for the Catalan utterance "la taula" (the table). Cognitive operations and semantic entities (predicates that start with **bind**) build a meaning network by connecting their variables.

an object prototype **table** and a selector **unique**. It also includes three cognitive operations: **get-context**, **filter-set-prototype** and **select-element**. Meaning predicates are connected using variables.

2.2.2 Execution of an IRL-network

Agents need to relate their internal representation of objects to their context. They do so by assigning values to the variables of an IRL-network in a process called *execution*. This process leads to a *solution* when (a) all the variables of the IRL-network are bound to a value and (b) all the cognitive operations have been executed. The dynamics of evaluating an IRL-Network are the following [123, p. 161]:

Execution of the network proceeds by executing all cognitive operations in the network. In each step, a random operation is picked from the list of not yet executed operations and it is checked whether the operation can be executed given the current set of bindings for its arguments, i.e. whether it has implemented a case for that particular combination of bound and unbound arguments. If such a case exists, then the operation is executed and newly established bindings are added to the list of bindings. If not, then another operation is tried.

It can occur that the execution of an operation (a) binds a variable to more than one possible value or (b) it is incompatible with the values of the IRL-network. In the first situation, IRL splits the search into multiple hypotheses and explores them at the same time. In the second, IRL judges the values of the meaning structure erroneous and abandon that hypothesis.

Execution is used both in conceptualisation and interpretation. In the former the speaker executes its IRL-network to judge if its meaning structure leads to a discriminative conceptualisation of its communicative goal, a process called *re-entrance* [130]. In the latter, the hearer recovers the IRL-network from an utterance and executes it in order to identify the communicative goal of the speaker. A detailed example of the execution of an IRL-network in conceptualisation and interpretation is presented in Appendix A.

2.2.3 Building meaning

IRL-networks are built with the *composer*, an algorithm responsible of assembling cognitive operations into semantic structures [123, p. 164]:

The composer is implemented as a standard best first search algorithm. Starting from an initial (usually empty) network, cognitive operations are recursively added and linked until a useful network is found. Moreover, the composer can also use complete or incomplete networks in the process of composition.

This algorithm is used in conceptualisation, with the goal of finding an adequate IRL-network that represents the communicative goal of the speaker, and employed by the hearer when it has an incomplete meaning structure (when the IRL-network has free, unlinked meaning predicates), where it tries to use the partial meaning recovered and its world model to identify possible communicative goals of the speaker. In every interaction, the composer (a) checks if the actual IRL-network is a good conceptualisation of the communicative goal (a solution) and (b) *extends* the present meaning structure with an additional cognitive operation if it is not a solution.

A significant feature of IRL is that it is *omnidirectional*. We have seen that cognitive operations introduce several arguments in the form of variables (Subsection 2.2.1). When an IRL-network is executed, the arguments that are already bound to values are the input of the cognitive operation, responsible for finding new bindings for its variables. Note that, depending on how the execution has been developed, the bound arguments of a cognitive operation may be different. IRL is a constraint language in the sense that it manages to execute these operations with different inputs. In other words, IRL's machinery permits the system to execute a cognitive operation with different entries.

2.3 Fluid Construction Grammar

Fluid Construction Grammar (FCG) [134, 135, 139] is an operational constructional grammar formalism that can be used either (a) to build and develop a grammar or (b) in experiments of language evolution. One of the most important features of FCG is that it is *bidirectional*: the same grammar can be used to *formulate* an utterance from a meaning structure or *comprehend* an utterance in order to recover the meaning it conveys.

2.3.1 Representing linguistic structures

In FCG, linguistic structures are represented as a set of *units*, each one containing a *feature structure*. Units are abstract groupings of linguistic information that usually correspond to individual words, morphemes or constituents. They have a *unit name*, which has to be unique, and a feature structure. Feature structures are abstract representations of linguistic information. They are formed by a set of *feature-value* pairs.

la-1
args: [?x-85] sem-cat: sem-class: selector syn-cat: lex-class: article gender: fem number: singular form: {string(la-1, "la")} meaning: {bind(selector, ?x-85, [unique])}

Figure 2.3: A possible linguistic structure for the Catalan article "la". The information is grouped in one unit called **la-1**. It contains five feature-value pairs: *meaning* and *form* are both sets of predicates that store an IRL-predicate and the string of the unit respectively, *sem-cat* and *syn-cat* are feature-sets that store the semantic and syntactic information of the unit and *args* keeps the set of arguments used in the unit.

Both units and feature structures are the outcomes of a combination of three different elements:

- **Symbols:** symbols are used at different levels of the linguistic representation, such as naming a unit, a feature or a value. An example of symbols are **np-unit** or **fem**. With some exceptions, symbols do not have any meaning in the FCG syntax. Nevertheless, it is useful to choose the name of the symbols according to the function they have in the grammar for comprehension reasons (for instance, using **np-unit** as a name of a unit that represents a noun-phrase).
- **Values:** the values of features can be of different types. The default one is an atomic value, but other kinds are also allowed. For instance, sets (represented with curly brackets: $\{arg_1, \dots, arg_n\}$), sequences (represented with square brackets: $[arg_1, \dots, arg_n]$), feature-sets (complex values that contain a set of feature-values), or predicate expressions (represented as a predicate with a set of arguments: $\text{predicate}(arg_1, \dots, arg_n)$), etc. [139, p. 8]. The non-default types must be explicitly declared, otherwise the FCG-interpreter will classify them as atomic values.
- **Variables:** variables start with a $?^3$ and may be used to represent units, feature-values or values that have not been identified yet in the linguistic structure. As symbols, variables have no meaning in the FCG syntax, but it is recommended to name them according to their function in the grammar.

An example of a linguistic structure in FCG is shown in Figure 2.3. The information corresponding to the Catalan article "la" is grouped in a unit called **la-1**. Its feature structure contains five features-value pairs: *meaning* has as a value a set of predicates containing one IRL-predicate, *form* (also a set of predicates) contains the corresponding string to the article, *sem-cat* and *syn-cat*

³Variables in FCG share the same representation with IRL, see Subsection 2.2.1.

are both feature-sets that store the semantic and syntactic information of the unit, respectively, and *args* stores the set of arguments used in the unit (in this example, just the variable ?x-85).

2.3.2 Transient Structures

The process of formulating or comprehending an utterance is seen as "a chain of consecutive operations over a linguistic structure, called the *transient structure*" [139, p. 5]. Transient structures (abbreviated as *TS*) as any other linguistic structure in FCG, are represented as a set of units that contain feature-value pairs. The result of applying an operation on a *TS* is a new transient structure *TS'* containing the information of *TS* and the information added by the operation.

TS have a special unit, called *root*, that acts as an input buffer: it carries the input information needed to start the execution of operations. In formulation the root unit contains the set of meaning predicates (if FCG is used as a component of the semiotic cycle these meaning predicates are the output of *conceptualisation*) and, in comprehension, it has syntactic information about the utterance to parse (normally its words and information about how they are ordered).

As "FCG is as neutral as possible with respect to the linguistic theory a linguist may want to explore" [139, p. 2], it does not impose any hierarchy on the units of a *TS*. However, if wanted, it is possible to express multiple hierarchical relations between units. A unit can use features that refer to other units using their unit names. For example, a *np-unit* can have a feature that specifies as its set of subunits *art-unit* and *noun-unit*.

2.3.3 Constructions

The operations that can be executed in a *TS* are called *construction schemes* or simply *constructions*. Constructions are linguistic structures that contain semantic and syntactic information, although they may also carry pragmatic or phonological information too. Their function is to expand the transient structure by adding information to it. This information depends on the construction, but it ranges from variable bindings or new-feature values for existing units in the *TS* to even enlarge the transient structure with new units. Constructions vary from particular operations that introduce words or morphemes to more general ones that carry more complex information, such as verbal or phrasal patterns.

Constructions are divided into a *contributing* and a *conditional* part. On the one hand, the contributing part stores the information (in the form of a set of units) that will be added to the *TS*. On the other hand, the conditional part contains the information that has to be in the transient structure in order to apply. Both conditional and contributing parts may involve one or several units. As a construction is bidirectional, the conditional part is split into two *locks*, named *formulation lock* and *comprehension lock*. Figure 2.4 illustrates the different parts of a construction.

The application of a construction consist of two phases, called *match* and *merge*:

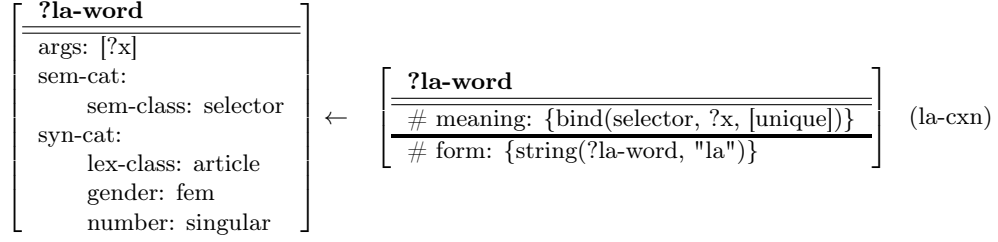


Figure 2.4: A possible construction for the Catalan article "la" (*the.F.SG*). In FCG, constructions are divided into a *contributing* (left-side) and a *conditional* (right-side) part. The former contains the linguistic information that should be added to the *TS* and the latter the information that should be in the transient structure before the construction is applied. The conditional part is further split into two *locks*, separated by a bold line. They store the information that needs to be present in the *TS* in formulation or comprehension in order to apply.

- **Match:** during this phase, a test is performed to check if the *TS* contains the information required by the construction. The information to test is stored either in the formulation or the comprehension lock, depending on which direction the construction is being applied.
- **Merge:** if the previous condition is fulfilled, the application proceeds by adding the information that the construction carries (the contributing part and the other conditional lock) to the transient structure. When the information is not contradictory (that is, there are no variable or value conflicts), the merge is successful and it generates a new transient structure which is the result of expanding the previous *TS* with that construction.

There are two significant characteristics of constructions that are not present in transient structures. Firstly, the name of the units in constructions are variables, not values. The reason is that during the match phase a construction looks for units in the *TS* that fulfil its requirements, but it cannot know in advance which are the exact units that will match its conditions or even if these units are present in the *TS* or not. Therefore, every time a construction is tried on a *TS*, it has to find the subset of units in that *TS* that match the conditions of the set of units of the conditional part of the construction.

Secondly, constructions need a way to interact with the *root*, as it stores the initial information necessary to start to apply constructions (for instance, the set of meaning predicates that should be used to formulate an utterance). Constructions use the reserved symbol # for this purpose: this symbol indicates that whatever follows should be found in the *root* unit during the match phase and moved from there to the unit where the # was located during the merge phase. For example, the lexical construction *la-cxn* (Figure 2.4), uses # two times: one in the formulation lock (for the meaning predicate `bind(selector, ?x, [unique])`) and the other in the comprehension lock (imposing the restriction that the string "la" has to be in the root unit).

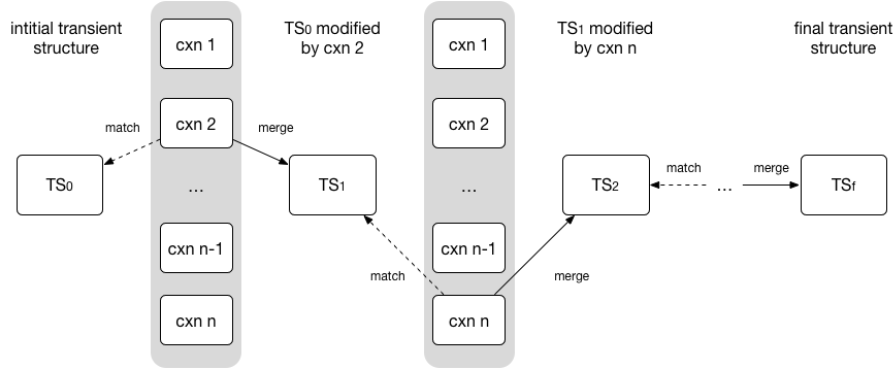


Figure 2.5: Construction application process (adapted from [134]). When a construction matches a TS_i , it merges its information with TS_i (adding new variable bindings, features-values or units) and generates a new TS_{i+1} . This permits other constructions to apply to the new TS_{i+1} in a recursive process, until the termination condition is reached.

An FCG grammar consists of a collection of constructions. This collection is stored in the *construction inventory*. Additionally, constructions can have a *label* assigned, useful in linguistic processing (see the following subsection). When FCG is used in experiments of language evolution, each agent in the population has its own construction inventory, which is never shared with the other individuals. Agents are then provided with mechanisms that help them modify their construction inventories.

2.3.4 Construction application process

How does the FCG grammar engine interact with the construction inventory or, in other words, how does the construction application process work? Given a construction set and an initial transient structure (TS_0), composed only of the root unit (which contains either (a) a set of meaning predicates in formulation or (b) a set of words and information about its order in comprehension), the construction application process iterates through the collection of constructions and tries to match them with TS_0 . If there is a construction that could match and merge, a new transient structure (TS_1) is created. The grammar engine then repeats the process for TS_1 : it tries to match all the constructions with TS_1 , and if it finds one that matches and merges will generate a new transient structure (TS_2), and so on and so forth.

The construction application process (Figure 2.5) is recursive (it will try to apply the constructions of the construction inventory at every step) and keeps executing until it reaches the termination conditions. By default, there is only one termination condition that is achieved when there are no more constructions to apply, that is, there is no construction in the construction inventory that could match with TS_n . The termination conditions can be modified by the FCG user to feed their needs. For instance, another common condition is to have a connected meaning (a meaning network where all predicates are linked through variables).

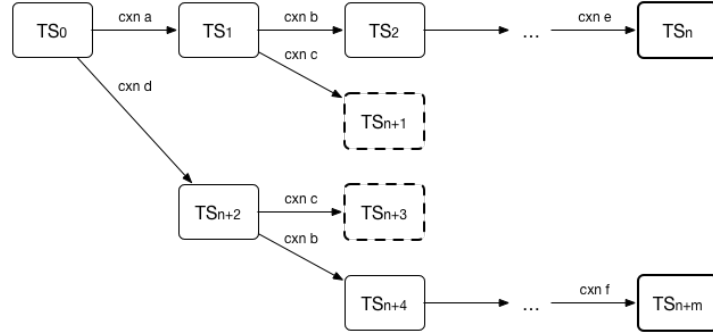


Figure 2.6: Example of construction application using a depth-first search: the grammar engine explores one path as far as possible until it reaches a solution (bold TS) or cannot continue (dotted TS), in which case it backtracks to a previous TS to search for other possible paths to explore.

It is important to remark that by iterating over all the constructions in the construction inventory for every intermediate TS , the FCG grammar tries to match constructions multiple times. This behaviour of the FCG engine is necessary because, in some cases, constructions may require information that at a certain time is still not in the TS but that other constructions may add later on. This behaviour has a drawback, namely that the computational cost of the construction application process rapidly grows and can become unmanageable with bigger grammars. How can the search space be reduced in order to speed up this process?

One way is to add *labels* to constructions. By doing this, the construction inventory is organised in subsets. The grammar engine can use labels to restrict the number of constructions that should be tested at every step and therefore speed up the construction application. The choice of which labels use is entirely on the FCG user's hands. A set of labels commonly employed is $\{\text{morph}, \text{lex}, \text{phrase}\}$, which is used to assign morphological, lexical and grammatical constructions, respectively. When using labels, it must be explicitly stated the order in which they should go, which can be different in formulation and comprehension. For instance, a possible order for the previous set could be $\{\text{lex}, \text{phrase}, \text{morph}\}$ in formulation and $\{\text{morph}, \text{lex}, \text{phrase}\}$ in comprehension.

The construction application process is implemented as a depth-first search (Figure 2.6): when a construction has been applied to a TS_n and generated a new TS_{n+1} the algorithm continues testing which constructions can be applied to TS_{n+1} , without trying all constructions in TS_n . If the path followed by the grammar engine does not lead to a *solution* (a TS that fulfils the termination conditions), it recursively backtracks to the transient structures of that path until (a) it finds one TS that can be expanded with another construction, exploring this new path also in a depth-first fashion, or (b) there are no constructions that can expand any known TS which stops the process.

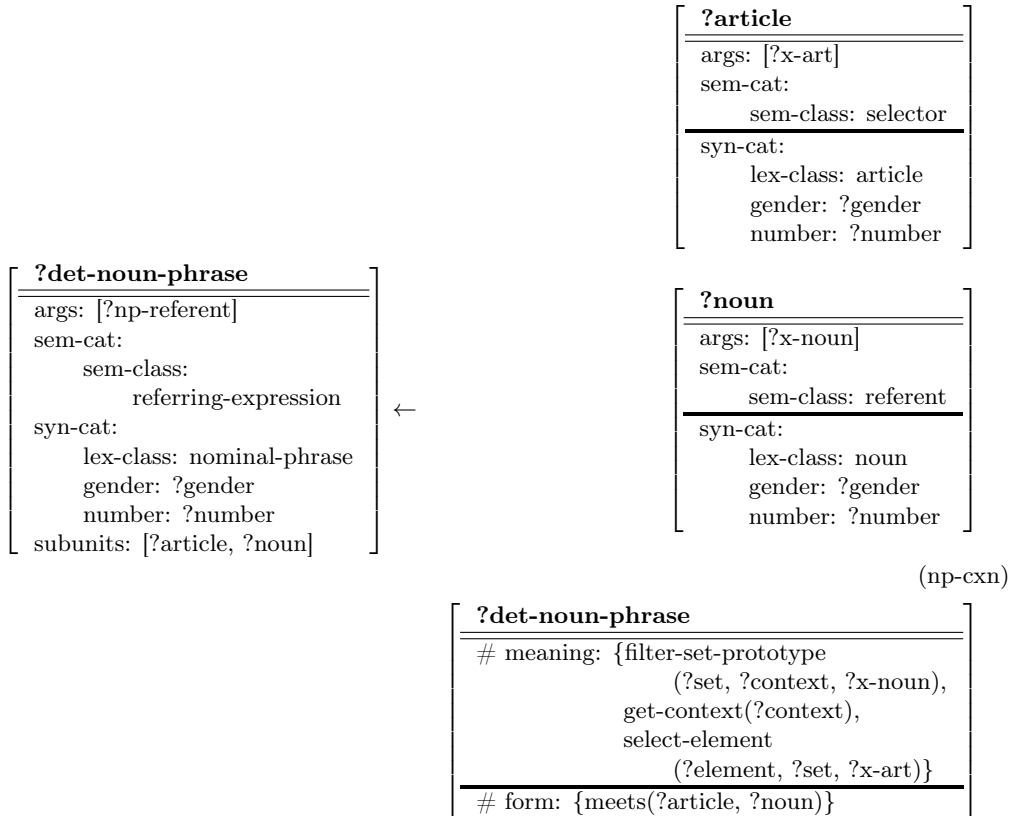
This process by default stops when it has found a solution. However, it may be useful to let it find more or even all the solutions possible. FCG can be

asked to retrieve the n -first solutions or even all the solutions of a process. These options should be used wisely, as they entail a higher computational cost.

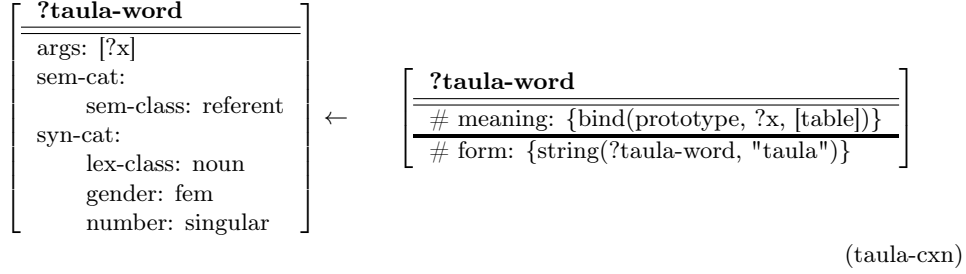
After the construction application process, when one or more solutions are found, the FCG grammar engine *renders* (in formulation) or *de-renders* (in comprehension) the set of solutions. On the one hand, rendering a TS consists of using the information stored in the **form** features of units (which introduce strings and order constraints) to construct an utterance. De-rendering a TS , on the other hand, consists of using the information stored in the **meaning** features of units (which introduce meaning predicates) to recover the meaning of an input utterance.

2.3.5 An example of how FCG works

I will illustrate how FCG operates with a simple noun-phrase example in Catalan. In this case, FCG has to formulate an utterance from using as an input the IRL meaning from Figure 2.2, replacing the variables with values. Let's assume that the construction inventory consists of three constructions: a lexical construction for the article "la" (introduced in Figure 2.4), a lexical construction for the noun "taula" (*table.F*) and a simple noun phrase construction⁴:



⁴As in other Latin languages, Catalan nouns and articles have associated a gender value. Their gender values should coincide in a noun-phrase. This behaviour is captured by the **np-cxn** with the restriction that the **gender** values should be the same for **?det-noun-phrase**, **?article** and **?noun** units.



The initial transient structure TS_0 only contains the root unit with the input meaning network. At this point only **la-cxn** or **taula-cxn** can apply, as (a) their formulation lock can match with the information in the root unit and (b) **np-cxn** requires that there are two units in the transient structure. Assuming that **la-cxn** applies, a new transient structure, TS_1 , is created in which the construction has added the information it carries (see Figure 2.3, with **unique** bound to ?x-85). At this point only **taula-cxn** can apply to TS_1 , as it contains only the unit **la-1**. After applying the construction, the resulting TS_2 contains the following information:

la-2	taula-2
args: [unique]	args: [table]
sem-cat:	sem-cat:
sem-class: selector	sem-class: referent
syn-cat:	syn-cat:
lex-class: article	lex-class: noun
gender: fem	gender: fem
number: singular	number: singular
form: {string(la-2, "la")}	form: {string(taula-2, "taula")}
meaning:	meaning:
{bind(selector, unique, [unique])}	{bind(prototype, table, [table])}

Note that at this point the values introduced by the meaning features of the two constructions (**table** and **unique**) are not linked. However, at this moment TS_2 contains two units, so **np-cxn** can try to match. As it should match in the formulation lock, the noun-phrase construction requires that (a) one unit has the feature **sem-class: selector** and the other has the feature **sem-class: referent**, (b) both units have one argument and (c) a set of meaning predicates to be available in the root. As the three conditions are met, the construction matches and can merge with TS_2 :

taula-2	la-2
args: [taula] sem-cat: sem-class: referent syn-cat: lex-class: noun gender: fem number: singular form: {string(taula-2, "taula")} meaning: {bind(prototype, taula, [table])}	args: [unique] sem-cat: sem-class: selector syn-cat: lex-class: article gender: fem number: singular form: {string(la-2, "la")} meaning: {bind(selector, unique, [unique])}

noun-phrase-11
args: [np-referent-4] sem-cat: sem-class: referring-expression syn-cat: lex-class: nominal-phrase gender: fem number: singular subunits: [la-2, taula-2] form: {meets(la-2, taula-2)} meaning: {filter-set-prototype(set, context, table), get-context(context), select-element(element, set, unique)}

The resulting TS_3 is a solution, as there are no more constructions that can apply. FCG can now *render* the transient structure using the information in the **form** features in units (**string(taula-2, "taula")**, **string(la-2, "la")** and **meets(la-2, taula-2)**), which results in the utterance "la taula" (*the table*). The other possible construction application path consists of alternating the order in which **la-cxn** and **taula-cxn** are applied. The resulting structure, however, will render the same utterance.

The same construction inventory can be used to comprehend "la taula". In this case, the construction application process will build a meaning representation equivalent to the one in Figure 2.2 with different variable names. If this process would be part of an interaction, the resulting meaning network would be then used by IRL (together with the hearer's world model) to interpret the meaning conveyed and would allow the hearer to identify the object in its context.

2.4 Summary

This chapter has introduced language games, the particular approach to study the emergence and evolution of language used in this thesis. Language games consist of recurrent communicative interactions between two agents (speaker and hearer) of a population that also take into account the context in which it takes place and can be used in computational simulation experiments.

As the experiments consist of computer simulations, it is important to understand the framework used to implement them. The general architecture of an

interaction involves several components that can be classified into three levels: sensorimotor (how agents can interact and perceive their surroundings), conceptual (how elements in the context can be internally represented as meaning structures) and linguistic (how meaning structures can become utterances and vice versa). This chapter has continued with a description of the two computational systems responsible for the conceptual and linguistic levels.

On the one hand, Incremental Recruitment Language is a computational system designed to provide agents with mechanisms to create discriminating meaning representations of elements in their context or identify objects in their scene from a meaning representation. It represents meaning as a semantic constrained network of predicates that are linked using variables.

Fluid Construction Grammar, on the other hand, is a constructional grammar formalism that consists of the process of applying a collection of constructions into a linguistic structure to map between meaning and linguistic structures. One of its principal features is that it is bidirectional: the same set of constructions can be used in formulation (build an utterance based on an input meaning representation) or comprehension (recover the meaning from a given utterance).

Part II

The Autotelic Principle

Chapter 3

History of Motivation: psychological and AI approaches

Despite the fact that it is a term widely used, it is not clear what motivation means. Psychologists have been studying the role of motivation since the 1930s and different definitions and classifications have been proposed. Inspired by these findings, artificial intelligence (AI) and robotics researchers have proposed several systems to provide agents with mechanisms to manage complexity and its development in an autonomous way.

So what is motivation? Are there different kinds of motivation? How can it be modelled? This chapter addresses these questions reviewing the major psychological theories of motivation and the computational models they inspired. We start by reviewing the psychological literature on motivation (Section 3.1), focusing on the distinction between *extrinsic*, where external forces influence the motivated behaviour, and *intrinsic* motivation, when an activity is inherently interesting and enjoyable. We then review the literature on computational models of motivation (Section 3.2), explaining reinforcement learning, the most popular technique used in computational models of motivation, and then focusing on those models that propose different approaches to intrinsic motivation.

3.1 Psychological theories on motivation

Psychologists have been studying the role of motivation for almost a century. During this time several theories have been proposed and various definitions of motivation have been suggested. This section reviews the most influential theories on motivation¹ in a chronological order and classifies them into different types.

¹It is important to make explicit that this section does not present a detailed overview of all motivation theories in psychology but rather a brief introduction to the most influential ones during the twentieth century.

3.1.1 Drive-reduction theory

Before the 1930s, psychologists were mainly interested in behaviour and conditioning. They thought that behaviour was the result of associations between stimulus and response [93]. Psychologists studied animals and animal behaviour as if they were inactive organisms unless special conditions of arousal appeared, such as hunger, pain or sexual excitement. It was under these assumptions that the first influential theory on motivation, the *drive-reduction* theory, was proposed by Hull in his book *Principles of behaviour: an introduction to behaviour theory* [62]. Heavily inspired by the notion of *homeostasis* from biology, which can be defined as a property of a system in which a set of variables is actively regulated to remain in equilibrium, Hull suggested that all behaviour of all organisms can be explained as a result of biological needs and lack of balance.

A *need* is a biological requirement of organisms. For example, thirst is seen as the need for more water, hunger as the need for more energy or tiredness as the need for some rest. Organisms are therefore motivated to perform actions as a response to their biological needs. Hull reasoned that behaviour could be regarded as an expression of the organism's goal of biological health.

Hull used the term *drive* to define the resulting state triggered by biological needs in organisms. In other words, drives are the forces that command behaviour. Organisms seek to eliminate these states, as any state coming from a biological need is experienced as discomfort. Following the previous example, when an organism is hungry it undergoes a *need* for energy. This necessity of energy acts as a *drive* to carry out a behaviour such as looking for food. If the same need occurs again, an animal would repeat any behaviour that previously reduced that drive. This *behaviour reinforcement* will eventually result in the *learning* of that behaviour (Figure 3.1). Hull defined this process as follows [62, p. 71]:

"Whenever a reaction (R) takes place in temporal contiguity with an afferent receptor impulse (\dot{s}) resulting from the impact upon a receptor of a stimulus energy (\dot{S}), and this conjunction is followed closely by the diminution in a need (and the associated diminution in the drive, D, and in the drive receptor discharge, s_D), there will result an increment, Δ ($\dot{s} \rightarrow R$), in the tendency for that stimulus on subsequent occasions to evoke that reaction. This is the "law" of primary reinforcement."

Hull wanted to make psychology as scientific in its predictions as physics or chemistry. Therefore, he proposed precise formulas to model behaviour. In its simplest form, Hull stated that behaviour (B) is the result of multiplying drive by habit (H): $B = D \times H$. But he also provided detailed formulas for more particular behaviours. For example, he defined (a) the increment of habit strength (ΔsH_R) as $\Delta sH_R = 1 - 1/10^{0.0305N}$, where N is the number of successive reinforcements that strengthen the stimulus and response, or (b) the reaction potential (ΔsE_R) as the product of habit strength and drive: $\Delta sE_R = sH_R \times D$.

His theory rapidly spread with other psychologists, becoming the most cited psychologist in the 1940s and 1950s. However, many predictions based on Hull's

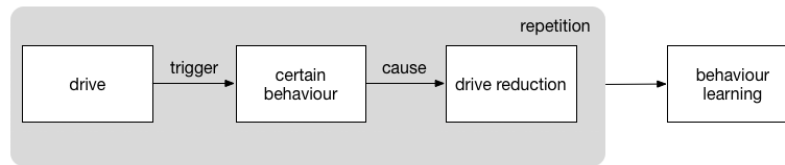


Figure 3.1: Visual representation of the Drive-reduction theory by Hull. Drives are experienced as uncomfortable by organisms. When a drive occurs, organisms react to it adopting a behaviour that reduces it. Organisms repeat behaviours that reduce drives and eventually learn them in a process called *behaviour reinforcement*.

equations were not correct and some of the most prominent researchers at the time proposed modifications to the theory in order to improve it. For example, Spence demonstrated how fundamental principles of conditioning could be applied to the analysis of complex learning tasks [119], Crespy suggested that reinforcement affects performance instead of learning [33] or Mowrer proposed avoidance learning, which also accounts for the fact that an organism can acquire a response to prevent unpleasant stimuli from happening [86].

In the 1950s researchers started to look at the relationship between drive, curiosity and learning, inspired by the pioneering work on curiosity by Nissen [87]. Montgomery carried out a series of studies on the exploratory behaviour of rats [84] and proposed the *exploratory drive*, stating that animal exploration of its environment is determined by the opposition of two motivations: curiosity and the fear of novelty. Another example is the research conducted by Harlow and his colleagues, who carried out several investigations on the manipulation and visual exploration observed on monkeys [54], where they found a strong drive of manipulation when learning how to solve puzzles in the absence of rewards.

3.1.2 Operant conditioning

Skinner proposed another behaviourist theory to explain behaviour, called *operant conditioning* [114, 115]. According to this theory, organisms *operate* in their environment, that is, act in a certain way with their surroundings. They are exposed to different external stimulus and one specific kind, called *reinforcing stimulus* or reinforcer, influences their behaviour. He called this *operant conditioning*, as reinforcing stimulus determine the behaviour of organisms.

Skinner identifies two kinds of reinforcers: positives and negatives. A *positive reinforcer* increases the probability of a behaviour to occur in the future. A *negative* reinforcer or *punishment* has the opposite effect, decreasing the possibility of observing that behaviour in the future. Based on Skinner's work *behaviour modification* emerged, a technique used to modify the behaviour of organisms. This method consists in using stimulus to reinforce or punish certain behaviours. It has been used to treat, for instance, addictions or depression [81, 118].

3.1.3 Reconsidering motivation

Drive-reduction theory dominated psychology in the 1940s and 1950s. In the 1950s some psychologists started to criticise Hull's emphasis on *drive-reduction*. Their main criticism was that this theory left out many factors that influence human behaviour by focusing only on biological factors.

White, like other psychologists, thought that drive-reduction theory could not successfully account for human behaviour and in 1959 he proposed a different approach to the study of motivation around the concept of *effectance motivation* [158], which he described as a "tendency to explore and influence one's environment". He suggested that the *master enforcer* for human beings is *competence*, the ability to successfully interact with the environment. Feeling competent (a) make humans experience a positive feeling and (b) asserts our capabilities within particular domains.

Competence motivation differs from biologically driven motivation in that it improves the abilities of an individual instead of regulating a process of need, such as hunger or thirst. Importantly, White not only decoupled motivation from biological needs but stated that competence based motivations help individuals to improve themselves. In his view, people do not only do activities because they need something but also to experience competence.

Another theory that took distance from the dominant behaviourist approach to motivation in those years was *cognitive dissonance*, proposed by Festinger [43]. The central hypothesis is that psychological tension acts as a motivational force. This theory is concerned with the conditions that cause dissonance in an individual and the ways it can be reduced.

The core of the theory relies on two relations in human cognition: *dissonance* and *consonance*. On the one hand, dissonance occurs when two internal cognitive structures such as opinions, behaviours or beliefs are inconsistent, meaning that these two items do not follow one from each other. In other words, dissonance is experienced when these cognitive structures are incompatible with the situations perceived. Consonance, on the other hand, is established when the relationship between two items is consistent. For example, a driver that thinks that private transportation is not good for the environment has an opinion that is dissonant with him continuing to drive. The existing dissonance between these two items (being a driver and thinking that private transport is negative for the climate) does not invalidate that possibly other items can be consonant with him driving, such as time efficiency or self-autonomy.

Festinger used the notion of dissonance to explain motivation. He argued that experiencing dissonance produces discomfort and therefore *motivates* humans to eliminate it or reduce it². According to him, attempts to reduce dissonance can take three forms (Figure 3.2): (a) try to change one or more beliefs, opinions or behaviours associated to the dissonance, (b) acquire new information that increases the existing consonance (and consequently reducing the dissonance) or (c) forget or reduce the cognitions that are dissonant. Moreover, Festinger found

²Note that Festinger used a similar argument to drive-reduction theory, where organisms are motivated to perform actions in order to reduce their needs.

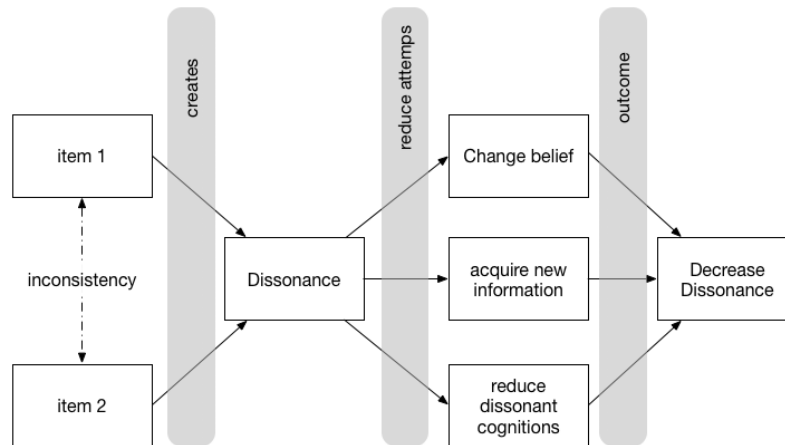


Figure 3.2: Visual representation of the Cognitive Dissonance Theory by Festinger (adapted from [53]). Two inconsistent items (incompatible cognitive structures) are experienced as dissonant. Organisms are motivated to reduce dissonances, which can be achieved by changing beliefs, acquiring new information or reducing dissonant cognitions.

a correlation between the magnitude of a dissonance and the motivation to reduce it: "The strength of the pressures to reduce the dissonance is a function of the magnitude of the dissonance" [62, p. 18].

3.1.4 Motivation and balance

Curiosity was usually assumed as an internally stimulated or *homeostatic* drive within the drive-theoretic view. In the 1950s Berlyne proposed an alternative drive-based perspective on curiosity called *Curiosity-drive theory*, suggesting that it is externally stimulated [17]. According to him, the curiosity drive is the result of external stimuli, particularly "stimulus conflict" or "incongruity", which is related to properties such as complexity or novelty. These stimuli cause unpleasant experiences of "uncertainty" which lead to a loss of cognitive and perceptual coherence. Coherence can be gained again by gathering new information about the stimulus and therefore reducing the incongruity. Interestingly, Berlyne proposed that curiosity drive is activated with intermediate levels of incongruity and not entirely new or familiar situations [17, p. 189]:

"Our theory of curiosity implies that patterns will be most curiosity arousing at an intermediate stage of familiarity. If they are too unlike anything with which the subject is acquainted, the symbolic response-tendencies aroused will be too few and too feeble to provide much conflict, while too much familiarity will have removed conflict by making the particular combination as an expected one."

The idea of intermediate incongruity by Berlyne, who was strongly influenced by behaviourism at the beginning of his career, was also shared with other theories

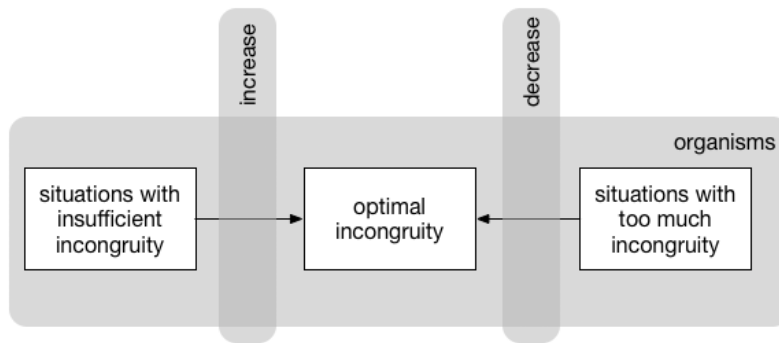


Figure 3.3: Visual representation of the Optimal Incongruity Theory by Hunt. Organisms seek situations that provide an optimal amount of incongruity. This motivates organisms to change their situation if its current situation provides them too much or too little "stimulus conflict". The incongruity level of a situation decreases when organisms have gathered more information about it.

that took more cognitive approaches. For instance, Hunt [63] proposed that humans search for *optimal incongruity*. Hunt used an information-processing approach to motivation. His central hypothesis is that organisms need an optimal amount of incongruity and look for situations that can provide them with that level (Figure 3.3).

Hunt suggested that organisms react to insufficient incongruity searching for situations that increase it. Similarly, organisms look for situations that decrease incongruity when there is too much of it. A situation loses incongruity after an organism has repeatedly been exposed to it, in a similar way to what Berlyne proposed with the information gathering concept, where having more information about a "stimulus conflict" reduces its incongruity. The main difference between both approaches is that in Hunt's view organisms look for incongruity, while in Berlyne's approach incongruity was caused by external stimuli. In other words, Hunt changes the perspective on motivation from external to internal causes.

Another related approach is that of Hebb [56], who put the focus on the need of *optimal arousal* in the central nervous system. This parallel approach to motivation puts its focus at the physiological instead of the psychological level, differing from optimal incongruity and curiosity-drive theories. The main idea is that organisms have a need for an optimal level of arousal and responses that conduct them towards that optimal level of arousal are reinforced. Optimal level are gratifying, while being over-aroused or under-aroused is experienced as unpleasant.

According to this model, stimuli too intense (such as being exposed to a new situation) introduce high levels of physiological arousal in organisms and motivate them to behave in a way that the arousal level decreases (for example by taking distance from a particular stimulus). On the contrary, organisms are motivated to increase their arousal level when it is too low, looking for more thrilling stimuli in their environment.

Despite their differences, the three approaches share the idea that motivation requires intermediate levels of stimuli, which means that organisms look for balanced situations, avoiding those that provide too low or too high a stimulus. This idea also entails an interesting effect, namely that organisms continuously look for the optimal level of stimuli. This level changes over time as organisms collect information about new stimulus, which results in a reduction of the incongruity or arousal it provokes.

3.1.5 Personal causation, or the opposition of intrinsic and extrinsic motivation

deCharms explicitly opposed the notions of extrinsic and intrinsic motivation in the *Personal causation theory* [28]. He defined personal causation as "the initiation by an individual of behaviour intended to produce a change in his environment" [28, p. 6]. In other words, he stated that people are motivated to modify their surroundings and the source or origin of this behaviour arises from the individuals themselves. That is, individuals *cause* things to happen.

In his view human beings continuously strive to remain the cause of their behaviour, fighting against being restricted or inhibited by external forces. Furthermore, he stated that individuals must be active and in control of their own behaviour instead of being controlled by other factors. Consequently, for deCharms there is a direct opposition between intrinsic and extrinsic forces for the control of human behaviour. To illustrate this differentiation between internal and external control he used the metaphor of *origin* and *pawn* [28, p. 273]:

"We shall use the terms "Origin" and "Pawn" as shorthand terms to connote the distinction between forced and free. An Origin is a person who perceives his behaviour as determined by his own choosing; a Pawn is a person who perceives his behaviour as determined by external forces beyond his control."

This distinction entails differences in the feelings that individuals experience: an Origin person has a feeling of personal causation, which is reinforced by the changes he or she causes in the environment. On the one hand, an origin person determines his or her behaviour based on intrinsic motivation. That human has freedom of action (as he or she has the control of his or her actions) and perceives the environment as challenging. On the other hand, the behaviour of a Pawn is induced by external forces and he or she experiences a feeling of powerlessness. This means that the behaviour of a Pawn person is determined by extrinsic motivation factors that regulate his or her actions. The "subordination" to external factors makes individuals perceive the environment as threatening instead of challenging.

3.1.6 Flow Theory

The explicit confrontation between intrinsic and extrinsic forces in personal causation together with deCharms' unambiguous view that intrinsic motivation is

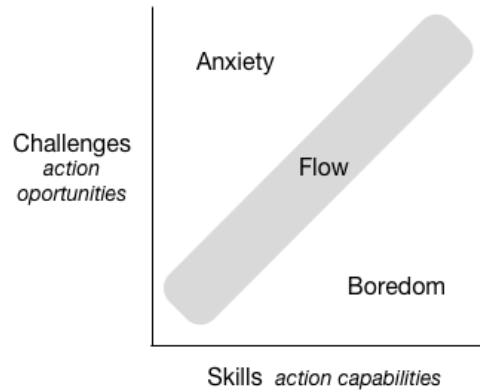


Figure 3.4: Visual representation of the Flow Theory by Csíkszentmihályi (adapted from [35]). A person enters a state of *flow* when opportunities for action are in balance with her skills. When the challenge is too demanding for her capabilities she experiences *anxiety*. *Boredom* is experienced when skills are greater than the current challenge.

more desirable for human wellness had an impact in the psychological study on motivation, increasing the focus of researchers towards intrinsic motivation. His influence can be seen in the work of Csíkszentmihályi, whose research on intrinsic motivation lead to the *Flow theory* [35, 36].

Csíkszentmihályi started studying creative processes in painters. He noticed that they greatly enjoyed their work and, sometimes, they were so concentrated that they could lose the notion of time. But what impressed him is that, once the painting was finished, they completely lost interest in it. He was intrigued by this and wanted to understand why people could get immersed in goal-focused, complex activities that do not provide any external reward.

He found that the reason was that participants found these activities inherently enjoyable. In other words, they were intrinsically motivated because becoming engaged in these activities was rewarding for them. He called these activities *autotelic*, as the motivational driving force (*telos*) comes from the individual itself (*auto*). It is necessary to distinguish autotelic activities such as playing chess or rock climbing from directly amusing activities like sliding down a hill on a toboggan. Autotelic activities are challenging and provoke a feeling of achievement and fulfilment once the individual has overcome its obstacles. In addition, these activities usually present challenges with progressively higher levels of difficulty. So, what makes an activity autotelic?

Autotelic activities can be explained as the interaction of two dimensions: *challenges* or action opportunities, a certain task to perform in order to achieve a goal, and *skill* or action capabilities, the capacity that an individual has to cope with the demands of that challenge. The relationship between them determines a range of different mental states that people experience when involved in an autotelic activity (Figure 3.4). According to him, *boredom* is experienced when the skills are greater than the current challenge and *anxiety* appears in the opposite

case, when the challenge is too difficult for the current action capabilities of an individual. Notably, a third state, named *flow*, can be experienced when there is a balance between both challenge and skills. Csíkszentmihályi identified flow as the optimal state of experience, as it provides optimal challenges for the skills of the participant in the autotelic activity [35, p. 36]:

"In the flow state, action follows upon action according to an internal logic that seems to need no conscious intervention by the actor. He experiences it as a unified flowing from one moment to the next, in which he is in control of his actions, and in which there is little distinction between self and environment, between stimulus and response, or between past, present, and future."

For example, an experienced rock climber will experience boredom climbing a route of average difficulty, whereas an inexperienced climber will experience anxiety when attempting to climb the same route. Both climbers will look for routes that match their skills, as it is in this equilibrium that they could experience the flow state.

Importantly, the flow state provides the best scenario to develop one's capabilities or skills. As a consequence, the state of flow is not stable but in continuous motion, as participants of an autotelic activity which are in a flow state continuously extend their skills and therefore unbalance their challenge-skill relation. As the flow state is enjoyable participants strive to remain in it, therefore becoming self-motivated [35, p. 48]:

"A flow activity allows people to concentrate their actions and ignore distractions. As a result, they feel in potential control of the environment. Because the flow activity has clear and non contradictory rules, people who perform in it can temporarily forget their identity and its problems. The result of all these conditions is that one finds the process intrinsically rewarding."

The Flow theory explains intrinsic motivation using the notions of intermediate levels of stimuli that first appeared in the 1960s (see Subsection 3.1.3) and the idea of perceiving the environment as challenging of deCharms (Subsection 3.1.5).

3.1.7 Self-determination theory

In the mid-1980s Edward L. Deci and Richard M. Ryan proposed a framework to study human motivation called the *Self-Determination Theory* (SDT) [39, 40]. SDT takes into account both intrinsic and extrinsic motivational forces and the influence of social conditions in order to explain motivation and its role in human development and wellness. The theory maintains that there are three universal psychological needs:

- **Autonomy.** People seek to feel free to set their own goals and behaviour. Extrinsic rewards undermine autonomy, as in such cases behaviour is controlled by external factors. Therefore, people feel more independent when their behaviour is intrinsically motivated.
- **Competence.** People need to satisfy their need for competence. This occurs when they feel challenged but also able to cope with the task. In these situations people experience positive feedback that increases their intrinsic motivation while reducing the extrinsic motivation for the task.
- **Relatedness.** It is essential for human well-being to develop and maintain satisfactory close personal relationships, both at sentimental and friendship levels.

According to SDT, conditions in which humans experience a fulfilment of these needs increase motivation and stimulate the engagement in activities. Equally, a situation in which some of these three psychological needs are not satisfied has a negative impact on wellness and decreases intrinsic motivation. SDT is articulated in six "sub-theories" that address different aspects of motivation:

- Cognitive Evaluation Theory (CET) addresses how social contexts impact intrinsic motivation.
- Organismic Integration Theory (OIT) addresses extrinsic motivation, its subtypes (external regulation, introjection identification and integration) and how it is influenced by social contexts.
- Causality Orientations Theory (COT) is concerned with individual differences in behaviour.
- Basic Psychological Needs Theory (BPNT) addresses the concept of psychological needs and their relation to wellness.
- Goal Contents Theory (GCT) focus on differences between intrinsic and extrinsic goals and their relationship to wellness.
- Relationships Motivation Theory (RMT) studies the relationship between close personal relationships and wellness.

SDT makes a distinction between *autonomous*, when someone acts according to his or her will, and *controlled* motivation, when someone acts conditioned by external forces. SDT states that the former provides better results in terms of quality of performance and well-being. This psychological framework has been applied in different domains such as education [40, 41], psychotherapy [100] or marketing [151].

SDT closes the overview of the different psychological theories of motivation. By presenting them chronologically, one can observe that the conception of motivation shifted from a purely external driving force in the first half of the XXth century to notions that progressively took into consideration the importance of intrinsic forces, which introduced concepts such as challenges, competence or balance.

3.2 AI approaches to motivation

This review was necessary to understand the psychological ideas that inspired computational motivation systems (CMS). The rest of the chapter continues with a description of reinforcement learning, the most common technique used in CMS, followed by an overview of some of the most relevant computational motivation systems³. This will provide a background to situate the autotelic principle, the motivational system implemented in this thesis, within the different computational models proposed in the literature. Interested readers on various typological classifications of computational motivational systems are referred to [13, 91, 90, 107, 82, 57].

3.2.1 Computation Reinforcement Learning

The most popular technique used in computational models of motivation is *reinforcement Learning* (RL). This technique consists in providing agents with a mechanism to decide which actions they should take according to their environment. The agent obtains a *reward*, either positive or negative, after each action has taken place. The decision process is based on the idea of maximising the cumulative reward. A more detailed definition can be found in the introduction of the 1998 book on RL by Sutton and Barto [150, p. 4]:

"Reinforcement learning is learning what to do —how to map situations to actions— so as to maximize a numerical reward signal. The learner is not told which actions to take, as in most forms of machine learning, but instead must discover which actions yield the most reward by trying them. In the most interesting and challenging cases, actions may affect not only the immediate reward but also the next situation and, through that, all subsequent rewards. These two characteristics—trial-and-error search and delayed reward—are the two most important distinguishing features of reinforcement learning."

Sutton and Barto [150] stated that RL is based on four elements:

- **Policy:** It defines the behaviour of the agent. It consists of a mapping between a set of actions and a set of conditions in the environment. The stronger a relationship is, the more likely a particular action will be performed by an agent when its associated environmental conditions are perceived.
- **Reward function:** It provides a number (reward) for each environment condition that represents the worth of an action. This value is received after each action the agent has performed. Rewards can be *external*, when they are provided by an external entity, *internal*, when they are computed by the agent itself, or a mix of both.

³As in the case of psychological theories on motivation, this section should not be taken as a detailed overview of all computational motivational systems, but rather as a summary of some of the most influential ones.

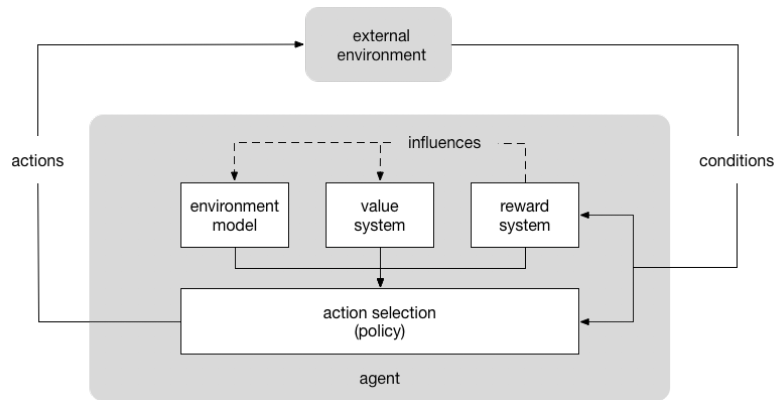


Figure 3.5: Schema of the basic elements of a Reinforcement Learning system. Given a goal, an agent learns a set of *policies*, a map between actions and environment conditions, by maximising a cumulative *reward*. Rewards are numbers that represent the worth of an action and can be positive or negative. They are used to estimate the long-term *value* of actions. Values can be combined with *environment models* to choose which actions should be performed given the environment conditions. It is important to stress that rewards can be computed by the system itself (internal) or by an external entity (external).

- **Value function:** it provides an estimation of the reward that a given state will provide in the future. In other words, it is a measure of the value of a particular action in the long-term. Values are evaluated and re-estimated based on the experience of the agent, namely the different observations it can gather. The decision process uses values to evaluate and decide which are good or bad actions and which actions should be attempted in the future.
- **Environment model:** it predicts (a) the outcome environment conditions of performing a particular action and (b) its associated reward. It is used for *planning*, taking into account the possible future environment conditions of performing an action to decide which action should be executed next. In early RL systems there were no environment models and the decision of which action to perform was based on trial-and-error.

It is important to point out that RL is designed as a goal-directed learning mechanism that does not require any supervision by the experimenter. Although, it is still possible that the learning is partially or entirely supervised in some cases, for instance when the agent interacts with a teacher (either human or another agent) that is present in its environment. Also, in RL the learning occurs without relying on a previous set of pre-processed examples, but rather from the data obtained performing actions and observing the changes that these actions produce in an agent's environment.

In AI and robotic research, RL is mainly used to develop *intrinsically motivated agents*, which are computational systems that can autonomously learn

and interact with their environment. *Intrinsic motivation*, though, is a difficult concept to define in computational motivation systems. As has been mentioned above, rewards can be intrinsic, extrinsic, or both, depending on how the reward is computed. A reward is *external* when its numerical value is calculated outside the system, and *internal* when it is generated by the system itself [90, p. 3]. This distinction makes it technically hard to establish a clear cut between intrinsically and extrinsically motivated systems, as reward functions usually combine information from the external environment with internal evaluations of the system, due to the fact that an activity can be both intrinsically and extrinsically motivated at the same time.

3.2.2 Classifying computational motivation systems

What should, therefore, be a good way to establish differences between CMS? Instead of following the most common classification used in psychological literature, articulated in terms of intrinsic and extrinsic motivation, Oudeyer and Kaplan [90] proposed that computational motivational systems should be classified according to their differences in the operational approaches they take.

Knowledge vs competence motivation systems

The most widely adopted classification is based on the distinction between knowledge and competence [90, 82]. *Knowledge based motivation systems* are those that evaluate the knowledge of the system. They can be further classified between *prediction* and *novelty* based. The former relies on the contrast between predictions of actions and the resulting environmental outcomes, while the latter ranks new situations on a scale from novel to already known, where novel is usually linked to being more interesting for the system. This classification is then used to drive the action selection process.

On the other hand, *Competence based motivation systems* measure the competence of the system to achieve self-determined goals, that is, they measure *how well* the system performs certain tasks. This information can be then used to select which tasks should be attempted in future actions.

Fixed vs adaptive motivation systems

According to Oudeyer and Kaplan [90], a motivational system is *fixed* when the resulting reward for an executed action is constant, that is, it does not change over time. Contrary to this, a motivational system is *adaptive* when the reward value, when performing an action, is not static but changes, usually depending on the learned skills of the agent and on how much that action has been executed.

Homeostatic vs heterostatic motivation systems

Computational motivation systems can also be classified according to their internal dynamics [90, 71]. A system is *homeostatic* when its goal is to maintain the system within a comfortable zone. Usually, these systems keep track of a

set of system variables that monitor the internal state of the system and strive to maintain them within a comfort range. An analogy can be made with the drive-reduction theory by Hull (Subsection 3.1.1), as an alteration of the system equilibrium will push the agent to execute actions to return to the comfort range.

Inversely, a system is identified as *heterostatic* when it avoids its equilibrium, moving out from a balanced situation. Such a system will consistently reward new situations and can be very useful if the system is designed to explore the different environmental conditions or states.

The section continues with an overview of some of the most influential CMS, explaining the key points of their architecture, situating them using the different classifications introduced in Subsection 3.2.2 and establishing links with the distinct psychological theories of motivation.

3.2.3 From Predictor Error to Compression Driven Progress

In 1991, Schmidhuber presented a motivational system based on prediction error [105]. The main idea is to develop a system that can learn to map between actions and consequences in the environment. The system improves by focusing on actions with a high prediction error.

The architecture of the system consists of (a) a predictor P of the states of an environment to a set of actions that the system can execute and (b) a confidence module C that evaluates the predictor reliability (both P and C implemented as Recurrent Neural Networks (RNN)). The system's goal is to improve P , that is, the mapping between actions and their consequences in the environment. The action selection is driven by the system's *adaptive curiosity*, probably inspired by Berlyne's notion of curiosity (see Section 3.1.4). This mechanism chooses the next action to execute based on its reliability, selecting those actions that present lower reliability in order to improve P .

This action selection algorithm sometimes pushed the system to continuously address actions for which the system could not improve P . Schmidhuber suggested that the system could be enhanced with the addition of a module to measure the improvement of P as a solution [104] or measuring the predictor's information gain [146].

Later, Schmidhuber reformulated his motivational system in terms of *Compression Driven Progress* (CDP) [106, 107]: the system stores the data obtained through interactions with the environment. This data is analysed in order to detect regularities and allows the system to *compress* it. More stored data increases the possibility to identify patterns in it and therefore improves its compression. The system monitors the improvements of the predictor P , which is seen as a data compressor module. It generates rewards in proportion to the compression progress and uses them in the action selector. The decision process maximises the expected reward, which can be seen as the expectation of augmenting the compression of the data. CDP can be classified as a prediction based, adaptive and heterostatic (as it continuously looks for new situations that permit the compression of data) motivation system.

3.2.4 Intrinsically Motivated Reinforcement Learning

The goal of Intrinsically Motivated Reinforcement Learning (IMRL) [13, 111, 148, 112] is to implement a system that can develop itself in a set of different situations without requiring problem specific reward functions. Within this system agents can develop skills or *options*, which are closed-loop control policies. The different options of an agent determine how it can interact with its environment. Each option has an *initialisation set*, which is a set of environment conditions or states of which an option can apply, and a *termination condition*, the set of environment conditions in which this option stops. For instance, an option `grasp(something)` may require that at least one articulated extremity of the agent should be free in order to perform that option and may have as a terminate condition that the extremity used is not available.

Importantly, options may involve other options, meaning that this structure provides a set-up that allows the rise of hierarchical control architectures, where complex options may reuse more basic options already known by the agent. Following the previous example, the option `hammer(something)` will initially require the option `grasp(hammer)`.

When a new option is discovered, it is added to a set of known options. In IMRL the decision-making component is formalised as a semi-Markov Decision Process (SMDP) to decide which options to execute. It learns probabilistic relations between options and the effects of running policies, which are called *option models*. Option models are learned and improved using the outcome of performed options. As the architecture permits hierarchical relations between options in order to keep consistency in the system, it is possible to update several policies at the same time that involve a certain option, even when these policies were not determining the behaviour of the agent at that point [13].

The development of a collection of options is driven by a *surprise* factor. It consists in a decrease of the reward that the system provides for executing an option when it has repeatedly been performed. The surprise factor takes into consideration (a) if the probabilistic relationship between the option and its effects has been learned and (b) it incentivises the agent to attempt lesser known options. Therefore, IMRL can be classified as a knowledge based (more specifically prediction based)⁴, adaptive, homeostatic motivation system.

The IMRL architecture was modified by Stout and Barto into a competence-based motivation system named *Competence Progress Intrinsic Motivation* (CPIM) [147]. The system is initialised with a set of options or skills for which it has to learn the appropriate policies. The system measures its *competence* in a similar way as option models in IMRL and combines it with a small penalty that increases over time to compute the reward obtained when performing that option. The system uses the expected improvement in competence, the estimated

⁴IMRL can be classified as a prediction based motivation system because the system focuses on addressing options for which the probabilistic relation with their effects is less known. The system reduces the probabilistic error on known options and therefore improving its predictions on what executing options cause in the environment. This view is shared with Oudeyer and Kaplan [90] but not with Mirolli and Baldassarre [82], who classified IMRL as competence based.

progress of an option's reward, to decide the future options to explore.

3.2.5 Temporal Difference Competence-Based Intrinsic Motivation

Temporal Difference Competence-Based Intrinsic Motivation (TD-CB-IM) [103, 102, 6] is a system developed within the IMRL framework that learns a set of skills based on the system's estimation of the improvement of its skills. The architecture of the system consists of (a) a given set of experts, which are RL components that can control its behaviour (in other words, components used to choose which action the system should execute) and (b) a *selector* component that decides which expert use for every interaction. Both are implemented as an actor-critic RL model [150].

Interestingly, TD-CB-IM proposes to split the system into two phases: *childhood* and *adulthood*. During childhood, the system is in an exploration phase where it tries to discover and improve skills without any particular goal. This differs from the adulthood phase, where the system has a clear task and uses the skills developed during its childhood to achieve it. While the selector learns in both phases, experts only learn during the childhood phase.

Experts learn to improve their predictions computing its TD-error. When an expert has been selected to perform an action, it improves its predictions using a TD learning algorithm [150] that takes into account the two last successive estimations of the expert's critic component and an expert reinforcer's reward. During childhood, the selector learns which expert it should use with a similar TD learning algorithm that considers the last two estimations of the selector's critic component and the resulting TD-error of the selected expert, which makes the system adaptive. At each interaction, the selector chooses to use the expert with the highest expected learning improvement (the expert with a maximum TD-error value of the set of experts). The goal of this phase is to allow the system to develop a set of skills, distributed over its set of experts, and to discover which expert the selector should choose depending on the environmental state.

The developed skills are used in the adulthood phase, where the system has to perform a specific task. In contrast with the expert components, the selector continues to learn with the same learning algorithm it used during the childhood phase but taking into account an external reward related to the system instead of using the TD-error of the selected expert. This allows the system to keep improving the selector with the feedback of how well it is performing. Although the authors stress the fact that it is a competence-based motivation system, this classification is not evident as the system's focus is to improve the predictions of the experts.

3.2.6 Intelligent Adaptive Curiosity

Intelligent Adaptive Curiosity (IAC) [89, 91, 69] is a motivation system created by Oudeyer and Kaplan inspired in the notion of curiosity drive by Berlyne (see Subsection 3.1.4). The main idea is that the system must maximize an abstract cognitive variable called *learning progress*. Repeating actions decrease

the learning progress of the system and therefore pushes it to explore new actions. It is a knowledge-based, adaptive, homeostatic motivation system.

The system stores all its interactions with the environment (called experiences) in the form of vector exemplars. Each vector represents the sensorimotor context in time t , which stores the concrete values of both its motor parameters $M(t)$ and its sensors $S(t)$ (noted $SM(t)$). Experiences are recursively grouped into *regions* of the sensorimotor space of the system. Each region R_i has an *expert* E_i , a learning component, which is trained with the experiences of that region. The function of E_i is to predict the outcome of $SM(t+1)$ when $SM(t)$ corresponds to the range of values of the sensorimotor space associated with the region R_i . New regions are created when a region surpasses the maximum number of vector exemplars. When this occurs, R_i is split into R'_i and R''_i , and two new experts E'_i and E''_i are associated to the new regions, respectively.

When an action $SM(t)$ covered by the region R_i is executed, the system stores the error difference between the predicted outcome $S(t+1)$ by E_i and the resulting outcome. All prediction errors are stored by R_i in a list. This list is then used to evaluate the learning progress of E_i , which calculates the decrease of its mean error rate in prediction. This way, the system has an estimation of the learning progress for each region. These evaluations are used to guide the development of the system: it selects the next action to perform taking into account the region where the expert has the highest learning progress estimation.

IAC presents some similarities with prediction error motivational systems. The fact of splitting the sensorimotor space and using the estimated learning progress of experts in the action selection mechanism, though, permits the system to focus on actions that are not or too complicated (as these actions will present a low learning progress) and therefore overcome the problems faced by Schmidhuber where the action selection is based on direct evaluations of the predictor.

Baranès and Oudeyer [8] proposed a novel formulation called *Robust-IAC* or simply R-IAC and demonstrated that it performs better than IAC. In this implementation, a novel action selection and region creation mechanisms were introduced. In R-IAC regions are still split into subregions, but the system also keeps the original regions active, instead of getting rid of them. This allows the system to estimate the learning progress of regions on different scales.

3.2.7 Self-Adaptive Goal Generation R-IAC

Baranès and Oudeyer also proposed an alternative competence-based motivation system called *Self-Adaptive Goal Generation - R-IAC* (SAGG-RIAC) [9, 10]. The system's architecture is similar to R-IAC, where the sensorimotor space of the agent is split into regions, but the focus of the system has changed. Instead of improving $M(t)$ so that the prediction error of $S(t+1)$ diminishes, SAGG-RIAC fixes as a goal a certain $S_g(t+1)$ and it explores the possible actions in $M(t)$ that produce a resulting state $S(t+1)$ that approximates $S_g(t+1)$.

When executing an action to accomplish a goal, the system measures its *competence* as the similarity between the goal and the resulting state. This informa-

tion is stored and used to compute the competence progress of a region R_i , which is a measure that evaluates the evolution of the different actions performed and their outcome. The local competence progress can be *positive*, which indicates that the expected competence gain is substantial, or *negative*, which designates that the system cannot reach the fixed goal in that region and suggests that it could potentially be achieved in a subregion of R_i .

Additionally, the system computes the *interest* value of a region R_i as the absolute value of the evolution of the local competence progress in a time window. This value is used in the goal selection process. Similarly to the action selection process in IAC, the system chooses new goals among those regions for which its associated interest value is higher.

As IAC and R-IAC, this motivation system is adaptive, as the reward for an action (in this case, the competence progress) decreases over time, and homeostatic, as the system continuously searches for situations with a higher expectation for competence improvement. Other researchers have proposed extensions to this system, such as the *FIMO* framework by Hervouet [58, 57].

3.2.8 The Autotelic Principle

Despite being explained in detail in the next chapter, the overview of computational motivation systems should also account for the *autotelic principle*. This motivational system, proposed by Steels [131], operationalises the *Flow Theory* by Csikszentmihály [35] (see Subsection 3.1.6). It is a competence-based, adaptive and homeostatic model of intrinsic motivation.

As in the psychological theory it is based on, the core of the system relies on the relationship between challenges and skills. Agents can self-determine their own goals (called *challenges*) and indirectly evaluate their abilities measuring their competence for these challenges. This evaluation is based on their performance in successive actions while attempting a challenge.

After each action for a specific challenge agents receive a reward, which is computed taking into account the outcome of the action⁵. The system employs the resulting reward to update the evaluation of the challenge attempted.

This information is used to decide the internal state the system has associated with each challenge, which corresponds to the mental states identified by Csikszentmihály when people are involved in an autotelic activity: *anxiety*, *boredom* and *flow*. The *anxiety* and *boredom* states indicate that the relationship between challenge and skills is unbalanced. In these situations, the system tries to return to a balanced situation by addressing challenges that match its abilities. The *flow* state indicates that the system can continue to attempt that challenge. Similarly to people engaged in an autotelic activity, the *flow* state allows the system to develop its skills. As a consequence, its abilities may improve over time and the system can sequentially address challenges of increasing complexity.

⁵In the original proposal, Steels calculates the reward considering only the result of an action. In the implementation presented in this thesis, however, the system also performs an internal evaluation of its abilities.

3.3 Summary

This chapter has first presented an overview of the major psychological theories of motivation. For almost a century, psychologists have proposed various explanations of what motivation is and how it can be formalised. Since the 1930s a shift from pure extrinsic accounts to motivation (as such of Hull or Skinner) to theories that progressively focus more on the intrinsic aspects of it (as de Charms or Csíkszentmihályi). More recently, these two (sometimes opposed) views have been articulated as complementary within the self-determination theory by Ryan and Deci.

Psychological theories have profoundly influenced computational motivation systems, designed to provide agents with an architecture to autonomously interact with their environment. The second part of the chapter has started introducing Computational Reinforcement Learning, the most common technique in CMS. It has continued presenting a selection of some of the most influential computational motivation systems and classified them according to their operational approaches.

Recently there have been some attempts to come up with a unified theory on motivation [50], that takes into account insights from psychology, neuroscience and machine learning. The question of what motivation is, however, has not been resolved yet and remains open.

Chapter 4

The Autotelic Principle

This chapter describes the computational motivational system studied in this thesis, the *Autotelic Principle*. This motivational system was proposed by Steels [131, 133] and is an operational version of the *Flow Theory* by Csíkszentmihály [35, 36]. It is used by artificial agents to manage the complexity of their actions and autonomously regulate their development. An earlier implementation of this system was used by Steels and Wellens [145] in a language evolution experiment. This chapter extends the previous contributions regarding the autotelic principle in that it presents a detailed implementation of the motivation system, explaining not only its general principles but also the different algorithms and the parameters associated to them that determine how it operates.

Although the basic principles of the autotelic principle are shared by both implementations (Sections 4.1 - 4.3), the algorithms that control the autotelic principle (Sections 4.5 and 4.6) have been specifically designed to regulate the complexity in linguistic interactions. The chapter includes a specific description of the process of challenge generation, a explicit characterization of how the improvement for each challenge is monitored after a series of simulations to determine the best parameter values for the system (see Appendix B), the addition of an internal evaluation in order to take into account the differences in knowledge between the interacting agents and a precise description of the challenge selection mechanism for both speaker and hearer. The resulting implementation is the outcome of an exhaustive redesign to adjust the motivation system to autonomously manage linguistic complexity in language evolution experiments.

Following the classification proposed by Oudeyer and Kaplan [90], the autotelic principle can be categorised as a competence-based model of intrinsic motivation, as agents (a) self-determine their own goals or challenges and (b) base the evaluation of these challenges on a measure of competence. It is an adaptive motivation system, as the resulting rewards for an executed action depend on the learned skills (taking into account how well it performs and the degree of certainty it has on its skills) and homeostatic, as it tries to maintain the system in a flow state (balancing the challenges it attempts with its skills).

As in the Flow theory, the core of the autotelic principle relies on the relation between challenges and skills. Artificial agents are able to set their own challenges and use this relation to identify how well they are performing for that certain

task. They do so by monitoring its evolution and use this information to decide when they should change the challenges to address in order to further develop their skills.

The chapter continues with a general view of the system, introducing the definition of challenge (Section 4.1) and its architecture (Section 4.2). It then presents the details of how it has been implemented (sections [4.3 - 4.6]), including a specific description of how the different values associated to a challenge allow the system to identify its internal state, how this information is used to navigate through the space of possible challenges, how the system sets its current challenge and how it generates new challenges.

4.1 The notion of challenge

The autotelic principle¹ (AP) permits agents to define their own *challenges*, which are certain tasks to be done in order to achieve a goal. Being a computational motivational model inspired by the ideas from the Flow theory, the AP must provide agents with mechanisms that allow them to (a) define their own challenges and (b) be able to evaluate their performance for each challenge in order to decide if they are capable or not of successfully achieving a certain task. For example, a robotic agent can have the goal of moving itself around its context or a linguistic agent the goal of communicate about objects in its context. In order to do so, it needs to coordinate the movement of different parts of its body or, in the case of a linguistic agent, be able to produce and understand utterances. Moreover, identifying that it is succeeding in its task requires a proper interpretation of the feedback that its various sensors provide.

In the autotelic principle, *challenges* are specific configurations of a set of parameters in a parameter space. Formally, given a multi-dimensional parameter space P , a challenge p_i is defined as a vector $\langle p_{i,1}, p_{i,2}, \dots, p_{i,n} \rangle$, where $p_{i,j}$ corresponds to the value of the parameter j in the challenge i . Agents are able to generate different challenges by changing the specific configuration of a challenge p_i . Given p_i , agents can create a set of new challenges $\{p_i^1, p_i^2, \dots, p_i^n\}$ by modifying one or more parameters $p_{i,j}$ of that challenge. The space of possible challenges depends on the number of parameters used to define a challenge and the potential different values of each parameter. The particular parameter space (the set of parameters that defines a challenge) depends on the nature of the ultimate goal an agent has to achieve and the different *components* involved in reaching that task.

Additionally, Each challenge has associated a *level* value l_i , which is an integer number in a range of $[1, l_{max}]$. This number represents the difficulty associated to that challenge. In the implementation used in the experiments in this thesis, this number corresponds to the sum of the values of its parameters. This relation

¹The term *autotelic principle* was first used in the context of learning by More and Anderson [85].

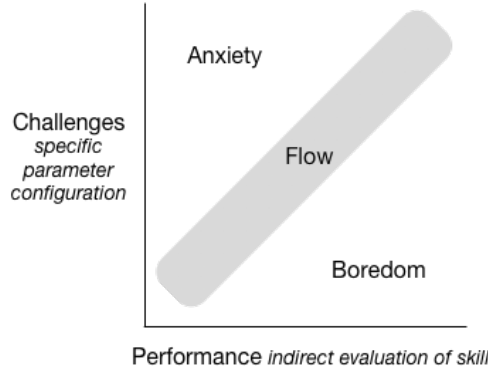


Figure 4.1: Visual representation of the Flow theory in terms of challenges and performance. An agent identifies that it is in a state of *flow* when its performance is neither too high or low for its current challenge. Agents undergo a state of *boredom* or *anxiety* when their performance is consistently too elevated or too poor for their actual task, respectively.

can be formalised as:

$$l_i = \sum_{j=1}^n p_{i,j}$$

4.2 Architecture of the autotelic principle

In the Flow theory, the concepts of *challenge* and *skill* and their relationship are used to determine the mental state of people involved in autotelic activities. We have already introduced how challenges are defined in its operational version, we therefore need to determine how the other element of the relation, *skill*, is modelled.

Unfortunately, skill cannot be directly measured, as there is no way to estimate the proficiency of an agent carrying out an action before it has been executed. Alternatively, the system can evaluate how well an agent is performing. *Performance* is an indirect measure of skill, as it uses the information of the outcome of an action to determine if the goal of a particular challenge was achieved or not in that particular attempt. For example, a robotic agent cannot evaluate how proficient it is in moving. Instead, it can use its sensors to evaluate *how much* its position changed doing a certain action (in this case, a particular set of signals executed in a set of motor controllers in a certain time span). As the different experiments presented in this manuscript use the autotelic principle to regulate the complexity of their linguistic interactions, the system evaluates the performance of a linguistic agent based on the outcome of the linguistic interactions it participates in and an internal evaluation of the agent's capacities to also consider the differences in knowledge between the interacting agents (Section 4.5).

We can now rephrase the general schema of the Flow theory in terms of

challenges and performance. This relation determines a range of different *states*, internal situations that agents undergo using this motivational system which are equivalent to the mental states that people experience when involved in an autotelic activity identified by Csíkszentmihály [35, 36] (Figure 4.1). An agent identifies that it is in a state of *flow* when its performance is neither too high or low for its current challenge. Agents undergo a state of *boredom* when their performance is constantly too elevated and go through a state of *anxiety* when their performance is too poor for their actual task.

4.2.1 Agent as a set of components

A component is an element or constituent of an agent in charge of a distinct, determined subtask. When an agent performs an action, its outcome (the result of its action) is determined by the resulting outputs of the components involved in that particular operation. In consequence, agents can be seen as a set of different components that interact in order to perform actions.

For example, in order to move a robotic agent different components are required, such as motors to control the specific position of the articulated members attached to its joints or sensors used to identify a variation of its position. The components used depend on the nature of the task to perform. If the goal is to move, the agent does not require components to communicate with other agents. These components, in charge of subtasks such as conceptualisation, formulation or interpretation, are essential when agents need to communicate about their context, as in the experiments reported in this thesis.

4.3 Development of skills

A central component of the Flow theory is that a state of flow (which may occur only when there is a balance between the challenge someone is addressing and his or her skills) facilitates the development or learning of skills. An improvement of skills unbalances the challenge-skill relationship and *motivates* the person to attempt a new challenge that will counterbalance his or her amelioration of capacities, seeking to experience the state of *flow* another time.

Similarly, an agent must be able to learn or develop new capabilities. In the autotelic principle, an action is the result of a chain of components (a subset of the elements that compose an agent), where the output of one is the input for the next one. It may occur that one or more components cannot provide the appropriate output, which affects the resulting action.

To overcome this problem, the system takes into account the outcome of the performed action as feedback, and uses this feedback to detect possible failures in components and repair them. The autotelic principle does not establish *how* the learning must take place [131, p. 10]:

They could range from methods to increase needed resources (for example increase the memory available to a component), simple learning mechanisms (such as various forms of neural networks), or sophisticated symbolic machine learning techniques.

The learning methods that have been implemented in this thesis are a collection of *diagnostics*, mechanisms used to identify problems during and after an action has taken place, and *repairs*, processes that attempt to solve diagnosed problems. These learning methods are common in language evolution experiments [120, 46] and were also used in the previous experiment on the autotelic principle by Steels and Wellens [145]. The specificities of the diagnostics and repairs used are explained in the description of the experimental set-ups.

4.4 Deciding what to do next

After an action has been executed, the system identifies the state of the current challenge (either *flow*, *anxiety* or *boredom*). This information is then used to determine the current *phase*, that is, the current situation of the system. Depending on its phase, the system will decide to either (a) keep attempting the current challenge or (b) address a different one. Changing the current challenge can be done in two ways: (a) attempt a challenge already known or (b) address a new one created by adjusting the current challenge parameter configuration.

The phase in which the system chooses to continue with the same challenge is called *operational phase* and occurs when the system is in a state of flow for its actual challenge. In this phase, the system tests itself in a particular parameter configuration. It executes actions and uses the feedback from the resulting actions to improve the performance of the components involved in those actions. If it accomplishes a stable high performance for its current task, the system will eventually turn into a state of boredom. It is also possible that the actual challenge is too difficult for the system, which means that the system failed in developing skills to cope with its current challenge. This results in a steady low performance for that task and ultimately leads the system into a state of anxiety. In both states, the system enters into a *shake-up phase*.

The system enters into the *shake-up phase* when there is no balance between its current challenge and its performance. The system attempts to re-establish an equilibrium again moving to a different challenge. When the system undergoes a state of anxiety for a challenge p_i with an associated level l_i , it first explores if it can move to an already known challenge for which it achieved high performance. Alternatively, it will generate a set of possible choices $\{p_i^1, \dots, p_i^n\}$ by modifying the parameter configuration of p_i . In both cases, the challenge that the agent will attempt in future actions has an associated level l_k , where $l_k < l_i$.

In a state of boredom the system will seek a more challenging task. It will first go through the known challenges and will select one challenge among those that presents a higher difficulty than the current one for which it had a track of low performance in the past. If it cannot find any, it will generate a set of new challenges adjusting the parameter configuration of p_i and will select one of them as the new challenge to attempt. In both cases, the chosen challenge to attempt p_i^i will have an associated level l_k , where $l_k > l_i$.

The combination of these two phases allows the system to self-regulate its own development, as it provides the system with tools to decide when it should continue addressing a task in order to develop its skills and improve its performance

but also a mechanism to adjust its challenge when there is a steady imbalance between challenge and performance.

When the system is used by agents that interact between them, as in the experiments reported in this thesis where agents play language games, the resulting performance of an action depends on the skills of the agents involved in that interaction. In these cases, as a consequence, the autotelic principle controls the skill development and the increase of complexity in the attempted tasks by the population. An agent will fail in more complex challenges if the other agents in the population are still struggling in less complex tasks, even when it has a high performance level for those tasks. Therefore, when this system is used by the interacting agents, it self regulates the development at both individual and population levels.

4.4.1 Generating new challenges

The system could potentially generate all challenges in the challenge space (that is, all the different parameter configurations possible) every time it is in the shake-up phase. In order to regulate this space, the system uses some heuristics to restrict the number of possible challenges that will be created from its current parameter configuration. The problem of how many challenges to create is linked with a second one, namely *how* to select which challenge to address from the set of known challenges. How can the system then (a) limit the set of new challenges it may generate and (b) how it determines the new challenge to attempt?

Steels [131] did not propose an explicit mechanism of how to restrict the number of new challenges. This implementation of the autotelic principle follows Steels and Wellens [145], restricting the number of new challenges created in a shake-up phase using its associated level. The set of new challenges will be formed only by parameter configurations where (a) its associated level value is one unit lower or higher than source challenge, depending on the state of the system (anxiety or boredom, respectively), and (b) the new challenges are created changing only one parameter from the reference challenge, increasing or decreasing its value by one unit. Formally, all the generated new challenges must fulfil two conditions:

- $|l_k - l_i| = 1$, where l_i corresponds to the level of the current challenge and l_k to the level of the challenge candidate.
- $1 = \sum_{j=1}^n |p_{k,j} - p_{i,j}|$, where $p_{i,j}$ and $p_{k,j}$ correspond to the value of the parameter j in p_i and p_k , respectively.

Challenges that fit these restrictions are added into a list of known challenges by the system. This list is used to determine the current challenge of the system (see Subsection 4.6.1).

4.5 Updating the internal state

So far, the chapter has explained how the system can create challenges and move between them depending on its associated internal state. This section describes

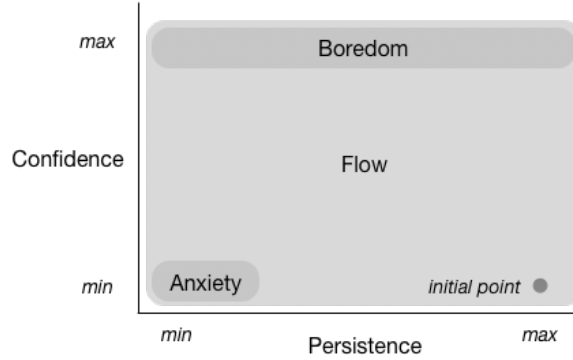


Figure 4.2: Visual representation of the relation between the confidence and persistence values of a challenge and its association to the different internal states. Challenges are initialised with a maximum value of *persistence* and a minimum value of *confidence*. The system reaches a state of *anxiety* when both persistence and confidence values are minimum, and a state of *boredom* when the confidence value has reached its maximum. In both cases, the system enters in a shake-up phase in order to change the challenge to attempt. In all other cases the system stays in the operational phase, which is identified as the state of flow.

how the internal state of a particular challenge is computed and updated. The system monitors the evolution of a challenge and uses the information it gathers to determine how well it performs and, eventually, resolve if it has achieved the goal of that challenge or not. This is done by assigning to each challenge two values that are updated after every action the challenge is attempted and oscillate on range of values $[min, max]^2$:

- **Confidence:** it represents the certainty the agent has of being proficient in a specific task, and it is related to its performance. The higher this value is, the higher the average performance attempting that challenge is.
- **Persistence:** it is a measure to ensure a minimum number of attempts before the system can decide to change its current task. The reason for such a measure is that usually it takes some time before the system is able to develop skills that increase its performance for its current task. This measure prevents the system from continuously entering into the shake-up phase.

Together, these two parameters are used to determine the *internal state* of a challenge (Figure 4.2). The first time a challenge is attempted it is initialised with a minimum value for confidence, as the system has no experience for that particular task, and a maximum value of persistence, to prevent the system from entering the shake-up phase during the first trials. These values were already part of the implementation of the autotelic principle by Steels and Wellens [145].

²In the experimental results reported in this thesis, the range has been set to $[0.0, 1.0]$ for both confidence and persistence values.

However, the way that the confidence value is updated in this implementation has been improved (a) incorporating an internal evaluation measure to the algorithm (see Section 5.5 for an analysis of the impact that performing an internal evaluation has in the development of a shared language) and (b) performing a series of simulations to determine the best parameter values (see Appendix B).

Both values are updated after each time the system performs an action in a given challenge. Confidence is updated taking into account the result of the action carried out. Additionally, the current implementation also considers an evaluation of the different components involved in that action.

Algorithm 1 Challenge update

```

procedure UPDATECHALLENGE(agent, chali, success?)
  confi  $\leftarrow$  CONFIDENCE(chali)
  persi  $\leftarrow$  PERSISTENCE(chali)
  if success? then
    confi  $\leftarrow$  confi +  $\delta_{inc\_conf}$ 
  else
    confi  $\leftarrow$  confi -  $\delta_{dec\_conf}$  + INTERNALEVALUATION(agent, chali, success?)
    if confi = minconf then
      persi  $\leftarrow$  persi -  $\delta_{dec\_pers}$ 
    end if
  end if
end procedure

```

On the one hand, persistence is decreased by a δ_{dec_pers} when (a) the outcome of the interaction is a failure and (b) the confidence value is in its minimum. In all other cases (when at least one condition is not satisfied) its value remains the same. When persistence reaches its minimum it triggers the anxiety state and the system enters into the shake-up phase. When this occurs its value is set to $-max$. A negative persistence value blocks the challenge to the system for a certain time span, that is, prevents the system from attempting it until it has a positive value again. While it has a negative value, persistence is updated at the end of every action as $pers_i(t) = pers_i(t - 1) + \delta_{dec_pers}$, where $pers_i(t)$ and $pers_i(t - 1)$ are the current and previous persistence values of the challenge p_i . Once it reaches the minimum value, the challenge becomes available again and the persistence is reset to its maximum value.

In all the experiments reported in this thesis the value of δ_{dec_pers} is set to 0.02. The reason for this is to provide agents with enough time to acquire the necessary skills for a particular challenge. This value forces agents to attempt a particular challenge configuration at least 50 times before considering it too difficult and thus triggering the anxiety state. Nevertheless, if the persistence value gets to its minimum, the same value is used to block this challenge for 50 interactions, in order to allow agents to explore other challenge parametrisations.

Confidence, on the other hand, is updated after every action. In this implementation, the system takes into account both (a) the outcome of an action and (b) an internal evaluation of the individual competences of the agent that per-

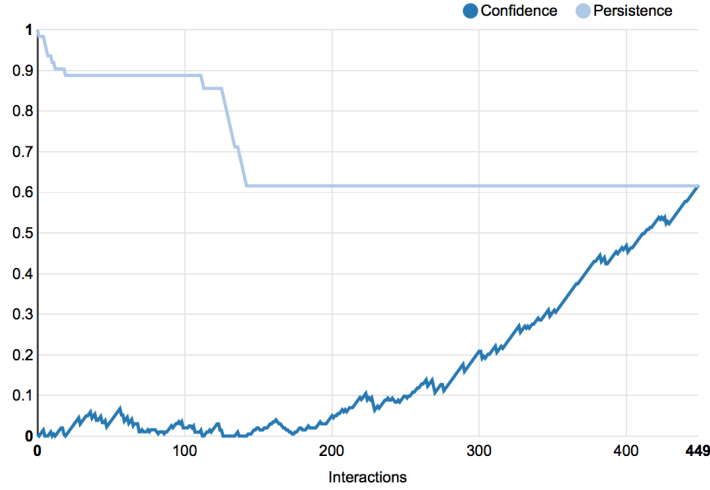


Figure 4.3: Illustration of the evolution of the confidence and persistence values of a challenge, obtained from data from the experimental results. Persistence and confidence values are initialised with its maximum and minimum values, respectively. After every interaction, both values are updated. In this example, during the first interactions when the system is developing new skills there is no stable confidence gain and the persistence value decreases. At some point this tendency changes, there is a stop of the persistence decline and the system starts increasing its confidence.

formed that action when updating the confidence value. The internal evaluation was not part of the original proposal by Steels [131] or the previous implementation of the motivation system [145] and its inclusion is a consequence of the field in which the autotelic principle is employed. The fact that in this thesis the system is used by a population of artificial agents that engage in language games (recurrent communicative interactions) means that the outcome of an interaction, which can be either communicative success or failure, depends on the skills of the two interacting agents (*speaker* and *hearer*). Therefore, the result depends on the abilities of both agents, which may have different linguistic abilities. An internal evaluation has been added for this reason, as it provides agents with a better evaluation of their skills³.

Confidence is updated differently depending on the result of an interaction. When the interaction is a success, meaning that the topic conveyed by the speaker as a sequence of words was correctly interpreted by the hearer⁴, the confidence value is updated as follows: $conf_i(t) = conf_i(t - 1) + \delta_{inc_conf}$, where $conf_i(t)$ and $conf_i(t - 1)$ are the current and previous confidence values. When the interaction is a failure, the confidence value is updated in this way: $conf_i(t) = conf_i(t - 1) - \delta_{dec_conf} + ind_{comp}$, where ind_{comp} corresponds

³See Section 5.5 for an analysis of the impact of performing an internal evaluation has in language evolution experiments.

⁴See "Language games and technical background" (Chapter 2) for an introduction to language games.

to the internal evaluation of the individual competences by the agent and always has a value inferior to δ_{dec_conf} . Formally, For every interaction $I(t)$, $|ind_{comp}| < |\delta_{dec_conf}|$. An example of the evolution of confidence and persistence values of a challenge is presented in Figure 4.3.

In all the experiments reported in this thesis the values of δ_{inc_conf} and δ_{dec_conf} have been set to 0.005 and 0.02, respectively. These values were determined after testing different parameter configurations in the first experimental set-up (Section 5.3). These parameter configuration demonstrated to provide the best conditions for the development of language abilities. Appendix B presents the results of the different simulations carried out to determine the values for δ_{inc_conf} and δ_{dec_conf} .

The internal evaluation, ind_{comp} , differs depending of the role agent in the interaction, as speaker and hearer have access to different information. If the agent acted as a *speaker*, the evaluation considers (a) the number of possible utterances that can be formulated given the conceptualised meaning, (b) if the speaker could express all meaning predicates in formulation and (c) a measure of how compact the lexicon of the agent is. In the case of the *hearer*, the evaluation takes into account (a) if the agent could comprehend all the words of the utterance and (b) if the recovered meaning was ambiguous. Each of these measures can get a maximum value of 0.01. The ind_{comp} value resulting of the addition of these measures, however, always has a value inferior or equal to δ_{dec_conf} . Therefore, the ind_{comp} value in a range of $[0.0, 0.02]$. The internal evaluation is shared among all the experiments reported in this thesis where agents make use of the autotelic principle.

4.6 Initialisation of the system

A key component of the system is how it is initialised. It may start with any challenge configuration possible and, by the dynamics explained in Section 4.4 and appropriate learning methods for its task, it will eventually attempt a challenge from which the system can begin developing its skills. Moreover, the system can start with a set hard-coded skills and the goal of further developing them.

In this implementation agents are initialised without any hard-coded skills. They are provided with a set of learning mechanisms that can be recruited and used when needed in order to acquire or develop skills. This is due to the fact that the system is used in language emergence experiments as a way to manage linguistic complexity, which is widely assumed to have developed across different stages ⁵. In other words, it is used as a tool to scale the complexity of recurrent communicative interactions between autonomous agents, where more complex descriptions of a topic reuse skills which have been developed in earlier stages.

Consequently, the system adopts a bottom-up approach. It is provided only with those challenges that are associated to the lowest challenge level possible. This approach is shared with the implementation by Steels and Wellens. This permits agents to develop the fundamental skills that will be required in further,

⁵ See Subsection 1.3.3.

more complex challenges later on without losing time (interactions) finding a good initial challenge. Once these skills have been learned the system can then move to more difficult challenges, which results in an incremental development of the agent's capabilities and a optimisation of the interactions needed to accomplish the main task of the experiment.

4.6.1 Deciding the current challenge

Given the set of known challenges by the system at a certain time, how does it decide *which* one should be addressed? In the implementation by Steels and Wellens [145] the speaking agent does not perform an evaluation to decide which challenge use in the next action but, instead, it randomly selects one from the list of known challenges that have a positive persistence value. In some cases, the agent may select to address a challenge below the highest level known, slowing down the development of the agent's skills. The algorithm implemented in this thesis (Algorithm 2) aims to improve this decision process and makes use of the level, confidence and persistence values to determine the challenge in the next action.

Algorithm 2 Challenge decision

```

procedure SELECTCURRENTCHALLENGE(system)
   $known_{chal} \leftarrow \text{CHALLENGES}(\textit{system})$   $\triangleright$  List of known challenges
   $pos_{chal} \leftarrow \emptyset$   $\triangleright$  List of possible challenges initialised to  $\emptyset$ 
   $cand_{chal} \leftarrow \emptyset$   $\triangleright$  List of candidate challenges initialised to  $\emptyset$ 
  for each  $chal_i \in known_{chal}$  do
     $conf_i \leftarrow \text{CONFIDENCE}(chal_i)$   $\triangleright$  Current value of  $conf$  for  $chal_i$ 
     $pers_i \leftarrow \text{PERSISTENCE}(chal_i)$   $\triangleright$  Current value of  $pers$  for  $chal_i$ 
    if  $conf < max_{conf_i}$  and  $pers > min_{pers_i}$  then
       $pos_{chal} \leftarrow pos_{chal} + chal_i$   $\triangleright$  Add  $chal_i$  to  $pos_{chal}$ 
    end if
  end for
  if  $pos_{chal}$  then
     $cand_{chal} \leftarrow \text{MINLEVEL}(pos_{chal})$   $\triangleright$  return challenges with lowest level
  else
     $cand_{chal} \leftarrow \text{MAXLEVEL}(known_{chal})$   $\triangleright$  return challenges with highest level
  end if
  return RANDOM( $cand_{chal}$ )
end procedure

```

This algorithm first restrains the selection to those challenges that are available, that is, that have a positive persistence value, and a confidence value below its maximum (those challenges for which the system does not undergo a state of boredom). It then ranks the resulting set of challenges according to their challenge level, and randomly selects one of the challenges with the lowest challenge level from the set. When the resulting set is empty the algorithm randomly

chooses one of the challenges with the highest value level from the set of known challenges.

A problem has to be considered when applying the autotelic principle to language games: the speaker selects its current challenge at the beginning of the interaction, which has an impact on the topic it may select or in the resulting utterance it produces. It is not straightforward, however, how the hearer should decide which is the communicative challenge for that interaction.

There are different ways to solve this problem. In the one used by Steels and Wellens [145] the hearer uses the same algorithm to select its current challenge and actualises its confidence and persistence values as the speaker does. This procedure, however, can easily misidentify the challenge of the interaction and, therefore, update the wrong challenge. For example, it could be that the hearer knows one challenge p_i that it is still unknown by the speaker, or that the hearer has not yet discovered the challenge the speaker is attempting. If the hearer chooses p_i as the communicative challenge of the interaction, it will update a challenge that does not correspond to the one of the speaker.

This implementation has taken an alternative approach. The hearer makes a guess of the communicative challenge of the speaker based on its known challenges, the meaning representation it could retrieve from the interaction and the context of the scene. In the reported experiments, relevant meaning components, such as different meaning classes and their number in the recovered meaning representation are related to (a) the different parameters that define a challenge and (b) the different challenge levels. The system uses the reconstructed meaning to identify its relevant meaning components. This information is then compared to the list of known challenges by the hearer in order to identify the known challenge most likely to have been picked by the speaker as the communicative challenge of that interaction.

4.7 Summary

This chapter has introduced the autotelic principle, a computational motivation system inspired by the Flow theory, and the implementation details of the version used in this thesis, explaining the specific algorithms that operate it.

The motivational force of the system consist in a continuous attempt to balance two elements: the current challenge of the system, which is defined with a specific configuration in the set of parameters that determine the space of possible challenges, and its performance, which is an indirect measure of the system's skills. The system is provided with learning mechanisms that enable the development of skills, which destabilizes the challenge-performance relation and allows the system to progressively attempt and succeed in more difficult, complex tasks.

Part III

Experiments on Language Emergence

Chapter 5

Experiments on the discrete domain

This chapter describes the first set of agent-based experiments in which a population of artificial agents self-organises a vocabulary to refer to objects in their context and extends it into multi-word utterances with primitive syntactic order using the autotelic principle. The autotelic principle allows agents to autonomously explore different communicative goals (varying the level of description of the objects described from the scene) and decide which of these tasks is better suited for their current skill level, which facilitates the development of communicative abilities. The group of experiments presented in this chapter is inspired by the pioneer work by Steels and Wellens [145]. They were the first to use this motivational system in a language evolution experiment. Regrettably, their code was not maintained and cannot be used anymore. The different experiments reported in this chapter are the result of a new implementation of the motivation system (explained in Chapter 4) and the experimental set-up, which includes the context, the interaction script and the different diagnostics and repairs used. Although a direct comparison between the experiment of Steels and Wellens and the current experimental set-up is not possible, this chapter includes an experiment (Section 5.5) that examines the effect of performing an internal evaluation when employing the autotelic principle in a language evolution experiment. Section 5.7 lists how the different experiments in this chapter extend the experiment by Steels and Wellens [145].

The chapter starts by describing the language game played, the set of different diagnostics and repairs that agents have available during the experiment and the mechanism that the population use to align their vocabularies (Section 5.1). The chapter continues with the baseline experiment in which the population does not use the autotelic principle to autonomously regulate their development (Section 5.2), followed by a second experiment with a similar experimental set-up in which agents do have access to the intrinsic motivation system (Section 5.3). Section 5.4 compares the results obtained in both experiments and Section 5.5 analyses the impact that performing an internal evaluation in the motivation system has on language development. It then presents a last experiment (Section 5.6) where the challenge parametrisation has been extended to express the existing spatial

relations between objects and concludes with a list of the contributions of this chapter to the previous work by Steels and Wellens (Section 5.7).

5.1 Description of the experiment

The goal of the experiment is to develop a shared language in a population of artificial agents in order to communicate about objects. The communication occurs in the form of interactions between two agents randomly picked from the population, one acting as a speaker and the other as a hearer.

5.1.1 World and context

In each interaction, both speaker and hearer have the goal to communicate about one topic (one or two objects) from the *context* or *scene*, the environment in which interactions take place. The set of all possible scenes in an experiment is called *world*. Objects have three different physical feature values which are divided between *prototypes* and *properties* (shape and colour):

- **prototype.** A discrete value that specifies the class of the object. It has seven possible values: *table*, *chair*, *glass*, *window*, *lamp*, *clock* and *box*.
- **shape.** A discrete value that specifies the shape of the object. It has six possible values: *squared*, *round*, *triangular*¹, *pentagonal*, *hexagonal* and *octagonal*.
- **colour.** A discrete value that specifies the colour of the object. It has eight possible values: *blue*, *green*, *red*, *yellow*, *orange*, *purple*, *white* and *black*.

Objects are unique, which means that in the world there is not a pair of objects that share the same feature values for the three physical characteristics. Both prototypes and properties are formally described using first order predicates. For example, an object obj_i with values *table*, *squared* and *green* is represented as follows:

$$obj_i = \{prototype(obj_i, table), shape(obj_i, squared), colour(obj_i, blue)\}$$

A scene is composed of two different objects and a spatial relation between them, also in the form of a first order predicate, which is restricted to *next-to*, *far-from* and *left-to* (the latter is equivalent to *right-to* changing the order of the elements). For instance, a possible scene s_k involving obj_i and obj_j (with values *table*, *round* and *blue*) and a spatial relation of the two objects of closeness could be described as follows:

$$s_k = \{prototype(obj_i, table), shape(obj_i, squared), colour(obj_i, blue), \\ prototype(obj_k, table), shape(obj_k, round), colour(obj_k, green), \\ next-to(obj_i, obj_k)\}$$

¹This value is not present in the objects used in the different experiments of this chapter (see Table 5.1) but it is present in the objects used in Appendix A, where the obj_7 has as values $\{prototype(obj_i, chair), shape(obj_i, triangular), colour(obj_i, black)\}$.

Although objects and scenes can be randomly generated at the beginning of each experimental run, in order to be able to analyse better and compare the results reported in this chapter, both the number of objects and the number of scenes that form the world have been fixed to ten (see Table 5.1). In the experiment, the context of the interaction is one randomly selected scene from the set of possible scenes Table 5.2.

object	prototype	shape	colour
<i>obj</i> ₀	glass	round	blue
<i>obj</i> ₁	lamp	pentagonal	red
<i>obj</i> ₂	window	round	red
<i>obj</i> ₃	box	round	green
<i>obj</i> ₄	glass	squared	orange
<i>obj</i> ₅	clock	pentagonal	white
<i>obj</i> ₆	clock	round	blue
<i>obj</i> ₇	chair	pentagonal	black
<i>obj</i> ₈	table	octagonal	purple
<i>obj</i> ₉	lamp	hexagonal	yellow

Table 5.1: Feature values of the different objects of the experiment in the discrete domain.

scene	0	1	2	3	4
objects	<i>obj</i> ₆ , <i>obj</i> ₉	<i>obj</i> ₇ , <i>obj</i> ₅	<i>obj</i> ₄ , <i>obj</i> ₀	<i>obj</i> ₃ , <i>obj</i> ₂	<i>obj</i> ₀ , <i>obj</i> ₁
relation	far-from	left-to	next-to	next-to	next-to
scene	5	6	7	8	9
objects	<i>obj</i> ₉ , <i>obj</i> ₁	<i>obj</i> ₆ , <i>obj</i> ₄	<i>obj</i> ₈ , <i>obj</i> ₂	<i>obj</i> ₀ , <i>obj</i> ₆	<i>obj</i> ₅ , <i>obj</i> ₄
relation	left-to	next-to	left-to	far-from	next-to

Table 5.2: Scenes of the world. This table contains the objects and the spatial relation for every scene.

5.1.2 Multi-word guessing game

The specific language game that the population of agents plays is a *multi-word guessing game*. In this game, the speaker produces an utterance containing one or more words with the goal of describing the selected topic of the scene and the hearer has to use this information to identify it. The dynamics of interactions in this language game work as follows:

1. Before starting the interaction, two agents are randomly chosen from the population, one assigned with the role of *speaker* and the other the role of *hearer*. Both agents share the same context, which consists of a scene randomly selected from the world.
2. The speaker chooses a topic from its context (either one or the two objects of it). It conceptualises the topic into a meaning network that the speaker uses

to *formulate*² an utterance (transmitted as text) which conveys information about the topic. Each word of the utterance either refers to one feature value of an object (prototype, shape or colour) or refers to the existing spatial relation between two objects. In this experiment the speaker has to, at least, refer to one prototype value.

3. The hearer *comprehends*³ the utterance and uses its world model to interpret it, creating hypotheses about what the topic may be. If it has only one hypothesis, it points to the hypothesised topic. If the hearer has multiple hypotheses, it communicates to the speaker that it could not identify the topic.
4. If the hearer's pointing turns out to be right, the speaker signs the hearer that the interpreted topic is correct. In this situation the game is successful. In all the other cases (when there is no pointing by the hearer or it does not correspond with the intended topic by the speaker) the game is a failure and the speaker provides feedback by pointing to the intended topic.
5. At the end of the interaction both agents align their vocabularies based on the outcome of the interaction (success or failure) and the feedback provided by the speaker.

5.1.3 Diagnostics and repairs

Each agent in the population starts the experiment with an empty lexicon. Agents can extend their vocabulary employing different *diagnostics*, used to identify problems during an interaction, and *repairs*, procedures to solve diagnosed problems.

Diagnostics and repairs allow agents to create and learn FCG constructions. Constructions are stored in the *construction inventory* or *lexicon* of the agent, which contains its vocabulary and grammar. The construction inventory is used to *formulate*, that is to verbalise a conceptualised meaning, and to *comprehend*, that is to extract a meaning representation from an input utterance.

The first two diagnostics and repairs available to agents correspond to the processes of creating and learning lexical words. These learning mechanisms are usually found in language evolution experiments [136] and were already present in the experiment by Steels and Wellens [145]. Nevertheless, they were reimplemented in the current version of Babel [143].

Create a word: the speaker does not have a word to refer to (a) a particular feature value of an object or to (b) a spatial relation between objects in formulation.

- *Diagnostic:* the speaker cannot express a meaning predicate that specifies a feature-value or a spatial relation in the topic conceptualisation.

²Formulation is the process by which an agent conveys a meaning network as an utterance, which may be composed of one or more words, using the constructions of its construction inventory.

³In comprehension an agent tries to reconstruct the meaning conveyed in an utterance using the constructions in its construction inventory.

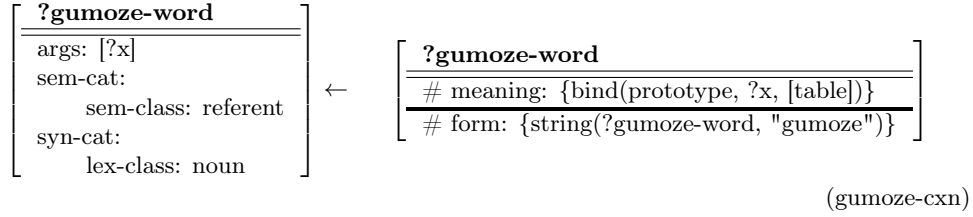


Figure 5.1: Example of a construction C that results from a process of word creation or adoption. In this case, C relates the feature-value *prototype-table* to the word "gumoze". Additionally to this meaning-form relation, the agent can infer from the feature type (*prototype*) that the semantic and syntactic classes should be *referent* and *noun*, respectively.

- *Repair*: the speaker invents a new term t for the unexpressed meaning predicate m and creates a new construction C that relates m with t .

Adopt a word: the hearer cannot identify the meaning predicate associated to an unknown word t , which may refer to (a) a feature value of an object or to (b) a spatial relation between objects.

- *Diagnostic*: the hearer encounters an unknown word in the input utterance.
- *Repair*: the hearer uses the feedback from the speaker and tries to determine the meaning predicate m linked to the word t . When it can identify m it creates a new construction C that relates m with t .

These two processes (creating or adopting a word) produce constructions with the same structure. An example relating the feature value *prototype-table* to the word *gumoze* is shown in Figure 5.1. Note that the resulting construction C does not only contain information about form and meaning but also about semantic and syntactic classes. These classes differ whether C refers to a prototype or a property (colour and shape). For mnemonic reasons these classes have comprehensible labels ($\{referent, noun\}$ and $\{modifier, adjective\}$, respectively).

It is important to note that adopting a word is only possible when the hearer can unequivocally deduce the meaning associated to that word. In certain cases this would not be possible. For example, if the hearer fails to comprehend an **n** word utterance because certain or all words are unknown it will not be able to infer the meaning associated to each word unambiguously and, therefore, will not learn any new lexical construction.

In addition to these two processes, agents can also detect grammatical problems. Problems can be of *word order*, how multiple words should be sequentially organised in an utterance, or of *variable equality*, how to identify which words refer to which object.

Grammatical problems are linked to a lack of information on different features of the resulting transient structure⁴ (TS) when formulating or comprehending an

⁴See Section 2.3 on FCG.

utterance. Agents can solve these problems by creating or learning grammatical constructions, which introduce additional semantic and syntactic information to the transient structure that is used to relate different lexical units. These constructions make use of (a) the different semantic and syntactic labels assigned to lexical units that refer to properties (colour and shape) and prototypes of objects and (b) the *args* feature-value, used to identify which object in the scene each lexical construction refers to.

On the one hand, a problem of word order will occur when there is not enough information in the form features of the TS about how the different lexical elements (words) should be sequenced. For example, it may be that the form features in the resulting TS of the speaker have the following information: $\{string(?x, "gumoze"), string(?y, "wizake")\}$. At this point, the speaker does not know how these words must be sequenced and can formulate two utterances: "gumoze wizake" and "wizake gumoze". This uncertainty can be solved with a construction that introduces information about how these two words should be ordered, for instance $meets(?x, ?y)$ ⁵.

On the other hand, a problem of variable equality will take place when the variables are not correctly connected in the meaning predicates of the TS. For example, it may be that a hearer has retrieved the following meaning: $\{prototype(?x, table), colour(?y, blue), prototype(?z, chair)\}$. With this information the hearer ignores which prototype should be linked to the colour "blue". It will be only in the interpretation phase that the agent will be able to infer this information, as long as the scene does not contain two objects with colour blue. It can use the information gathered in the interpretation process and the feedback from the speaker to add a grammatical construction to its lexicon that correctly links the *args* values of the lexical constructions in the TS. An example of a grammatical construction is shown in Figure 5.2.

Grammatical problem during formulation phase: the speaker formulates an utterance that contains two or more words and does not know how to sequence them.

- *Diagnostic*: the speaker can formulate multiple utterances with the same lexical elements, as the information about the form is not explicit enough about how these elements should be ordered.
- *Repair*: the speaker invents a new syntactic construction C that imposes new form restrictions on the lexical elements of the utterances.

The same problem may occur in comprehension. The order of the words of an utterance may be incorrect according to the grammatical constructions known by the hearer or the recovered meaning may be not correctly linked. In this case, the hearer can diagnose a problem and learn the grammatical construction used by the speaker in that interaction.

Grammatical problem during comprehension phase: an input utterance containing two or more words (a) does not match the grammatical order

⁵Note that another possible order may be $meets(?y, ?x)$.

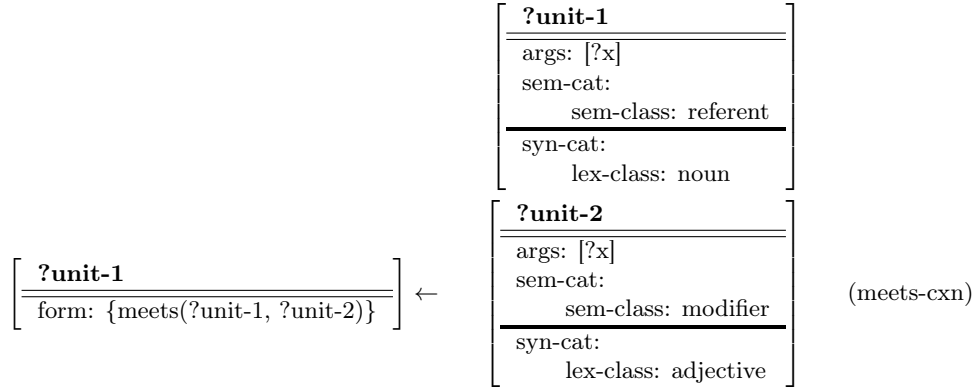


Figure 5.2: Example of a construction C that introduces a restriction in the word order between two lexical units. In this case, C will introduce a *meets* relation between units *?unit-1*, *?unit-2* when (a) the first has as semantic and syntactic classes *referent*, *noun* and the second *modifier*, *adjective* and (b) both refer to the same object *?x* (denoted by using the same variable in the *args* feature in two different lexical units).

known by the hearer or (b) the *args* values in the lexical units are not correctly related.

- *Diagnostic*: the input utterance is not grammatically correct according to the hearer's knowledge, as the lexical elements are ordered in an unknown way or there the resulting meaning structure is not properly linked.
- *Repair*: the hearer creates a new syntactic construction C with the form and meaning restriction information observed gathered from (a) the input utterance, (b) its interpretation process and (c) the speaker's feedback.

5.1.4 Alignment

The mechanisms explained in the previous subsection allow agents to create and learn new lexical and grammatical constructions. The collection of form-meaning mappings known by an agent determines its *construction inventory* or *lexicon*. Within a lexicon it may happen that some of its constructions are in *competition*. This problem occurs when (a) constructions with different form convey the same meaning (*meaning competitors*) or (b) the same form is used in distinct constructions to express different meanings (*form competitors*).

We can illustrate this competition with an example: suppose that an agent has in its lexicon two lexical constructions, C_1 and C_2 , that refer to the same meaning *[table]* but have different form values (for instance "gumoz" and "tepa-li"). In formulation the agent will have to decide which one to use to convey the meaning of *[table]*, which makes the two constructions meaning competitors. The same problem will occur with form competitors in comprehension, as the same

form will convey different meanings. The agent, which in this case is acting as a hearer, will not know which was the intended meaning by the speaker.

Agents have a mechanism to avoid competition among constructions, called *alignment*. The goal of alignment is to drive the preference of agents on which constructions they should use. Each construction is assigned with a *score*, a number in the range $[0.0, 1.0]$. The score is initialised at 0.5, independently of whether the construction has been created or learned. When the lexicon contains competing constructions, their score is used to decide which one should be used, selecting that with the highest score. After each interaction, the interacting agents update the scores of the constructions used (and their competitors). When the score of a construction reaches its minimum (set to 0.0), that construction is removed from the construction inventory of the agent.

The alignment used in this experiment takes into account the outcome of the interaction (whether it is a communicative success or a failure) and the role of the interacting agents (speaker or hearer) to update the scores of constructions, according to the dynamics of lateral inhibition [155]. When the interaction is a communicative success, both agents increase the constructions used by δ_{inc} and decrease their competing constructions by δ_{dec} . If the interaction was a communicative failure, the speaker will decrease the score associated with the constructions used by δ_{dec} only when the utterance was of length one, as it is the only situation in which it can undoubtedly identify that the lexical construction used was not comprehended. In the case of the hearer, it aligns its lexicon in an unsuccessful interaction as it will do if it was a communicative success when the speaker's topic was among its hypotheses. In the experiment both δ_{inc} and δ_{dec} are set to 0.1. A pseudo code of the algorithm used in alignment is presented in Algorithm 3.

Algorithm 3 Alignment

```

procedure ALIGNMENT(agent)
   $used_{cxs} \leftarrow \text{CONSTRUCTIONS-USED}(\textit{agent})$ 
   $comm_{succ} \leftarrow \text{COMMUNICATIVE-SUCCESS?}(\textit{agent})$ 
   $role \leftarrow \text{ROLE}(\textit{agent})$   $\triangleright$  Role in the interaction: speaker or hearer
  if  $comm_{succ}$  or
     $\{role = \textit{hearer} \text{ and } \text{TOPIC-AMONG-HYPOTHESES?}(\textit{agent})\}$  then
    for each  $cxi \in used_{cxs}$  do
       $comp_{cxi} \leftarrow \text{COMPETITORS}(cxi)$ 
      INCREASE-SCORE( $cxi$ , 0.1)
      DECREASE-SCORE( $comp_{cxi}$ , 0.1)
    end for
  else if  $role = \textit{speaker}$  and  $\text{ONE-WORD-UTTERANCE?}(\textit{agent})$  then
    for each  $cxi \in used_{cxs}$  do
      DECREASE-SCORE( $cxi$ , 0.1)
    end for
  end if
end procedure

```

5.2 Baseline experiment

In order to analyse the impact of a motivational system that permits agents to regulate the complexity of their linguistic interactions and thus their shared language development we need first to establish a baseline experiment. In this case, we study how many interactions it takes in a population to converge on a shared language for different communicative tasks.

The baseline experiment consists of a population of agents playing the guessing game described in the previous section. These agents start with an empty lexicon. They enlarge it by adding lexical and grammatical constructions using the different diagnostic and repair mechanisms explained in the Subsection 5.1.3. They align their construction inventories as in Algorithm 3, converging into a minimal, shared number of form-meaning mappings.

In each interaction the speaker formulates a multi-word utterance that refers to one or two objects in the scene that, at least, expresses the prototype of the object(s). The speaker may additionally refer to several properties of the objects to describe. The learning tasks (also named communicative tasks) are defined by the maximum number of properties that an agent can refer to in its topic description:

- **Learning task 1:** agents can describe up to two objects only referring to their prototypes.
- **Learning task 2:** agents can describe up to two objects only referring to their prototypes and one property of a described object.
- **Learning task 3:** agents can describe up to two objects only referring to their prototypes and two properties of the described object(s).
- **Learning task 4:** agents can describe up to two objects only referring to their prototypes and three properties⁶ of the described object(s).
- **Learning task 5:** agents can describe up to two objects only referring to their prototypes and four properties of the described object(s).

The complexity of the communicative tasks is determined by the maximum number of physical features agents can refer to. Therefore, in the list of communicative tasks these are ordered according to their complexity, from the lowest to the most complex task. Importantly, a higher communicative task includes the previous ones. For example, the communicative task 4, where agents can describe one or two objects referring to their prototypes and up to three properties also includes the communicative tasks 1, 2 and 3, as agents can refer to fewer properties than the maximum allowed (that is none, one or two, which corresponds to the previous communicative tasks).

⁶An object has three different physical values (one prototype and two properties). A topic description containing more than two properties obligatorily (a) describes two objects and (b) refers to properties of both objects.

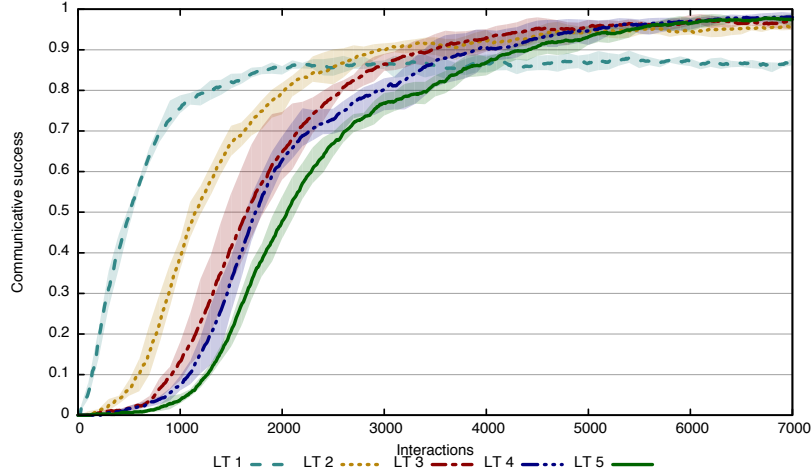


Figure 5.3: Results of a population of 10 agents attempting the different learning tasks, averaged over ten runs. The x-axis represents the number of guessing games played by the population and the y-axis the resulting communicative success, which is smoothed by a sliding window of 100 interactions.

It is also important to note that the number of objects of the topic is unspecified (agents can choose to refer to one or two objects) and, in consequence, the number of prototypes of the resulting utterance may vary. This is important for communicative tasks that allow agents to express more than one property, as in these cases the hearer will have to decide to which prototype each property expressed refers to. For example, an agent can refer to two properties in three cases: (a) describing one object using one prototype and two properties, (b) describing two objects using two prototypes and two properties, both referring to the same prototype or (c) describing two objects using two prototypes and two properties, each one referring to a different prototype.

5.2.1 Experimental results

Each learning task has been tested in a population of ten agents and the results have been averaged over ten runs. The outcome of these runs is shown in Figure 5.3. The x-axis represents the number of interactions in the population (that is the number of guessing games that have been played) and the y-axis represents the percentage of communicative success, which can be either 1 (success) or 0 (failure). In order to better understand the image, communicative success has been smoothed by a sliding window of 100 interactions.

From our analysis it can be said, on the one hand, that communicative tasks with lower complexity (those that allow the population to refer to a smaller number of properties) arise faster at a shared language, in contrast to higher complexity ones, where more interactions are required to achieve a stable, shared language. But converging rapidly to a common language does not mean that the resulting language is better in terms of communication: those communicative tasks that converge faster to a shared language present a lower percentage of

Communicative task	LT 1	LT 2	LT 3	LT 4	LT 5
Communicative success	85.82%	94.95%	96.98%	98.06%	98.30%

Table 5.3: Percentage of communicative success obtained in the last 2000 interactions for each learning task. As at this point the population has already converged to a shared language, the reason why some interactions fail is due to ambiguous description of the topic by the speaker.

communicative success. In other words, there is a relation between the maximum number of properties that can be expressed by the population and the resulting performance of that language.

There is an explanation for such behaviour: shorter utterances consist of a smaller number of lexical words (and therefore are built using a smaller number of constructions). This facilitates agents to learn and agree on the different form-meaning mappings used to create those utterances. As it has been explained in Subsection 5.1.3, agents can only adopt a construction when there is no ambiguity on its meaning-form association. Interactions with long utterances have a bigger chance that the hearer does not know the meaning of some of its words. In such cases, when there are two or more unknown words, the hearer cannot learn any new construction. As a consequence it takes more time for the population to converge to a shared language for more complex communicative tasks.

On the other hand, the possibility to refer to more properties of objects reduces the ambiguity in the topic descriptions. Consequently, longer utterances have more discriminative power than shorter ones. For example, referring only to one prototype to discriminate one object in scene five (Table 5.2) will not be sufficient to identify the topic of the scene, as both objects are lamps (one with properties *hexagonal, yellow* and the other *pentagonal, red*).

In order to analyse this effect we have tracked the ambiguity of the scenes described in each learning task (Table 5.3). The results show that there is a direct relation between the ambiguity of a topic description and the percentage of games where communicative success could not be achieved. For example, for the first communicative task around 14,2% of topic descriptions were ambiguous for only 1.7% in the communicative task 5. These results correspond to the percentage of unsuccessful interactions at the end of the simulations for each communicative task.

Descriptions that refer to all properties of the topic (two if it consists of only one object or four if the topic is composed of two objects) are unambiguous. The reason then for small percentages of ambiguity in all learning tasks is due to the fact that each learning task also includes the lower ones. As a consequence some of the utterances formulated will not include all properties of the topic, producing ambiguities in some cases.

5.3 Addition of the autotelic principle

We have seen (a) that more complex communicative tasks are harder, as they require more interactions between the agents in the population to develop a

shared language, and (b) how a bigger discriminative power leads to a higher percentage of communicative success in the population. In this section we study the case of agents provided with the autotelic principle⁷ and test how the learning tasks in the previous section can act as *challenges* of increasing complexity.

How are these challenges formalised? In the autotelic principle, challenges are defined as a particular instance of a vector made of parameters. In the baseline experiment, learning tasks are defined on the basis of the maximum number of properties that an agent can refer to in its topic description. Challenges can, therefore, be formalised as a vector of one parameter that represents the number of properties that should be used in a topic description ($\langle num_{prop} \rangle$), going from no properties at all (level 1) to a maximum of four properties (level 5). Table 5.4 presents the existing relation between challenges and the learning tasks of the baseline experiment.

It is important to note that there is a difference in the number of properties that agents refer to in learning tasks and communicative challenges: in the former the number of properties that a speaker can express addressing a learning task varies from zero to the maximum number of properties allowed, while in the latter the number of properties to express is fixed. There is an important exception to this restriction associated to the number of objects of the topic: if the speaker refers to only one object the maximum number of properties it can refer to is two, as objects have a fixed number of physical features (one prototype and two properties values).

Learning Task	LT 1	LT 2	LT 3	LT 4	LT 5
Challenge	$\langle 0 \rangle$ [1]	$\langle 1 \rangle$ [2]	$\langle 2 \rangle$ [3]	$\langle 3 \rangle$ [4]	$\langle 4 \rangle$ [5]
$\langle num_{prop} \rangle$ [level]					

Table 5.4: Conversion of the learning tasks of the baseline experiment into challenges, in this case a vector of one parameter with an associated level of complexity.

The goal of the experiment is the same as in the previous section, namely that the population develops a shared language to refer to objects in its environment. The experiment starts with all agents in the population attempting the less complex challenge (level 1), as the results obtained in the baseline experiment show that it is the easiest one to learn.

After each interaction, each interacting agent updates their confidence and persistence values associated to the challenge used as explained in Section 4.5. *Confidence* is updated taking into account the outcome of the interaction (whether it was successful or not) and an internal evaluation of each agent's performance (see Algorithm 1).

While attempting a challenge C_i , agents will autonomously decide to address a more complex challenge C_{i+1} when they experience *boredom* (when they have reached a maximum value of confidence for the challenge C_i) or decide to move back to a less complex task C_{i-1} when C_i results in a too difficult target, undergoing a state of *anxiety* (when both confidence and persistence values associated to

⁷See Chapter 4 on the autotelic principle.

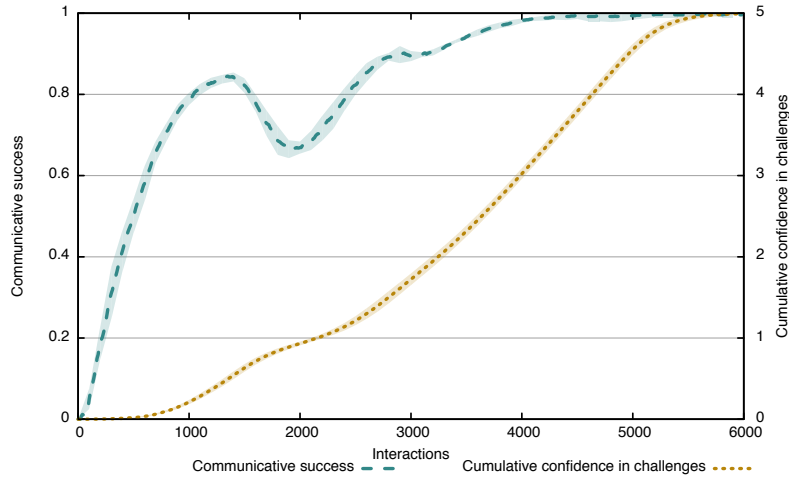


Figure 5.4: Result of ten runs for a population of ten agents provided with the autotelic principle. The x-axis represents the number of guessing games played by the population, the left y-axis the rate of communicative success (smoothed by a sliding window of 100 interactions) and the right y-axis the number of challenges mastered by the population (computed as the average confidence score of each agent in the population over all possible challenges).

C_i are at their minimum). If none of these situations occurs, agents will continue attempting C_i .

5.3.1 Experimental results

The results of ten runs for a population of ten agents provided with the autotelic principle are shown in Figure 5.4. The x-axis represents the number of interactions in the population. The left y-axis represents the rate of communicative success (smoothed by a sliding window of 100 interactions) and the right y-axis indicates the number of challenges that the population masters, measured as the average confidence score of each agent in the population over all challenges.

As was expected from the results obtained in the baseline experiment, the goal of developing a shared language in order to refer to the objects in their environment is achieved. At the end of the runs the population reaches a 100% communicative success, which differs from the outcome obtained in the baseline experiment. It is a consequence of the way challenges are defined, as in each challenge the number of properties that the speaker should express is fixed. Therefore, when (a) agents address and master the higher complexity communicative tasks and (b) their lexicons are aligned, their utterances are not ambiguous and the outcome of all interactions is a communicative success.

The initial challenge for the population is set to the challenge of level 1, which consists of developing a shared language to refer to prototypes. They successfully generate such lexicon in approximately 1200 interactions. At that moment (a) the population has agreed on a shared lexicon for prototypes, (b) the average communicative success of the population is set to a value around 85% (similar

to the resulting communicative success for the first learning task in the baseline experiment) and (c) some agents reach the maximum confidence score for their first challenge, experience boredom and move to the second challenge.

The change of the current challenge in some agents results in a drop of the communicative success in the population. This is due to the fact that, when attempting the challenge of level 2, agents not only refer to prototypes of objects anymore but also to one property. In order to communicate successfully they have to develop a vocabulary for shapes and colours.

Additionally, multi-word utterances present reference issues that can only be managed by developing grammatical constructions that introduce restrictions in the word order and the meaning network. When agents reach maximum confidence for the second challenge (a) their lexicon, which allows them to also refer to both colour and shape properties, is entirely developed and (b) they have also developed and agreed on a grammatical construction to prevent reference problems. Agents thus have a shared lexicon for both properties and prototypes, which will drastically speed up learning the remaining challenges. They will only need to develop other constructions that will introduce new grammatical constraints to refer to multiple properties of objects.

Figure 5.4 presents the results at the population level, but it must be kept in mind that the language development process of each agent is shaped only by the interactions it participates in. This means that some agents can reach maximum confidence for a particular challenge faster and, therefore, move to higher challenges at different times than other agents. For instance, it may be that some agents are still on the first challenge while others have already moved to the second one. This mismatch of current challenge in agents produces interactions in which the communication has a high chance to fail and decreases the confidence score gain in phases where the population gradually shifts from one challenge to another. This effect is best visible at the transition between the two first challenges (see interactions 1500-2500 in Figure 5.4), as it is this transition that requires the most time.

The evolution of the competition between the different terms created by the population and its lexicon size is shown in Figure 5.5. During the initial interactions there is an explosion of words created by agents. This is due to the fact that (a) agents start with an empty lexicon and (b) the interactions between them are still quite limited, which causes that agents are continually creating new terms and learning them from their interaction peers to name the different prototypes in the world. By interaction 500 (around 100 interactions per agent) this tendency stops and the number of words per meaning starts to decrease as a consequence of the alignment process, which reduces the size of the lexicon of agents by deleting the less used constructions.

At approximately the interaction 2000 the competition among constructions rises again as can be seen from the shape of the green line in Figure 5.5. This growth is caused by the population gradually moving to the second challenge: at this point new competing terms are introduced to refer to properties (colour and shape) of objects. This tendency continues until interaction 3000, where a second peak in the number of words per meaning is observed and the average

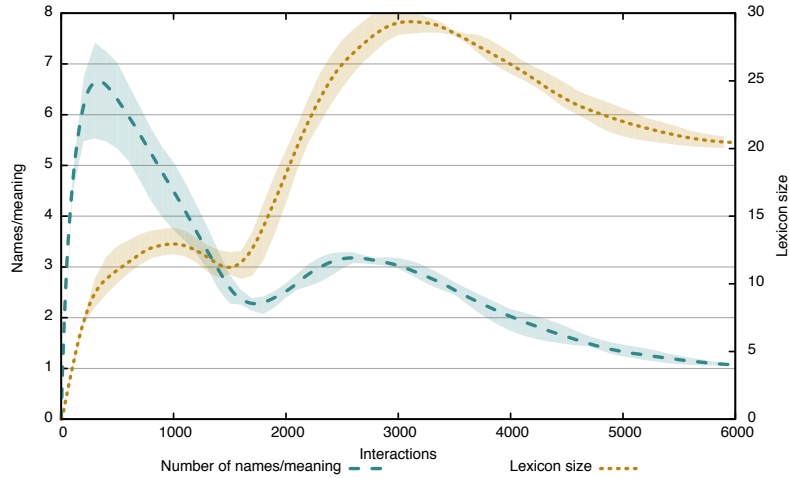


Figure 5.5: Result of ten runs for a population of ten agents provided with the autotelic principle. The x-axis represents the number of guessing games played by the population, the left y-axis the rate of names per meaning and the right y-axis the average lexicon size. Initially the population creates competing names for the concepts to express, but by the end of the runs they have aligned their lexicon to a minimum set. As a consequence, the number of words per meaning decreases until one, which means that there is no lexicon competition.

lexicon size of an agent has 30 constructions.

After this point both the number of words per meaning and the lexicon size gradually decrease, as the alignment process slowly removes the less utilised word-meaning mappings in the population. At the end of the runs the lexicon size has been reduced to a minimal set, causing the number of words per meaning to descend to one, as there is no lexical competition anymore.

5.4 Incremental learning of skills

So far the results obtained, both in the baseline experiment and in the experiment that includes the autotelic principle, have shown that a population of agents is able to develop a shared language (that includes lexical concepts as well as primitive syntactic restrictions) in order to communicate about their context in the form of multi-word utterances. In the previous section we have seen how agents can autonomously manage the complexity of their linguistic interactions by trying to stay in a state of flow. Despite the fact that each agent independently decides when to change the communicative challenge to address, by the end of the runs all agents in the population have maximum confidence for all challenges.

The fact that by using the autotelic principle agents can gradually increase the complexity of their language in a step by step fashion prevents them from attempting utterances that are too complex for their current skill level. This avoids interactions in which the resulting utterance would be too difficult for the hearer and, as a result, it would not be able to infer information from them.

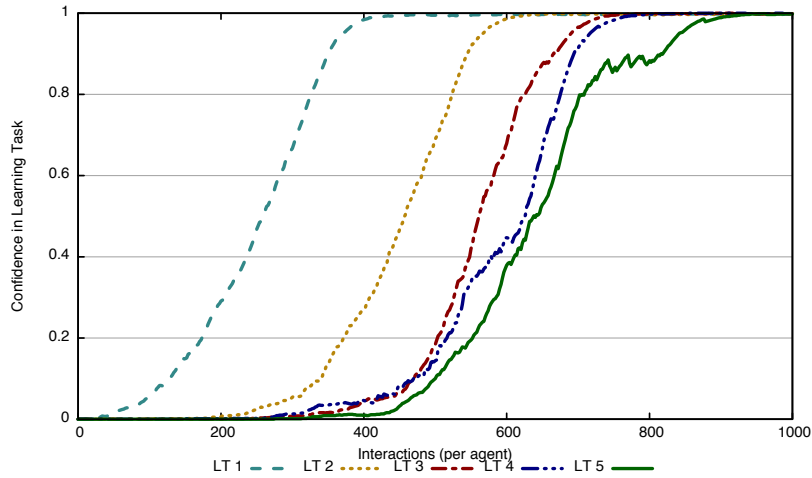


Figure 5.6: Average number of interactions required for reaching maximum confidence in the different learning tasks of the baseline experiment. The results show a relation between the complexity of a learning task and the number of interactions needed to be mastered. LT 1 only requires about 400 interactions, while LT 5 requires almost 1000.

These kinds of interactions occur much more often in the baseline experiment, as some utterances may contain two or more words unknown to the hearer. In such situations the hearer cannot establish the meaning associated to each unknown word and it will not be able to infer the correct word-meaning mappings (see subsection 5.1.3). This poses the following question: do the staging of skills speed up the development of more complex communicative tasks, in comparison to the baseline experiment where the learning of the skills required for a certain task is not organised?

In order to answer this question we have computed how many interactions are required by each learning task in the baseline experiment (Figure 5.6) and in each challenge (Figure 5.7) to reach maximum confidence. The results presented here have been obtained computing the confidence gain for each learning task in the same way as it is calculated in the autotelic principle, despite the fact that *confidence* is a value associated to each challenge generated in the autotelic principle and it is not used at all in the baseline experiment.

The results obtained show that more complex learning tasks require more interactions to reach maximum confidence than less complex ones and therefore are more difficult tasks to learn. This relation is incremental, as an agent attempting the first learning task only needs around 400 interactions to achieve maximum confidence, while an agent addressing the fifth one requires nearly 1000 interactions to master that task.

In contrast, when agents regulate the complexity of their interactions with the autotelic principle this relation is inverted. It is during the first challenges that agents need more interactions to reach maximum confidence. As this motivational system allows them to stage skill learning, in more complex challenges they can

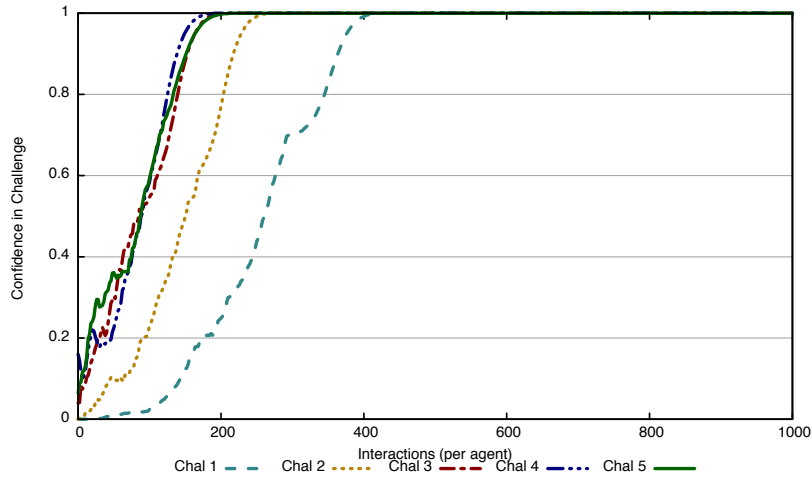


Figure 5.7: Average number of interactions required for reaching maximum confidence for each challenge in the experiment where agents are provided with the autotelic principle. The results show an inverse relation between the time required to reach maximum confidence for a challenge and its complexity. This is a consequence of (a) the reuse of skills learned in previous challenges and (b) the reduction of interactions where the utterance would be too difficult for the hearer to learn from.

reuse the skills previously learned, which speeds up the confidence gain for such challenges. For instance, while the first challenge requires as many interactions as the first learning task (400), the challenge of level 5 only takes 200 interactions to be mastered.

5.5 Analysing the impact of internal evaluation

After examining (a) the impact that the intrinsic motivation system has in the process of language development and (b) how its usage reduces the time devoted to more complex linguistic tasks as agents can rely on skills developed in previous challenges, the question of how the results obtained with the current implementation compare to the ones presented in Steels and Wellens [145] arises.

Unfortunately, the results of both implementations cannot be directly compared. The reasons that do not allow the contrast between both experiments are threefold:

- Firstly, the experimental set-up is not the same in both experiments. These differences are not only the result of using different implementations of the autotelic principle, but also differences in (a) the contexts the linguistic agents should communicate about (including the physical feature values of objects and the composition of scenes), (b) the specific communicative challenges and (c) the kind of language game played: in Steels and Wellens the population plays a *description game*, while in all the experiments reported

in this chapter agents play *multi-word guessing games*⁸.

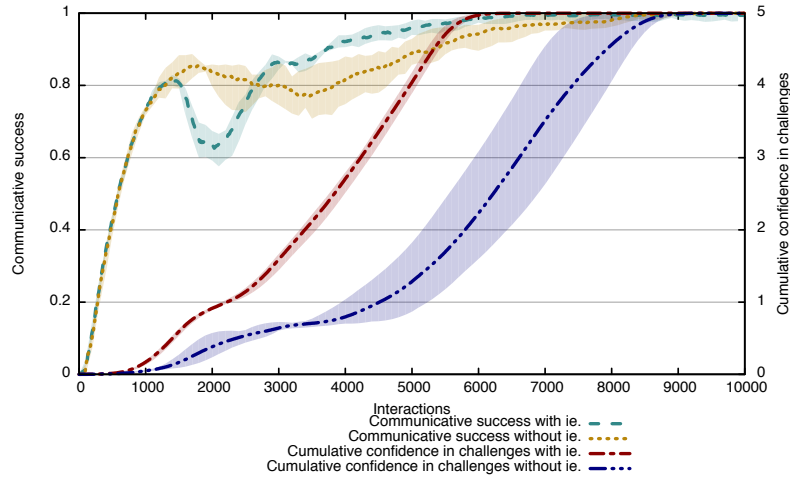
- Secondly, despite being developed in an earlier version of Babel [143], the framework used to implement the experiments of this thesis, the code by Steels and Wellens was not maintained and cannot be run anymore.
- Thirdly, the data used in their experiment is lost, which means that even if the current experimental set-up is modified to match the one used by Steels and Wellens there is no data available to compare the performance of both motivation system implementations.

However, there is a feature in the current implementation of the autotelic principle that was not present in the implementation used by Steels and Wellens and its impact can be analysed: the addition of an internal evaluation measure. This measure has been introduced in the current implementation to take into account the differences in knowledge between the interacting agents, as the outcome of a linguistic interaction is determined by the skills of the agents that participate in it. This section compares the experimental results obtained in Sections 5.3 and 5.4 to an experimental set-up that only differs from the previous one in that it does not perform an internal evaluation when updating the internal state of challenges (Section 4.5). The results of ten runs for a population of ten agents provided with the autotelic principle with and without internal evaluation are shown in Figure 5.8a. The x-axis represents the number of guessing games played by the population, the left y-axis the rate of communicative success (smoothed by a sliding window of 100 interactions) and the right y-axis the average confidence value associated to the different challenge levels.

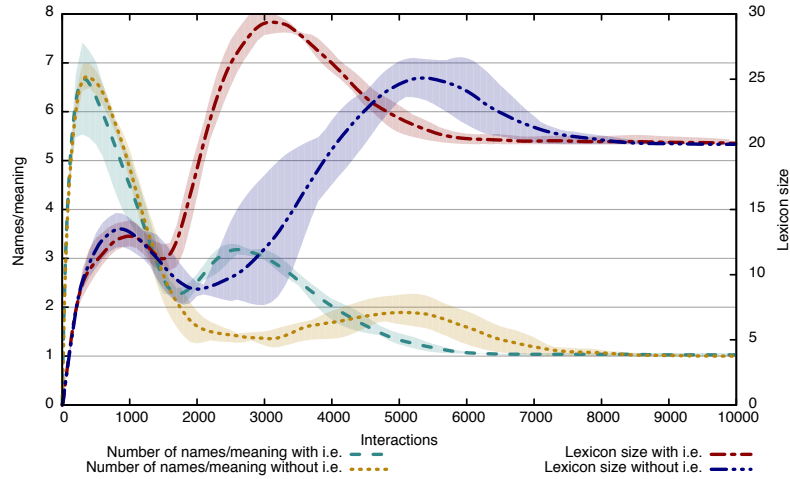
As was expected from the results obtained in the previous sections, the population manages to develop a shared language in both experimental set-ups (i.e., when an internal evaluation is performed and when it is not taken into account). The shape of the communicative success rates for both experimental set-ups are similar: in both cases the communicative success grows during the experiment, presents a drop in the transition between the communicative challenges of level 1 and 2 and reaches a 100% rate by the end of the interactions. However, the results show that performing an internal evaluation increases the communicative success of the population during the experiment, excluding the before mentioned transition from challenges of level 1 and 2.

In addition, performing an internal evaluation allows the interacting agents to gain confidence in the communicative challenges they address faster, which speeds up the development of the shared language. This finding is confirmed by the evolution of lexicon size and the rate of names per meaning during the experiment (Figure 5.8b). As the population makes the transition between challenges of level 1 and 2 faster when an internal evaluation is performed, the different terms to refer to object properties are created and learned earlier. This allows the population to start aligning their inventories sooner. As a consequence, their lexicon reaches a minimal set of word-meaning mappings faster and the population gets rid of lexical competition in less interactions.

⁸See Section 2.1 for a description of the existing language games.



(a) The x-axis represents the number of guessing games played by the population, the left y-axis the rate of communicative success (smoothed by a sliding window of 100 interactions) and the right y-axis the average confidence value associated to the different challenge levels.



(b) The x-axis represents the number of guessing games played by the population, the left y-axis the rate of names per meaning and the right y-axis the average lexicon size.

Figure 5.8: Comparison of the result of ten runs for a population of ten agents provided with the autotelic principle with and without performing the internal evaluation. The results show that when performing the internal evaluation the population (a) gains confidence in the different communicative challenges faster and (b) achieve a higher rate of communicative success during the experiment.

It can be concluded that the inclusion of an internal evaluation measure to the autotelic principle is advantageous when the motivation system is used in language evolution experiments, as it (a) allows the population to move to higher level communicative challenges faster and (b) speeds-up the creation and

alignment of words and the convergence of the shared lexicon to a minimal set.

5.6 Increasing the challenge dimensions

The experiments have focused, so far, on the development of a language to describe objects in a context, where agents can refer to both prototypes and properties. In this experimental set-up, challenges have been defined as a one-dimensional vector, where their only parameter is the number of properties to verbalise. The expressing power of the resulting language can be extended by including the notion of spatial relationship between the objects in the scene⁹. In order to do this, we need to extend the definition of challenges to a two-dimensional vector ($\langle num_{prop}, num_{rel} \rangle$), where one extra parameter is used to specify if the spatial relation between two objects should be expressed or not.

Extending the number of dimensions that determine a challenge configuration requires a new definition of complexity that takes into account both parameters. As the specific parameter values of a challenge determine the number of meaning predicates of the conceptualised meaning to be formulated by the speaker, the complexity of a certain challenge (and consequently its associated level) must be computed based on these meaning predicates.

Level	[1]	[2]	[3]	[4]	[5]
Challenge	$\langle 0, 0 \rangle$	$\langle 1, 0 \rangle$	$\langle 2, 0 \rangle$	$\langle 3, 0 \rangle$	$\langle 4, 0 \rangle$
$\langle num_{prop}, num_{rel} \rangle$			$\langle 0, 1 \rangle$	$\langle 1, 1 \rangle$	$\langle 2, 1 \rangle$

Table 5.5: Challenges and its associated level as a vector of two parameters: number of properties and if a relation between two objects is expressed. The level associated to a specific challenge is determined by the number of linked variables of the conceptualised meaning network it will generate.

We have chosen to define the complexity based on the number of *linked variables* in the meaning network, as it is a measure that reflects not only the number of meaning predicates of a meaning network but also how interconnected they are. There are two kinds of meaning predicates: while those introducing a prototype or property introduce one variable (for example *prototype(?var_i, table)* or *shape(?var_i, squared)*), relations express spatial associations between two different variables (as in *next - to(?var_i, ?var_k)*). Following this definition, the different challenge configurations can be classified in relation to the complexity of the meaning networks they generate (see Table 5.5). For example, two challenges correspond to level 3, as there are two possible challenge parametrisations of equal complexity¹⁰: one introducing meaning predicates that refer to two

⁹As the physical properties of objects, the number of different spatial relations is predetermined. During a scene the spatial relation between two objects does not change. More information can be found in Subsection 5.1.1.

¹⁰Note the number of linked variables in the resulting meaning structures of both challenge parametrisations of level 3 is the same. In $\langle 2, 0 \rangle$, either if the properties refer to the same (*prototype(X, table), colour(X, blue), shape(X, squared)*) or to different objects (*prototype(X, table), colour(X, blue), prototype(Y, chair), colour(Y, green)*) the number of

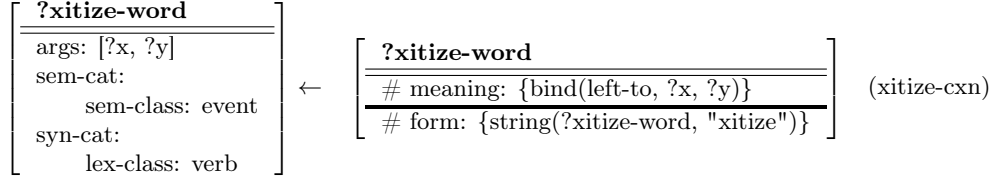


Figure 5.9: Example of a lexical construction C that results from a process of word creation or adoption of a spatial relation. In this case, C introduces a spatial relation *left-to* to the word "xitize". Note that two arguments, $?x$, $?y$, are introduced which latter should be linked to the objects of the relation.

properties of the topic and another that expresses the existing spatial relation between two objects.

As in the case of properties and prototypes, agents need lexical constructions in order to refer to spatial relations. Agents can create and learn these lexical constructions with the diagnostics and repairs presented in Subsection 5.1.3. There is a small difference between a lexical construction to express a spatial relation in comparison to those that refer to a prototype or a property, since its meaning predicate will introduce two arguments instead of one (see Figure 5.9).

Additionally, agents also need to learn new grammatical constructions in order to (a) link the variables introduced by lexical constructions that express spatial relations and (b) insert word-order constraints that facilitate the formulation and comprehension of the resulting utterance. As in the previous grammatical constructions the word order constraints are not fixed but rather randomly decided when the construction is created, which introduces competition between grammatical constructions at the word-order level. An example of such constructions is shown in Figure 5.10.

5.6.1 Experimental results

The challenge configuration and the ability to create and learn lexical and grammatical constructions to refer to spatial relations are the only additions made with respect to the previous experiment configuration. The results of ten runs for a population of ten agents provided with the autotelic principle for a two-dimensional challenge configuration are shown in Figure 5.11. The x-axis represents the number of interactions in the population. The left y-axis represents the rate of communicative success (smoothed by a sliding window of 100 interactions) and the right y-axis indicates the number of challenges that the population masters grouped by levels, measured by calculating the average confidence score of each agent in the population for each challenge level.

The shape of the resulting communicative success resembles the ones obtained in the former experiment configurations. As in the previous experiments,

linked variables is two. The same happens in $< 0, 1 >$, as the meaning predicate expressing the spatial relation introduces two variable linkings (*prototype*($X, table$), *left-to*(X, Y), *prototype*($Y, chair$)).

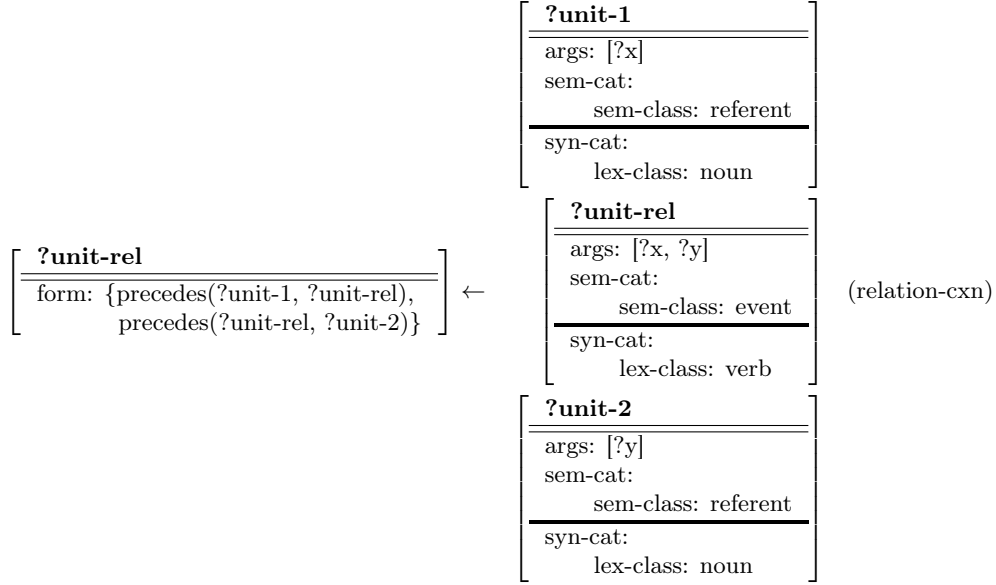


Figure 5.10: Example of a grammatical construction C that (a) links the variables introduced by a relation to two prototype units and (b) introduces word-order restrictions between these units. In this case, C will introduce two *precedes* relations between units *?unit-1*, *?unit-2* and *?unit-rel* when (a) both *?unit-1* and *?unit-2* have as semantic and syntactic classes *referent*, *noun* and (b) their variables refer to different objects which correspond to the arguments of the relation unit *?unit-rel*.

agents start with an empty construction inventory and with the goal of developing a shared language to refer to prototypes as a first challenge. They generate it rapidly, increasing their confidence on the first challenge up to its maximum value around interaction 2000. Note that the communicative success starts to drop before the average confidence value in the population has reached its maximum, as observed in the results of the first experiment using the motivation mechanism. This is due to the fact that some agents have already reached the highest confidence score for the first challenge and they have moved to the next one, where they start to refer to the properties of objects. In order to achieve its new communicative task, these agents need to create new lexical and grammatical constructions to refer to the different colours and shapes in their context. This also causes a decrease in the average confidence gain for the first challenge, as agents that have not yet reached maximum confidence are exposed to more complex utterances, which reduces the number of successful interactions and has an impact on how the confidence score associated to the first challenge is updated.

The communicative success rate rises again once the population starts to share the lexical constructions to refer to colour and shape as a response to the second communicative task, and so does the average confidence value associated with the challenge of level two. As in the previous transition between challenges,

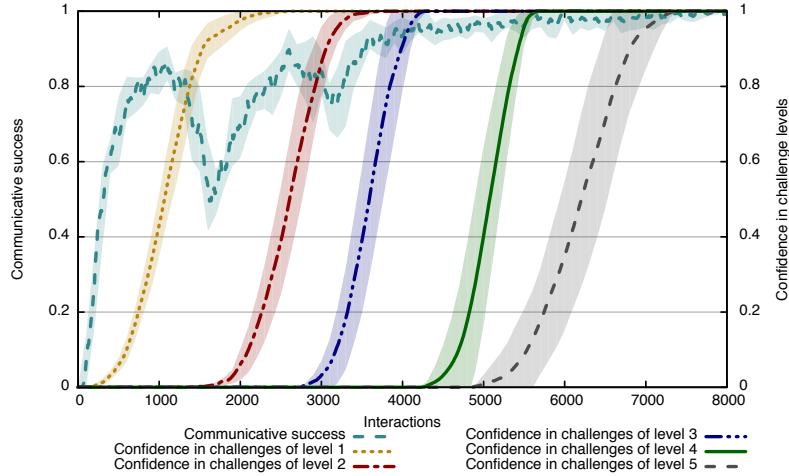


Figure 5.11: Result of ten runs for a population of ten agents provided with the autotelic principle attempting the challenges described in Table 5.5. The x-axis represents the number of guessing games played by the population, the left y-axis the rate of communicative success (smoothed by a sliding window of 100 interactions) and the right y-axis the average confidence value associated to the different challenge levels. The results show that communicative success rate increases as the population attempts more complex challenges, as the shared language progressively gains expressive power. There are two drops in the communicative success gain, which correspond to the periods where the population is enlarging its lexicon to refer to (a) properties and (b) spatial relations. By the end of the simulations all utterances formulated are unambiguous, reaching 100% communicative success.

a drop of communicative success rate and confidence gain is observed again when the first agents move to the challenges of level three. At this point agents can address two challenges: one allows them to increase the expressive power of their lexicon by stating more properties of the topic, while the other extends their construction inventory with new lexical and grammatical constructions to express spatial relations between objects. As long as both challenges have not reached maximum confidence, the decision of which of the two challenges should be attempted is randomly chosen at the beginning of the interaction¹¹. The percentage of communicative success increases again when the population has reached a consensus on how to name the new lexical and grammatical concepts, plateauing to a percentage above 95% by the time that the average confidence score in challenges of level three has reached its maximum.

The four remaining challenges (two of level four and two of level five) extend the construction inventory of the population by adding new grammatical constructions that allow more complex topic descriptions. Once these constructions have been learned and spread over the population, the percentage of ambiguous utterances is reduced to zero, which means that the population can now

¹¹See Subsection 4.6.1 in the chapter on the autotelic principle.

unambiguously describe topics. This can be observed in the percentage of communicative success, which reaches 100% by the end of the simulations. Soon after interaction 7000 all agents in the population reach the maximum confidence value for each possible challenge.

5.7 Contribution to previous experiments

As already mentioned before, Steels and Wellens [145] carried out the first experiment in which the autotelic principle was used in a language evolution experiment. The set of experiments presented in this chapter are inspired on their work and are the result of a new design and implementation of both the motivation system and the experimental set-up, including the scenes, the interaction script and the different diagnostics and repairs used.

Although a direct comparison between the two experimental set-ups is not possible because there is no data available of their experiment, it cannot be run anymore and the experimental set-ups are different (e.g., the population do not play the same language game: a *description game* in Steels and Wellens and *multi-word guessing games* in the experiments of this chapter), both experimental set-ups explore the role of intrinsic motivation in language evolution experiments in a discrete world, where objects have discrete property values. The goal of this chapter was not to reimplement the experiment by Steels and Wellens, but to extend their findings.

This chapter accomplishes this goal by (a) explicitly analysing and comparing the results obtained without using the autotelic principle in a language evolution experiment to those when the motivation system is used (Sections 5.2 and 5.3, respectively), (b) investigating the impact the autotelic principle has in the learning of communicative skills (Section 5.4), (c) proving that the addition of an internal evaluation to the autotelic principle has a positive impact in the development of a shared language (Section 5.5) and (d) testing the motivation system in a bi-dimensional challenge configuration which increases the expressive power of the resulting communication system (Section 5.6).

5.8 Summary

This chapter presented a series of experiments where a population of agents made use of the autotelic principle to self-regulate the complexity of their linguistic interactions while developing a shared language to describe objects in a scene. The first part of the chapter describes the specificities of the world and the language game played: the set of physical features per object, the set of scenes that conform the world, the different mechanisms to create and learn new lexical and grammatical constructions and the way agents can align their vocabularies.

In the second part, the results of a baseline experiment are compared to those obtained when the population is provided with the autotelic principle. The results show that in both experimental configurations the population manages to create a language to describe objects in the scenes. The results obtained in the second

experimental set-up show that agents successfully increase the complexity of the communicative tasks they attempt, as a result of trying to stay in a state of flow, which reduces the number of ambiguous utterances they formulate. Additionally, the number of interactions to achieve maximum confidence for each challenge decreases for more complex tasks, as (a) they can rely on skills developed in previous challenges and (b) reduce the number of utterances that are too complex to learn from. A comparative analysis on the impact of internal evaluation in the motivation system has shown that it improves the development and convergence of the shared lexicon to a minimal set.

Finally, a third experiment is reported where the challenge definition has been extended in order to allow the population to express relations between objects. This case is interesting because, for certain levels of complexity, agents have two communicative challenges to master. As in the results obtained in the second part, agents progressively develop their communicative skills when trying to stay in a state of flow, increasing the communicative success of their interactions until they reach 100%.

Chapter 6

Experiments on a continuous domain: a case study on colour

This chapter describes the second set of agent-based experiments, where a population of artificial agents becomes engaged in linguistic interactions with the goal of developing a shared language for the domain of colour. Similar to the previous chapter, agents need to self-organise a vocabulary to successfully refer to colour samples in their context. They manage the complexity of their interactions using the autotelic principle, starting with simple colour sample descriptions and extending them into multi-word utterances making use of different colour strategies.

The first part of the chapter (sections 6.1 and 6.2) is devoted to the research on the colour domain, explaining what colour prototypes and colour systems are, introducing the different colour strategies used in the experiments and how they can be modelled as a network of cognitive operations. It continues with a description of the experiments carried out (Section 6.3), explaining the different diagnostic and repairs available to the population and the alignment process that agents use to adjust their different sets of prototypes. The last part of the chapter presents three experiments that test the motivation system under different situations. In the first one (Section 6.4) agents use the autotelic principle as a mechanism to regulate the complexity of their utterances. The second experiment (Section 6.5) evaluates the motivation system as a mechanism to control several colour strategies at the same time. Finally, the third experiment (Section 6.6) evaluates the impact of perceptual deviation in the emerged communicative system.

6.1 The domain of colour

The domain of colour has been and still is of great interest to a lot of researchers. This attention is caused by the observed differences in how colours are described in human languages [16]. It is commonly accepted that a *colour space*, the space of colours that can be perceived, is organised in different *colour categories*, subsets of this space (Figure 6.1). *Colour prototypes* are points in the colour space that

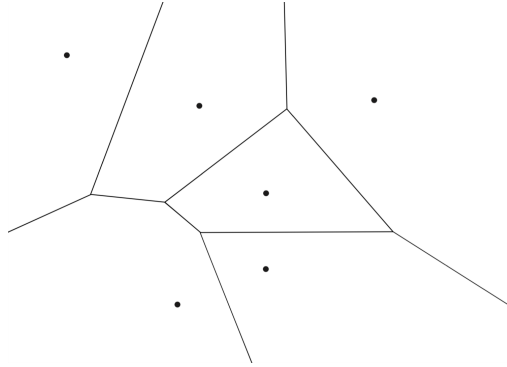


Figure 6.1: A colour space is divided into colour categories that represent subsets of that space. These categories are usually represented by colour prototypes, which are points in the colour space. Image extracted from Bleys [21].

represent a particular colour category in a colour space [98]. It is important to stress that, despite the fact that all human languages divide the colour space into colour categories, the number of colour prototypes and their position in the colour space differs among languages. For instance, while English has eleven basic colours (*red, blue, green, yellow, pink, purple, orange, brown, black, white* and *grey*), Himba has only five (*serandu, zoozu, dumbu, vapa* and *burou*) [95].

Formally, human languages divide the colour space into a set of colour prototypes $\{c_1, c_2, \dots, c_n\}$. Given the colour prototype c_k , its associated cell R_k , which determines the associated colour category, contains every point whose distance to c_k is shorter or equal to the distance to any other prototype c_i (Figure 6.1).

Different colour systems have been proposed to represent colour spaces¹. In all the experiments reported in this chapter I use the CIE 1976 $L^*a^*b^*$ [96] reference. In this colour system, colour samples (also named colour chips) are represented in three dimensions: the L^* dimension represents lightness, the a^* dimension roughly redness-greenness and the b^* approximately yellowness-blueness. The difference between two colour samples is determined by their Euclidean distance. The values of the L^* dimension range from 0 (black) to 100 (white), while both a^* and b^* can have positive or negative values and are only limited by the physical properties of materials [21, p. 192].

Most of the studies on colour have focused on how languages use single terms to refer to colours samples. However, usually this is not the preferred way to describe them: experiments by Simpson and Tarrant [110] and Lin et al. [76] have shown that only 15% of colour samples are characterised using a single colour term when human subjects are asked to describe colour samples without any restriction. These results provide evidence to the fact that commonly people choose to express more information about colour samples than what is possible by using only single terms.

Researchers working on artificial language evolution have shown a particular

¹Interested readers are referred to the Appendix "Colour Spaces and Systems" in [21] for an overview of different colour systems.

interest in the domain of colour. Most of the models proposed have focused on the emergence of single colour terms [140, 15, 11, 12], although there have been some attempts to model more complex colour descriptions.

In this respect, the most advanced contribution to the domain of colour has been made by Bleys. In his doctoral thesis [21] he explains and studies different *language strategies*, a particular method to express one area of meaning. These language strategies are then tested in artificial language evolution experiments, showing how these models can emerge and be learned by a population of artificial agents.

The research reported in this chapter stands on his work, as Joris Bleys kindly gave me access to the original implementation of the language strategies he used in his research in order to do my own experiments. Readers interested in Joris Bleys' research are referred to his multiple articles in conferences and books [24, 25, 22, 23]. In the next section the different language strategies to describe colour samples used in the experiments reported in this chapter are presented. The basic colour, graded membership and colour combination strategies (subsections 6.2.1, 6.2.2 and 6.2.3, respectively) have been adapted from Joris Bleys' code.

6.2 Language strategies for the domain of colour

Human languages employ different methods to describe a colour sample, ranging from the usage of a single term to more elaborate descriptions that involve colour modifiers or combination of colour prototypes. This section does not present an exhaustive record of all language strategies observed in natural languages but rather the part of them which is used in the computer simulations of this chapter.

6.2.1 Basic colour strategy

In the basic colour strategy, a single term is used to describe a colour sample. In this strategy, the speaker first (a) identifies the colour prototype that is the closest to the colour sample he or she wants to mention and (b) uses the term associated with that category to characterise the chosen colour sample (Figure 6.2a).

For instance, English splits the colour space into eleven basic colour categories, represented by the prototypes $\{c_1, c_2, \dots, c_{11}\}$. When using the basic colour strategy, English speakers employ the term associated to the closest prototype c_k to the colour sample they want to refer to. For example, colour samples will be described using the word "blue" when their closest colour prototype corresponds to the blue colour category ($L^*a^*b^*$ values of [51.576, -0.41787, -52.648] according to [149]).

6.2.2 Graded membership strategy

The graded membership strategy (see Figure 6.2b) characterises a colour sample by expressing both the closest colour prototype and the distance between the colour sample to it. As in the basic colour strategy, the speaker has to (a) identify

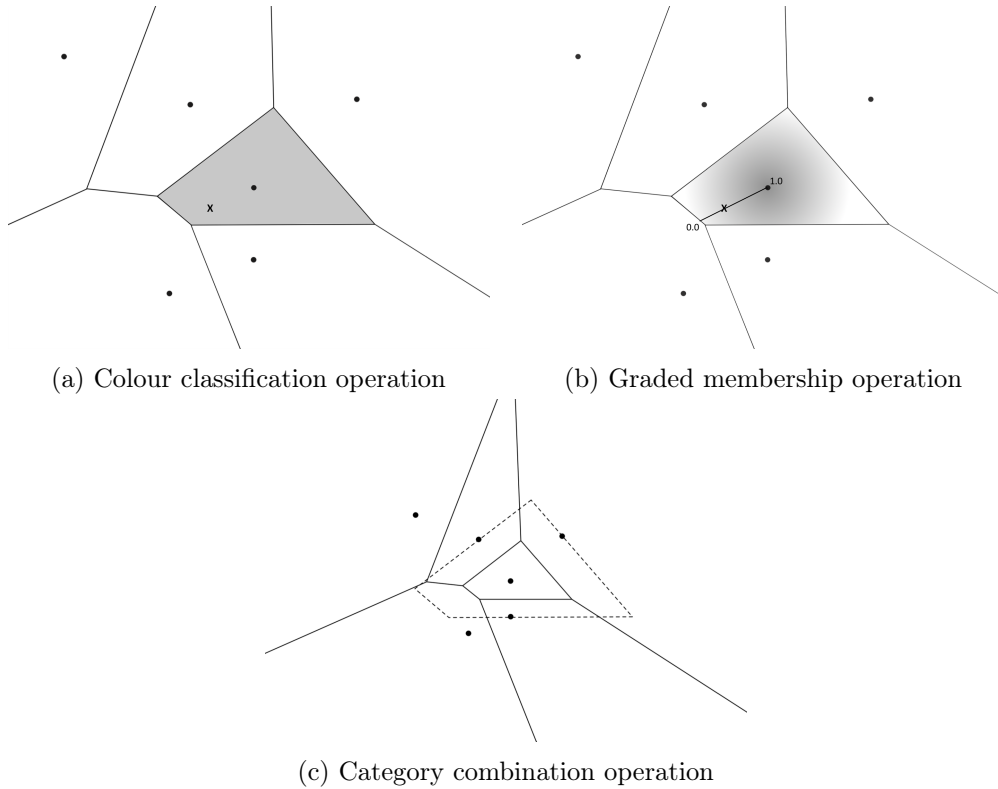


Figure 6.2: Visual representations of different operations used in colour strategies. The basic colour strategy (Figure 6.2a) describes a colour sample referring only to its closest basic colour prototype. In the graded membership strategy (Figure 6.2b) the distance between the colour sample and the closest colour prototype is also expressed. In a category combination operation (Figure 6.2c) the colour space is transformed towards the main colour prototype of the colour sample in order to perform a second classification. Images adapted from Bleys [21].

the colour prototype closer to the colour sample to describe but additionally (b) he or she also refers to the distance between them, which is captured by a set of modifiers.

This strategy is also observed in English, where it is possible to describe a colour sample by combining a basic colour term with adverbs such as “very” (when the sample is close to the basic colour prototype) or the suffix “-ish” (when the distance between the two points is bigger), as in “very blue” or “greenish”.

6.2.3 Colour combination strategy

Another strategy observed in human languages is colour combination. As its name suggests, this strategy consists of describing a colour sample using two basic colours known by the speaker. Different models have been proposed [74, 83], but I chose to use a strategy originally proposed by Bleys [21], that consists in classifying colour samples twice, the second classification being done after a colour

space transformation.

In this model, speakers first (a) identify the closest colour prototype to the colour sample, as in the basic colour strategy, and then (b) transform the colour space towards that prototype (Figure 6.2c). Finally, they (c) classify the colour sample again on the transformed colour space, obtaining a second colour prototype. The colour space transformation shrinks the area of the colour category identified in (a) but does not remove its prototype from the colour space, allowing the classification of the colour sample to the same colour prototype twice, before and after the colour space transformation.

As for the previous language strategies, this one also exists in English. English speakers can describe a colour sample using two colour terms, as in "green-blue" or "yellow-green".

6.2.4 Lightness strategy

The lightness strategy describes a colour sample expressing its closest colour prototype and the variance in lightness between the colour sample and the prototype. There are different approaches to model this strategy. For instance, Bleys [21] modelled it as a combination of colour categories with white and black colour prototypes. For example, "light blue" would be the resulting utterance of the combination of the colour categories "blue" and "white". I opted for a different approach, in order to be able to specify the difference in the value of the lightness dimension between the colour sample and the closest colour prototype.

Similar to the basic colour strategy, the speaker first has to (a) identify the closest colour prototype to the colour sample to describe, but additionally (b) he or she also needs to make explicit the difference in lightness between them, represented as a float number, which is captured by a set of modifiers. In the CIE 1967 $L^*a^*b^*$ colour system the L^* dimension is used to express the lightness of colour chips. Using a lightness modifier on a colour sample changes its value for the L^* dimension, increasing or decreasing it within the range of L^* possible values. Taking this approach the lightness modifier in the utterance "light blue" performs an operation that increases the value of the L^* dimension of the colour category "blue".

6.2.5 Chromaticity strategy

This strategy states how saturated a colour sample is. The level of saturation is expressed by a modifier and has been implemented in a way similar to the lightness strategy. First, the speaker (a) identifies the colour category of the colour sample to describe, and additionally (b) uses a modifier to express the difference in chroma, which is a float number as in the lightness strategy, between the colour category and the colour sample.

The main difference with the lightness strategy is that chromatic modifiers act on the values of two dimensions (a^* and b^*) instead of only one. What a chromatic modifier does is to increase or decrease the *chroma* (C_{ab}^*) value of a colour sample while keeping the same *hue* (h_{ab}^*) value. These values are not

directly represented in the $L^*a^*b^*$ colour system, but can be computed using the following formulas [21, 192]:

$$C_{ab}^* = \sqrt{a^{*2} + b^{*2}}$$

$$h_{ab}^* = \arctan \frac{b^*}{a^*}$$

This strategy is also observed in English. For example, "pale blue" decreases the saturation of the colour category associated to "blue", while "bright blue" increases it. This operation, although, cannot be performed on all basic colours of English, as the colour categories "brown" and "orange" have very similar hue values.

6.2.6 Colour strategies as a network of meaning predicates

Language strategies have been implemented as *semantic constraint networks* using the Incremental Recruitment Language (see Section 2.2). These networks contain meaning predicates that perform *cognitive operations* such as object selection, categorisation or filter, and *semantic entities*, which in this experimental set-up correspond to prototypes of different kind (colour, membership, lightness or chromaticity). This section briefly describes the cognitive operations used in the different colour strategies:

- **equal-to-context**: this operation introduces a variable containing the set of objects in the context, which in the experiments reported in this chapter consists of a set of colour samples.
- **get-XXXX-category-set**: this primitive introduces a variable containing the prototypes of the specified category. Depending on the chosen category, the variable may contain a set of colour, lightness, membership or chromatic prototypes.
- **profile-XXXX-dimensions**: this operation gets a set of colour samples and filters them on particular dimensions of the colour system, returning a filtered set. As the CIE 1967 $L^*a^*b^*$ is the colour system used in all the experiments of this chapter, profiling according to (a) *lightness* will block the a^*b^* dimensions of each colour sample (producing a set of samples on the grey scale as an outcome), (b) *chromaticity* will filter the set cutting off the L^* dimension and (c) *colour* will return the original set.
- **categorise-by-XXXX**: this primitive filter takes as input a set of entities and a given category, and returns the set of entities having the given category as their prototype. For instance, categorising a set of colour samples according to a colour category c_n will return the subset of samples that have c_n as their closest category with an activation value for each of them representing how similar the colour sample is to the colour category. When categorising according to *lightness* or *chromaticity* an additional argument is required, as these prototypes have different dimension values

depending on the entity they are applied to. Taking an example from English, the lightness modifier "dark" will correspond to different L^* values when applied to the colour categories "yellow" or "brown".

- **draw-category-set-to-category**: this operation transforms category sets, shrinking them towards a specified category. It returns a transformed category set where the prototypes values of each category have been modified in the direction of the input category. This cognitive operation is used in the colour combination strategy (Subsection 6.2.3) to transform a colour category set after a first classification of a colour sample in order to perform a second categorisation of it.
- **select-most-activated**: this primitive selects the highest activated entity from a given set. It uses the activation values obtained in the cognitive operations of categorisation to determine the most prominent entity from the input set.

These meaning predicates are then combined into IRL-networks that capture the different cognitive operations involved in colour strategies. We illustrate the relationship between the different meaning predicates in a network taking as example the basic colour strategy (Figure 6.3). The meaning predicate **equal-to-context** introduces the colour samples of the context, represented by the variable $?s1$, in the agent's meaning representation. This set is profiled on the colour dimensions (variable $?s2$). The colour prototypes known by the agent are obtained with the **get-basic-colour-category-set** operation (variable $?bccs$). On the one hand, when this strategy is used in formulation the speaker must find the colour category $?cc$ in its colour category set that is most activated given $?topic$, and will use the word associated with it to describe the topic. On the other hand, in comprehension the hearer makes use of the colour category $?cc$ recovered from the utterance and uses it to identify the colour sample on the context that is more activated.

6.3 Description of the experiments

The goal of the experiments is, as in the experiments presented in the previous chapter, to make a population of artificial agents develop a shared language to communicate about objects in their environment (in this case colour samples). Agents face different communicative challenges and decide which one to address making use of the autotelic principle. A shared language emerges from successive interactions between two agents randomly picked from the population, one acting as a speaker and the other as a hearer.

6.3.1 World and context

In each interaction, both the speaker and the hearer aim to communicate about one colour sample from their shared *context* or *scene*, the environment in which

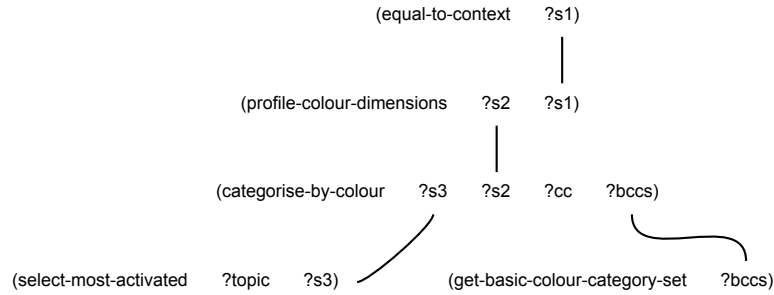


Figure 6.3: Example of a colour strategy as a connected network of meaning predicates. This example corresponds to the basic colour strategy. The context of the interaction and the colour categories known by the agent are captured in the *?s1* and *?bccs* variables. In formulation, a topic *?topic* is described with the word associated to the colour category *?cc*, which corresponds to the most activated colour category for that colour sample.

interactions take place, which consists of a set of colour samples. The set of all possible colour samples in an experimental set-up is called the *world*.

In each interaction, the speaker selects the context, which is a subset of the colour samples present in the world. It chooses both the size of the context and the different colour samples that are part of it. The choice depends on the communicative challenge the speaker wants to attempt. For example, if the world consists of the focal colour samples for English [149], a possible context could contain the colour samples "red", "green" and "blue" if contexts are generated by randomly picking colour samples of the world.

6.3.2 Multi-word guessing game

The particular language game that the population of agents plays, as in the experiments on the discrete domain, is a multi-word guessing game. The speaker produces an utterance containing one or more words describing the selected colour sample of the context and the goal of the hearer is to use this information to identify the colour chip the speaker refers to. A detailed description of interactions in this language game can be found in the previous chapter (Subsection 5.1.2).

6.3.3 Diagnostics and repairs

Agents start the simulations with an empty lexicon and no prototypes. They can expand their construction inventory using different *diagnostics*, methods to identify problems during an interaction, and *repairs*, procedures to correct problems, and also enlarge the list of known prototypes by creating or adopting new ones. Constructions are stored in the *construction inventory* or *lexicon* of the agent, which defines its vocabulary and grammar. Agents make use of their lexicon to *formulate*, verbalise a conceptualised meaning, and to *comprehend*, extract the meaning representation of an input utterance.

As in the previous chapter, the two first diagnostics and repairs that agents need in order to build a shared language are those that allow them to create and learn lexical constructions (Figure 6.4):

Create a word: the speaker does not have a word precise enough to refer to a colour, membership, lightness or chromatic prototype during the process of conceptualisation.

- *Diagnostic:* the speaker cannot come up with a discriminative conceptualisation of the topic.
- *Repair for lack of relevant colour prototype:* the speaker creates a new colour prototype C and sets the colour sample as its value. Additionally, the speaker invents a new term t for the colour prototype and creates a new construction relating C with t .
- *Repair for lack of relevant membership prototype:* the speaker creates a new membership prototype M and sets its value to the distance between the colour sample and the prototype of its closest colour category (see Subsection 6.2.2). Additionally, the speaker invents a new term t for the membership prototype and creates a new lexical construction relating M with t .
- *Repair for lack of relevant lightness prototype:* the speaker creates a new lightness prototype L and sets its value to the difference between the L^* dimension values of the colour sample and its closest colour prototype. Additionally, the speaker invents a new term t for the lightness prototype and creates a new lexical construction relating L with t .
- *Repair for lack of relevant chromatic prototype:* the speaker creates a new chromatic prototype Ch and sets its value to the difference in *chroma* (see Subsection 6.2.5) between the colour sample and its closest colour prototype. Additionally, the speaker invents a new term t for the chromatic prototype and creates a new lexical construction relating Ch with t .

Adopt a word: the hearer cannot identify the meaning predicate associated with an unknown word t , which may refer to a colour, membership, lightness or chromatic prototype.

- *Diagnostic:* the hearer encounters an unknown word in the input utterance.
- *Repair for unknown word that refers to a colour prototype:* the hearer uses the feedback from the speaker to create a new colour prototype C with the colour sample of the topic as its value. Additionally, the hearer creates a new construction relating C with t .
- *Repair for unknown word that refers to a membership prototype:* the hearer uses the feedback from the speaker to create a new membership prototype M and sets its value to the distance between the colour sample and its closest colour prototype. Additionally, the hearer creates a new lexical construction relating M with t .

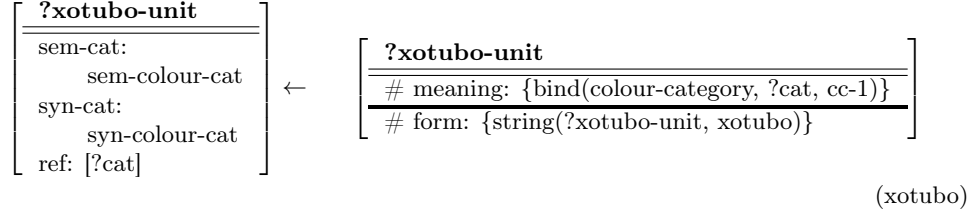


Figure 6.4: Example of a lexical construction C resulting of the process of creating or adopting a word. In this case, C expresses a relation between the colour category $cc-1$ and the word "xotube". From the meaning-form relation the agent can infer that "xotube" will be used to refer to a colour prototype and therefore assigns the corresponding semantic and syntactic classes to it.

- *Repair for unknown word that refers to a lightness prototype*: the hearer uses the feedback from the speaker to create a new lightness prototype L and sets its value to the difference in the L^* dimension between the colour sample and its closest colour prototype. Additionally, the hearer creates a new lexical construction relating L with t .
- *Repair for unknown word that refers to a chromatic prototype*: the hearer uses the feedback from the speaker to create a new chromatic prototype Ch and sets its value to the difference in *chroma* between the colour sample and its closest colour prototype. Additionally, the hearer creates a new lexical construction relating Ch with t .

These diagnostics and repairs allow agents to build a repertoire of different prototypes and lexical constructions. However, only with lexical constructions agents cannot express or reconstruct the complete meaning network issued from the process of conceptualisation of the topic. Moreover, in some contexts, lexical items alone could not carry enough discriminative information about the topic. This problem occurs especially when different colour strategies can be used to describe a colour sample.

As an example, in English a topic description combining the graded membership and the colour combination strategies could describe a colour sample as "blueish green". This utterance contains two lexical words ("blue" and "green") and one suffix ("-ish"). Without any further information the hearer would not know how to combine these prototypes to reconstruct the meaning network. For instance, which colour prototype refers to the first classification of the colour sample or to which colour prototype the speaker is expressing its membership with the suffix. Grammatical constructions are needed to solve this uncertainty, as they convey non-lexical information in colour sample descriptions.

Grammatical constructions introduce additional semantic and syntactic information to the transient structure, the linguistic structure in Fluid Construction Grammar on which constructions are applied in order to formulate an utterance from a meaning conceptualisation or comprehend the meaning conveyed given an input utterance, while connecting the open variables introduced by lexical units

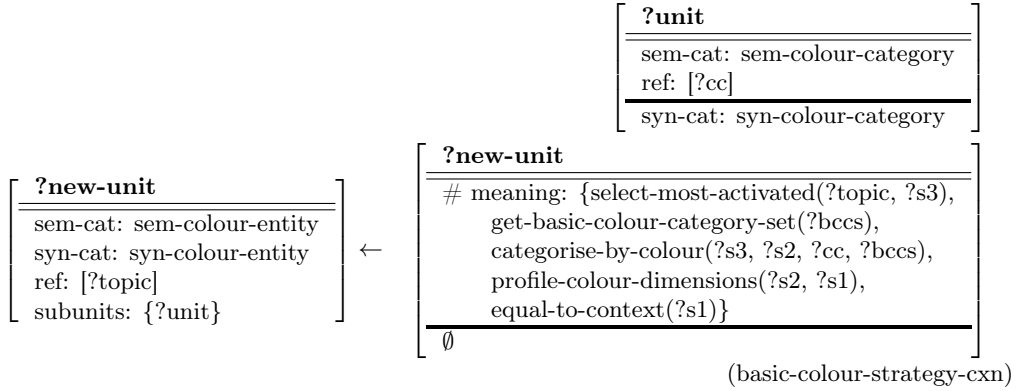


Figure 6.5: Example of a grammatical construction: C captures the missing semantic information of a colour sample conceptualised using the basic colour strategy. It introduces a new unit $?new-unit$ with four meaning predicates to the transient structure. Note that, in order to apply, C requires that a unit expressing a colour category is already present in the TS.

to other meaning predicates, resulting in a linked meaning network. Whenever problems of word-order or unconnected meaning network are detected, agents will try to solve them creating a new grammatical construction (Figure 6.5):

Unconnected meaning network: This problem occurs when the resulting meaning structure in the transient structure is not connected. This problem may occur in both communicative roles.

- *Diagnostic for the speaker:* the resulting transient structure after formulation does not contain all the meaning predicates of the meaning network resulting from the conceptualisation process.
- *Diagnostic for the hearer:* the variables of the meaning predicates in the resulting transient structure after comprehension are unlinked.
- *Repair for the speaker:* the speaker (a) identifies the unexpressed meaning predicates comparing the meaning networks in the resulting transient structure to those in conceptualisation and (b) randomly determines the word order of the lexical items. It then stores this additional semantic and syntactic information into a new construction that is added to its lexicon.
- *Repair for the hearer:* the hearer uses the feedback from the speaker to (a) determine the intended topic and, based on the utterance, identify the missing meaning predicates in its recovered meaning and (b) the word order of the sentence. It then stores this additional semantic and syntactic information into a new construction that is added to its lexicon.

Unknown word order: This problem occurs when the order of words in the utterance is not recognised by the hearer, preventing it from reconstructing a connected meaning structure.

- *Diagnostic*: The word order of the input utterance is not known by the hearer, impeding it from applying a grammatical construction that will lead to a linked meaning network.
- *Repair*: as in the repair for an unconnected meaning network, the hearer uses the feedback from the speaker to (a) determine the intended topic, based on the utterance, (b) identify the missing meaning predicates in its recovered meaning and (b) detect the word order in the utterance. It then stores this additional semantic and syntactic information into a new construction that adds to its lexicon².

6.3.4 Alignment

These lexical and grammatical constructions are stored in the construction inventories of agents. Similar to what was observed in the experiments on the discrete domain, it may happen that some of these constructions are in competition, either because two or more constructions (a) convey the same meaning with different forms or (b) use the same form to refer to different meanings.

This competition is dealt with a mechanism called *alignment*, which is used to tune the preference of agents on which constructions are used. Constructions have a *score*, a number that ranges from 0.0 to 1.0, and is initialised at 0.5. Agents use this score to decide which constructions to employ when there is competition in their lexicon, picking the competing construction with the highest score. After each interaction, the interacting agents update the scores of the constructions used and their competitors following the dynamics of lateral inhibition [155]. When the score of a construction gets to its minimum value (set to 0.0), the construction is removed from the lexicon of the agent. The details of the alignment used are explained in the previous chapter (Subsection 5.1.4).

When adopting a lexical construction, the hearer also has to decide when it should add a new prototype to one of its sets (colour, membership, lightness and chromatic) or associate the unknown word to an already existing prototype. This decision is taken based on how close the observed prototype is to the closest prototype in the inventory. The hearer will add a new prototype when the Euclidian distance to the closest prototype in the inventory is bigger than 0.05. This check is done to prevent the creation of new prototypes each time an unknown word is heard. When the distance is equal or smaller, the hearer introduces competition in its construction inventory, as at least two lexical constructions refer to the same prototype. For example, if an agent has two lexical constructions, one for the English term "blue" and the other Spanish term "azul", these would be competitors if they are associated to the same colour prototype.

Additionally, the different prototypes involved in a successful interaction also undergo a process of alignment (see Figure 6.6). The idea is that the dimension values of prototypes change over time, as agents are exposed to more occurrences of those prototypes. For example, when an agent learns a new colour prototype

²Note that it may be the case that the hearer already has a grammatical construction to express the same semantic information. In this case, the repair will introduce a competitor at the grammatical level.

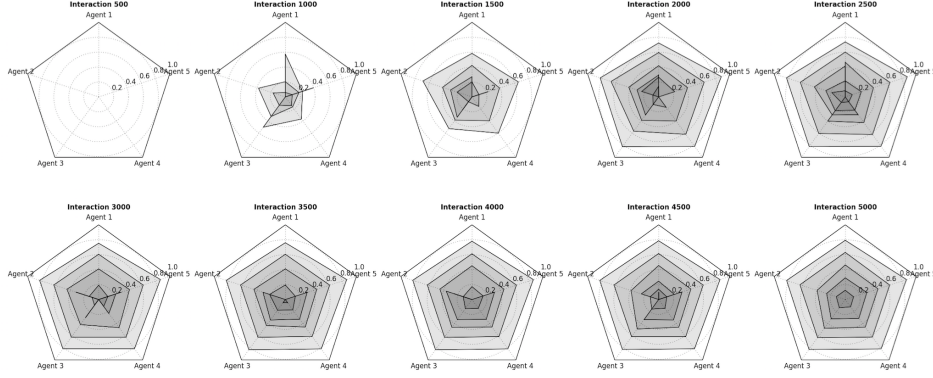


Figure 6.6: Example of the alignment of membership prototypes in a population of 5 agents. Initially each agent creates prototypes with different values. After each successful interaction the involved membership values of the interacting agents are adjusted. At the end of the simulation the population converges to a shared set of membership prototypes.

associated with an unknown word w_i , its value corresponds exactly to the values of the colour sample named using w_i in that interaction. The value of this colour prototype, however, should represent the average colour sample that is labelled with that word instead of having its value fixed to the values of the first colour sample named using that word. Therefore, the value of a prototype changes as it is used in more interactions.

The value of a prototype is computed slightly different depending on the nature of the prototype. In the case of membership prototypes, the value m associated to a membership prototype M is updated by both speaker and hearer with the following equation: $m_i = m_{i-1} - \delta rate(m_{i-1} - act_i)$, where m_i and m_{i-1} are the current and previous values of M and act_i the activation of the topic. In the case of the other two unidimensional prototypes (lightness and chromatic), the same formula is used, replacing act_i for the values of the L^* dimension or h_{ab}^* , respectively. This formula is also used to update colour prototypes. For each dimension of the prototype (L^* , a^* and b^*), the same computation is made, changing act_i for the current dimension value of the colour sample. In the experiment the $\delta rate$ is set to 0.05 for all prototypes.

Finally, as it has been explained in the previous chapter, after each interaction the interacting agents update their confidence and persistence values associated with the challenge used in that interaction. The process of how challenges are updated is described in Section 4.5.

6.4 Experiment 1: regulating linguistic complexity

This experiment mirrors the baseline experiment in which the autotelic principle is used by a population of agents to autonomously regulate the linguistic com-

Level	[1]	[2]	[3]
Challenge	< 1, 0 >	< 1, 1 >	< 2, 1 >
< num_{col}, num_{memb} >			

Table 6.1: Challenges of the the first experiment and their associated level as a vector of two parameters that correspond to the maximum number of colour and membership categories expressed. The level value designates the complexity of a specific challenge, which is computed based on its number of *cognitive operations*.

plexity of their interactions in the preceding chapter (see Section 5.3). However, the two experiments differ as the one reported here takes place in a continuous domain, that of colour, while the other occurs in a discrete world, where object features and spatial relations are predetermined. This difference has a direct impact on the sensori-motor and conceptual levels of the linguistic interactions (see Chapter 2), as agents not only create and learn lexical and grammatical constructions but they also need to create and learn prototypes of different kinds that encode the meaning associated to lexical constructions.

Agents play a multi-word guessing game in which they communicate about colour samples. They start the experimental runs with an empty lexicon, and enlarge it making use of three language strategies: *basic colour*, *graded membership* and *graded category combination*, which are identified by agents as communicative challenges of increasing complexity. The complexity of a communicative task is determined by the number of *cognitive operations* it requires. This number differs among the different challenges and is used to establish its *level*: the *basic colour* strategy has the smallest meaning network, as it involves less meaning predicates than the other strategies, while the *graded category combination* strategy the biggest one. Agents initially address the less complex challenge, and use the *level* value to move between them, depending on the internal state (*boredom*, *anxiety* or *flow*) associated to each challenge.

Moreover, more complex colour descriptions can reuse skills developed at earlier stages. When addressing the *basic colour* strategy agents need to converge both on a classification of the colour space into colour prototypes and on the terms associated to each colour prototype. Once this has been achieved agents can move to the second challenge, which corresponds to the *graded membership* strategy. Since at this point the colour prototypes and their associated terms are already known, the population will focus on the creation and alignment of membership prototypes and the terms related to them. Lastly, in the *graded category combination*, a combination of the *graded membership* and the *colour combination* strategies³, both colour and membership prototypes need to be known. Table 6.1 presents the different communicative challenges as a vector of two parameters.

³In the *graded category combination* agents first identify the closest colour prototype to the colour sample and transform the colour space towards that prototype. Agents then classify the colour sample again on the transformed colour space, obtaining a second colour prototype. Finally, they express how close the colour sample is to the identified colour prototype in the transformed colour space using a graded membership term.

6.4.1 Context

In this experiment, the world consist of 268 different colour samples in the CIE 1967 $L^*a^*b^*$ colour space (see Figure 6.7). The world contains the focal colours and the consensus samples⁴ for English and colour samples created by combining two focal colours in different percentages: 25%, 45%, 55% and 75%, respectively.

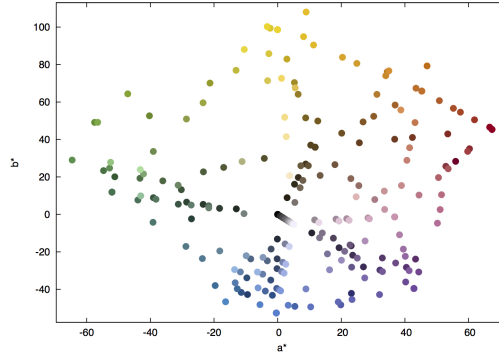


Figure 6.7: The world of the first experiment consist of 268 different colour samples in the CIE 1967 $L^*a^*b^*$ colour space: the focal and consensus colours for English (see [149]) and colour samples created combining two focal colours with percentages 25%, 45%, 55% and 75%.

The context of the interaction consists of a subset of the colour samples present in the world. It is determined at each interaction by the speaker, both its size and the colour samples that are part of it, depending on its current challenge. In the basic colour challenge, the context is created by randomly picking three focal colours of English. In the graded membership challenge, the speaker chooses five random samples from the consensus samples for English. Lastly, in the graded category combination challenge, the speaker picks six colour samples that correspond to the combination of two focal colours for English. Figure 6.8 provides an example of each context.

⁴Colour samples that were consistently named in English by all participants. See Sturges & Whitfield [149].

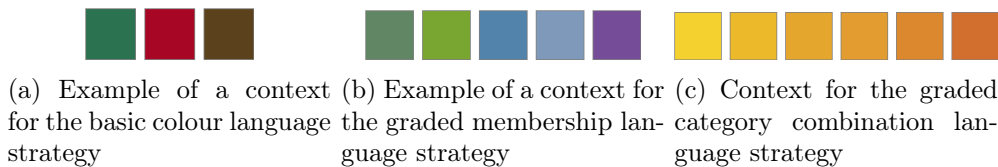


Figure 6.8: Example of the different contexts speakers can create for the first experiment, depending on their current challenge.

6.4.2 Experimental results

To ensure the consistency of the results, they have been tested over ten runs. In each trial, a population of ten artificial agents starts with an empty lexicon, colour and membership inventories. They enlarge and adjust it to the other agents using diagnostics, repairs and alignment (see sections 6.3.3 and 6.3.4). We report four measures:

- **Communicative success** measures the average performance of the population in the communicative challenge. When the communication is successful a value of 1.0 is recorded, 0.0 otherwise.
- **Alignment success** measures the average cohesion of the lexicon in the population. A value of 1.0 is recorded when (a) there was communicative success and (b) both agents would have used the same constructions to refer to the topic of that interaction, 0.0 otherwise.
- **Lexical stability** measures the average score of lexical constructions of the population. A value of 1.0 means that all lexical constructions on each agent have the maximum score. This measure shows how stable the lexicon is, independently of its size.
- **Confidence in challenge** measures the average confidence value of the population for a particular challenge level. It has a value between 0.0 and 1.0.

The resulting dynamics of the experiment are shown in Figure 6.9. Agents start addressing the first challenge, for which the population has to create and coordinate its (a) lexicon for basic colour terms and (b) a set of colour prototypes. Agents gain confidence for this challenge rapidly, as both the communicative success and the confidence value for the first challenge increase quickly. Around interaction 2000 a sudden drop on both measures is observed as a result of agents starting to reach maximum confidence. As this situation corresponds to the internal state of boredom, agents enter in the shake-up phase and move to the second challenge, in which agents need to develop lexical constructions and membership prototypes in order to express the distance between colour samples and basic colour prototypes.

In the course of the second challenge, agents are exposed to a bigger diversity of contexts, which makes the alignment of membership prototypes and its associated lexical constructions more difficult. This is also reflected in the evolution of lexical stability, as the average score of lexical constructions drops despite the fact that agents are converging to an optimal lexicon for basic colour terms.

By interaction 5000 a fraction of the population has already reached maximum confidence for the second challenge, causing an overlap of the current challenge in the population. At this point, some agents identify their internal state as boredom and are motivated to start attempting the third challenge while another part of the population is still addressing the second one. Eventually, all agents move to the third challenge, and communication success progressively improves

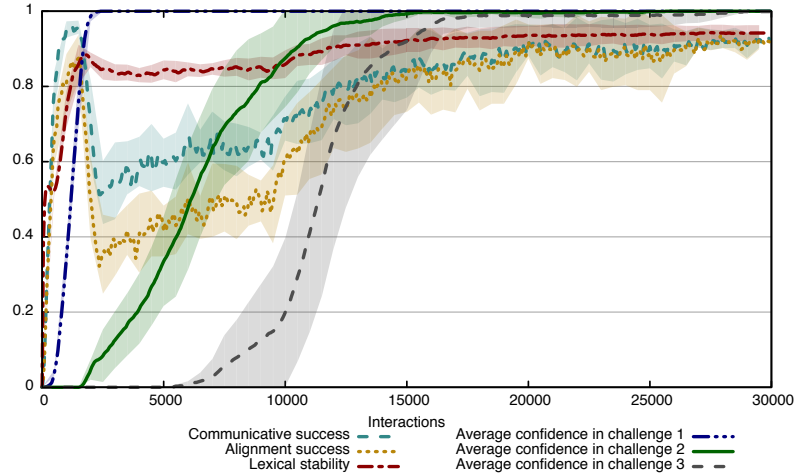


Figure 6.9: Resulting dynamics of the experiment for a population of 10 agents averaged over 10 runs of 30000 interactions. By the end of the simulation all agents in the population reach a steady communicative success value above 90% and maximum confidence for the three challenges. Error bars represent the maximum and minimum across the different experimental runs.

as the population succeeds in aligning its lexicon and membership prototypes. As a consequence, the alignment success rate also increases until it reaches the same percentage as communicative success. By interaction 30000 all agents in the population have reached maximum confidence for the three challenges and a steady communicative success value above 90%.

The fact that 100% communicative success is not achieved even when all agents got maximum confidence scores for all challenges is due to the fact that the language strategies available (*basic colour*, *graded membership* and *graded category combination*) are not discriminative enough in certain situations, causing the speaker to fail occasionally when trying to conceptualise a topic in particular contexts. This effect was also observed by Bleys [21] when he looked at each language strategy independently.

However, lexical stability also stabilises to a value around 95% (not reaching 100%), which means that some of the lexical constructions in the population have not achieved a score of 1.0. In order to explain this outcome it is necessary to look at the lexicon and membership categories that agents have by the end of an experimental run: some agents keep membership categories and their associated lexical constructions that are no longer used in the linguistic interactions. Therefore, population has not fully converged to a minimal lexicon although agents manage to communicate successfully in most contexts. As already pointed before, by the end of the simulations the observed failures in communication are caused by an unsuccessful conceptualisation of the topic and not by a problem of alignment.

This effect (the lack of convergence to a minimal lexicon) was unexpected and, in order to understand it better, I looked at the relation between lexical

stability and communicative success, testing the same configuration on different population sizes. Figure 6.10a presents the resulting communicative success, alignment and lexical stability for a population of two, five, ten and twenty agents (3000, 10000, 50000 and 150000 interactions, respectively). Results show a slight reduction of communicative success as the population size increases. Additionally, a little disparity between communicative success and alignment is observed in bigger populations.

This effect can be explained by the fact that larger populations converge to systems with more membership prototypes (Figure 6.10b). An increased number of membership prototypes requires more time to align: this helps prototypes which are not spread over the population to stay longer in individual inventories as they are less used. The decrease in communicative success is therefore explained by (a) a lower alignment of agents' construction inventories and membership prototypes and (b) longer presence of non-spread membership prototypes among the population which are still used in conceptualisation. These results suggest that smaller populations could be able to develop more consistent communicative systems for the domain of colour.

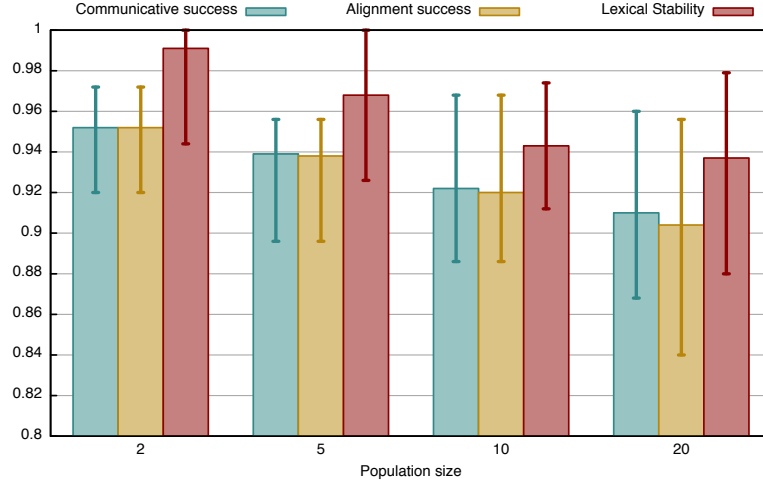
6.5 Experiment 2: Managing multiple challenges

For the second experiment I wanted to test the usefulness of the motivation system to choose not only *when* agents change their current challenge but also to select *which* challenge an agent should address next, based on the challenges it already knows and the characteristics of the contexts in which interactions take place. In order to do so I have integrated the basic colour, colour combination, lightness and chromatic language strategies for the domain of colour presented in Section 6.2 (and the possible combinations of two of these strategies) into one experimental set-up.

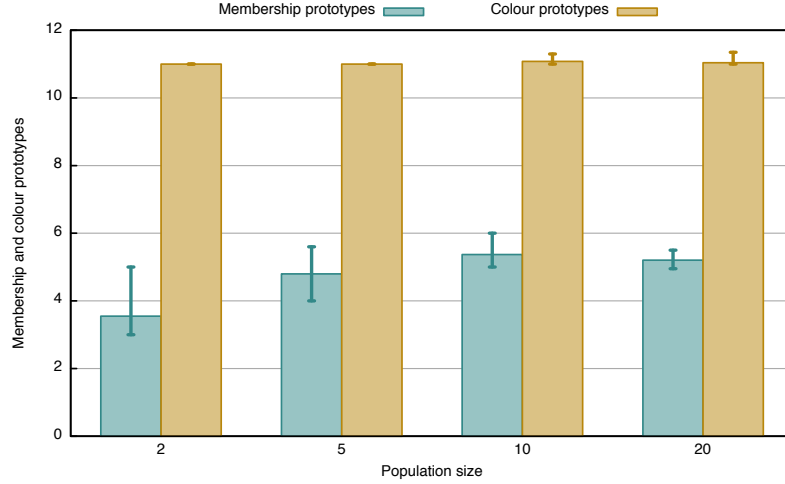
As in the first experiment (Section 6.4), agents play a multi-word guessing game in which they have to communicate about colour samples. The population starts the experimental runs with an empty lexicon (no lexical or grammatical constructions) and no colour, lightness or chromatic categories. They enlarge their construction inventory and the number of different categories using several language strategies, identified by agents as communicative challenges of a particular complexity.

The complexity of a communicative task is again computed based on its number of *cognitive operations*. This number differs among the different challenges and is used to determine its *level*. Agents start attempting the less complex challenge, and make use of the level value to move between them, depending on the internal state (*boredom*, *anxiety* or *flow*) associated with each challenge. However, in this experiment, some language strategies have the same number of cognitive operations and therefore are assigned with the same level of complexity. This entails that agents can address multiple challenges and, in order to be able to develop a shared language, must be able to manage various communicative tasks at the same time successfully.

As in the previous experiment, the first level of complexity corresponds to



(a) Resulting communicative success, alignment and lexical stability for different population sizes averaged over 10 runs.



(b) Resulting number of membership and colour categories for different population sizes averaged over 10 runs.

Figure 6.10: Experimental results for different population sizes. Figure 6.10a shows the resulting communicative success, alignment and lexical stability scores in a population of two, five, ten and twenty agents. The scale on the y-axis is set to the range $[0.8, 1.0]$. Figure 6.10b displays the average number of membership and colour prototypes for the same populations.

the *basic colour* strategy (see Subsection 6.2.1), as it has the smallest meaning network. The second level of complexity is assigned to three different language tasks: *colour combination*, *lightness* and *chromaticity* language strategies (see subsections 6.2.3, 6.2.4 and 6.2.5, respectively), as they involve the same number of meaning predicates. Finally, the third level of complexity corresponds to a combination of two language strategies of the second level. Once an agent has reached a maximum confidence value for a challenge of level 2, it generates two

Level	[1]	[2]	[3]
Challenge	$\langle 1, 0, 0 \rangle$	$\langle 1, 1, 0 \rangle$	$\langle 2, 1, 0 \rangle$
$\langle num_{col}, num_{light}, num_{chrom} \rangle$		$\langle 1, 0, 1 \rangle$	$\langle 2, 0, 1 \rangle$
		$\langle 2, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$

Table 6.2: Challenges and their associated level for the second experiment on the continuous domain. Challenges are defined as a vector of three parameters that correspond to the maximum number of colour, lightness and chromatic properties expressed. The level associated to a specific challenge designates its complexity, which is computed based on the number of *cognitive operations*.

new challenges introducing and mixing another language strategy of the same level to that challenge.

For example, from the colour combination strategy (level 2) an agent will generate two challenges combining it with the lightness and chromaticity strategies. As a result, two challenges of level 3 are created: the *lightness colour combination* and the *chromaticity colour combination*. An example of these language strategies on the English language would be "dark yellow green" or "bright blue green". Table 6.2 presents the different communicative challenges as a vector of three parameters.

6.5.1 Context

In this experiment, the world consists of 335 different colour samples in the CIE 1967 $L^*a^*b^*$ colour space. The world contains the focal colours for English and colour samples obtained (a) modifying the L^* value of these focal colours (increasing or decreasing it by 20.0 within its limits: $[0.0, 100.0]$), (b) modifying the chroma dimension (see Subsection 6.2.5) of the focal colours (increasing or decreasing it by 0.2), (c) combining two focal colours for English in different percentages (25%, 45%, 55% and 75%, respectively) and (d) combining two of the previous modifications ((a), (b) and (c)). The colour samples from (d) are created by performing either the lightness and chroma modifications to the focal colours for English or by applying lightness or chroma modifications to the resulting colour samples of (c).

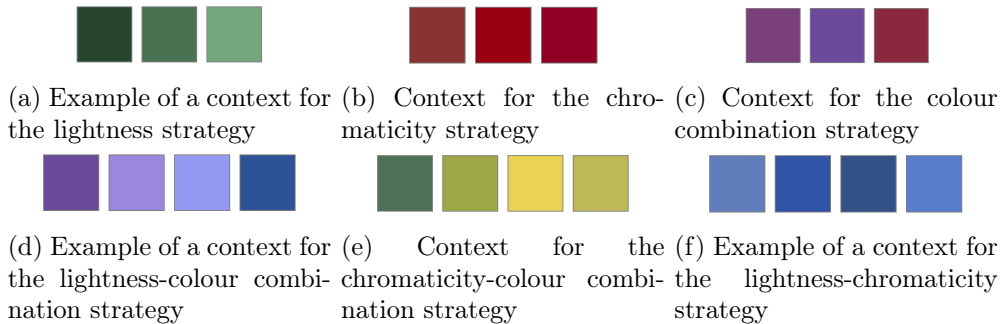


Figure 6.11: Example of the contexts speakers create for the second experiment when addressing the different challenges of level 2 and 3.

The speaker selects the context, which is a subset of the colour samples present in the world, at the beginning of each interaction. The agent chooses both the size of the context and the different colour samples that are part of it. This choice depends on the current challenge of the speaker. In the basic colour challenge the context is created by randomly picking three focal colours of English. When addressing challenges of level 2 the speaker composes different contexts of three colour samples depending on which challenge it wants to attempt. For the *lightness strategy* the agent selects colour samples that vary in their L^* value, for the *chromaticity strategy* colour samples with similar *hue* values and for the *colour combination strategy* the speaker randomly picks three colour samples from the combination of two focal colours.

Finally, for the challenges of level 3 the speaker puts together four colour samples. For the *lightness-colour combination strategy* the colour samples consist of lightness modifications on combinations of two focal colours, for the *chromaticity-colour combination* chromatic modifications on combinations of two focal colours and for the *lightness-chromaticity strategy* both lightness and chromatic modifications to focal colours of English. Figure 6.11 provides an example of each context.

6.5.2 Misestimation of the speaker's challenge

One of the main difficulties in applying the autotelic principle to language games is to decide which challenge the hearer should select as the communicative challenge of the interaction. This is due to the fact that the interacting agents may differ in (a) the communicative challenges they know and (b) the selection of their current challenges.

This problem is particularly relevant for this experimental set-up, as agents must deal with several communicative challenges of similar complexity at the same time. As explained in the chapter devoted to the autotelic principle (subsection 4.6.1), the hearer tries to guess what challenge was attempted by the speaker based on (a) the challenges it knows and (b) the information it can gather from the interaction, focusing on the utterance produced by the speaker and the meaning representation it could retrieve from it.

Despite using all the information at its disposal, the hearer has to face two situations in which it cannot successfully identify the communicative challenge attempted by the speaker. It is important to highlight these cases, as misinterpretations of the challenge attempted by the speaker have an impact on the agent's confidence gain for the different communicative challenges in the experiment.

The first one occurs when the recovered meaning network is not fully connected. This happens when the hearer does not know how to combine all the lexical items (in the experiment words used to refer to colour, chromatic or lightness categories) of the input utterance. This indicates that the hearer does not know the challenge used by the speaker. In this situation, the hearer identifies among its known challenges those that provide the more connected meaning network possible and randomly selects one of them as the challenge to update (Subsection 4.6.1). For example, the speaker uses the lightness-chromatic strat-

egy to describe a topic and produces a three-word utterance that refers to one colour, chromatic and lightness category. The hearer, however, does not know this strategy yet as it only knows the communicative challenges of level 1 and 2 and therefore it is not able to build a connected meaning network. Among these four strategies, the lightness and chromaticity strategies are the ones that minimise the number of unconnected meaning predicates given the input utterance. The hearer then randomly selects one of them as the challenge to update, which does not correspond with the communicative challenge attempted by the speaker.

The second situation where the challenge addressed by the speaker is misidentified by the hearer is a consequence of agents having multiple conceptualisations possible for a topic. In an interaction, the speaker first selects the communicative challenge and then creates a context in which the language strategy associated with that challenge may be used. Sometimes it occurs that the most discriminative conceptualisation for the topic is not the one corresponding to the attempted communicative challenge but a conceptualisation already known before. In these situations, the speaker produces an utterance that uses less lexical words than the maximum allowed by the challenge, as the goal of interactions in language games is to communicate about colour samples successfully. However, this affects the hearer's guess of the challenge attempted by the speaker, as the agent will identify a less complex challenge instead of the actual challenge addressed by the speaker. For example, the speaker decides to attempt the lightness strategy (level 2) and creates a context for it. However, it may occur that the most discriminative conceptualisation for the topic in this context corresponds to the basic colour strategy (level 1). The speaker will then use the basic colour strategy to formulate an utterance, which will contain only one word referring to a colour category. In this case, the hearer will identify the basic colour strategy as the communicative challenge of the interaction, which differs from the attempted challenge by the speaker.

6.5.3 Experimental results

In the same manner as in the other experiments reported in this thesis and in order to ensure the consistency of the results, the experimental set-up has been tested on ten runs. In each trial, a population of ten artificial agents start with empty lexicon, colour, lightness and chromatic inventories, which are gradually increased and adjusted using diagnostics, repairs and alignment. Similarly to the first experiment of this chapter, we analyse four measures: communicative and alignment success, lexical stability and the confidence values for each challenge.

The resulting dynamics of the experiment are shown in Figure 6.12. Agents start addressing the basic colour challenge (level 1), for which they need to create and coordinate a set of prototypes and lexical items to refer to basic colour terms. As in the first experiment of this chapter (Section 6.4), the population accomplish this task quite fast. They require around 500 interactions to successfully communicate about the English focal colours in the 80% of the cases, and by interaction 1000 (about 200 interactions per agent) it has raised up to more

than 95%. As a consequence of this, the average confidence value of the first challenge increases rapidly and, quickly after the 2000th interaction, all agents in the population have reached maximum confidence for this task.

In parallel, a drop in the communicative and alignment success is observed. This is a consequence of agents experiencing an internal state of boredom as the basic colour challenge is not challenging anymore. This causes agents to enter into the shake-up phase and generate three more demanding communicative tasks (those labelled with a complexity of level 2), that correspond to the colour-combination, lightness and chromaticity strategies.

With the aim of recovering from the reduction of communicative success and being able to cope with the new challenges, the population creates lightness and chromatic prototypes in order to discriminate the colour samples in the new contexts successfully. However, facing three communicative challenges at once slows down the alignment of these prototypes and their associated lexical items, partially because of the problems hearers undergo when they estimate the challenge attempted by the speaker, explained in Subsection 6.5.2. Nevertheless, the population manages to increase the communicative success above 80% once the population starts to align their lightness and chromatic inventories.

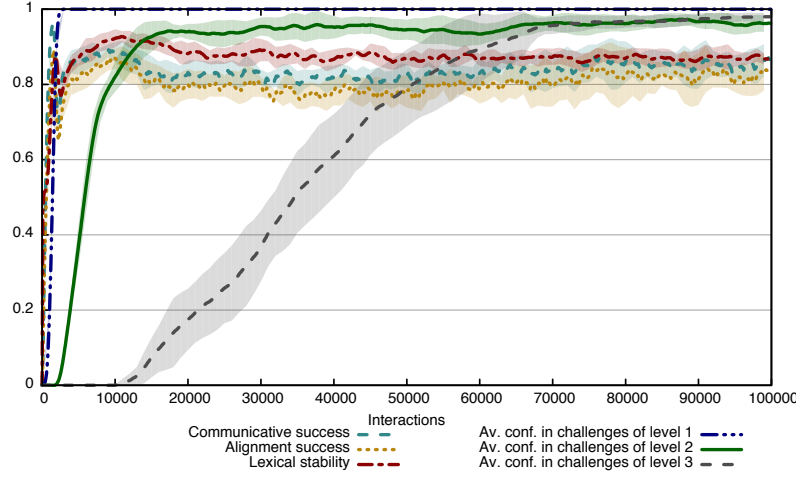
A similar behaviour is observed in the evolution of lexical stability. Although the population is converging to an optimal lexicon for basic colour terms, the creation of new lightness and chromatic terms produces a decrease in the average score of lexical constructions⁵, which progressively rises as agents start aligning these terms.

When an agent reaches maximum confidence for a challenge of level 2 it undergoes a state of boredom and uses this challenge to generate new ones of increased complexity, in the way explained in the introduction of this section. However, it chooses to focus on the remaining challenges of level 2 for which it has not gained maximum confidence yet instead of addressing the recently created ones of level 3. It is not until interaction 15000 that some agents in the population have already reached maximum confidence for the three challenges of the second level of complexity and start addressing challenges of level 3.

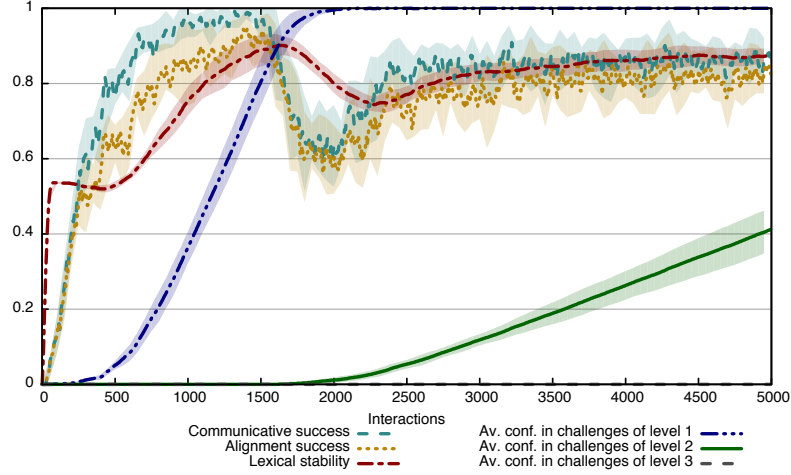
This situation results in an overlap of challenges of level 2 and 3 attempted by the population. Notice that it is not until the end of the simulations that the population reaches highest confidence values for challenges belonging to these two levels. In order to understand this behaviour we need to look at the evolution of the average confidence for each communicative task, presented in Figure 6.13.

It turns out that those strategies that involve chromatic categories are more difficult for agents than the other communicative tasks. It is particularly interesting in the case of the chromaticity strategy (which corresponds to the challenge parametrisation $< 1, 0, 1 >$), as the population does not reach a maximum confidence value for this challenge but stabilises to a value around 90%. Besides, the fact of being aware of more communicative tasks makes the hearer's decision of which challenge to update at the end of an interaction more difficult, leading

⁵A score is associated with each construction which is used to drive the preference of the constructions that agents will use when there is competition. See Subsection 5.1.4 for detailed information about how the alignment of constructions has been implemented in this thesis.



(a) Resulting dynamics of the experiment for a population of 10 agents averaged over 10 runs of 100000 interactions.



(b) Zoom in on the first five thousand interactions of the population.

Figure 6.12: Resulting dynamics of the experiment for a population of 10 agents averaged over 10 runs of 100000 interactions (Figure 6.12a). By the end of the simulation all agents in the population reach a steady communicative success value above 85% and high confidence values for all the communicative challenges. Error bars represent the maximum and minimum across the different experimental runs. Figure 6.12b shows the same results zooming in on the first five thousand interactions.

to a higher number of misestimations. These errors also have an impact on the confidence value associated with challenges of level 2, slowing down the confidence gain of the population and even decreasing it in a recurrent manner for short periods, as is observed for the lightness strategy (challenge parametrisation $< 1, 1, 0 >$). Eventually, the population overcomes these difficulties, ending the simulations with high confidence values for all challenges.

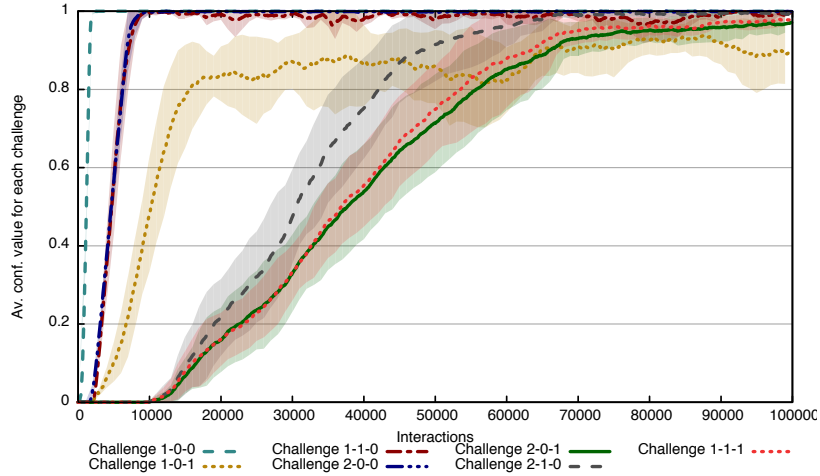


Figure 6.13: Evolution of the average confidence value in the population for each communicative challenge. The population achieves maximum confidence for all challenges except for $\langle 1, 0, 1 \rangle$, for which it only reaches a value around 90%.

The chromaticity strategy (challenge parametrisation $\langle 1, 0, 1 \rangle$), however, presents a different behaviour. As can be seen in Figure 6.13, the population starts gaining confidence for this task at a slower pace when agents start addressing the communicative challenges of level 2. Contrary to the other communicative tasks, for this challenge the population does not reach a maximum confidence value but instead its value stabilises around 90%. This outcome was unexpected but can be explained by the combination of two factors. First, the set of colour samples speakers choose when addressing this challenge result, in some cases, in conflicting contexts. As a consequence, this strategy cannot produce discriminative utterances in some interactions, as the English focal colours "orange" and "brown" have similar hue values. Secondly, the misestimation of the speaker's attempted challenge by the hearer, as most interactions in which speakers attempted challenges $\langle 2, 0, 1 \rangle$ and $\langle 1, 1, 1 \rangle$ that result in a communication failure are identified by hearers as the challenge $\langle 1, 0, 1 \rangle$.

As already seen with the first experiment, at the end of the experimental runs the emerging language does not allow the population to reach 100% communicative success in their linguistic interactions but reaches a steady value slightly below 90% (specifically 88.40%). This outcome is caused by a failure in the conceptualisation of the selected topic of the interaction, which implies that in some cases the different language strategies of the experiment cannot come up with discriminative meaning networks for specific colour samples in particular contexts.

Similarly, the alignment success stabilises to a value slightly below the communicative success (86.00%). The fact that both measures do not settle to the same percentage means that by the end of the simulations the population has slight variances in their lexicon and prototypes as they keep lexical constructions

and prototypes no longer used. These differences are presented in Table 6.3.

Constructions	Value	SD (σ)	Entities	Value	SD (σ)
Colour cxns	11.56	0.91	Colour entities	11.56	0.91
Chromatic cxns	4.59	1.09	Chromatic entities	3.51	0.44
Lightness cxns	8.52	1.65	Lightness entities	5.95	0.54
Lexical cxns	24.67	2.42	Entities	21.02	1.34
Grammatical cxns	6.96	0.07			

Table 6.3: Results for different construction and entity measures for the second experiment on the continuous domain.

While the colour constructions and prototypes match completely, the results exhibit a mean number of lightness and chromatic constructions higher than their associated prototypes. This mismatch is particularly pronounced in the case of lightness constructions, as in average the population has around 2.5 more constructions than prototypes. This divergence implies that the resulting language is not entirely aligned and explains why the lexical stability gets a steady value of 87.2% by the end of the experimental runs.

6.6 Experiment 3: Simulating an embodied experiment

The experiments reported so far have all taken place in a simulated world, in which all agents perceive the objects and colour samples of their context in the same way. This differs from our personal experience, as we usually get involved in linguistic interactions in which each participant observes its surroundings from different points, causing different perceptions of the spacial, dimension, light and colour dimensions of the objects in our environment.

We wanted to study the impact that embodiment would cause to our experimental results. Unfortunately, during my research I did not have access to robots and did not have enough time to be able to design and execute an experiment with embodied agents. What I did, however, was to simulate differences in how the interacting agents perceive the objects in the context.

6.6.1 Perceptual deviation

This experiment reproduces the experiment carried out in Section 6.5, introducing variance in how colour samples are perceived by the agents taking part in a linguistic interaction. This is done by modifying the $L^*a^*b^*$ values of each colour sample in the context at the start of the interaction by p_{dev} , a value in the range $[0.0, 1.0]$ where 0.0 means no deviation and 1.0 a random number for each dimension of the colour sample. This value represents the percentage of maximum value modification for the $L^*a^*b^*$ values, which is directly related to the range of possible values for each dimension⁶. The pseudo code of how perceptual deviation is computed is presented in Algorithm 4.

⁶In the CIE $L^*A^*B^*$ colour system only the L^* has a delimited range (see Section 6.1). The limits for the a^*b^* dimensions have been set to the maximum and minimum values observed in

Algorithm 4 Perceptual deviation

```

procedure PERCEPTUALDEVIATION( $p_{dev}, input_{cs}$ )
   $MaxChange_{dev} \leftarrow \text{RANGE OF POSSIBLE VALUE CHANGE PER DIMENSION}(p_{dev})$ 
   $res_{cs} \leftarrow input_{cs}$   $\triangleright$  Resulting colour sample set to input colour sample
  for each  $maxValue_i \in MaxChange_{dev}, res_{cs-i} \in res_{cs}$  do
     $actChange_i \leftarrow \text{RANDOM}(maxValue_i)$   $\triangleright$  Uniformly distributed
     $res_{cs-i} \leftarrow res_{cs-i} \pm actChange_i$ 
  end for
  return  $res_{cs}$ 
end procedure

```

We can illustrate how the perceptual deviation algorithm works with an example. The focal colour of English "purple" has [41.22, 42.83, -39.57] as $L^*a^*b^*$ values and we want to modify its perception using a p_{dev} of 0.05. First, we need to compute how much the value can change for each dimension. This is done by multiplying the range of each dimension by p_{dev} , which corresponds to the ranges [0.0, 5.0], [0.0, 6.60] and [0.0, 8.04] for the L^* , a^* and b^* dimensions respectively.

Secondly, the actual variance for each dimension should be determined, which is computed by applying the random function $X \sim U(d_r)$, where d_r corresponds to the ranges for the $L^*a^*b^*$ dimensions computed in the previous step. This operation could give, for example, the values 3.93, 2.75 and 5.49 for each dimension. Finally, the values of the dimensions that will be perceived by the agent consist of the randomly chosen addition or subtraction of the values obtained in the second step to the original values of the colour sample. In this example, the resulting perceived colour sample could have [45.14, 40.08, -45.06] as $L^*a^*b^*$ values. The effect of different perceptual deviation values on the English focal colour "red" is shown in Figure 6.14.



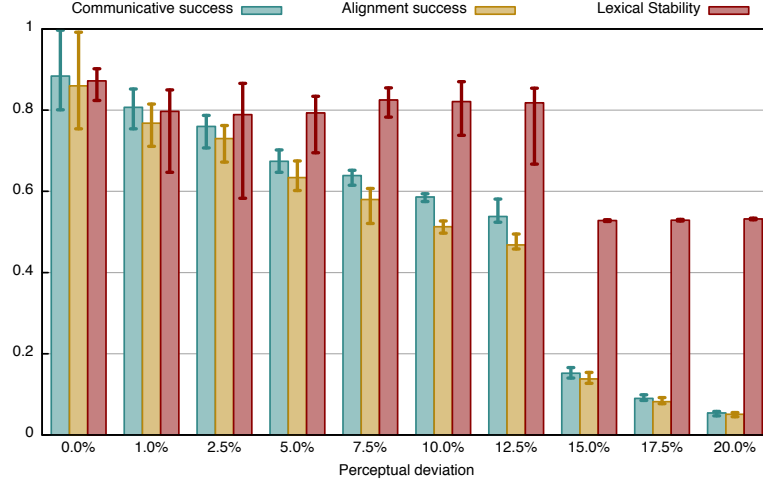
Figure 6.14: Effect of perceptual deviation on the English focal colour "red" ($L^*a^*b^*$ values of [41.22, 67.38, 45.36]). The first colour sample corresponds to the focal colour without deviation. The following chips correspond to the same colour sample modified with p_{dev} values of 0.01, 0.025, 0.05, 0.075, 0.10, 0.125, 0.15, 0.175 and 0.20, respectively.

6.6.2 Experimental results

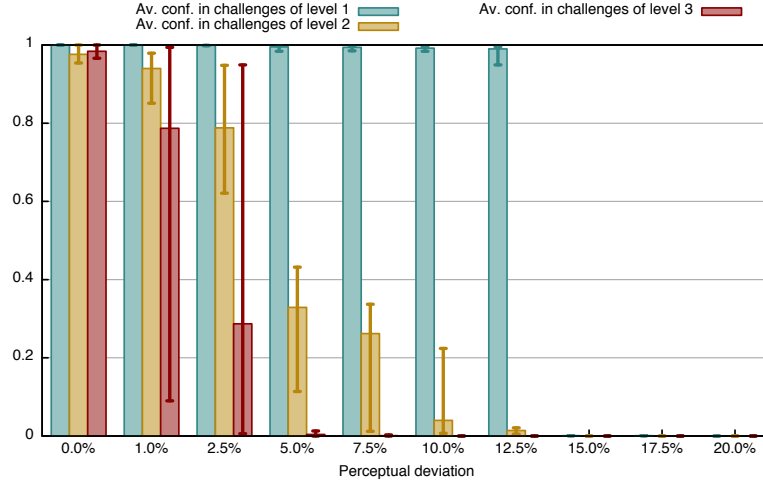
We have run a series of experimental set ups over ten runs, which vary in the value of p_{dev} used to create the perceptual deviation. The values of p_{dev} tested

the colour samples that conform the world, which in this experiment correspond to [-64.59, 67.38] for a^* and [-52.65, 108.06] for b^* .

are 0.01, 0.025, 0.05, 0.075, 0.10, 0.125, 0.15, 0.175 and 0.20. As in the previous experiments, a population of ten artificial agents start with empty lexicon, colour, lightness and chromatic inventories, which increase and adjust it using diagnostics, repairs and alignment. We also report four measures: communicative and alignment success, lexical stability and the confidence values for each challenge.



(a) Resulting communicative success, alignment success and lexical stability for different p_{dev} values.



(b) Resulting average confidence for the different challenge levels for different p_{dev} values.

Figure 6.15: Results of the experiment for a population of 10 agents averaged over 10 runs of 100,000 interactions for different p_{dev} values. As can be seen in Figure 6.15a the average communicative success, alignment success and lexical stability decrease as the p_{dev} value increases. Figure 6.15b shows the average confidence for the different challenge levels, which present a similar tendency, as higher p_{dev} entail lower confidence values.

The results of the experiment are shown in Figure 6.15. As expected, the higher the p_{dev} value is, the harder results to communicate successfully about colour samples. As can be seen in 6.15a even a perceptual deviation of 2.5% reduces the average communicative and alignment success of the population. The communicative success, alignment success and lexical stability scores progressively decrease as the p_{dev} tested increases, with a sudden drop in the average score of the three measures in the change of a p_{dev} value of 0.125 to 0.15. This effect is a consequence of a perceptual deviation that is so high that the population is not even able to agree on a lexicon for basic colours, ending up with less than 10% of communicative success for a perceptual deviation of 20%.

A similar cause-effect is observed in the average confidence score for the different challenge levels (Figure 6.15b). Even small differences in perceptual deviation have an enormous impact on the experimental outcome. A p_{dev} value of 0.025 reduces the average confidence for challenges of level 3 from almost 100% to less than 30%. This number is further reduced to less than 1% for a p_{dev} value of 0.05. The most sensible challenges to perceptual deviation are $< 1, 1, 1 >$ and $< 2, 0, 1 >$, as the population almost does not gain any confidence for these tasks with a p_{dev} value of 0.025.

This effect is also noticeable in the average confidence for challenges of level 2. A p_{dev} value of 0.025 reduces the confidence value for these challenges to a number around 80%, and a p_{dev} value of 0.05 to one third. The impact is considerable for the challenges $< 1, 0, 1 >$ and particularly $< 1, 1, 0 >$, for which the population does not even reach 1% confidence values with a perceptual deviation of 5%. For higher p_{dev} values this trend persists, leading to less than 10% confidence score on average for the communicative tasks of level 2 for a p_{dev} of 0.10.

Finally, for bigger perceptual deviation values communicating about colour samples becomes too difficult for the population. The last p_{dev} value for which agents manage to develop a shared language for basic colours is 0.125. For p_{dev} values of 0.15 or higher the population cannot gain confidence for the first communicative task.

Challenge	Average confidence for different p_{dev} values								
	0.01	0.025	0.05	0.075	0.10	0.125	0.15	0.175	0.20
Level 1									
$< 1, 0, 0 >$	1.00	1.00	1.00	1.00	0.99	0.99	0.01	0.00	0.00
Level 2									
$< 1, 0, 1 >$	0.95	0.84	0.06	0.02	0.01	0.00	0.00	0.00	0.00
$< 1, 1, 0 >$	0.87	0.53	0.01	0.01	0.01	0.00	0.00	0.00	0.00
$< 2, 0, 0 >$	1.00	1.00	0.92	0.77	0.11	0.02	0.00	0.00	0.00
Level 3									
$< 1, 1, 1 >$	0.79	0.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$< 2, 0, 1 >$	0.74	0.24	0.01	0.00	0.00	0.00	0.00	0.00	0.00
$< 2, 1, 0 >$	0.83	0.32	0.01	0.00	0.00	0.00	0.00	0.00	0.00

Table 6.4: Resulting confidence value for each communicative challenge for different p_{dev} values.

Table 6.4 presents the average confidence for each communicative task individually. The perceptual deviation affects the confidence value for challenges

Measure	Average number for different p_{dev} values								
	0.01	0.025	0.05	0.075	0.10	0.125	0.15	0.175	0.20
Constructions									
Col. cxns	11.97	16.82	14.84	13.20	16.09	13.63	21.99	22.21	22.64
Chrom. cxns	6.98	6.86	5.64	5.00	6.65	6.43	0.00	0.00	0.00
Light. cxns	15.45	11.69	7.07	6.05	5.74	5.20	0.00	0.00	0.00
Lex. cxns	34.40	35.38	27.54	24.26	27.48	25.26	21.99	22.21	22.64
Gram. cxns	6.96	6.98	6.73	5.96	4.06	4.00	1.00	1.00	1.00
Entities									
Col. ent.	11.97	13.12	13.49	12.63	14.99	12.32	9.27	9.11	9.04
Chrom. ent.	4.36	4.26	3.28	3.40	3.82	4.19	0.00	0.00	0.00
Light. ent.	8.74	8.05	4.93	4.53	4.76	4.56	0.00	0.00	0.00
Total	25.07	25.43	21.70	20.56	23.57	21.07	9.27	9.11	9.04

Table 6.5: Results for different construction and entity measures for different p_{dev} values.

of level 3 evenly, as they a portion similar values for each p_{dev} value tested. In contrast, it has more impact on specific challenges of level 2. While the confidence value for $\langle 2, 0, 0 \rangle$ does not vary for p_{dev} values of 0.01 and 0.025, the confidence value for $\langle 1, 1, 0 \rangle$ gets reduced to fifty percent for a p_{dev} value of 0.025. With a perceptual deviation of 7.5% the population almost does not gain any confidence for the challenges $\langle 1, 0, 1 \rangle$ and $\langle 1, 1, 0 \rangle$ but it still reaches a value above 75% for challenge $\langle 2, 0, 0 \rangle$. With a p_{dev} of 0.125 all confidence values for challenges of level 2 are almost zero, but agents still manage to gain confidence for the first challenge. For p_{dev} values of 0.15 or higher the population fails to master the basic colour challenge.

Table 6.5 shows several construction and entity measures for the p_{dev} values tested. While the number of chromatic and lightness constructions and entities gets reduced for low p_{dev} values (0.01, 0.025, 0.005 and 0.075), the number of colour constructions and entities increases. These numbers become steady for p_{dev} values of 0.10 and 0.125, despite the fact that the population almost does not gain any confidence for challenges of level 2. For higher perceptual deviations agents only create and learn constructions and entities for colours.

6.7 Summary

This chapter presented a series of experiments where a population of agents make use of the autotelic principle to self-regulate the complexity of their linguistic interactions while developing a shared language to describe colours. The first part introduces the subject of study, a domain of particular interest given the differences that human languages present when communicating about colour samples. It also presents the colour system employed to represent colour spaces in this chapter (the CIE 1967 $L^*a^*b^*$ system), the set of language strategies that are used by agents in the different experiments reported and how they have been implemented using IRL as semantic constraint networks composed of meaning predicates that perform cognitive operations.

It then defines the various worlds considered in the different experimental

setups, a set of all possible colour samples in the experiment, and the language game played, which is a multi-word guessing game. Agents start with an empty lexicon and no prototypes and try to communicate about one colour sample in the context, a subset of the world chosen by the speaker at the beginning of each interaction. They enlarge the size of their lexicon and prototypes using several diagnostics and repairs and employ alignment to tune their preference on which constructions to use.

The second part reports on three different experiments. The first one explores the usefulness of the autotelic principle as a mechanism to regulate linguistic complexity. Agents face challenges of increasing difficulty and successfully manage to develop an artificial language of increasing expressive power to refer to colours when trying to remain in a state of flow: they start (a) converging on a language for colour prototypes and then extend it developing (b) membership categories, which express the degree of similarity between a colour sample and a colour prototype, and eventually (c) combining colour prototypes. In addition, simulations with different population sizes show that bigger populations converge to systems with more membership prototypes on average.

The second experiment studies the performance of the motivation system when it is required to manage several language strategies simultaneously. In this set up the population develops chromatic and lightness prototypes while learning to combine colours at the same time, and later combine them into more complex conceptualisations. The population can handle these multiple challenges and successfully develops a shared language to communicate about colour samples. Interestingly, the population does not reach maximum confidence for every challenge, as the chromaticity strategy stabilises to a value around 90% by the end of the interactions.

Finally, a third experiment was conducted to study the impact of perceptual deviation on the second experimental set-up. The results obtained show that the higher the perceptual deviation is, the more difficult it is for agents to communicate successfully about colour samples. The average confidence value for the more complex challenges is significantly affected by perceptual variance, but for low deviation values the population still manages to develop and align their colour, chromatic and lightness inventories to some degree. However, for perceptual deviations of 15% or higher agents the communication becomes extremely difficult and they fail to gain confidence even for the basic colour strategy.

Part IV

Conclusion and Future Directions

Chapter 7

Conclusion

The hypothesis of this thesis was that artificial agents are capable of successfully managing the development of successive stages of a language system by regulating the complexity of their interactions with a particular computational model of motivation called autotelic principle (see Section 1.4). The thesis achieved the two objectives defined in Section 1.5:

- The first achievement of this thesis is the concrete implementation of the autotelic principle for its usage in language evolution experiments, after presenting an overview of different psychological theories on motivation and a review of the computation models they have inspired. The core of this motivation system relies on the relation between challenges and skills: agents keep track of their performance for a set of challenges and use this information to decide which communicative tasks to address in order to continue developing their skills.
- The second achievement of this thesis is the demonstration that the autotelic principle can be used as a general mechanism to regulate complexity in language emergence in an autonomous way. The experiments carried out for both discrete and continuous domains have shown that the motivation system successfully regulates the complexity of linguistic interactions by allowing the population of artificial agents to progressively increase the expressive power of the shared language, going from one word to multi-word utterances. Moreover, Section 6.5 has provided evidence that the autotelic principle can manage several language strategies of similar complexity at the same time.

These two achievements have demonstrated that the autotelic principle is a useful mechanism in the development of language systems by a population of artificial agents. This thesis has extended the previous experiments on the autotelic principle by (a) providing experimental results that prove that the motivation system can be used by agents to successfully regulate the complexity of linguistic interactions in several stages in an autonomous way, proving that it allows agents to coordinate multiple challenges and its associated learning mechanisms in order to improve their shared language, (b) testing the motivation system in

multiple challenge dimensions for both discrete and continuous domains and (c) the integration of concept creation (prototypes) to these experiments (Chapter 6) and how differences in perception affect the creation and learning of these prototypes.

Together, the experiments reported in Chapters 5 and 6 have shown that the motivation system is capable of managing communicative tasks of various levels of complexity in both discrete and continuous environments. It has also demonstrated that it is a valid mechanism to coordinate distinct language strategies of the same complexity level. As a consequence, in all experimental set-ups the resulting shared language allows the population to communicate about the objects in their environment successfully.

Future research on the role of the autotelic principle in the emergence and development of communicative systems in populations of artificial agents should focus on the strengths and weaknesses of the autotelic principle, in order to better understand the limitations and preconditions of the system. Once this is achieved, research an interesting path to follow is the impact of embodied experiments in open-ended environments. Such experimental set-ups provide the ideal conditions to examine the autotelic principle managing multiple language strategies of different language domains. In addition, such experiments would allow the removal of certain constraints in the current implementation. Moreover, they will allow to observe the performance of the motivation system within a dynamic challenge extension and the resulting language strategies that could derive from this process.

7.1 Overview of publications

The work carried out during this thesis has led to three peer-reviewed publications that have been published during my time as a PhD student. The first two of them report on my experiments on the usage of the autotelic principle in the discrete domain. In these papers, I investigated how this motivational system could be employed by a population of artificial agents to set their own communicative challenges in order to develop their linguistic skills autonomously and how this mechanism allows them to make faster transitions between different learning phases.

The third publication studies the usefulness of the motivation system in a continuous domain. It shows how several language strategies of different complexity can be managed by a population of artificial agents using the autotelic principle in order to develop a shared language for the domain of colour, increasing the expressive power of the resulting system.

- **Miquel Cornudella**, Paul Van Eecke, and Remi van Trijp. How intrinsic motivation can speed up language emergence. In *Proceedings of the European Conference on Artificial Life*, pages 571–578, 2015.
- **Miquel Cornudella** and Thierry Poibeau. Language emergence in a population of artificial agents equipped with the autotelic principle, in *Proceed-*

ings of the Sixth Workshop on Cognitive Aspects of Computational Language Learning, pages 40–44. Association for Computational Linguistics, 2015.

- **Miquel Cornudella**, Thierry Poibeau, and Remi van Trijp. The role of intrinsic motivation in artificial language emergence: a case study on colour. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1646–1656. The COLING 2016 Organizing Committee, 2016.

Chapter 8

Future Directions

In the experiments reported in this thesis, we have studied the performance of the autotelic principle as a mechanism to regulate the complexity of linguistic interactions, facilitating the development of language systems of increasing complexity. The next research step should focus on the strengths and weaknesses of the current implementation, in order to better understand the limitations of the system. Once this is done, there have been issues that have not been addressed in these experiments but would be, however, highly interesting as future research paths. This concluding chapter explores some of these potential continuations, explaining their importance and sketching how they could be implemented.

The first part of the chapter (Section 8.1) addresses possible extensions to language evolution experiments that make use of the autotelic principle. This research could be continued in embodied experiments in open-ended environments, where different language domains are involved in the linguistic interactions. In such experimental set-ups, a dynamic challenge extension could be explored, where agents can autonomously extend the dimensions of the vector used for challenge configuration, and also the resulting language strategies that could emerge from this process.

The second part of the chapter (Section 8.2) enumerates other potential uses of the motivation system. They range from incorporating it into Intelligent Tutoring systems to improve language learning, exploring its performance in developmental robotic experiments and employing it to model the behaviour of computer-controlled characters in video games.

8.1 Open-Ended Learning Environments

This thesis has tested the autotelic principle on different environmental conditions. Despite the existing differences in the experimental design for artificial language emergence in discrete and continuous domains, namely the need of additionally creating, learning and aligning prototypes of different classes for the latter, both experimental set-ups differ in the kind of contexts in which the linguistic interactions occur:

- **Externally-imposed contexts:** In the chapter on the discrete domain,

the population has faced externally-imposed contexts. In these experiments, there was a predefined set of possible environments (a fixed set of scenes) in which the interacting agents were randomly situated. The speaker randomly chose the topic of the interaction, but neither of the interacting agents had any influence on the composition of the scene.

- **Individually-selected contexts:** The motivation system has also been studied in individually selected contexts, as in the experiments on the continuous domain of colour. In this second set of experiments, the speaker actively selects the colour samples that determine the context of the interaction depending on the communicative challenge that the agent wants to attempt. As in the first set of experiments, the topic is randomly chosen: one colour chip among the set of colour samples in the composed context.

In both cases, the population is exposed to a limited set of stimuli. Since the context is defined as (a) a randomly picked scene from a predetermined group for the experiments on the discrete domain or (b) a number of colour samples selected from a fixed set of colour chips by the speaker for the experiments on the domain of colour, there is always a maximum number of possible scenes in which agents can interact. This restriction should be lifted up in future experiments if agents interact in the real world.

8.1.1 Multi dimensional environments

So far, all experiments carried out have occurred in simulated environments where the relevant properties of objects for the emerging language system are determined in advance. For instance, in the experiments on the continuous domain, the environment is made up only of colour samples. These contexts do not contain any information about the size, the position, the distance, the sound, the temperature or the shape of the objects in the scene, as they are not important for the language strategies tested and therefore are not present in the emerging communication system.

These kinds of environments do not correspond with the situations in which human language developed, despite having been proved as appropriate for experiments on the evolution of language [136]. This restriction can be removed if agents are situated in *multi dimensional environments* where objects have attributes on several perception domains.

These type of situated interactions would enable experiments in which agents could analyse the context, determine the most relevant dimensions of the topic and describe it referring to the values of these significant dimensions. An experimental set-up of this kind would remove a limitation present in all the experiments reported in this thesis, namely the exploration of the autotelic principle as a mechanism to develop complex language systems that produce concepts for several domains.

Such experiments, however, require (a) a perception system to provide information about several cognitive domains and (b) a mechanism to decide which communicative challenges are relevant and worth exploring given the internal

state of an agent, its lexicon and the contexts it has been exposed to. The first requirement is already met by state-of-the-art robots used in research, like the SoftBank's NAO [51] and Pepper [97], the SONY QRIO [64] or the humanoid MYON [59]. All of these robots have (a) vision systems, mandatory for developing a communication system for colour, spatial categories, shapes or sizes, (b) microphones and speakers for perceiving and producing sounds and (c) a set of motors combined with several degrees of freedom that allow them to produce gestures and allow them to move in the given environment. The information obtained from the different sensors can then be processed using different algorithms (such as contour detection and image segmentation [4], edge detection [78], discrete differentiation [68] or anchoring identified objects [30] for the input images) to identify the most salient characteristics of the objects in the environment. It may be that for specific domains other sensors may be required, such as heat, temperature, humidity, distance or orientation detectors, but all current robots used in research are equipped with sensors to retrieve information from multiple cognitive domains.

Concerning the second requirement, the choice of the communicative challenges to explore can be addressed by modifying the algorithm used for challenge selection. In the experiments reported in this thesis speakers base their election on the complexity measure (level value) associated with each communicative task: agents choose to attempt the less complex challenges for which they have not yet achieved maximum confidence. This mechanism could be problematic in language evolutionary experiments that tackle multiple language domains, as it could get stuck attempting communicative challenges that address non-relevant dimensions (dimensions not salient in the different contexts it has interacted so far) in the environment instead of trying more difficult challenges for significant domains. This problem can be solved if agents make use of their perception systems to identify important dimensions in the environment. As this decision determines the "challenge path" to be explored, it should not be based only on the current interaction, but it should also take into account (a) the different contexts the agent has been exposed to and (b) the confidence levels associated by the agent with each communicative challenge.

8.1.2 Dynamic challenge configuration

In the autotelic principle, challenges are defined as vectors of several parameters that additionally are associated with a complexity level value. These parameters have a relation with some of the dimensions in the environment. For example, among the communicative challenges for the colour domain both *colour* and *membership* parameters introduce categories that act on the three colour dimensions perceived ($L^*a^*b^*$), while *lightness* or *chromatic* parameters do it only on a subset of these dimensions (L^* and a^*b^* , respectively). Nevertheless, both in the experiments for the discrete and colour domains this parametrisation was fixed and settled by the experimenter.

The dimensions that agents can perceive depend on their sensors and the algorithms they apply to the input data. For example, in order to be able to

perceive shapes, agents need an input stream of images and an algorithm to detect contours. Each algorithm provides information about different features of the input data. Depending on the environment, these features may capture potentially distinctive characteristics of objects, and therefore may be relevant to achieve successful communication. This information can be used to dynamically modify the dimension vector that determines the possible communicative challenges. The idea is that agents can extend the parametrisation values used to determine the different challenges once they identify important features not captured by their current challenge dimensions.

There are two possible situations in which the dynamic parametrisation can be explored: (a) one in which agents are initialised with a given set of parameters as a basis to explore different challenge configurations and (b) another where the population starts with no parameters identified. The latter increments the difficulty of the experiment, as it may be that agents start identifying different salient features in the environment. I would start exploring the first experimental set-up, providing agents with a basic set of parameters to generate challenges that they could extend afterwards, to study the resulting dynamics of such configuration. If this first set of experiments prove to be successful I would then move onto testing the second situation in which the population starts with no parameters in the challenge vector.

Either way, a dynamic challenge configuration may change completely the behaviour of the autotelic principle in comparison to the implementation used in this thesis, and therefore should be carefully studied before being implemented. So far new challenges are generated only when agents enter into the shake-up phase for an already known challenge. If agents can modify the challenge configuration once they identify a new relevant characteristic, they may then change the parametrisation at any time, messing up their current challenges without being in the shake-up phase.

To prevent this problem, I would create a list of unexplored features in the context. When a new salient feature in the environment is identified by an agent as potentially salient, it is stored in that list together with an associated value, a number that represents the estimation of how *relevant* a feature is in the environment. The relevant values of the "candidate" features (the potential features to extend the challenge parameters) would be updated after every interaction, increasing or decreasing their value in function of how prominent they have been in the contexts the agent has been exposed to so far.

This list would allow agents to rank potential features to add to the vector of parameters and use this information to extend the challenge dimensions under certain conditions. It is hard to determine these conditions in advance, but in order to stop an explosion of possible challenge configurations I would (a) force the agents to first focus on the different *challenge paths*, the different unexplored challenge parametrisations they can generate from their current vector of parameters, before exploring new ones and (b) allow them to add a new feature to the vector of parameters when they can no longer progress in any of their communicative tasks, selecting the one with the highest relevant value.

8.1.3 Generation of new language strategies

A dynamic challenge configuration modifies the initial vector of parameters and permits agents to create new challenges based on the new challenge configuration. As a consequence, agents need to create language strategies to associate to the new challenges. Each communicative task is associated with a particular language strategy. These strategies do not (a) impose any restriction to how or when lexical items should be created or (b) enforce certain word-order constraints in multi-word utterances, but provide agents with means to cope with the different communicative challenges. However, if agents can autonomously increase the vector of parameters used to define challenges, they also need a mechanism to generate new strategies to handle these new communicative tasks dynamically.

In this thesis language strategies have been implemented as semantic constraint networks using the Incremental Recruitment Language (IRL) (see Section 2.2). Language strategies are represented in IRL as meaning networks that combine several cognitive operations using variable linking. Cognitive operations are sequences of coded instructions, programmed by the experimenter, that perform some cognitive activity such as categorise, classify or filter. Such networks contain unconnected variables that are linked by agents to semantic entities from their inventories in order to build good conceptualisations and interpretations for the topic of the interaction.

It should be mentioned that a language strategy specifies the kind of semantic entities it can be combined with but does not impose any restriction on the number or the characteristics of these entities. The resulting semantic entities that will be part of an agent's lexicon do not depend only on a language strategy but also on the environment in which it is used. For example, all human languages use the basic colour strategy to split the colour space into colour categories, but the particular categories created differ among languages [149, 67, 45]. An example of a language strategy as a meaning network was already introduced in Figure 6.3 for the basic colour strategy. This meaning network contains several cognitive operations and one unconnected variable, `?cc`. This open variable allows agents to link the meaning network to the different colour categories in their inventory through a semantic entity.

How could agents be able to create new language strategies that were not planned in advance by the experimenter? IRL can generate new semantic networks as the result of a combinatorial search process of cognitive operations [122, 123]. This mechanism has already been used to study the origins of language strategies for the domain of colour by Bleys ([21], Chapter 11). The idea is to provide agents with a pool of basic cognitive operations. When they face a new challenge configuration for which they do not have any language strategy, agents can build a new one through a process of combinatorial search of these cognitive operations. When this process leads to a successful semantic network it can be stored by the agent as a language strategy for that challenge configuration.

It is important to stress that the more general the distinct cognitive operations are, the more useful they would be. Taking the meaning network for the basic colour strategy previously mentioned as an example, we observe that it is

composed of five cognitive operations. Some of these predicates¹ can be transformed into more general cognitive operations if an additional argument is added. For instance, the meaning predicate **profile-colour-dimensions** can be transformed into **profile-dimensions** if an extra argument *?dimensions* is provided. With such transformations, cognitive operations could become more powerful, as they could employ the same operations to different perceptual dimensions.

As previously stated, each dimension in the vector employed to determine challenges corresponds to a distinct feature in the environment. A communicative challenge is defined as a particular configuration of these vector dimensions, with a specific value associated with each feature. Features stand for distinct kinds of semantic entities. These values are used to determine the number of times each kind of semantic entity should appear in the language strategy associated with that challenge. For example, from the language strategies studied for the domain of colour, the basic colour strategy is associated to challenge configurations that only refer to one colour category, the membership strategy is associated to challenge configurations that may involve one membership and one colour strategy or the colour combination strategy to challenge configurations that include at least two colour categories.

Accordingly, a language strategy for a new challenge configuration should combine the set of different semantic entities associated with that challenge into a connected meaning network. Such meaning networks are obtained through a combinatorial search process that uses the pool of cognitive operations. A proper design of such operations should allow agents to build discriminative meaning networks for several language domains, but it may happen that agents are unable to build successful semantic networks for specific communicative challenges. This problem will occur when the cognitive operations available cannot be combined to create a connected meaning network for the set of semantic entities of the challenge. Unfortunately, in the current implementation agents do not have a way to autonomously create new cognitive operations when their set of cognitive operations fails to assemble a connected meaning network. It will be fascinating to explore this issue in the future but at this moment it constitutes a restriction of the system.

8.2 Other applications

Apart from exploring the usage of the autotelic principle in open-ended experimental set-ups, where some of the restrictions in the experiments carried out in this thesis can be removed, there are other applications in which this motivation system can be useful. For instance, in its origins, the autotelic principle was proposed as an alternative to the behaviourist approaches to developmental robotics [131] or, more recently, Beuls suggested to use it to model the knowledge of second language learners ([18], Chapter 9). This section briefly discusses three of these applications.

¹ *Cognitive operations* are IRL predicates that represent a cognitive activity that the agent has to perform. IRL contains another kind of meaning predicates, named *semantic entities*. See Subsection 2.2.1.

8.2.1 Intelligent tutoring systems

In her thesis, Beuls [18] point out that language learning applications, despite presenting several advantages in contrast to learning a language in a classroom, lack a proper personalised analysis of the language abilities of their users. In order to solve this problem, she proposed to build artificial tutors with which second language learners could interact in an individualised way. Such system would analyse the performance of students and provide exercises adapted to their current knowledge, facilitating their language improvement.

Her idea is that artificial tutors build a model of each user of the application to identify his level, strengths and flaws when learning a second language. With this information, the system can select the best exercises to increase the level of the student while avoiding falling into repetitive, tedious activities. Part of this task lies in identifying exercises that are challenging enough for the current knowledge of the user but not too much: in other words, that allow students to stay in a state of *flow* as much as possible during the lesson.

The autotelic principle can be used by artificial tutors in the learner's knowledge model of a user. Similarly to other autotelic activities, such as rock climbing or learning to play chess, in order to develop our proficiency for a second language we need to address exercises of different complexity. The set of different lessons of a language program can be labelled and ranked according to their difficulty, assigning a level value to each task. The artificial tutor can evaluate the level of a new student after an initial test consisting of exercises of different complexity. It can use the results to identify the proficiency of the student and create a knowledge model for him. With this model, the tutor can propose exercises according to the level of the student. This model is continually updated with the user's feedback, keeping track of his evolution and adapting the estimation of the student's level over time.

The model mimics the dynamics of a course in a language school: when students start, they have to face an exam to evaluate their level. The school uses these results to assign each student to different groups, which are usually split up according to various proficiency levels. The level of the students is regularly evaluated in class and they eventually change their course when they have learned the set of language skills taught in that classroom. The same approach can also be applied to other educational areas. For instance, the same architecture can be used to implement an application to learn mathematics, biology, physics or chemistry.

8.2.2 Developmental robotics

As stated in the introduction of this section, the autotelic principle was first proposed by Steels as an alternative to the behaviourist approaches to developmental robotics [131]. However, the research carried out so far has explored its utility as a mechanism to regulate the complexity of linguistic interactions [145, 32, 31].

The idea is to use the motivation system to self-regulate the developmental process of an embodied agent. As already explained in Chapter 4, the autotelic principle assumes that agents are a set of interconnected components, each of

them responsible for a particular task. In order to produce an action, such as move or point, the agent needs to combine a subset of its components. For example, in order to point to an object, an agent needs first to process the input images to identify the object and then coordinate the group of motors with which it can modify the position of its arm to the direction of that object.

It is possible for agents to achieve such goals if they can explore their sensorimotor space in order to build a mapping between instructions on a set of components and the resulting actions. The system can first monitor the performance of each component individually. For instance, the ability of the agent to use a motor to move an articulated member into the direction chosen and then evaluate the performance of a set of components to perform actions, as in the pointing example. This approach allows agents to first focus on simpler challenges and later move to more complex ones that require the interaction of several components.

8.2.3 Character modelling in video games

Recently, Artificial Intelligence researchers have looked at the possible application of motivation systems to model the behaviour of computer-controlled characters in video games. They aim to use AI techniques to improve the game experience, providing a personal, more enjoyable gameplay. Their attention is linked to the boost that this industry has undergone for the last three decades.

Nowadays, video games have become one of the most popular and fruitful businesses among the entertaining industries, with associated incomes that have been consistently growing between 9% and 15% annually for a period of 25 years [161]. The number of players has increased accordingly, with around 1.8 billion gamers worldwide [113].

The idea, as explained by Merrick [80] in the introduction of her book, is to "use information gathered by game data mining researchers about players to inform the design of novel self-motivated game-playing agents to control non-player characters". In other words, the goal of this research is to design algorithms that adapt the behaviour of the characters controlled by the game, both those that help the player and those that play against the player, to the style and level of the user of the game in order to achieve more enjoyable, diversified, involving gaming experiences.

Usually, such computer programs require players to develop a set of skills that they must use correctly in order to advance through the story. In some cases, the game may become repetitive and boring once the player has acquired those skills. In order to avoid this, the game can keep track of the level of proficiency of the player for each ability required. This information can then be used to react to her improvements, adapting the situations the player has to undergo to provide more exciting scenarios and also facilitate the improvement of her less developed skills.

The autotelic principle can be used for this purpose, as it already has a way to monitor the performance of an agent for different challenges. It will be necessary to determine the different game challenges, which may correspond to the mastery of different abilities, and the appropriate monitors to follow the evolution of those

skills during the gameplay. All this information can be processed by the game engine to dynamically adjust the game conditions to the advances of the player.

8.3 Summary

This chapter mentioned future paths for extending the research on the autotelic principle. The first part focused on what I think the next generation of language evolution experiments using the AP should explore. I argue that the next step should be embodied experiments primarily because it will allow the testing of the motivation system in environments where several language domains (space, colour, form, size, etc.) are prominent for communication, despite the difficulties that may arise from differences in perceptual deviation. Such experimental setups are also interesting to investigate (a) a dynamic extension of the challenge dimensions and (b) the language strategies agents will need to come up in order to communicate about new challenges. The former would allow agents to modify the vector of parameters used to define challenges when they identify a relevant feature in the environment not captured with their current challenge configuration, while the latter would consist of generating new meaning networks by a combinatorial search process that uses the pool of cognitive operations available to agents.

The second part focused on research routes aside from experiments of language development. The usage of the autotelic principle in developmental robotic experiments is maybe the most straight forward to apply, as the system was originally designed for such experiments. Recently, artificial intelligence researchers have explored the utility of motivation systems as mechanisms to provide precise information about users of large scale applications. The idea is to use the information gathered to personalise the user experience, facilitating the learning of second languages in Intelligent Tutoring systems or adapting the conditions of a video game dynamically to match the player's skill.

Appendix A

Example of execution of an IRL-network

This appendix demonstrates how an IRL-network is executed. I illustrate the *execution* of an IRL-network using as example the basic colour strategy used in the experiments in Chapter 6.

As the other language strategies in Chapter 6, the basic colour strategy has been modelled as a connected network of meaning predicates. The corresponding IRL-network for basic colour strategy is shown in Figure 6.3. Nevertheless, it is reproduced in Figure A.1 in order to help the reader. This network combines five cognitive operations: **equal-to-context**, **profile-colour-dimensions**, **categorise-by-colour**, **get-basic-colour-category-set** and **select-most-activated** (see Subsection 6.2.6 for a description of each operation).

In IRL-networks meaning predicates are connected using variables. For example, in the basic colour strategy the meaning predicate **equal-to-context** introduces the variable *?s1*, which represents the colour samples in the context, and **profile-colour-dimensions** gets a set of colour samples (*?s1*) and filters them according to colour dimensions, which in this case returns the original set on a new variable *?s2*.

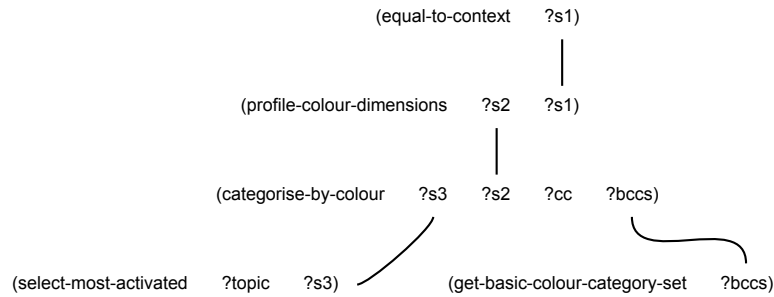


Figure A.1: Example of a language strategy as a connected network of cognitive operations. This example corresponds to the basic colour strategy.

The goal *execution* of an IRL-network is to bind the different variables to actual objects (see Subsection 2.2.2). In conceptualisation, the meaning structure is executed to determine if the IRL-network leads to a discriminative conceptualisation of the topic (in the example a colour sample from the context). In interpretation, the hearer reconstructs the IRL-network from an utterance, which introduces different semantic entities linked to lexical constructions, and executes it in order to identify the communicative goal of the speaker. The rest of the appendix illustrates the execution of the IRL-network for the basic colour strategy in both conceptualisation and interpretation.

Execution of an IRL-network in conceptualisation

Both conceptualisation and interpretation are parts of the *semiotic cycle* (see Section 2.1). In order to show the execution of the basic colour strategy we determine that the context of the interaction between two agents consist of three different colour samples, which correspond to the following focal colours for English¹: *green*, *black* and *pink* (Figure A.2). The speaker has chosen as a topic of the interaction the *black* colour sample.



Figure A.2: Context in which for the basic colour strategy will be executed. In this case, the context contains three colour samples that correspond to the English focal colours *green*, *black* and *pink*.

The goal of the speaker is to find if the basic colour strategy is able to create a discriminative conceptualisation of the topic. In other words, the speaker will execute the IRL-network in order to find a colour category (variable *?cc*) that successfully conceptualise the topic *?topic* (the colour sample *black* in the example). The first primitives that are evaluated are those that introduce only one variable in the network. For instance, it may start executing the cognitive operation *equal-to-context*, which binds the variable *?s1* to the colour samples in the context (Figure A.2).

The execution can continue evaluating the primitive **profile-colour-dimensions**. A profiling primitive filters on certain dimensions of colours and returns a filtered set. However, profiling on colour dimensions returns the original set. Therefore, the resulting variable *?s2* contains the same colour samples as *?s1*.

Before categorising by colour, the network must execute the **get-basic-category-set** to obtain the set of basic colour categories of the speaker. This primitive binds this set to the variable *?bccs*. In this case, *?bccs* contains the eleven basic colours for English (Figure A.4).

So far, the execution of the IRL-network has bound the variables *?s1* and *?s2* to sets containing the colour samples in the context (Figure A.2) and the

¹See Sturges & Whitfield [149]

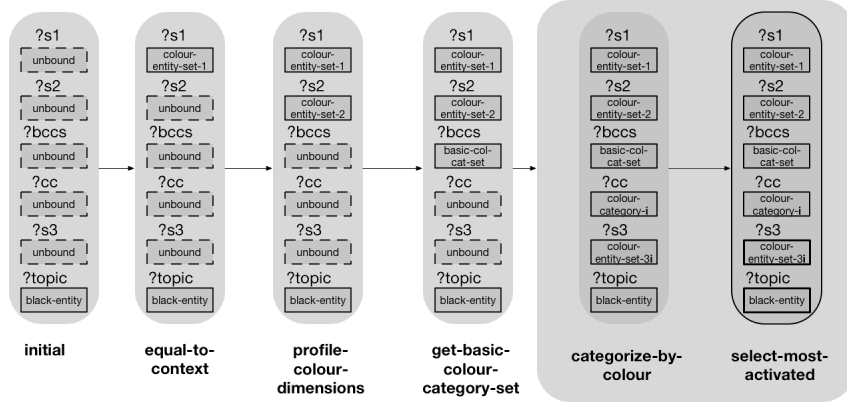


Figure A.3: Visual representation of the execution process for the basic colour strategy in conceptualisation. The variables in the meaning network get bound as the different cognitive operations are evaluated. The categorisation by colour leads to three different hypotheses. The only hypothesis that fulfils that the most activated entity in *colour-entity-set-3i* corresponds to the topic *black-entity* leads to a *solution*.



Figure A.4: Basic colour categories known by the speaker. In this case, the set contains colour samples that correspond to all English focal colours.

variable $?bccs$ to the set of basic colour categories known by the agent. The **categorise-by-colour** primitive categorises the input set $?s2$ according to a colour category $?cc$, which can be one of the colour categories in $?bccs$. The variable $?cc$ introduces a semantic entity to the IRL-network, which can only be one of the basic colour categories known by the agent.

At this point the execution process splits the search into three hypotheses, as there are three possible bindings to semantic entities for the variable $?cc$ for which the resulting set $?s3$ is not empty: *green*, *black* and *pink*. For each hypothesis, $?s3$ contains the set of colour samples in $?s2$ that have as a closest colour prototype the colour category $?cc$ with an associated activation value that represents how similar the colour sample is to the colour category introduced by $?cc$. Therefore, the resulting set of each hypothesis only contains one colour sample. For example, $?s3$ contains the colour sample *green* when $?cc$ is bound to the colour category of the agent that represents the English focal colour labelled as "green".

There is only one cognitive operation that has not been evaluated yet: **select-most-activated**. This primitive selects the colour sample in $?s3$ with the high-

est activation value. As in each of the three hypotheses $?s3$ only has one colour sample, it returns that colour sample. From the three hypotheses, the only one that fulfils that $?topic$ is bind to the selected topic (the colour sample *black*) is when $?cc$ is bound to the colour category "black".

As the IRL-network of the basic colour strategy can successfully discriminate the colour sample chosen by the speaker in the context referring to the colour category "black", the execution lead to a *solution*. In the semiotic cycle, the speaker will use this information as the meaning for which it has to formulate an utterance. A visual representation of the execution just described is shown in Figure A.3.

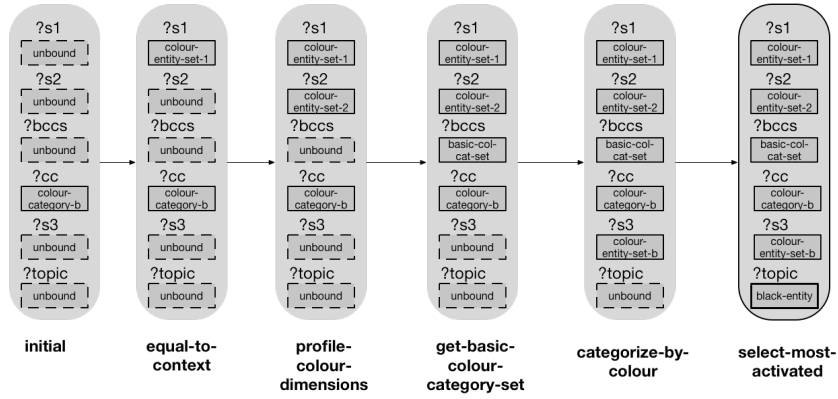


Figure A.5: Visual representation of the execution process for the basic colour strategy in interpretation. The network is initialised with the variable $?cc$ bound to the colour category "black". The variables of the meaning network get bound as the different meaning predicates are evaluated. At the end of the execution $?topic$ is bound to the colour sample *black* of the context, and therefore to a solution for this IRL-network. In this example, the $?topic$ corresponds to the intended topic of the speaker.

Execution of an IRL-network in interpretation

The goal in interpretation is to execute the recovered meaning (in the form of an IRL-network) from the input utterance in order to identify the communicative goal of the speaker. In the current example, as we are focusing on the basic colour strategy, the goal is to identify the topic of the interaction (the colour sample bound to $?topic$) given the colour category expressed by the speaker. Therefore, assuming that the interpretation process is part of the same interaction, the IRL-network is initialised with the $?cc$ bound to the colour category "black".

The execution start with these cognitive operations that introduce only one variable in the network: **equal-to-context** binds $textit?s1$ to a set containing the colour samples in the context, and **get-basic-colour-category-set** binds

textit?bccc to a set composed of the colour categories known by the hearer. The execution continues with the evaluation of **profile-colour-dimensions**. This cognitive operation binds *?s2* to a set containing the colour samples in the context, as it occurred in conceptualisation.

At this point the cognitive operation **categorise-by-colour** can be executed, as the only unbound variable is *?s3*. As in the initialisation of the network the variable *?cc* is bind to the colour category "black", the result of evaluating this meaning predicate is a resulting set, bind to the variable *?s3*, that contains only the colour sample *black* with the associated activation value. Therefore, evaluating **select-most-activated** binds the colour sample *black* to the variable *?topic*, which in this case corresponds to the topic chosen by the speaker at the beginning of the interaction. A visual representation of the execution just described is shown in Figure A.5.

Conclusion

This appendix has presented an example of how an IRL-network is executed in both conceptualisation and interpretation. The interested reader on IRL is referred to [141, 152, 122, 123] for further information on the system.

Appendix B

Parametrization of the autotelic principle

This appendix explains how the different parameters in the implementation of the autotelic principle used in this thesis have been selected. When Steels proposed the autotelic principle in 2004 [131] he mentioned two characteristics about how the motivation system should be implemented:

- "In our experiments to date we have found that the system should start with the lowest challenge levels possible for all components (instead of starting with a random configuration) so as to build up steadily in a bottom-up fashion." [131, p. 11]
- "We also know from our experiments that a conservative strategy (where only one repair is executed in the Operation Phase, and one parameter is changed in the Shake Up Phase) is much more desirable than drastic and rapid change." [131, p. 14]

In other words, he identified two situations that facilitate the development of skills: (a) agents should start attempting those challenges with the lowest complexity levels instead of random parameter configurations and (b) during both operation and shake-up phases, slight changes are preferred over drastic modifications. The latter suggests that the system performs better with limited changes.

I adapted these pieces of advice to employ the autotelic principle in experiments of language emergence. During the operation phase of a challenge, agents only use one repair for each time they participate in an interaction: (a) when the speaker has to conceptualise a topic to produce an utterance, (b) when the hearer comprehends an input utterance in order to determine the topic of the interaction or (c) when the hearer receives feedback from the speaker. Also, in the shake-up phase new challenges are generated increasing or decreasing by one unit one parameter in the challenge parametrisation, instead of randomly creating new challenge configurations.

However, despite Steels stating that (a) the internal state of a challenge should be determined by monitors that track the performance of the system over a time

span and (b) this period should be long enough for letting the system acquire the skills to reach a certain performance level for that challenge, he did not specify how it should be implemented [131, p. 10]:

The cost of a parameter configuration cannot simply be computed by applying a simple function (as in the travelling sales man for example, where cost is basically the length of a path) but must be derived from monitoring actual performance of the system over a particular period of time, including enough time to achieve the acquisition of the necessary skills to reach a certain performance level, (2) this monitoring period must include enough time for the system to acquire the necessary skills to reach a certain performance level. It is to be noted that the objective is not to get optimal performance, but rather to explore the landscape of possibilities in such a way that a higher degree of complexity is reached.

In the implementation used in this thesis, there are two parameters associated with each challenge: *confidence*, the certainty the system has of being proficient in a particular task, and *persistence*, a measure to ensure a minimum number of attempts before the system decides to change its current task. Section 4.5 describes how these parameters are captured by a set of variables (δ_{inc_conf} , δ_{dec_conf} and δ_{dec_pers}) and explains how the system updates these variables after it performs an action in a given challenge.

The value δ_{dec_pers} , related to the parameter *persistence* limits the minimum number of interactions the system will attempt on a particular challenge. This value has been set to 0.02 in order to provide enough time to acquire the necessary skills for a particular challenge.

The values used to update the parameter *confidence*, δ_{inc_conf} and δ_{dec_conf} , have been determined after testing their performance in 400 different configurations of these values on the experiment described in Section 5.3¹. To ensure the consistency of the results each configuration has been tested on four runs of 7000 interactions. The outcome of these simulations is presented in table B.1, which present the resulting communicative success, alignment success, the average cumulative confidence in challenges and the average number of grammatical and lexical constructions for each pair of δ_{inc_conf} and δ_{dec_conf} values examined.

The results show that for most configurations the system obtains considerably good results, with almost perfect outcomes for each measure. These results are related to the experimental set-up: the different communicative challenges are pretty easy to achieve when addressed as learning tasks of increasing complexity, as agents have to agree on how to name a set of 21 possible characteristics of objects (during the first two learning tasks) and a small set of grammatical constructions.

However, from all configurations tested the values 0.005 for δ_{inc_conf} and 0.020 for δ_{dec_conf} provide the best results. With these values, the population of

¹The objects and scenes used correspond to the ones described in Subsection 5.1.1, except for the shape value of obj_7 , which has been modified to *triangular* ($obj_7 = \{prototype(obj_i, chair), shape(obj_i, triangular), colour(obj_i, black)\}$).

agents achieves 100% communicative and alignment success and the maximum number of cumulative confidence for the five communicative challenges. Moreover, it is the only configuration for which all agents converge to the minimum set of lexical and grammatical constructions. As a consequence, δ_{inc_conf} and δ_{dec_conf} have been set to $[0.005, 0.020]$ in all the experiments reported in this thesis.

It is necessary to point out that there are other combinations that also provide extremely good results, specifically those with δ_{inc_conf} and δ_{dec_conf} value pairs of $[0.010, 0.025]$, $[0.035, 0.095]$, $[0.085, 0.075]$ and $[0.100, 0.095]$. Such configurations have been discarded for two reasons. The first one is that $[0.005, 0.020]$ is, as has been set before, the only configuration that allows the population to converge to a minimal set of lexical and grammatical constructions. The second reason is that gaining confidence at a lower pace provides the system with more time to properly develop the skills required for that challenge. This is particularly relevant for more challenging communicative challenges, such as developing a lexicon (and its associated prototypes) for the different colour language strategies used in chapter 6.

δ_{inc_conf}	δ_{dec_conf}	Comm. succ.	Align. succ.	Chall. mast.	Gram. cxns.	Lex. cxns.
0.005	0.005	0.997	0.991	5.0	4.0	21.187
0.005	0.010	0.999	0.996	5.0	4.0	21.207
0.005	0.015	0.998	0.995	5.0	4.0	21.183
0.005	0.020	1.0	1.0	5.0	4.0	21.0
0.005	0.025	0.992	0.977	4.992	4.0	21.597
0.005	0.030	0.995	0.983	4.937	4.0	21.655
0.005	0.035	0.984	0.958	4.626	3.999	22.094
0.005	0.040	0.975	0.940	4.112	3.934	22.324
0.005	0.045	0.965	0.935	3.078	3.51	21.263
0.005	0.050	0.929	0.908	1.627	2.732	19.806
0.005	0.055	0.865	0.865	0.472	1.949	16.017
0.005	0.060	0.871	0.871	0.242	1.233	9.113
0.005	0.065	0.869	0.869	0.117	1.0	7.0
0.005	0.070	0.862	0.862	0.067	1.0	7.0
0.005	0.075	0.869	0.869	0.059	1.0	7.0
0.005	0.080	0.863	0.863	0.047	1.0	7.0
0.005	0.085	0.868	0.868	0.048	1.0	7.0
0.005	0.090	0.865	0.865	0.041	1.0	7.0
0.005	0.095	0.874	0.873	0.044	1.0	7.062
0.005	0.100	0.860	0.860	0.037	1.0	7.0
0.010	0.005	0.989	0.985	5.0	4.0	21.235
0.010	0.010	0.988	0.986	5.0	4.0	21.237
0.010	0.015	0.996	0.989	5.0	4.0	21.385
0.010	0.020	0.999	0.998	5.0	4.0	21.155
0.010	0.025	1.0	1.0	5.0	4.0	21.007
0.010	0.030	0.995	0.986	4.999	4.0	21.448
0.010	0.035	0.998	0.994	4.999	4.0	21.21
0.010	0.040	0.993	0.984	4.999	4.0	21.506
0.010	0.045	0.991	0.980	4.999	4.0	21.496
0.010	0.050	0.994	0.981	4.999	4.0	21.487
0.010	0.055	0.995	0.985	4.999	4.0	21.535
0.010	0.060	0.990	0.966	4.996	4.0	21.848
0.010	0.065	0.990	0.968	4.980	4.0	21.527
0.010	0.070	0.994	0.979	4.851	3.993	21.359

150 APPENDIX B. PARAMETRIZATION OF THE AUTOTELIC PRINCIPLE

δ_{inc_conf}	δ_{dec_conf}	Comm. succ.	Align. succ.	Chall. mast.	Gram. cxns.	Lex. cxns.
0.010	0.075	0.984	0.968	4.400	3.894	22.024
0.010	0.080	0.954	0.921	3.259	3.551	21.792
0.010	0.085	0.949	0.935	2.848	3.411	21.064
0.010	0.090	0.871	0.862	0.868	2.264	16.072
0.010	0.095	0.877	0.877	0.648	2.684	17.728
0.010	0.100	0.863	0.863	0.403	2.209	15.413
0.015	0.005	0.989	0.987	5.0	4.0	21.230
0.015	0.010	0.986	0.979	5.0	4.0	21.435
0.015	0.015	0.998	0.996	5.0	4.0	21.185
0.015	0.020	0.999	0.999	5.0	4.0	21.071
0.015	0.025	0.998	0.996	4.999	4.0	21.193
0.015	0.030	0.999	0.996	4.999	4.0	21.149
0.015	0.035	0.997	0.992	4.999	4.0	21.314
0.015	0.040	0.999	0.997	4.999	4.0	21.131
0.015	0.045	0.996	0.991	4.999	4.0	21.395
0.015	0.050	0.995	0.988	4.999	4.0	21.312
0.015	0.055	0.997	0.991	4.999	4.0	21.382
0.015	0.060	0.996	0.989	4.999	4.0	21.353
0.015	0.065	0.999	0.996	4.999	4.0	21.171
0.015	0.070	0.993	0.985	4.999	4.0	21.621
0.015	0.075	0.996	0.991	4.999	4.0	21.411
0.015	0.080	0.992	0.977	4.997	4.0	21.482
0.015	0.085	0.995	0.986	4.998	4.0	21.311
0.015	0.090	0.993	0.978	4.996	4.0	21.778
0.015	0.095	0.996	0.992	4.998	4.0	21.285
0.015	0.100	0.995	0.989	4.988	4.0	21.234
0.020	0.005	0.903	0.898	5.0	4.0	21.524
0.020	0.010	0.990	0.990	5.0	4.0	21.037
0.020	0.015	0.988	0.987	5.0	4.0	21.127
0.020	0.020	0.997	0.998	4.999	4.0	21.139
0.020	0.025	0.996	0.989	4.999	4.0	21.432
0.020	0.030	0.997	0.992	4.999	4.0	21.320
0.020	0.035	0.997	0.993	4.999	4.0	21.271
0.020	0.040	0.995	0.988	4.999	4.0	21.468
0.020	0.045	0.999	0.999	4.999	4.0	21.023
0.020	0.050	0.998	0.993	4.999	4.0	21.248
0.020	0.055	0.999	0.997	4.999	4.0	21.114
0.020	0.060	0.996	0.990	4.999	4.0	21.367
0.020	0.065	0.995	0.989	4.999	4.0	21.492
0.020	0.070	1.0	0.999	5.0	4.0	21.046
0.020	0.075	0.998	0.996	4.998	4.0	21.152
0.020	0.080	0.995	0.986	4.999	4.0	21.380
0.020	0.085	0.998	0.995	4.999	4.0	21.222
0.020	0.090	0.995	0.986	4.997	4.0	21.409
0.020	0.095	0.999	0.999	4.999	4.0	21.067
0.020	0.100	0.987	0.967	4.993	4.0	21.995
0.025	0.005	0.946	0.942	5.0	4.0	21.520
0.025	0.010	0.975	0.965	5.0	4.0	21.789
0.025	0.015	0.997	0.992	5.0	4.0	21.395
0.025	0.020	0.995	0.990	4.999	4.0	21.386
0.025	0.025	0.999	0.999	4.999	4.0	21.094
0.025	0.030	0.989	0.989	4.999	4.0	21.076
0.025	0.035	0.997	0.992	4.999	4.0	21.295
0.025	0.040	0.999	0.998	4.999	4.0	21.094
0.025	0.045	0.995	0.989	4.999	4.0	21.402
0.025	0.050	0.988	0.973	4.999	4.0	21.990

δ_{inc_conf}	δ_{dec_conf}	Comm. succ.	Align. succ.	Chall. mast.	Gram. cxns.	Lex. cxns.
0.025	0.055	0.999	0.999	4.999	4.0	21.080
0.025	0.060	0.995	0.991	4.999	4.0	21.383
0.025	0.065	0.998	0.995	4.999	4.0	21.152
0.025	0.070	0.997	0.990	4.999	4.0	21.344
0.025	0.075	0.995	0.984	4.999	4.0	21.380
0.025	0.080	0.999	0.998	4.999	4.0	21.093
0.025	0.085	0.999	0.998	4.999	4.0	21.104
0.025	0.090	0.997	0.992	4.996	4.0	21.342
0.025	0.095	0.990	0.972	4.998	4.0	21.786
0.025	0.100	0.999	0.999	4.999	4.0	21.069
0.030	0.005	0.970	0.968	5.0	4.0	21.350
0.030	0.010	0.972	0.960	5.0	4.0	21.623
0.030	0.015	0.939	0.930	5.0	4.0	21.658
0.030	0.020	0.995	0.990	4.999	4.0	21.504
0.030	0.025	0.996	0.991	4.999	4.0	21.447
0.030	0.030	0.998	0.996	4.999	4.0	21.176
0.030	0.035	0.987	0.981	4.999	4.0	21.490
0.030	0.040	0.998	0.995	4.999	4.0	21.267
0.030	0.045	0.993	0.985	4.999	4.0	21.500
0.030	0.050	0.998	0.996	4.999	4.0	21.204
0.030	0.055	0.999	0.999	4.999	4.0	21.051
0.030	0.060	0.996	0.992	4.999	4.0	21.401
0.030	0.065	0.997	0.992	4.999	4.0	21.298
0.030	0.070	0.992	0.980	4.999	4.0	21.663
0.030	0.075	0.998	0.995	4.999	4.0	21.257
0.030	0.080	0.999	0.998	4.999	4.0	21.104
0.030	0.085	0.998	0.994	4.999	4.0	21.204
0.030	0.090	0.9989	0.996	4.998	4.0	21.179
0.030	0.095	0.997	0.991	4.999	4.0	21.255
0.030	0.100	0.999	0.996	4.999	4.0	21.145
0.035	0.005	0.947	0.942	5.0	4.0	21.621
0.035	0.010	0.976	0.970	5.0	4.0	21.593
0.035	0.015	0.977	0.972	5.0	4.0	21.701
0.035	0.020	0.977	0.976	4.999	4.0	21.379
0.035	0.025	0.989	0.989	4.999	4.0	21.072
0.035	0.030	0.997	0.992	4.999	4.0	21.354
0.035	0.035	0.978	0.973	4.999	4.0	21.504
0.035	0.040	0.998	0.995	4.999	4.0	21.265
0.035	0.045	0.995	0.989	4.999	4.0	21.574
0.035	0.050	0.998	0.996	4.999	4.0	21.186
0.035	0.055	0.996	0.990	4.999	4.0	21.432
0.035	0.060	0.993	0.982	4.999	4.0	21.735
0.035	0.065	0.999	0.997	4.999	4.0	21.113
0.035	0.070	0.999	0.999	4.999	4.0	21.073
0.035	0.075	0.997	0.992	4.999	4.0	21.341
0.035	0.080	0.998	0.995	4.999	4.0	21.176
0.035	0.085	0.999	0.997	4.999	4.0	21.164
0.035	0.090	0.997	0.990	4.999	4.0	21.383
0.035	0.095	1.0	1.0	5.0	4.0	21.012
0.035	0.100	0.999	0.999	4.999	4.0	21.045
0.040	0.005	0.890	0.884	5.0	4.0	21.930
0.040	0.010	0.956	0.949	5.0	4.017	21.434
0.040	0.015	0.907	0.901	5.0	4.0	21.554
0.040	0.020	0.990	0.989	4.999	4.0	21.104
0.040	0.025	0.999	0.998	4.999	4.0	21.140
0.040	0.030	0.968	0.956	4.999	4.001	21.663

152 APPENDIX B. PARAMETRIZATION OF THE AUTOTELIC PRINCIPLE

δ_{inc_conf}	δ_{dec_conf}	Comm. succ.	Align. succ.	Chall. mast.	Gram. cxns.	Lex. cxns.
0.040	0.035	0.996	0.991	4.999	4.0	21.304
0.040	0.040	0.979	0.977	4.999	4.0	21.275
0.040	0.045	0.997	0.993	4.999	4.0	21.266
0.040	0.050	0.994	0.984	4.999	4.0	21.580
0.040	0.055	0.976	0.973	4.999	4.0	21.310
0.040	0.060	0.996	0.990	4.999	4.0	21.397
0.040	0.065	0.999	0.998	4.999	4.0	21.122
0.040	0.070	0.997	0.991	4.999	4.0	21.313
0.040	0.075	0.997	0.993	4.999	4.0	21.240
0.040	0.080	0.997	0.992	4.998	4.0	21.315
0.040	0.085	0.998	0.994	4.999	4.0	21.324
0.040	0.090	0.997	0.993	4.99	4.0	21.332
0.040	0.095	0.994	0.982	4.999	4.0	21.462
0.040	0.100	0.999	0.999	4.998	4.0	21.081
0.045	0.005	0.882	0.879	5.0	4.0	21.584
0.045	0.010	0.969	0.964	5.0	4.0	21.439
0.045	0.015	0.980	0.979	5.0	4.0	21.224
0.045	0.020	0.975	0.962	4.999	4.0	21.608
0.045	0.025	0.979	0.972	4.999	4.0	21.491
0.045	0.030	0.998	0.994	4.999	4.0	21.222
0.045	0.035	0.993	0.983	4.999	4.0	21.767
0.045	0.040	0.998	0.996	4.999	4.0	21.177
0.045	0.045	0.987	0.984	4.999	4.0	21.352
0.045	0.050	0.989	0.989	4.999	4.0	21.089
0.045	0.055	0.999	0.997	4.999	4.0	21.096
0.045	0.060	0.990	0.989	4.999	4.0	21.060
0.045	0.065	0.992	0.980	4.999	4.0	21.461
0.045	0.070	0.998	0.994	4.999	4.0	21.217
0.045	0.075	0.997	0.992	4.999	4.0	21.289
0.045	0.080	0.998	0.997	4.999	4.0	21.171
0.045	0.085	0.999	0.996	4.999	4.0	21.127
0.045	0.090	0.999	0.997	4.999	4.0	21.126
0.045	0.095	0.995	0.990	4.999	4.0	21.348
0.045	0.100	0.997	0.993	4.999	4.0	21.290
0.050	0.005	0.922	0.920	5.0	4.0	21.426
0.050	0.010	0.986	0.979	5.0	4.0	21.450
0.050	0.015	0.968	0.959	5.0	4.0	21.529
0.050	0.020	0.959	0.952	4.999	4.0	21.566
0.050	0.025	0.999	0.999	4.999	4.0	21.045
0.050	0.030	0.968	0.964	4.999	4.0	21.421
0.050	0.035	0.987	0.982	4.999	4.0	21.457
0.050	0.040	0.999	0.999	4.999	4.0	21.072
0.050	0.045	0.987	0.986	4.999	4.0	21.200
0.050	0.050	0.981	0.980	4.999	4.0	21.220
0.050	0.055	0.992	0.979	4.999	4.0	21.736
0.050	0.060	0.996	0.988	4.999	4.0	21.173
0.050	0.065	0.994	0.983	4.999	4.0	21.292
0.050	0.070	0.999	0.996	4.999	4.0	21.171
0.050	0.075	0.998	0.996	4.998	4.0	21.175
0.050	0.080	0.996	0.988	4.999	4.0	21.556
0.050	0.085	0.995	0.991	4.998	4.0	21.320
0.050	0.090	0.999	0.998	4.999	4.0	21.093
0.050	0.095	0.999	0.998	4.999	4.0	21.086
0.050	0.100	0.998	0.996	4.999	4.0	21.161
0.055	0.005	0.863	0.855	5.0	4.0	22.125
0.055	0.010	0.938	0.933	5.0	4.0	21.497

δ_{inc_conf}	δ_{dec_conf}	Comm. succ.	Align. succ.	Chall. mast.	Gram. cxns.	Lex. cxns.
0.055	0.015	0.950	0.946	5.0	4.0	21.423
0.055	0.020	0.997	0.993	4.999	4.0	21.335
0.055	0.025	0.912	0.906	4.999	4.0	21.543
0.055	0.030	0.989	0.986	4.999	4.0	21.303
0.055	0.035	0.977	0.972	4.999	4.0	21.392
0.055	0.040	0.978	0.976	4.999	4.0	21.166
0.055	0.045	0.985	0.980	4.999	4.0	21.380
0.055	0.050	0.997	0.992	4.999	4.0	21.370
0.055	0.055	0.999	0.999	4.999	4.0	21.045
0.055	0.060	0.996	0.990	4.999	4.0	21.335
0.055	0.065	0.998	0.994	4.999	4.0	21.299
0.055	0.070	0.996	0.993	4.999	4.0	21.407
0.055	0.075	0.993	0.981	4.999	4.0	21.814
0.055	0.080	0.996	0.988	4.999	4.0	21.507
0.055	0.085	0.998	0.995	4.999	4.0	21.162
0.055	0.090	0.997	0.994	4.999	4.0	21.304
0.055	0.095	0.989	0.989	4.999	4.0	21.102
0.055	0.100	0.998	0.995	4.997	4.0	21.290
0.060	0.005	0.906	0.906	5.0	4.0	21.507
0.060	0.010	0.956	0.951	5.0	4.0	21.679
0.060	0.015	0.955	0.948	5.0	4.0	21.797
0.060	0.020	0.937	0.937	4.999	4.0	21.142
0.060	0.025	0.976	0.969	4.999	4.0	21.551
0.060	0.030	0.983	0.978	4.999	4.0	21.605
0.060	0.035	0.9845	0.977	4.999	4.0	21.683
0.060	0.040	0.987	0.983	4.999	4.0	21.298
0.060	0.045	0.997	0.991	4.999	4.0	21.401
0.060	0.050	0.994	0.987	4.999	4.0	21.636
0.060	0.055	0.999	0.997	4.999	4.0	21.144
0.060	0.060	0.987	0.981	4.999	4.0	21.418
0.060	0.065	0.991	0.980	4.999	4.0	21.699
0.060	0.070	0.998	0.994	4.999	4.0	21.242
0.060	0.075	0.957	0.947	4.900	4.004	21.523
0.060	0.080	0.998	0.996	4.999	4.0	21.169
0.060	0.085	0.989	0.985	4.999	4.0	21.350
0.060	0.090	0.996	0.989	4.998	4.0	21.372
0.060	0.095	0.995	0.989	4.999	4.0	21.489
0.060	0.100	0.990	0.990	4.999	4.0	21.104
0.065	0.005	0.937	0.930	5.0	4.0	21.717
0.065	0.010	0.947	0.946	5.0	4.0	21.329
0.065	0.015	0.937	0.934	5.0	4.0	21.396
0.065	0.020	0.957	0.948	4.999	4.0	21.627
0.065	0.025	0.950	0.949	4.999	4.0	21.174
0.065	0.030	0.932	0.928	4.999	4.0	21.557
0.065	0.035	0.996	0.992	4.999	4.0	21.287
0.065	0.040	0.982	0.980	4.999	4.0	21.146
0.065	0.045	0.994	0.986	4.999	4.0	21.666
0.065	0.050	0.999	0.997	4.999	4.0	21.179
0.065	0.055	0.999	0.999	4.999	4.0	21.070
0.065	0.060	0.988	0.986	4.999	4.0	21.166
0.065	0.065	0.989	0.988	4.998	4.0	21.160
0.065	0.070	0.987	0.982	4.998	4.0	21.418
0.065	0.075	0.999	0.996	4.999	4.0	21.161
0.065	0.080	0.987	0.985	4.999	4.0	21.275
0.065	0.085	0.997	0.992	4.995	4.0	21.248
0.065	0.090	0.990	0.987	4.997	4.0	21.317

δ_{inc_conf}	δ_{dec_conf}	Comm. succ.	Align. succ.	Chall. mast.	Gram. cxns.	Lex. cxns.
0.065	0.095	0.996	0.988	4.999	4.0	21.429
0.065	0.100	0.990	0.978	4.999	4.0	21.991
0.070	0.005	0.874	0.871	5.0	4.035	21.538
0.070	0.010	0.852	0.850	5.0	4.0	21.734
0.070	0.015	0.975	0.965	4.999	4.001	21.825
0.070	0.020	0.923	0.920	4.999	4.0	21.439
0.070	0.025	0.969	0.965	4.999	4.0	21.352
0.070	0.030	0.984	0.977	4.999	4.0	21.600
0.070	0.035	0.985	0.979	4.999	4.0	21.504
0.070	0.040	0.989	0.984	4.999	4.0	21.307
0.070	0.045	0.989	0.986	4.999	4.0	21.360
0.070	0.050	0.995	0.986	4.999	4.0	21.407
0.070	0.055	0.964	0.954	4.999	4.0	21.810
0.070	0.060	0.939	0.935	4.97	4.0	21.426
0.070	0.065	0.970	0.969	4.998	4.0	21.195
0.070	0.070	0.995	0.983	4.999	4.0	21.460
0.070	0.075	0.994	0.984	4.999	4.0	21.703
0.070	0.080	0.997	0.991	4.999	4.0	21.439
0.070	0.085	0.964	0.958	4.974	4.0	21.359
0.070	0.090	0.997	0.994	4.998	4.0	21.337
0.070	0.095	0.995	0.987	4.999	4.0	21.448
0.070	0.100	0.997	0.991	4.999	4.0	21.164
0.075	0.005	0.859	0.858	5.0	4.0	21.571
0.075	0.010	0.768	0.767	5.0	4.0	21.870
0.075	0.015	0.904	0.898	5.0	4.0	21.690
0.075	0.020	0.914	0.910	4.999	4.0	21.695
0.075	0.025	0.815	0.809	4.999	4.0	21.922
0.075	0.030	0.931	0.929	4.999	4.0	21.416
0.075	0.035	0.990	0.988	4.999	4.0	21.226
0.075	0.040	0.968	0.961	4.999	4.0	21.582
0.075	0.045	0.997	0.993	4.999	4.0	21.373
0.075	0.050	0.997	0.992	4.999	4.0	21.335
0.075	0.055	0.991	0.991	4.999	4.0	21.049
0.075	0.060	0.987	0.984	4.999	4.0	21.393
0.075	0.065	0.985	0.977	4.999	4.0	21.710
0.075	0.070	0.999	0.997	4.999	4.0	21.172
0.075	0.075	0.996	0.991	4.999	4.0	21.318
0.075	0.080	0.993	0.984	4.999	4.0	21.632
0.075	0.085	0.985	0.978	4.999	4.0	21.399
0.075	0.090	0.993	0.983	4.999	4.0	21.544
0.075	0.095	0.999	0.997	4.999	4.0	21.199
0.075	0.100	0.997	0.993	4.999	4.0	21.326
0.080	0.005	0.769	0.765	5.0	4.082	22.227
0.080	0.010	0.868	0.867	5.0	4.003	21.349
0.080	0.015	0.936	0.931	5.0	4.0	21.621
0.080	0.020	0.943	0.934	4.999	4.0	21.985
0.080	0.025	0.876	0.870	4.999	4.0	21.747
0.080	0.030	0.904	0.901	4.998	4.0	21.458
0.080	0.035	0.978	0.966	4.999	4.0	21.928
0.080	0.040	0.960	0.948	4.999	4.0	22.016
0.080	0.045	0.979	0.975	4.999	4.0	21.397
0.080	0.050	0.980	0.978	4.999	4.0	21.252
0.080	0.055	0.997	0.993	4.999	4.0	21.299
0.080	0.060	0.987	0.985	4.999	4.0	21.285
0.080	0.065	0.998	0.996	4.999	4.0	21.214
0.080	0.070	0.988	0.985	4.999	4.0	21.350

δ_{inc_conf}	δ_{dec_conf}	Comm. succ.	Align. succ.	Chall. mast.	Gram. cxns.	Lex. cxns.
0.080	0.075	0.985	0.981	4.999	4.0	21.398
0.080	0.080	0.989	0.986	4.999	4.0	21.241
0.080	0.085	0.998	0.995	4.999	4.0	21.272
0.080	0.090	0.998	0.995	4.999	4.0	21.216
0.080	0.095	0.980	0.977	4.998	4.0	21.215
0.080	0.100	0.995	0.986	4.999	4.0	21.614
0.085	0.005	0.857	0.852	5.0	4.018	21.656
0.085	0.010	0.914	0.906	5.0	4.0	21.829
0.085	0.015	0.864	0.860	4.999	4.030	21.757
0.085	0.020	0.923	0.917	4.999	4.045	21.772
0.085	0.025	0.966	0.960	4.999	4.0	21.559
0.085	0.030	0.982	0.968	4.999	4.0	21.742
0.085	0.035	0.978	0.971	4.999	4.0	21.601
0.085	0.040	0.958	0.955	4.999	4.0	21.597
0.085	0.045	0.987	0.981	4.999	4.0	21.396
0.085	0.050	0.963	0.957	4.982	4.0	21.669
0.085	0.055	0.989	0.988	4.999	4.0	21.237
0.085	0.060	0.989	0.986	4.999	4.0	21.263
0.085	0.065	0.976	0.970	4.999	4.0	21.549
0.085	0.070	0.972	0.961	4.998	4.0	21.763
0.085	0.075	1.0	1.0	5.0	4.0	21.017
0.085	0.080	0.986	0.979	4.999	4.0	21.323
0.085	0.085	0.998	0.995	4.999	4.0	21.255
0.085	0.090	0.997	0.992	4.999	4.0	21.327
0.085	0.095	0.995	0.987	4.999	4.0	21.329
0.085	0.100	0.999	0.997	4.999	4.0	21.163
0.090	0.005	0.881	0.871	5.0	4.0	22.097
0.090	0.010	0.876	0.870	5.0	4.042	21.498
0.090	0.015	0.851	0.848	5.0	4.0	21.897
0.090	0.020	0.915	0.915	4.999	4.0	21.25
0.090	0.025	0.913	0.908	4.999	4.0	21.628
0.090	0.030	0.985	0.973	4.999	4.0	21.900
0.090	0.035	0.855	0.852	4.996	4.0	21.529
0.090	0.040	0.986	0.979	4.999	4.0	21.601
0.090	0.045	0.981	0.981	4.99	4.0	21.121
0.090	0.050	0.966	0.958	4.998	4.0	21.892
0.090	0.055	0.979	0.975	4.999	4.0	21.438
0.090	0.060	0.995	0.986	4.999	4.0	21.553
0.090	0.065	0.986	0.980	4.999	4.0	21.456
0.090	0.070	0.967	0.964	4.978	4.0	21.356
0.090	0.075	0.987	0.982	4.998	4.0	21.456
0.090	0.080	0.999	0.995	4.999	4.0	21.220
0.090	0.085	0.977	0.974	4.998	4.0	21.314
0.090	0.090	0.997	0.991	4.999	4.0	21.322
0.090	0.095	0.992	0.979	4.999	4.0	21.805
0.090	0.100	0.999	0.998	4.999	4.0	21.165
0.095	0.005	0.766	0.762	5.0	4.002	22.172
0.095	0.010	0.918	0.912	5.0	4.039	21.730
0.095	0.015	0.902	0.896	4.999	4.044	21.934
0.095	0.020	0.931	0.929	4.999	4.0	21.290
0.095	0.025	0.930	0.925	4.899	4.007	21.477
0.095	0.030	0.986	0.982	4.999	4.0	21.361
0.095	0.035	0.929	0.921	4.999	4.0	21.802
0.095	0.040	0.986	0.979	4.999	4.0	21.451
0.095	0.045	0.958	0.947	4.999	4.0	21.710
0.095	0.050	0.980	0.979	4.999	4.0	21.307

δ_{inc_conf}	δ_{dec_conf}	Comm. succ.	Align. succ.	Chall. mast.	Gram. cxns.	Lex. cxns.
0.095	0.055	0.998	0.995	4.999	4.0	21.328
0.095	0.060	0.982	0.973	4.999	4.0	21.715
0.095	0.065	0.983	0.973	4.999	4.0	21.700
0.095	0.070	0.957	0.951	4.998	4.0	21.493
0.095	0.075	0.989	0.985	4.998	4.0	21.363
0.095	0.080	0.954	0.947	4.997	4.0	21.458
0.095	0.085	0.995	0.990	4.999	4.0	21.576
0.095	0.090	0.997	0.992	4.999	4.0	21.292
0.095	0.095	0.998	0.994	4.999	4.0	21.270
0.095	0.100	0.997	0.992	4.999	4.0	21.282
0.100	0.005	0.783	0.776	5.0	4.058	22.247
0.100	0.010	0.875	0.865	5.0	4.026	21.871
0.100	0.015	0.947	0.944	5.0	4.0	21.446
0.100	0.020	0.882	0.876	4.975	4.021	21.610
0.100	0.025	0.946	0.933	4.999	4.0	21.658
0.100	0.030	0.916	0.911	4.992	4.017	21.490
0.100	0.035	0.916	0.906	4.995	4.002	21.626
0.100	0.040	0.939	0.932	4.998	4.0	21.562
0.100	0.045	0.989	0.987	4.999	4.0	21.188
0.100	0.050	0.960	0.958	4.999	4.0	21.251
0.100	0.055	0.985	0.977	4.999	4.0	21.497
0.100	0.060	0.982	0.982	4.999	4.0	21.048
0.100	0.065	0.979	0.976	4.999	4.0	21.351
0.100	0.070	0.986	0.981	4.999	4.0	21.459
0.100	0.075	0.977	0.970	4.998	4.0	21.608
0.100	0.080	0.972	0.969	4.997	4.0	21.347
0.100	0.085	0.999	0.999	4.999	4.0	21.054
0.100	0.090	0.959	0.947	4.904	4.005	21.727
0.100	0.095	1.0	1.0	4.998	4.0	21.021
0.100	0.100	0.984	0.976	4.998	4.0	21.620

Table B.1: Experimental results of 400 different configurations of δ_{inc_conf} and δ_{dec_conf} values for the experimental set-up in Section 5.3. The table shows the resulting values for communicative success, alignment success, the average cumulative confidence in challenges and the average number of grammatical and lexical constructions.

Résumé long de la thèse en français

Résumé long de la thèse en français

Remarque liminaire

Ce document constitue une présentation en français des travaux contenus dans la thèse. Cette présentation n'est pas exhaustive : elle est au contraire limitée à certains résultats qui nous semblent les plus intéressants et les plus caractéristiques parmi ceux obtenus. Le manuscrit en anglais est évidemment plus complet et rapporte l'essentiel des expériences effectuées pendant la thèse dans les domaines discrets et continus, ainsi qu'un état de l'art des systèmes de motivation computationnels, une description des outils utilisés dans les expériences et une réflexion sur les possibles continuations des travaux présentés ici. Nous espérons cependant que cette présentation rapide permettra au lecteur francophone qui ne pourrait lire la version anglaise d'avoir une vision synthétique et précise des travaux effectués.

Introduction

Cette thèse étudie le rôle de la motivation intrinsèque dans l'émergence et le développement des systèmes de communication dans des populations d'agents artificiels. Plus précisément, nous avons étudié comment les populations d'agents peuvent utiliser un système de motivation pour réguler le développement d'un langage partagé qui devient de plus en plus élaboré. Le cœur de cette recherche consiste à relier deux domaines d'études différents : la linguistique évolutive et les approches computationnelles de modélisation de la motivation.

La linguistique évolutive essaye d'expliquer l'évolution du langage en tant que processus de négociation culturelle. Dans ce domaine, les chercheurs étudient l'émergence et le changement de systèmes de communication similaires à ceux qu'on observe dans les langues humaines, sur la base d'interactions récurrentes entre pairs, dans une population d'agents artificiels, dans un environnement contrôlé et délimité [116, 129]. Les différents systèmes de communication présentés dans cette thèse suivent le paradigme des jeux du langage ("language games" en anglais) [125, 128], notion introduite par [160]. Cette approche postule que le langage est un système adaptatif complexe.

Avec cette approche, les chercheurs en linguistique évolutive ont examiné

différents phénomènes linguistiques. Ils ont d'abord appliqué cette méthode pour étudier l'auto-organisation du vocabulaire [125, 126, 127], mais pendant les vingt dernières années elle a aussi été utilisée dans une variété de domaines tels que la couleur [15, 24], l'espace [124], la notion de quantité [92], le changement des sens associés aux mots [157, 156], les systèmes casuels [154], l'accord grammatical [19, 94], l'aspect [48] ou la syntaxe [37, 46]. Bien que cette approche ait aidé à éclairer ces domaines, le contrôle de la complexité des systèmes de communication qui en résultent est la plupart du temps aux mains de l'expérimentateur, qui normalement explore un certain domaine en utilisant plusieurs configurations expérimentales.

La question suivant se pose alors : comment les agents peuvent-ils gérer la complexité des expériences eux-mêmes. Une solution possible est proposée par les chercheurs en intelligence artificielle (IA) et en robotique développementale, qui, inspirés par des études de psychologie sur la notion de motivation, ont proposé différents modèles qui permettent aux agents de gérer de façon autonome la complexité de leurs actions. L'idée de base consiste à pourvoir les agents de mécanismes qui évaluent le résultat d'une action et utilisent cette information pour décider la prochaine action à exécuter. Ces mécanismes permettent aux agents d'évaluer la précision d'une action, par exemple en comparant le résultat attendu et le résultat réel, et d'utiliser ce renseignement pour prédire l'effet des actions, décider des prochaines actions à réaliser, explorer l'espace des différentes actions possibles ou améliorer leur performance sur une action spécifique.

Ce travail contribue à la compréhension du rôle de la motivation intrinsèque dans l'émergence et le développement du langage de deux façons. Tout d'abord, il présente une description détaillée d'un modèle computationnel de motivation appelé *Autotelic principle* (principe autotélique). Deuxièmement, il teste son rôle dans le développement du langage en appliquant ce modèle de motivation dans des expériences de linguistique évolutive appliquées à différents domaines linguistiques. La thèse soutient que la complexité du langage résultant peut être gérée par les agents artificiels qui y participent et montre que le principe autotélique peut être utilisé comme un mécanisme général dans ce genre d'expériences.

Jeux de langage (Language Games)

L'approche adoptée dans ce travail pour étudier l'émergence et le développement du langage au sein d'une population d'agents artificiels consiste à procéder au moyen de simulations computationnelles d'interactions linguistiques. L'idée de base est (a) d'implémenter un ensemble de fonctions pour créer, apprendre et aligner des mots et des concepts et (b) de fournir aux agents artificiels ces fonctions afin de simuler l'émergence de systèmes de communication [137].

Les jeux de langage sont des interactions récurrentes et simplifiées entre deux agents d'une population qui ont un but communicationnel [137, p. 71] :

A language game is embedded in a cooperative activity in which communication is useful. It attempts to model situated dialogue in contrast to the isolated sentences that are commonly used today in formal

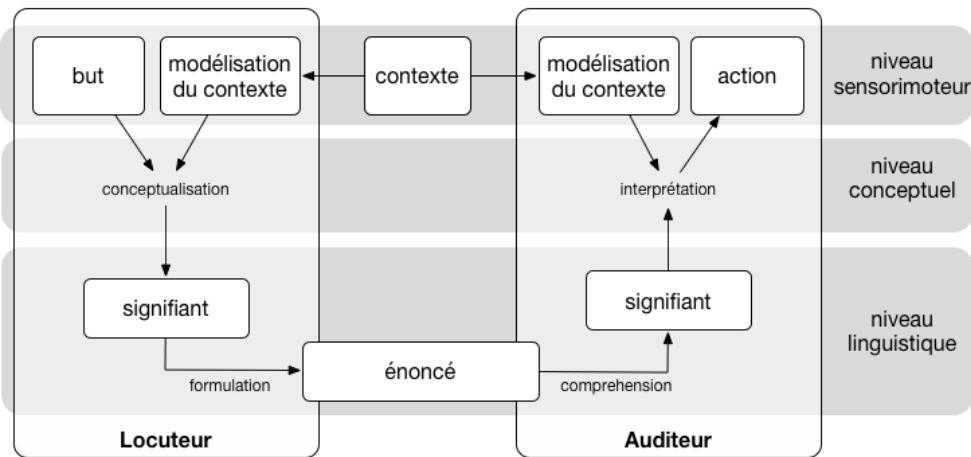


Figure B.1: Le cycle sémiotique représente une interaction entre un locuteur et un auditeur dans un contexte (figure adaptée de [156]). Cette représentation implique trois niveaux de processus différents : sensorimoteur, conceptuel et linguistique.

linguistics. Consequently, language games introduce a population of individuals (instead of an idealized speaker), a context and a communicative purpose, so that pragmatics is part of the modelling effort from the start.²

La manière dont les interactions entre un locuteur et un auditeur dans un contexte sont modélisées est définie par le *cycle sémiotique* (figure B.1). Cette représentation implique trois niveaux de processus différents : (a) le niveau *sensorimoteur*, car les deux agents perçoivent le monde et interagissent avec lui en utilisant leur système sensorimoteur, (b) le niveau *conceptuel*, responsable de la correspondance entre les éléments dans le contexte et leurs représentations conceptuelles et (c) le niveau *linguistique*, en charge de faire la liaison entre les représentations conceptuelles et les phrases que les agents peuvent formuler, et vice versa.

Il y a différentes implémentations des jeux de langage, qui vont des *naming games* aux *syntax games*. Dans toutes les expériences présentées dans cette thèse, les agents prennent part à des *guessing games* [38], un jeu de langage particulier dans lequel il y a compétition au niveau du signifiant (un mot peut signifier plusieurs choses) et au niveau de la forme (plusieurs mots peuvent être utilisés pour transmettre un sens).

Dans tous les jeux de langage, cependant, chaque interaction suit un même schéma : au début, l'auditeur et le locuteur construisent leurs modèles du con-

²Un *language game* fait partie d'une activité coopérative dans laquelle la communication est utile. Il essaye de modéliser une situation de dialogue, par opposition aux phrases isolées qui sont couramment utilisées aujourd'hui en linguistique formelle. Par conséquent, les jeux de langage introduisent une population d'individus (au lieu d'un locuteur idéalisé), un contexte et un but communicationnel, de sorte que la pragmatique a été partie prenante de l'effort de modélisation depuis le début.

texte en utilisant leurs systèmes sensorimoteurs. Cette information est utilisée par le locuteur pour *conceptualiser* son but communicationnel dans une représentation conceptuelle du sens à transmettre. Cette représentation est utilisée pour *formuler* un énoncé. L'auditeur *comprend* l'énoncé pour reconstituer la représentation conceptuelle que le locuteur a essayé de transmettre. Il utilise ensuite le modèle du contexte pour *interpréter* le sens de la représentation conceptuelle dans cet environnement et exécute une action. Si l'action de l'auditeur correspond au but du locuteur, l'interaction est considérée comme un succès. Dans tous les autres cas, l'interaction n'est pas satisfaisante et, donc, elle est considérée comme un échec. Les agents sont équipés avec différents *mécanismes d'apprentissage* qu'ils utilisent à la fin des interactions pour aligner leur vocabulaire.

Implémentation du cycle sémiotique

Les différents *guessing games* qui font partie de cette thèse ont été implémentés avec Babel [77, 143], une plateforme d'expérimentation open-source qui permet de concevoir et d'implémenter l'ensemble du cycle sémiotique dans des expériences informatiques d'émergence du langage. Parmi les multiples systèmes computationnels de Babel il y en a deux qui sont particulièrement importants pour cette thèse : *Incremental Recruitment Language* (IRL), qui est utilisé pour modéliser le niveau conceptuel, et *Fluid Construction Grammar* (FCG), qui sert à modéliser le niveau linguistique.

Incremental Recruitment Language

IRL [141, 152, 122, 123] est un système qui permet aux agents (a) de créer des représentations sémantiques discriminantes des objets dans leur contexte ou (b) d'identifier des objets dans leur scène en utilisant une représentation sémantique particulière. Ce système représente le sens comme des réseaux de contraintes sémantiques appelés *IRL-networks*. Ces réseaux sont construits en combinant deux types différents de prédicats. Il existe d'une part des méthodes qui représentent une certaine activité cognitive que l'agent doit exécuter (par exemple catégoriser, filtrer ou sélectionner des objets dans le contexte). Ces méthodes s'appellent *opérations cognitives*. D'autre part il y a des *entités sémantiques*, c'est-à-dire des prédicats qui font référence à une entité particulière (par exemple une caractéristique distinctive des objets ou spécifique d'un objet donné).

Les opérations cognitives sont représentées comme des prédicats munis d'un ensemble d'arguments. Ils sont identifiés par leur nom (par exemple, **filter-set**) et introduisent un ensemble de variables (éléments commençant avec ?). Les entités sémantiques, par contre, n'introduisent qu'une seule variable qui est utilisée comme argument par les opérations cognitives. Ce deuxième type de prédicats est introduit par un opérateur spécial appelé **bind**. Un réseau de contraintes sémantiques est donc constitué d'un ensemble de prédicats qui sont liés entre eux en utilisant des variables.

Fluid Construction Grammar

Le niveau linguistique est modélisé avec FCG [134, 135, 139], un formalisme de grammaire de construction opérationnel qui peut être utilisé soit (a) pour construire et développer une grammaire ou (b) dans des expériences d'évolution du langage. L'une des caractéristiques les plus importantes de FCG est que ces grammaires sont *bidirectionnelles* : la même grammaire peut être utilisée pour *formuler* un énoncé à partir d'une représentation sémantique ou pour *comprendre* un énoncé afin de modéliser le sens qu'il contient.

Le processus de formulation ou de compréhension d'un énoncé est considéré comme une chaîne d'opérations consécutives sur une *structure linguistique* [139, p. 5] appelée *transient structure* (structure transitoire). Les structures transitaires (dite *TS*), comme toute autre structure linguistique dans FCG, sont représentées comme un ensemble d'unités formées de paires de caractéristiques et de valeurs.

En FCG, les *unités* sont des groupes abstraits d'information linguistique qui correspondent habituellement aux mots, aux morphèmes ou à d'autres éléments linguistiques. Elles contiennent un *nom*, qui doit être unique, et une *structure de caractéristiques*. Ces structures sont des représentations abstraites de l'information linguistique. Elles sont formées de paires {caractéristique / valeur}.

Les opérations qui peuvent être exécutées sur une *TS* s'appellent *constructions*. Les constructions sont des structures linguistiques ayant la fonction d'élargir la structure transitoire en y ajoutant une information, qui peut être sémantique, syntaxique, pragmatique ou phonologique. L'information ajoutée à une *TS* dépend de la construction et peut juste lier les variables dans la structure transitoire ou introduire de nouvelles unités dans la *TS*. Le résultat de l'application d'une opération sur une *TS* est une nouvelle structure transitoire *TS'* qui contient l'information de *TS* et l'information ajoutée par la construction.

Les constructions sont divisées en deux parties : une partie *contributive* et une partie *conditionnelle*. La partie contributive contient l'information (sous forme d'un ensemble d'unités) qui sera ajoutée à la *TS*. La partie conditionnelle contient les informations qui doivent se trouver dans la structure transitoire pour que la construction puisse être appliquée. Les deux parties (conditionnelle et contributive) peuvent contenir une ou plusieurs unités. Comme une construction est bidirectionnelle, la partie conditionnelle est divisée en deux *locks* (serrures), nommés *formulation lock* et *comprehension lock*. Chaque lock présente l'information qui doit se trouver dans la *TS* quand la construction est utilisée pour formuler ou comprendre un énoncé. La figure B.2 illustre les différentes parties d'une construction.

Les structures transitaires ont une unité spéciale, appelée *root*, qui est utilisée pour sauvegarder l'information d'entrée nécessaire pour commencer l'exécution d'opérations sur une *TS*. En formulation, cette unité contient le réseau de contraintes sémantiques à partir duquel on veut construire un énoncé (ce réseau correspond au résultat du processus de conceptualisation dans le cycle sémiotique). En compréhension, par contre, cette unité encode l'information syntaxique sur l'énoncé à analyser (normalement les mots et leur ordonnance-

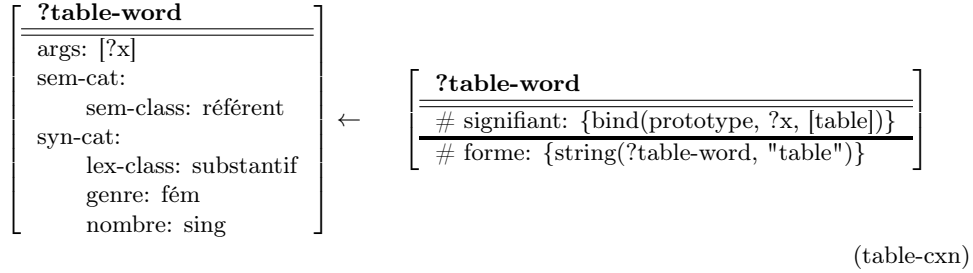


Figure B.2: Une construction possible pour "table". Dans FCG, les constructions sont divisées en une partie *contributive* (côté gauche) et une partie *conditionnelle* (côté droit). La première contient l'information linguistique qui doit être ajoutée à la *TS* et la seconde l'information qui devrait se trouver dans la structure transitoire avant l'application de la construction. La partie conditionnelle est ensuite divisée en deux *locks*, séparés par une ligne noire. Chaque *lock* contient l'information qui doit se trouver dans la *TS* quand la construction est utilisée pour formuler ou comprendre un énoncé.

ment dans l'énoncé) afin d'identifier le sens qu'il véhicule.

Cette première partie du résumé a présenté la notion de jeu de langage, approche fondée sur la notion de cycle sémiotique pour modéliser les interactions linguistiques. Nous avons ensuite introduit la notion de *guessing game*, le jeu de langage employé dans les différentes expériences présentées dans cette thèse. On a conclu avec une explication en quelques mots des deux systèmes computationnels clés dans l'implémentation de ce cycle : *Incremental Recruitment Language* et *Fluid Construction Grammar*. La section suivante définit le système de motivation computationnel étudié, l'*autotelic principle*.

Autotelic principle

Ce système de motivation a été proposé par Steels [131, 133] comme une version opérationnelle de la théorie du *flow* de Csíkszentmihály [35, 36]. Dans cette thèse le système est utilisé par des agents artificiels pour gérer la complexité de leurs actions et réguler de façon autonome leur développement.

Csíkszentmihály voulait comprendre ce qui motive les gens à être absorbés dans des activités complexes qui ne comportent pas de récompense extérieure, comme par exemple l'escalade, la peinture ou la sculpture. Sa conclusion est que ces activités sont intrinsèquement agréables pour ceux qui les pratiquent. Il a appelé ces activités *autotéliques*, parce que la source de la force motivationnelle (*telos*) provient de l'individu lui-même (*auto*).

À partir de ces observations, Csíkszentmihály a élaboré la théorie du flow [36]. Selon lui, les activités autotéliques peuvent être décrites en considérant la relation entre deux dimensions. D'une part le défi (*challenge*), une certaine tâche à accomplir. D'autre part la compétence (*skill*), les capacités qu'une personne a pour aborder cette tâche. Cette relation explique l'éventail d'états mentaux que

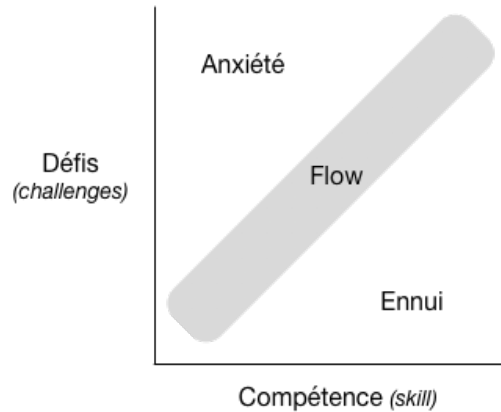


Figure B.3: Représentation visuelle de la théorie du flow de Csíkszentmihályi (adaptée de [35]). Une personne entre dans un état de *flow* quand les challenges sont en équilibre avec ses compétences. Lorsque le défi est trop exigeant pour ses capacités, elle ressent de l'*anxiété*. Par contre, elle expérimente l'*ennui* quand ses compétences sont plus grandes que le défi actuel.

les personnes ressentent lorsqu'elles sont impliquées dans une activité autotélique (figure B.3) : l'*ennui*, lorsque les compétences d'un individu sont trop élevées pour son défi actuel, l'*anxiété*, lorsque le défi est trop difficile par rapport aux compétences de la personne et, finalement, le *flow*, quand il y a un équilibre entre les valeurs des deux dimensions.

Csíkszentmihályi a estimé que la dernière situation correspond à l'état optimal d'expérience, car c'est l'état qui fournit le meilleur scénario pour développer davantage les compétences des participants d'activités autotéliques. L'état de *flow* est en mouvement continu car les compétences des individus évoluent avec le temps. Les participants cherchent à rester dans l'état de *flow*, restant ainsi motivés.

L'*autotelic principle* est une version opérationnelle de cette théorie [131, 133]. Comme dans la théorie psychologique sur laquelle il se fonde, le noyau de ce système de motivation repose sur la relation entre les défis et les compétences. Les agents peuvent déterminer eux-mêmes leurs propres objectifs (appelés *challenges*) et évaluer indirectement leurs capacités (*skills*) en mesurant leurs performances pour ces défis. Cette évaluation est basée sur leur rendement dans des actions successives lors d'un défi donné.

Après chaque action faite pour un défi particulier les agents reçoivent une récompense. Dans la proposition originale [131], Steels calcule la récompense en considérant seulement le résultat de l'action, mais dans l'implémentation proposée dans cette thèse, la récompense est établie en tenant compte aussi de l'évaluation interne des capacités de l'agent. Cette récompense est employée pour actualiser l'évaluation du défi tenté, ce qui permet aux agents de décider quand il faut changer d'objectif afin de développer davantage leurs compétences.

Définition des notions de défi et de performance

Dans le système de motivation défini dans notre thèse, les défis correspondent à des configurations spécifiques d'un ensemble de paramètres dans un espace de paramètres. Formellement, étant donné un espace de paramètres multidimensionnel P , un défi p_i est défini comme un vecteur $\langle p_{i,1}, p_{i,2}, \dots, p_{i,n} \rangle$, où $p_{i,j}$ correspond à la valeur du paramètre j dans le défi i . Les agents sont capables de générer différents défis en changeant la configuration spécifique d'un challenge p_i . Prenant p_i , les agents peuvent créer un ensemble de nouveaux défis $\{p_i^1, p_i^2, \dots, p_i^n\}$ en modifiant un ou plusieurs paramètres $p_{i,j}$ de ce défi (p_i). L'espace des défis possibles dépend (a) du nombre de paramètres utilisés pour définir un défi et (b) des différentes valeurs potentielles de chaque paramètre.

De plus, chaque défi est associé à une valeur du *niveau* (*level* en anglais) l_i , qui est un nombre entier dans une gamme de valeurs $[1, l_{max}]$ et représente la difficulté associée à ce défi. Dans l'implémentation utilisée dans cette thèse, ce nombre correspond à la somme des valeurs des paramètres. La relation entre un défi et son niveau peut être formalisée comme suit :

$$l_i = \sum_{j=1}^n p_{i,j}$$

En regard des défis, la *compétence* ne peut pas être mesurée directement, car on ne peut pas estimer la compétence d'un agent qui exécute une action avant qu'elle ne soit exécutée. Le système peut, par contre, évaluer le rendement d'un agent. La *performance* est donc la mesure indirecte de la compétence, car le système peut utiliser l'information obtenue à la suite d'une action particulière pour déterminer si le but a été atteint ou non.

On peut donc reformuler la théorie du flow en termes de défis et de performances. La relation entre ces deux termes permet aux agents d'identifier une gamme d'états différents, c'est-à-dire des situations internes équivalentes aux états mentaux identifiés par Csíkszentmihály. Un agent identifie qu'il se trouve dans un état de *flow* quand sa performance n'est ni trop élevée ni trop faible pour son défi actuel. Les agents se trouvent dans un état d'*ennui* lorsque leur performance est constamment trop élevée et passent par un état d'*anxiété* lorsque leur performance est trop faible pour leur tâche actuelle.

Développement des compétences

Un élément central de la théorie est qu'un état de flow facilite le développement et l'apprentissage des compétences. De la même façon, un agent artificiel doit être capable d'apprendre ou développer des nouvelles capacités.

Les méthodes d'apprentissage qui ont été mises en place dans cette thèse sont une collection de *diagnostics*, des mécanismes utilisés pour identifier les problèmes pendant et après qu'une action a eu lieu, et une collection de *réparations*, des processus qui essaient de résoudre les problèmes diagnostiqués. Comme le système de motivation est utilisé dans des expériences d'évolution du langage, les

diagnostics et réparations que les agents ont à leur disposition dans les simulations leur permettent de créer ou apprendre des mots, des prototypes différents (par exemple des couleurs) ou des constructions grammaticales.

Phases du système

Après qu'une action a été exécutée, le système identifie l'état associé au défi tenté (qui peut être *flow*, *anxiété* ou *ennui*). Cette information est ensuite utilisée pour déterminer la *phase* ou situation actuelle du système. En fonction de sa phase, le système décidera de (a) continuer à tenter le défi actuel ou (b) d'aborder un autre défi.

La phase dans laquelle le système choisit de continuer avec le même défi s'appelle phase *opérationnelle* et se manifeste quand le système est dans un état de flow pour son défi actuel. Dans cette phase, le système exécute les actions et utilise le feedback, l'information obtenue après l'exécution d'une action, pour améliorer sa performance.

S'il arrive à obtenir une haute performance pour son défi actuel, le système finira par tomber dans un état d'ennui. Par contre, si le défi se révèle être trop difficile le système n'arrive pas à développer les compétences nécessaires, ce qui peut conduire le système à un état d'anxiété. Dans les deux états le système entre dans une phase de *réorganisation*.

Dans cette phase le système essaye de rétablir l'équilibre entre défis et compétences en changeant le défi actuel. Quand le système se trouve dans un état d'anxiété pour un challenge p_i avec un niveau associé l_i , il explore d'abord s'il peut tenter un défi déjà connu pour lequel il a obtenu une haute performance. Sinon, il générera un ensemble de choix possibles $\{p_i^1, \dots, p_i^n\}$ en modifiant la configuration des paramètres de p_i . Dans les deux cas, le défi que l'agent tentera de relever dans les actions futures présente un niveau associé l_k , où $l_k < l_i$.

Par contre, si le système se trouve dans un état d'ennui il cherchera une tâche plus difficile. D'abord il examinera les défis déjà connus et sélectionnera un défi qui présente une difficulté plus élevée que le défi actuel pour lequel il a obtenu une mauvaise performance dans le passé. S'il n'en trouve aucun, le système générera un ensemble de nouveaux défis en ajustant la configuration des paramètres de p_i puis en sélectionnera un comme nouveau défi à tenter. Dans les deux cas, le nouveau défi choisi aura un niveau associé supérieur au niveau du défi antérieur.

La combinaison de ces deux phases permet au système d'autoréguler son propre développement, car elles permettent au système de décider quand il doit continuer avec un défi afin de développer ses compétences ou changer le défi lorsqu'il y a un déséquilibre constant entre le défi et la performance.

Génération des nouveaux défis

Le système pourrait potentiellement générer tous les défis possibles (c'est-à-dire toutes les configurations de paramètres différentes) à chaque fois qu'il est en phase de réorganisation. On examine ici comment le système (a) limite l'ensemble des nouveaux défis qu'il peut créer et (b) sélectionne le nouveau challenge à tenter.

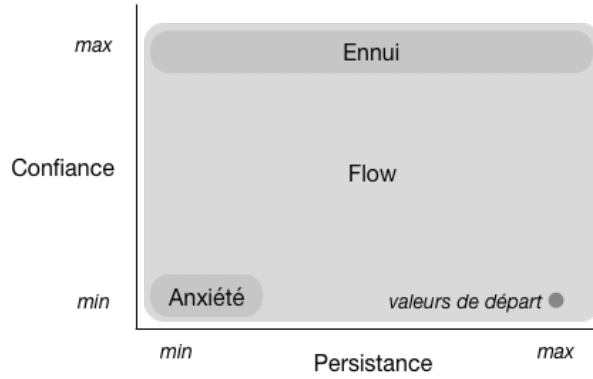


Figure B.4: Représentation visuelle de la relation entre les valeurs de confiance et de persistance d'un défi et sa relation aux différents états internes. Les défis sont initialisés avec une valeur maximale de *persistance* et une valeur minimale de *confiance*. Le système atteint un état d'*anxiété* quand les deux valeurs sont au minimum, et un état d'*ennui* quand la valeur de confiance a atteint son maximum. Dans ces deux cas le système entre dans la phase de réorganisation afin de modifier le défi à tenter. Si aucune de ces conditions n'est remplie, le système reste en phase opérationnelle, qui correspond à l'état de *flow*.

Dans l'implémentation proposée dans notre thèse, le nombre de nouveaux défis créés dans une phase de réorganisation est limité en fonction du niveau associé. L'ensemble de nouveaux défis ne sera formé que par des configurations de paramètres où (a) la valeur de niveau associée à un nouveau défi est une unité inférieure ou supérieure au défi source, selon l'état du système (anxiété ou ennui, respectivement), et (b) les nouveaux défis sont créés en ne changeant qu'un seul paramètre du défi de référence, augmentant ou diminuant la valeur du paramètre d'une unité. Formellement, tous les nouveaux défis générés doivent remplir deux conditions :

- $|l_k - l_i| = 1$, où l_i correspond au niveau du défi actuel et l_k au niveau du challenge candidat.
- $\sum_{j=1}^n |p_{k,j} - p_{i,j}| = 1$, où $p_{i,j}$ et $p_{k,j}$ correspondent à la valeur du paramètre j dans p_i et p_k , respectivement.

Les défis qui satisfont ces restrictions sont ajoutés dans la liste des défis connus par le système. Cette liste est utilisée pour déterminer le défi actuel.

Actualisation de l'état interne

Le système surveille l'évolution d'un défi et utilise l'information qu'il recueille après chaque action pour évaluer son rendement et, finalement, déterminer s'il a atteint ou non le but visé. Ce processus est fait en assignant à chaque défi deux valeurs qui sont actualisées après chaque action que le défi est tenté et oscillent sur une plage de valeurs $[min, max]$:

- **Confiance** : c'est une valeur liée à la performance de l'agent qui représente la certitude qu'il a d'être compétent dans une tâche spécifique. Plus la valeur est élevée, plus le rendement moyen pour le défi est élevé.
- **Persistance** : c'est une mesure pour assurer un nombre minimum de tentatives avant que le système puisse décider de changer sa tâche actuelle. Cette mesure répond au fait que normalement il faut un certain temps avant que le système développe les compétences nécessaires pour une nouvelle tâche. Grâce à cette mesure, le système est empêché d'entrer en continu dans la phase de réorganisation.

Ces deux paramètres sont utilisés pour déterminer l'*état interne* d'un défi (figure B.4). La première fois qu'un défi est essayé, il est initialisé avec une valeur minimale de confiance, parce que le système n'a aucune expérience pour cette tâche particulière, et une valeur maximale de persistance, pour prévenir le système d'entrer dans la phase de réorganisation dès les premières actions.

Les deux valeurs sont actualisées chaque fois que le système exécute une action pour un défi donné, comme suit :

Algorithme 5 Actualisation du défi

```

procedure ACTUALISATION DU DÉFI(agent, chali, succ?)
  confi ← CONFIANCE(chali)
  persi ← PERSISTANCE(chali)
  if succ? then
    confi ← confi +  $\delta_{inc\_conf}$ 
  else
    confi ← confi -  $\delta_{dec\_conf}$  + ÉVALUATION INTERNE(agent, chali, succ?)
    if confi = minconf then
      persi ← persi -  $\delta_{dec\_pers}$ 
    end if
  end if
end procedure

```

D'une part, la persistance est diminuée par la valeur δ_{dec_pers} lorsque (a) le résultat de l'interaction est un échec et (b) la valeur de confiance est dans son minimum. Dans les autres cas cette valeur reste la même. Lorsque la persistance atteint son minimum, elle déclenche l'état d'anxiété et le système entre dans la phase de réorganisation. Lorsque cela se produit, la valeur de persistance est mise à $-max$. Une valeur de persistance négative bloque le défi pour le système pendant une certaine période de temps jusqu'à la persistance ait à nouveau une valeur positive. Tant qu'elle a une valeur négative, la persistance est actualisée à la fin de chaque action avec $pers_i(t) = pers_i(t - 1) + \delta_{dec_pers}$, où $pers_i(t)$ et $pers_i(t - 1)$ sont les valeurs de persistance actuelle et précédente associés au défi p_i . Une fois que la persistance atteint la valeur minimale, le défi redevient disponible et la persistance est réinitialisée à sa valeur maximale.

Par contre, la valeur de confiance est mise à jour après chaque action. Pour actualiser cette valeur le système tient compte à la fois (a) du résultat d'une

action et (b) d'une évaluation interne des compétences individuelles de l'agent qui a réalisé l'action. La confiance est actualisée différemment selon le résultat de l'interaction linguistique : lorsque l'interaction est un succès (le sens transmis par le locuteur sous la forme d'une séquence de mots a été correctement interprété par l'auditeur) la valeur de confiance est mise à jour comme suit : $conf_i(t) = conf_i(t-1) + \delta_{inc_conf}$, où $conf_i(t)$ et $conf_i(t-1)$ sont les valeurs de confiance actuelle et précédente. Quand l'interaction est un échec, la valeur de confiance est mise à jour de cette façon : $conf_i(t) = conf_i(t-1) - \delta_{dec_conf} + ind_{comp}$, où ind_{comp} correspond à l'évaluation interne des compétences individuelles par l'agent et a toujours une valeur inférieure à δ_{dec_conf} . Formellement, pour chaque interaction $I(t)$, $|ind_{comp}| < |\delta_{dec_conf}|$.

Initialisation

Le système adopte une approche ascendante. Il est initialisé uniquement avec les défis qui ont associé un niveau l_i le plus bas possible. Cela permet aux agents de développer les compétences fondamentales qui seront nécessaires pour faire face à d'autres défis plus complexes plus tard sans perdre de temps (interactions) pour trouver un bon défi initial. Une fois ces compétences acquises, le système peut passer à des tâches plus difficiles. En conséquence, le système est fondé sur (a) un développement incrémental des capacités de l'agent et (b) une optimisation des interactions nécessaires à l'accomplissement des tâches principales de l'expérience.

Déterminer le défi actuel

Étant donné l'ensemble des défis connus par le système à un moment donné, comment décide-t-il lequel devrait être abordé? Cette décision est prise par un algorithme (algorithme 6) qui prend en compte le niveau et les valeurs de persistance et confiance pour choisir le défi de la prochaine interaction.

L'algorithme limite d'abord la sélection du défi à choisir aux défis associés à un état de flow, c'est-à-dire ceux qui ont une valeur de persistance positive et une valeur de confiance inférieure à son valeur maximale. Après, il classe l'ensemble des défis qui en résultent selon le niveau du défi courant et sélectionne aléatoirement un des défis parmi ceux qui présentent une valeur inférieure. S'il n'y a aucun défi candidat, l'algorithme choisit aléatoirement un des défis parmi l'ensemble des défis connus avec une valeur supérieure.

Un problème se pose lorsque l'*autotelic principle* est appliqué aux jeux de langage : le locuteur choisit son défi actuel au début de l'interaction, ce qui a un impact sur le but communicationnel et aussi sur l'énoncé qu'il produit. Par contre, on ne sait pas comment l'auditeur doit déterminer quel est le défi communicationnel de cette interaction. Ceci est dû au fait que le locuteur et l'auditeur peuvent différer (a) sur les défis qu'ils connaissent et (b) sur leurs défis actuels.

Dans l'implémentation présentée dans notre thèse, l'auditeur essaie de deviner le défi du locuteur sur la base des défis qu'il connaît, de la représentation du sens qu'il a pu récupérer de l'énoncé et du contexte. Les différents paramètres utilisés

Algorithme 6 Détermination du défi actuel

```

procedure DÉTERMINATION DU DÉFI ACTUEL(sys)
  connusdef  $\leftarrow$  DÉFIS(sys)
  potdef  $\leftarrow$   $\emptyset$ 
  canddef  $\leftarrow$   $\emptyset$ 
  for each defi  $\in$  connusdef do
    confi  $\leftarrow$  CONFIANCE(defi)
    persi  $\leftarrow$  PERSISTANCE(defi)
    if conf < maxconfi and pers > minpersi then
      potdef  $\leftarrow$  potdef + defi
    end if
  end for
  if potdef then
    canddef  $\leftarrow$  NIVEAUMIN(potdef)
  else
    canddef  $\leftarrow$  NIVEAUMAX(connusdef)
  end if
  return RANDOM(canddef)
end procedure

```

dans l'espace de paramètres pour définir les défis communicationnels et le niveau associé à un défi sont liés à plusieurs caractéristiques du sens, parmi lesquelles les différentes catégories de signification et leur nombre. Le système utilise la représentation du sens qu'il a pu récupérer de l'énoncé pour identifier les éléments de sens pertinents. Cette information est comparée à la liste des défis connus de l'auditeur afin d'identifier le défi le plus proche de la tâche communicative que le locuteur a probablement choisi comme défi pour cette interaction.

Expériences

Les différentes simulations expérimentales réalisées dans ce travail testent le rôle de la motivation intrinsèque dans l'émergence du langage. Dans ces expériences, une population d'agents artificiels qui participent aux *guessing games* est équipée avec l'*autotelic principle*. Ce système de motivation leur permet de déterminer leurs compétences actuelles pour un ensemble de tâches communicationnelles, c'est-à-dire une estimation de leur performance pour chaque tâche. L'autotelic principle est utilisé pour auto-réguler les buts communicationnels de chaque agent afin d'améliorer leurs compétences, alors que dans le même temps la population converge vers une langue partagée.

Ces expériences ont été divisées en fonction de la nature du contexte dans lequel elles se réalisent. La première série d'expériences concerne le domaine discret, à partir duquel une population d'agents artificiels doit développer un langage pour communiquer à propos de l'ensemble d'objets de la scène. Ces objets ont des propriétés discrètes, c'est-à-dire des valeurs définies pour certaines caractéristiques physiques.

La deuxième série d'expériences concerne le domaine continu de la couleur. Comme dans les simulations précédentes, le but des expériences est de déterminer un langage partagé pour communiquer sur les objets dans leur contexte (dans ce cas, des échantillons de couleurs). Ces expériences prolongent les précédentes dans la mesure où les agents doivent non seulement se mettre d'accord sur les mots associés aux propriétés ou à leur ordre dans l'énoncé, mais aussi sur la valeur associée à ces propriétés.

Expériences dans le domaine discret

La première série d'expériences est inspirée des travaux antérieurs de Steels et Wellens [145]. Ils ont été les premiers à utiliser un système de motivation comparable au nôtre dans une expérience d'évolution du langage. Dans cette section, nous présentons deux expériences : dans la première ou "expérience de base" la population n'utilise pas le mécanisme de motivation interne pour réguler de façon autonome son développement. La deuxième expérience a la même configuration expérimentale que la précédente, avec la différence que cette fois les agents sont pourvus d'un système de motivation intrinsèque.

Dans chaque interaction, le locuteur et l'auditeur ont pour but de communiquer sur un sujet (un ou deux objets) qui se trouvent dans le *contexte* ou *scène*, l'environnement dans lequel se situent les interactions. L'ensemble de toutes les scènes possibles d'une expérience s'appelle un *monde*. Les objets ont trois caractéristiques physiques différentes qui se divisent entre *prototypes* et *propriétés* (forme et couleur) :

- **prototype.** Une valeur discrète qui spécifie la classe de l'objet, à choisir parmi sept valeurs possibles : *table*, *chair*, *cup*, *window*, *lamp*, *drawer* et *box*.
- **forme.** Une valeur discrète qui spécifie la forme de l'objet, à choisir parmi six valeurs possibles : *squared*, *round*, *triangular*, *pentagonal*, *hexagonal* et *octagonal*.
- **couleur.** Une valeur discrète qui spécifie la couleur de l'objet, à choisir parmi huit valeurs possibles : *blue*, *green*, *red*, *yellow*, *orange*, *purple*, *white* et *black*.

Les objets sont uniques, ce qui veut dire que dans le monde il n'y a pas deux paires d'objets qui partagent les mêmes valeurs pour les trois caractéristiques physiques. Les prototypes et les propriétés sont formellement décrits par des prédicats de premier ordre. Par exemple, un objet obj_i avec les valeurs de caractéristiques *table*, *squared* et *green* est représenté de la façon suivante :

$$obj_i = \{prototype(obj_i, table), shape(obj_i, squared), colour(obj_i, green)\}$$

Une scène est composée de deux objets différents et d'une relation spatiale entre eux, également sous la forme d'un prédicat du premier ordre, qui se limite

à trois choix possibles : *next-to*, *far-from* and *left-to*³. Par exemple, une scène possible s_k comprenant l'objet que nous venons de décrire obj_i et obj_j (avec les valeurs *table*, *round* et *blue*) et une relation spatiale de proximité est décrite comme suit :

$$s_k = \{prototype(obj_i, table), shape(obj_i, squared), colour(obj_i, blue), \\ prototype(obj_k, table), shape(obj_k, round), colour(obj_k, green), \\ next - to(obj_i, obj_k)\}$$

Bien que des objets et des scènes puissent être générés aléatoirement au début de chaque simulation, afin de pouvoir mieux analyser et comparer les résultats obtenus, le nombre d'objets et le nombre de contextes qui composent le monde ont été fixés à dix.

Diagnostiques, réparations et alignement

Dans la description de l'*autotelic principle*, il a été mentionné que les agents sont dotés de deux mécanismes, appelés *diagnostics* et *réparations*, qui leur permettent de créer et d'apprendre des constructions lexicales et grammaticales. Les constructions créées sont stockées dans l'*inventaire de constructions* ou *lexicon*. Il est possible que certaines de ces constructions soient en *compétition*. Ce problème se produit lorsque (a) des constructions avec des formes différentes véhiculent le même sens (*compétiteurs au niveau du sens*) ou (b) la même forme est utilisée dans plusieurs constructions pour transmettre un sens différent (*compétiteurs au niveau de la forme*).

Les agents ont un mécanisme pour éviter la compétition entre les constructions qui s'appelle l'*alignement*. Le but de l'alignement est de guider la préférence des agents vers les constructions à utiliser. Chaque construction est dotée d'un *score*, un nombre dans l'intervalle [0.0,1.0]. Le score est initialisé à 0.5, que la construction ait été créée ou apprise. Lorsque le lexique contient des constructions en compétition, ce nombre est utilisé pour décider laquelle doit être utilisée, c'est-à-dire que le système choisit celle avec le score le plus élevé. Après chaque interaction, les agents mettent à jour les scores des constructions utilisées (et de leurs concurrents) en appliquant la dynamique de l'inhibition latérale [155]. Lorsque le score d'une construction atteint son minimum (fixé à 0.0), cette construction est retirée de l'inventaire des constructions de l'agent.

Expérience de base

Pour analyser l'impact du système de motivation intrinsèque sur la régulation de la complexité des interactions linguistiques, il faut d'abord établir une expérience de base. Cette expérience étudie le nombre d'interactions nécessaires pour qu'une population converge vers un langage commun pour différentes tâches de communication.

³La relation *left-to* équivaut à *right-to* en changeant l'ordre des éléments du prédicat.

La population des agents joue un *guessing game* et le jeu commence sans qu'aucune construction ne soit connue. Dans chaque interaction, le locuteur formule un énoncé composé de plusieurs mots qui désignent un ou deux objets de la scène. Cette énoncé doit faire référence, au minimum, au prototype de l'objet (ou objets) sélectionné(s). le locuteur peut également désigner plusieurs caractéristiques des objets à décrire. Les différentes tâches de communication sont définies par le nombre maximum de propriétés auxquelles un agent peut se référer dans la description du but communicationnel :

- **Tâche de communication 1** : les agents peuvent décrire jusqu'à deux objets uniquement en faisant référence à leurs prototypes.
- **Tâche de communication 2** : les agents peuvent décrire jusqu'à deux objets uniquement en faisant référence à leurs prototypes et à une propriété d'un objet décrit.
- **Tâche de communication 3** : les agents peuvent décrire jusqu'à deux objets uniquement en faisant référence à leurs prototypes et à deux propriétés de l'objet ou des objets décrits.
- **Tâche de communication 4** : les agents peuvent décrire jusqu'à deux objets uniquement en faisant référence à leurs prototypes et à trois⁴ propriétés de l'objet ou des objets décrits.
- **Tâche de communication 5** : les agents peuvent décrire jusqu'à deux objets uniquement en faisant référence à leurs prototypes et à quatre propriétés de l'objet ou des objets décrits.

La complexité des tâches de communication est déterminée par le nombre maximal d'éléments physiques que les agents peuvent désigner. Par conséquent, les tâches communicationnelles sont ordonnées en fonction de leur complexité, allant de la plus simple à la plus complexe. Il est important de souligner qu'une tâche plus élevée englobe les tâches précédentes. Par exemple, la tâche de communication 4, où les agents peuvent décrire un ou deux objets en faisant référence à leurs prototypes et jusqu'à trois propriétés, comprend également les tâches de communication 1, 2 et 3, puisque les agents peuvent se référer à moins des propriétés que le maximum autorisé (c'est-à-dire aucune, une ou deux, ce qui correspond aux tâches de communication précédentes).

Chaque tâche a été testée avec une population de dix agents et les résultats ont été calculés sur une moyenne de dix simulations. Le résultat de ces simulations est présenté sur la figure B.5. L'axe des abscisses représente le nombre d'interactions dans la population (c'est-à-dire le nombre de *guessing games* qui ont été joués) et l'axe des ordonnées représente le pourcentage de réussite de la communication, qui peut se situer entre 1 (succès) et 0 (échec).

⁴Un objet a trois caractéristiques physiques différentes (un prototype et deux propriétés). Une description contenant plus de deux propriétés obligatoirement (a) décrit deux objets et (b) se réfère aux propriétés des deux objets.

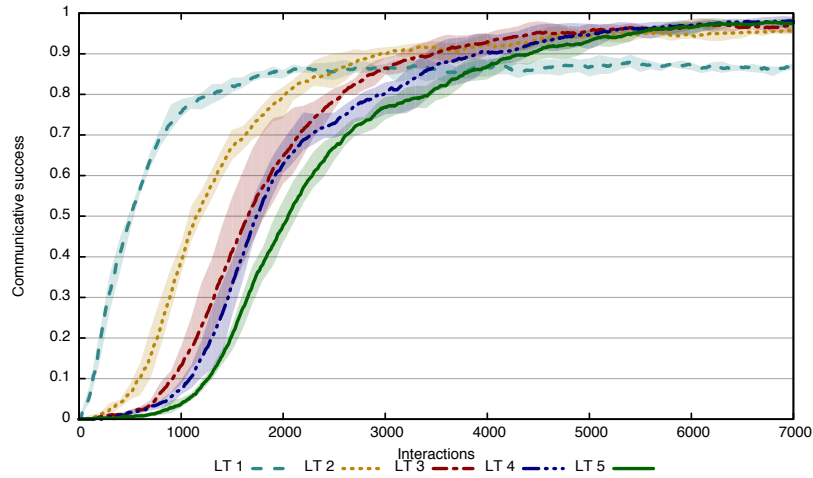


Figure B.5: Résultats d’une population de 10 agents effectuant les différentes tâches de communication, sur une moyenne de 10 répétitions. L’axe des abscisses représente le nombre de guessing games joués par la population et l’axe des ordonnées le succès communicationnel résultant.

D’une part, on peut observer que les tâches moins complexes (celles qui permettent à la population de se référer à un plus petit nombre de propriétés) génèrent plus rapidement une langue partagée, contrairement aux tâches plus complexes, où plus d’interactions sont nécessaires pour obtenir une langue commune. Mais une convergence rapide vers une langue commune n’implique pas que la langue résultante soit plus efficace en termes de communication : les tâches communicationnelles qui convergent plus rapidement vers une langue partagée présentent un pourcentage plus faible de réussite communicative.

Il existe une explication à ce comportement : les énoncés plus courts sont composés d’un plus petit nombre de mots. Cela permet aux agents d’apprendre et de convenir des différentes correspondances entre forme et sens, c’est-à-dire des constructions, utilisées pour créer ces énoncés. Les agents ne peuvent adopter une construction que s’il n’y a pas d’ambiguïté sur l’association entre sens et forme. Pourtant, les énoncés longs ont une probabilité plus grande que l’auditeur ne connaisse pas le sens de certains mots. Dans ces situations, quand il y a deux mots inconnus ou plus, l’auditeur ne peut apprendre aucune nouvelle construction. Par conséquent, la population a besoin de plus de temps pour converger vers une langue commune pour des tâches de communication plus complexes.

Par contraste, la possibilité de faire allusion à plusieurs propriétés des objets considérés réduit l’ambiguïté dans les énoncés. En conséquence, les énoncés plus longs ont un pouvoir descriptif plus grand que les énoncés plus courts. Afin d’analyser cet effet, nous avons mesuré l’ambiguïté des scènes décrites pour chaque tâche (tableau B.2). Les résultats indiquent qu’il y a une relation directe entre l’ambiguïté de l’énoncé et le pourcentage de jeux où la réussite communicative n’a pas pu être atteinte.

La raison pour laquelle certaines interactions sont un échec est causée par

Tâche de communication	LT 1	LT 2	LT 3	LT 4	LT 5
Réussite communicative	85.82%	94.95%	96.98%	98.06%	98.30%

Tableau B.2: Pourcentage de réussite communicative obtenue dans les 2000 dernières interactions pour chaque tâche de communication.

une description ambiguë du but communicationnel par le locuteur. Les descriptions qui font référence à toutes les propriétés du but communicationnel (deux propriétés si le but communicationnel ne comprend qu'un seul objet ou quatre si le sujet de la communication concerne deux objets) n'ont pas d'ambiguïté. La raison pour laquelle il y a de faibles pourcentages d'ambiguïté dans toutes les tâches est liée au fait que chaque tâche de communication inclut également les tâches moins complexes. Conséquemment, quelques uns des énoncés formulés ne comprendront pas toutes les propriétés du but communicationnel, ce qui produit des ambiguïtés dans certaines situations.

Intégration du principe d'auto-motivation dans l'architecture

Dans cette expérience nous étudions comment les tâches de communication dans l'expérience précédente peuvent agir comme *défis* d'une complexité croissante pour une population d'agents dotés d'un système de motivation intrinsèque.

Les défis peuvent être formalisés sous la forme d'un vecteur de paramètres qui représente le nombre de propriétés ($< nom_{prop} >$) à utiliser dans la description d'un but communicationnel, allant d'aucune propriété (niveau 1) à un maximum de quatre propriétés (niveau 5) (tableau B.3).

Tâche de communication	LT 1	LT 2	LT 3	LT 4	LT 5
Défi $< nom_{prop} > [niveau]$	$< 0 > [1]$	$< 1 > [2]$	$< 2 > [3]$	$< 3 > [4]$	$< 4 > [5]$

Tableau B.3: Conversion des tâches de communication de l'expérience de base en défis sous la forme d'un vecteur de paramètres avec un niveau de complexité associé.

L'objectif de l'expérience est le même que dans la section précédente : il faut que la population développe un langage partagé pour se référer aux objets de son environnement. Les agents de la population n'ont aucune construction et s'intéressent d'abord au défi le moins complexe (niveau 1). Après chaque interaction, les agents en interaction actualisent les valeurs de confiance et de persistance associées au défi abordé (voir algorithme 5).

Pendant qu'ils tentent un challenge C_i , les agents décident de manière autonome d'aborder un challenge plus complexe C_{i+1} quand ils entrent dans un état d'*ennui* ou décident de revenir à une tâche moins complexe C_{i-1} lorsque C_i se révèle être un défi trop complexe, ce qui provoque un état d'*anxiété*. Si aucune de ces situations ne se produit, les agents poursuivent leur tentative de C_i .

Les résultats obtenus sur une série de dix simulations pour une population de dix agents munis avec l'autotelic principe sont présentés dans la figure B.6a.

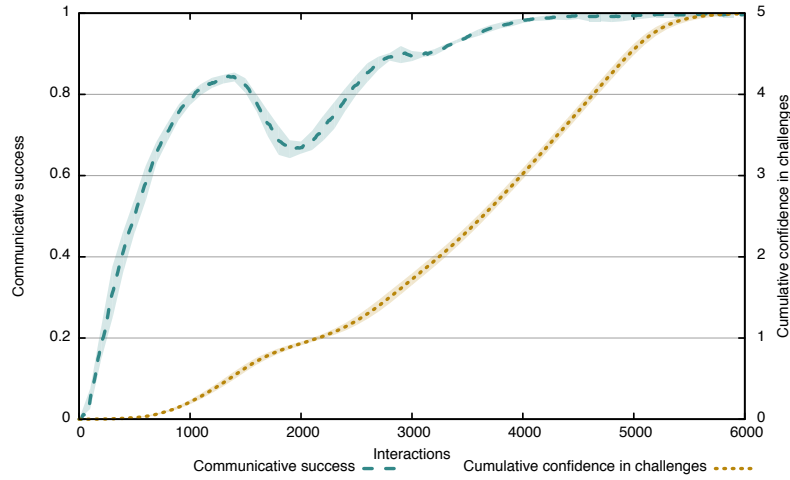
L'axe des abscisses représente le nombre de *guessing games* jouées par la population, l'axe des ordonnées à gauche indique le pourcentage de succès communicationnel et l'axe des ordonnées à droite montre le nombre de défis maîtrisés par la population, nombre qui correspond à l'addition de la valeur de confiance moyenne des agents dans la population pour tous les défis possibles. La figure B.6a présente les résultats au niveau de la population, mais il faut prendre en compte le fait que le processus de développement du langage de chaque agent n'est déterminable que par les interactions auxquelles il participe. Cela implique que certains agents peuvent atteindre plus rapidement une confiance maximale pour un défi particulier que d'autres agents et, par conséquent, passer à des défis plus complexes à des moments différents.

Comme on pouvait s'y attendre à partir des résultats obtenus dans l'expérience de base, l'objectif de développer un langage commun pour faire référence aux objets dans leur environnement est accompli. À la fin des simulations, la population obtient un pourcentage de réussite communicative de 100%, ce qui est différent des résultats obtenus dans la première expérience. Cette différence est une conséquence de la façon dont les défis sont définis, car le nombre de propriétés que le locuteur doit décrire est fixé pour chaque défi. Ainsi, quand (a) les agents maîtrisent les défis communicationnels plus complexes et (b) leurs lexiques sont alignés, les énoncés qu'ils formulent ne sont pas ambigus et toutes les interactions aboutissent à un succès communicationnel.

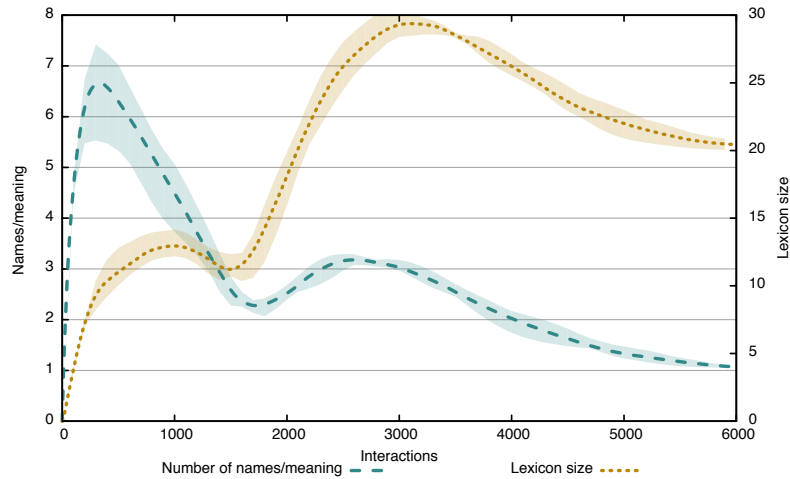
Le défi initial consiste à développer un langage partagé pour faire référence aux prototypes. Ce défi est accompli après environ 1200 interactions. À ce moment-là (a) la population s'est mise d'accord sur un lexique commun pour les prototypes et (b) le succès communicationnel moyen des interactions est établi à une valeur d'environ 85%, taux similaire au succès communicationnel résultant pour la première tâche de communication dans l'expérience de base. Certains agents atteignent la valeur de confiance maximum pour leur premier défi, entrent dans un état d'ennui et passent au second défi.

Le changement du défi actuel de certains agents provoque une baisse de la réussite communicationnelle dans la population. Ce phénomène est causé par le fait que, quand les agents tentent le défi du niveau 2, ils ne désignent plus seulement les prototypes des objets mais aussi une de ses propriétés (soit la forme ou la couleur). Afin de communiquer avec succès, ils doivent développer un vocabulaire pour les formes et les couleurs.

De plus, les énoncés de plusieurs mots présentent des problèmes de référence (à quel prototype une propriété fait référence) qui ne peuvent être gérés qu'en développant des constructions grammaticales qui introduisent des restrictions dans l'ordre des mots et la représentation sémantique. Une fois que les agents ont acquis une confiance maximale pour le deuxième défi (a) leur *lexique* leur permet de faire référence avec efficacité aux différentes couleurs et formes et (b) ils ont également développé et adopté une construction grammaticale pour éviter certains problèmes de référence. A ce stade, les agents ont donc un lexique commun pour les propriétés et les prototypes, ce qui accélère considérablement le développement des compétences liées aux derniers défis. Ils ont seulement besoin de développer d'autres constructions qui introduiront de nouvelles contraintes



(a) L'axe des abscisses représente le nombre de *guessing games* jouées par la population, l'axe des ordonnées à gauche indique le pourcentage de succès communicationnel et l'axe des ordonnées à droite montre le nombre de défis maîtrisés par la population.



(b) L'axe des abscisses représente le nombre de *guessing games* jouées par la population, l'axe des abscisses à gauche le nombre de termes par sens et l'axe des ordonnées à droite la taille moyenne du *lexicon* de la population.

Figure B.6: Résultats obtenus sur une série de dix simulations pour une population de dix agents muni de l'autotelic principe

grammaticales pour faire référence aux multiples propriétés des objets.

L'évolution de la concurrence entre les différentes constructions créées par la population et la taille du *lexique* est montrée dans la figure B.6b. Au cours des premières interactions il y a une prolifération de mots créés par les agents. Cette situation est attribuable au fait que (a) les agents commencent sans aucune construction et (b) les interactions entre eux sont encore très limitées, ce qui fait que les agents créent et apprennent continuellement de nouveaux termes auprès de leurs interlocuteurs pour nommer les différents prototypes existants. Autour de l'interaction 500 (environ 100 interactions par agent) cette tendance

cesse et le nombre de mots par sens commence à diminuer à cause du processus d'alignement, qui réduit la taille du lexique des agents en supprimant les constructions les moins utilisées.

À l'interaction 2000 environ, la compétition entre les constructions s'intensifie à nouveau. Cette situation s'explique par la bascule progressive de la population vers le second défi : à ce stade, de nouveaux termes concurrents apparaissent pour désigner les propriétés (couleur et forme) des objets. Cette tendance se maintient jusqu'à l'interaction 3000, où un deuxième pic correspondant à un nombre important de mots par sens est observé et l'inventaire de constructions d'un agent a alors en moyenne 30 constructions.

Après ce point, le nombre de mots par sens et la taille du lexique diminuent progressivement, à mesure que le processus d'alignement élimine lentement les constructions les moins utilisées. À la fin des simulations, la taille du lexique a été réduite au minimum. Ceci fait descendre le nombre de mots par sens à un, parce qu'il n'y a plus de compétition entre les constructions.

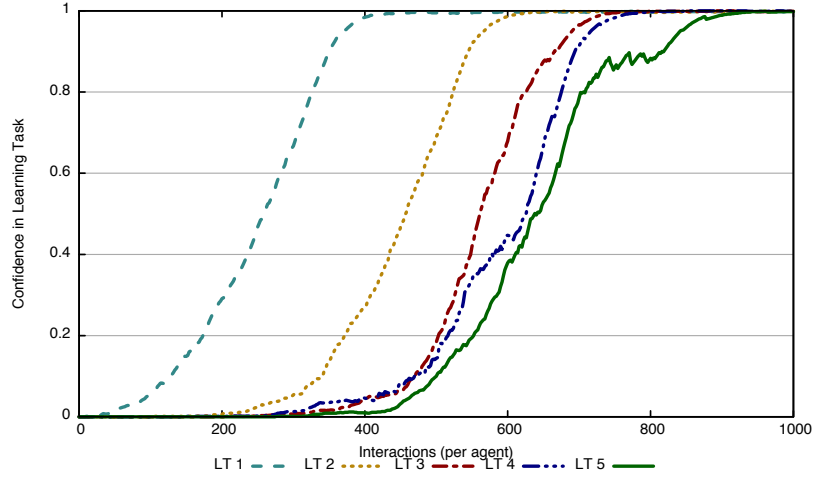
Acquisition progressive de compétences

En utilisant l'autotelic principe, les agents peuvent augmenter graduellement la complexité de leur système de communication, tout en évitant les tentatives de formulation d'énoncés trop complexes pour leur niveau de compétence actuel. Cela permet d'éviter les interactions autour d'un énoncé trop difficile pour l'auditeur, qui ne serait pas capable d'extraire l'information utile. Ce genre d'interactions se produit beaucoup plus souvent dans l'expérience de base. Dans ces situations, l'auditeur ne peut pas établir le sens associé à chaque mot inconnu et il n'est pas capable de déduire les correspondances correctes entre les mots et les sens. Est-ce que l'organisation des tâches de communication en étapes de complexité croissante accélère le développement des compétences, par comparaison avec l'expérience de base où l'apprentissage des compétences requises pour une tâche n'est pas organisé ?

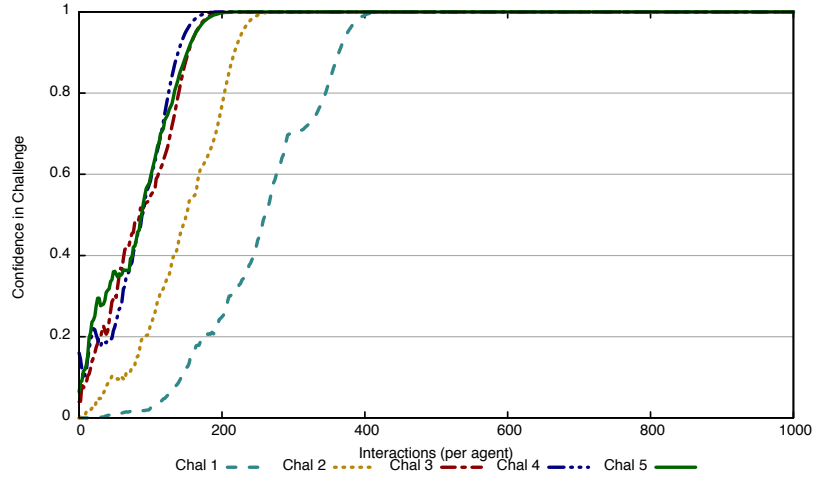
Afin de répondre à cette question, nous avons calculé le nombre d'interactions nécessaires pour chaque tâche de communication dans l'expérience de base (figure B.7a) et dans chaque défi (figure B.7b) pour obtenir une confiance maximale. Ces résultats ont été obtenus en calculant le gain de confiance pour chaque tâche de communication de la même façon qu'il est calculé dans le système de motivation, bien que la valeur de *confiance* soit associée à chaque défi généré dans le cas de l'expérience implémentant l'autotelic principe et que cette valeur n'est pas utilisé du tout dans l'expérience de base.

Les résultats obtenus montrent que les tâches de communication plus complexes nécessitent plus d'interactions que les tâches moins complexes pour obtenir un niveau de confiance maximal et, par conséquent, ce sont des tâches plus difficiles à réaliser. Cette relation est incrémentale, parce qu'un agent qui tente la première tâche de communication n'a besoin que d'environ 400 interactions pour arriver à une confiance maximale, alors qu'un agent qui effectue la cinquième tâche aura besoin de près de 1000 interactions pour la maîtriser.

En comparaison, cette relation est inversée quand les agents contrôlent la



(a) Nombre moyen d'interactions nécessaires pour obtenir une valeur de confiance maximale pour les différentes tâches de communication de l'expérience de base.



(b) Nombre moyen d'interactions nécessaires pour obtenir une valeur de confiance maximale pour chaque défi de l'expérience où les agents sont dotés de l'autotelic principe.

Figure B.7: Nombre moyen d'interactions nécessaires pour obtenir une confiance maximale pour les différentes tâches de communication (figure B.7a) et les défis (figure B.7b).

complexité de leurs interactions avec l'autotelic principe. C'est au cours des premiers défis que les agents ont besoin de plus d'interactions pour arriver à une valeur de confiance maximale. Comme ce système de motivation leur permet d'organiser le développement des compétences en étages, ils peuvent réutiliser les compétences acquises dans des situations moins complexes, ce qui accélère le gain de confiance pour les défis plus exigeants. Par exemple, si le premier défi demande autant d'interactions que la première tâche (400), celui du niveau 5 ne nécessite que 200 interactions pour être maîtrisé.

Ces résultats expérimentaux montrent que les agents qui essayent de rester dans un état de flow réduisent le nombre d'énoncés ambigus formulés. De plus, le nombre d'interactions qui permettent d'atteindre une valeur de confiance maximale pour chaque défi diminue pour les tâches plus complexes, car (a) les agents disposent des compétences acquises lors des défis précédents et (b) le nombre d'énoncés trop complexes pour l'apprentissage diminue.

Expériences dans un domaine continu : les couleurs

Dans la deuxième série d'expériences, la population des agents a pour objectif de développer un langage commun pour le domaine de la couleur. Comme dans le cas des expériences précédentes, les agents doivent organiser eux-mêmes un vocabulaire pour faire référence aux différentes couleurs dans leur contexte. Ils gèrent la complexité de leurs interactions linguistiques avec l'autotelic principe. Les agents commencent par produire des descriptions de couleurs simples et les complexifient progressivement en énoncés de plusieurs mots afin de pouvoir désigner des réalités plus complexes.

Avant de présenter les simulations réalisées, il faut d'abord introduire quelques notions de ce domaine. Il est communément admis qu'un *espace de couleurs*, l'espace de couleurs qui peut être perçu, est organisé en plusieurs *catégories de couleurs*, différentes subdivisions dans cet espace. Les *prototypes de couleur* sont des points dans l'espace de couleurs qui représentent une catégorie de couleur particulière dans cet espace [98]. Il faut souligner que, malgré le fait que toutes les langues humaines divisent l'espace de couleurs en catégories de couleurs, le nombre de prototypes et leur position dans l'espace de couleurs varient d'une langue à l'autre.

Formellement, les langues humaines séparent l'espace de couleurs en un ensemble de prototypes de couleurs $\{c_1, c_2, \dots, c_n\}$. Étant donné le prototype de couleur c_k , sa cellule associée R_k , qui détermine la catégorie de couleur correspondante, contient tous les points pour lesquels la distance à c_k est plus courte ou égale à la distance à tout autre prototype c_i (figure B.8).

Différents systèmes de couleurs ont été proposés pour représenter les espaces de couleurs⁵. Dans toutes les expériences dans ce domaine nous utilisons le système CIE 1976 $L^*a^*b^*$ [96]. Dans ce système, les couleurs sont représentées selon trois dimensions : la dimension L^* représente la luminosité, la dimension a^* la proportion de "rouge-vert" et la dimension b^* la proportion de "jaune-bleu".

⁵ Les lecteurs intéressés sont invités à consulter l'annexe "Colour Spaces and Systems" dans [21] pour un récapitulatif des différents systèmes de couleurs.

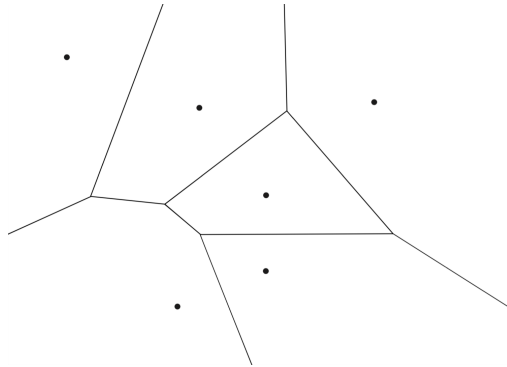


Figure B.8: Un espace de couleurs est divisé en catégories qui représentent des sous-ensembles de cet espace. Ces catégories sont généralement représentées par des prototypes, qui sont des points dans cet espace (figure extraite de Bleys [21]).

La différence entre deux couleurs est déterminée par leur distance euclidienne. Les valeurs de la dimension L^* vont de 0 (noir) à 100 (blanc), mais les deux valeurs des dimensions restantes peuvent avoir des valeurs positives ou négatives et ne sont limitées que par les propriétés physiques des matériaux [21, 192].

Stratégies linguistiques pour le domaine de la couleur

Les langues humaines emploient différentes méthodes pour décrire les couleurs, allant de l'utilisation d'un seul terme à des descriptions plus élaborées qui impliquent des modificateurs ou une combinaison de prototypes. Cette section présente les stratégies linguistiques utilisées dans les simulations informatiques. Les stratégies de couleur basique, d'appartenance graduée et de combinaison de couleurs ont été adaptées du code de Joris Bleys, qui m'a généreusement donné accès à l'implémentation originale des stratégies linguistiques qu'il a utilisées pour réaliser mes propres expériences.

Stratégie de couleur basique

Dans la stratégie de couleur basique, un seul terme est utilisé pour décrire une couleur. Dans cette stratégie, le locuteur (a) identifie d'abord le prototype de couleur qui est le plus proche de la couleur qu'il veut désigner et (b) il ou elle utilise le terme associé à cette catégorie pour désigner la couleur choisie. Un exemple est l'utilisation des mots "blue" ou "green" pour décrire une couleur en anglais.

Stratégie d'appartenance graduée

La stratégie d'appartenance graduée décrit une couleur en exprimant à la fois le prototype de couleur le plus proche et la distance existante entre la couleur et le prototype. Comme dans la stratégie précédente, le locuteur doit (a) identifier le prototype de couleur qui est le plus proche de la couleur à décrire et (b) il ou elle

doit également rapporter la distance entre la couleur et le prototype de couleur le plus proche, qui est désignée par un ensemble de modificateurs. Un exemple de cette stratégie est l'utilisation des énoncés "very green" ou "blue-ish" pour décrire une couleur en anglais.

Stratégie de combinaison de couleurs

Une autre stratégie observée dans les langues humaines est la combinaison de couleurs. Comme le nom le suggère, cette stratégie consiste à décrire une couleur en utilisant deux couleurs choisies par le locuteur. Différents modèles ont été proposés, mais j'ai choisi d'utiliser la stratégie proposée par Bleys [21] qui consiste à classer deux fois une couleur, la seconde classification ayant lieu après une transformation de l'espace des couleurs.

Dans ce modèle, les locuteurs (a) identifient d'abord le prototype de couleur le plus proche de la couleur à décrire, puis (b) transforment l'espace de couleurs vers ce prototype (c'est-à-dire ils modifient les valeurs associées à chaque prototype de couleur vers le prototype sélectionné) et, finalement, ils (c) classifient à nouveau la couleur sur l'espace de couleurs transformé, obtenant ainsi un deuxième prototype. La transformation de l'espace de couleurs réduit la surface de la catégorie de couleur identifiée en (a) mais n'enlève pas son prototype de l'espace de couleurs, ce qui permet de classer la couleur deux fois par rapport au même prototype, avant et après la transformation de l'espace. Un exemple de cette stratégie en anglais est l'énoncé "yellow-green" pour décrire une couleur.

Stratégie de luminosité

La stratégie de luminosité décrit une couleur par rapport à son prototype de couleur le plus proche en y intégrant la différence de luminosité entre la couleur et le prototype. Il y a différentes approches pour modéliser cette stratégie. Par exemple, Bleys [21] l'a modélisé comme une combinaison de catégories de couleurs avec les prototypes de couleurs blanc et noir. Dans cette implémentation, l'énoncé "bleu clair" serait l'expression résultante de la combinaison des catégories "bleu" et "blanc". L'approche utilisée ici spécifie la différence de valeur de luminosité entre la couleur et le prototype de couleur le plus proche.

De façon similaire à la stratégie de couleur basique, le locuteur doit d'abord (a) identifier le prototype le plus proche de la couleur à décrire, mais en plus (b) il ou elle doit aussi indiquer la différence de luminosité entre le prototype et la couleur à décrire, qui dans l'implémentation prend la forme d'un nombre décimal et qui est exprimé par un ensemble de modificateurs. Dans le système de couleurs CIE 1976 $L^*a^*b^*$, la dimension L^* indique la luminosité des couleurs. L'utilisation d'un modificateur de luminosité sur une couleur modifie la valeur de cette dimension, l'augmentant ou la diminuant dans la plage des valeurs possibles de L^* . Dans cette approche, le modificateur de luminosité dans l'énoncé "bleu clair" effectue une opération dans laquelle la valeur de la dimension L^* de la catégorie de couleur "bleu" est augmentée.

Stratégie de chromaticité

Cette stratégie indique la saturation d'une couleur. Le niveau de saturation est indiqué par un modifieur (aussi modelé sous la forme d'un nombre décimal) et a été implémenté d'une façon similaire à la stratégie de luminosité. Tout d'abord, le locuteur (a) identifie la catégorie de couleur la plus proche à la couleur à décrire et (b) utilise un modificateur pour décrire la différence de chroma entre la catégorie et la couleur.

La principale différence par rapport à la stratégie de luminosité est que les modificateurs chromatiques opèrent sur les valeurs à deux dimensions (a^* et b^*) au lieu d'une seule (L^*). Ce qu'un modificateur chromatique fait est augmenter ou diminuer la valeur de *chroma* (C_{ab}^*) d'une couleur tout en conservant la même valeur de *hue* (h_{ab}^*). Ces valeurs ne sont pas directement représentées dans le système de couleurs CIE 1976, mais peuvent être calculées en utilisant les formules suivantes [21, 192] :

$$C_{ab}^* = \sqrt{a^{*2} + b^{*2}}$$

$$h_{ab}^* = \arctan \frac{b^*}{a^*}$$

Cette stratégie est également observée en anglais. Par exemple, l'énoncé "pale blue" diminue la saturation de la catégorie associée à "blue", alors que l'énoncé "bright blue" l'augmente.

Stratégies linguistiques en tant que réseaux de contraintes sémantiques

Les différentes stratégies linguistiques ont été implémentées en tant que *réseaux de contraintes sémantiques* utilisant le système computationnel *Incremental Recruitment Language*. Ces réseaux contiennent (a) des prédicats qui exécutent des *opérations cognitives* telles que la sélection, la catégorisation ou le filtrage d'objets, et (b) des *entités sémantiques*, qui dans ce groupe d'expériences correspondent à des prototypes de différents types (couleur, appartenance, luminosité ou chromaticité). Cette section décrit succinctement les opérations cognitives utilisées dans les différentes stratégies linguistiques des expériences :

- **equal-to-context** : cette opération introduit une variable qui contient l'ensemble des objets dans le contexte, ce qui dans ces expériences consiste en un ensemble de couleurs.
- **get-XXXX-category-set** : cette opération introduit une variable contenant les différents prototypes de la catégorie spécifiée. Selon la catégorie choisie, la variable peut contenir un ensemble de prototypes de couleur, de luminosité, d'appartenance ou chromatiques.
- **profile-XXXX-dimensions** : cette fonction prend un ensemble de couleurs et les filtre sur certaines dimensions du système de couleurs, retournant

un ensemble filtré. Comme nous utilisons le système de couleurs CIE 1967 $L^*a^*b^*$, le profilage selon (a) la *luminosité* bloquera les dimensions a^*b^* de chaque couleur (ayant pour résultat un ensemble d'échantillons de couleur sur l'échelle des gris), (b) la *chromaticité* filtrera l'ensemble de couleurs enlevant les valeurs de la dimension L^* et (c) la couleur retournera l'ensemble original.

- **categorise-by-XXXX** : cette fonction prend comme entrée un ensemble d'entités et une catégorie donnée, et retourne l'ensemble des entités ayant la catégorie donnée comme prototype. Par exemple, classer un ensemble d'échantillons de couleur selon une catégorie de couleur c_n renverra le sous-ensemble d'échantillons dont la catégorie la plus proche est c_n et une valeur d'activation pour chacun d'entre eux qui représente la similarité entre l'échantillon de couleur et la catégorie de couleur.
Lors de la catégorisation selon la *luminosité* ou *chromaticité*, un argument supplémentaire est nécessaire, car ces prototypes ont des valeurs dimensionnelles différentes en fonction de l'entité sur laquelle ils sont appliqués. Prenant un exemple en anglais, le modificateur de luminosité "dark" correspondra à différentes valeurs de la dimension L^* lorsqu'il est appliqué aux catégories "yellow" ou "brown".
- **draw-category-set-to-category** : cette opération transforme les groupes de catégories en les réduisant vers une catégorie spécifiée. Elle renvoie un ensemble de catégories transformées où les valeurs de leurs prototypes ont été modifiées vers la catégorie d'entrée. Cette opération cognitive est utilisée dans la stratégie de combinaison des couleurs pour transformer un ensemble de catégories après une première classification d'une couleur afin de réaliser une seconde classification.
- **select-most-activated** : cette fonction sélectionne l'entité la plus activée parmi un ensemble donné. Elle utilise les valeurs d'activation obtenues dans les opérations cognitives de catégorisation pour déterminer l'entité la plus prédominante de l'ensemble d'entrée.

Ces prédicats sont ensuite combinés dans des réseaux IRL qui représentent les différentes stratégies linguistiques pour décrire les couleurs. Nous illustrons ces réseaux en prenant comme exemple la stratégie de couleur basique (figure B.9). Le prédicat de signification **equal-to-context** introduit les couleurs présents au contexte, représentés par la variable $?s1$, dans la représentation du sens de l'agent. Cet ensemble est filtré sur les dimensions $L^*a^*b^*$ (variable $?s2$). Les prototypes de couleur que l'agent connaît sont obtenus avec l'opération **get-basic-colour-category-set** (variable $?bccs$). Quand cette stratégie est utilisée en formulation, le locuteur doit trouver la catégorie $?cc$ dans son ensemble de catégories qui est la plus activée étant donné le $?topic$, et utilisera le mot associé à cette catégorie pour décrire la couleur. En compréhension, par contre, l'auditeur utilise la catégorie de couleur $?cc$ obtenue à partir de l'énoncé et l'utilise pour identifier l'échantillon de couleur dans le contexte qui est le plus actif.

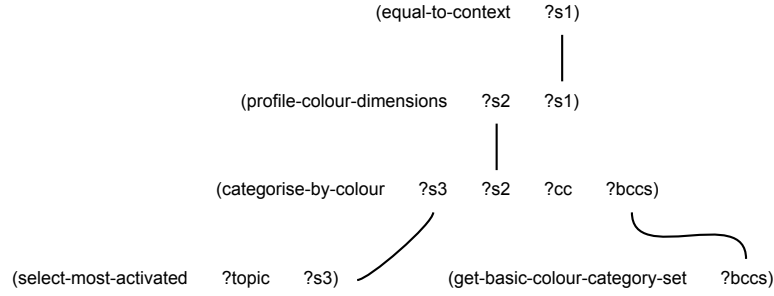


Figure B.9: Exemple d’une stratégie linguistique en tant que réseau connecté de prédicats.

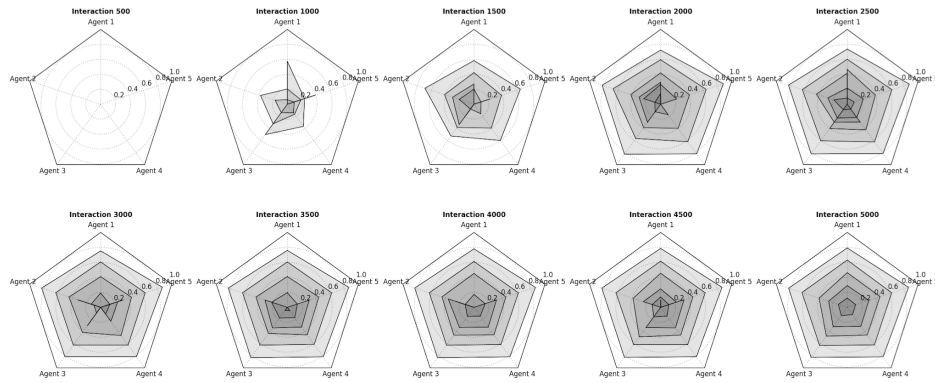


Figure B.10: Exemple d’alignement de prototypes d’appartenance dans une population de 5 agents.

Alignement des prototypes

Outre le processus d’alignement des constructions, les différents prototypes impliqués dans une interaction réussie suivent également un processus d’alignement (figure B.10). L’idée est que les valeurs dimensionnelles des prototypes changent au fil du temps, à mesure que les agents sont exposés à un plus grand nombre de situations concernant ces prototypes. Par exemple, lorsqu’un agent apprend un nouveau prototype de couleur associé à un mot inconnu w_i , sa valeur correspond exactement aux valeurs dimensionnelles de l’échantillon de couleur dans cette interaction. La valeur de ce prototype de couleur doit néanmoins représenter la couleur moyenne qui est identifiée par ce mot w_i au lieu d’être fixée par les valeurs du premier échantillon de couleur nommé avec mot. Par conséquent, la valeur d’un prototype change au fur et à mesure qu’il est utilisé dans un plus grand nombre d’interactions.

Régulation de la complexité linguistique

Cette expérience reproduit l'expérience de base dans laquelle l'autotelic principe est utilisé par une population d'agents pour réguler de façon autonome la complexité linguistique de leurs interactions. Cependant, les deux expériences diffèrent car l'expérience décrite ici se déroule dans un domaine continu, celui de la couleur, alors que l'autre se déroulait dans un monde discret, où les caractéristiques des objets et les relations spatiales sont prédéterminées. Cette différence a un impact direct sur les niveaux sensorimoteur et conceptuel des interactions linguistiques, car les agents ont non seulement à créer et à apprendre des constructions lexicales et grammaticales, mais ils doivent aussi créer et apprendre des prototypes différents qui représentent le sens associé aux constructions lexicales.

Les agents participent à un *guessing game* dans lequel ils communiquent sur des échantillons de couleurs. Ils commencent les simulations avec un *inventaire de constructions* vide et y ajoutent des constructions grâce à trois stratégies linguistiques, celles de *couleur basique*, d' et une *combinaison de catégories graduées*, qui sont identifiés par les agents comme des défis communicationnel d'une complexité croissante. La complexité d'une tâche de communication est déterminée par le nombre d'opérations cognitives requises. Ce nombre diffère selon les différents défis et est utilisé pour établir son *niveau*. La stratégie de *couleur basique* a le plus petit réseau de contraintes sémantiques, car elle nécessite moins de prédicats que les autres stratégies, alors que la stratégie *combinaison de catégories graduées* a le plus grand réseau de contraintes sémantiques. Les agents commencent par adresser le défi moins complexe, et utilisent la valeur de *niveau* pour se déplacer entre les différents défis, en fonction de l'état interne (*ennui*, *anxiété* ou *flow*) associé par chaque agent à chaque défi.

Une description de couleur plus complexe peut en plus réutiliser les compétences acquises à un stade plus précoce. En abordant la stratégie de *couleur basique*, les agents doivent converger à la fois sur une classification de l'espace de couleurs en différents prototypes et sur les termes associés à chaque prototype. Une fois que cela a été réalisé, les agents peuvent passer au deuxième défi, qui correspond à la stratégie d'. Comme les prototypes de couleurs et les termes associés sont déjà connus, la population peut se concentrer sur la création et l'alignement des prototypes d'appartenance et leurs termes associés. Finalement, dans la stratégie *combinaison de catégories graduées*, une combinaison des stratégies d' et de *combinaison de couleurs*, il est nécessaire de connaître les prototypes de couleurs et d'appartenance. Le tableau B.4 présente les différents défis communicationnels comme vecteur de deux dimensions.

Dans cette expérience, le monde est fait de 268 échantillons de couleurs différents dans l'espace de couleurs CIE 1967 L^*a^*b . Le monde contient les couleurs focales⁶ et les échantillons consensuels⁷ pour l'anglais ainsi que des échantillons

⁶Une couleur focale est une valeur d'une certaine catégorie de couleur qui représente le meilleur exemple de cette catégorie.

⁷Échantillons de couleurs qui ont été nommés de façon uniforme en anglais par tous les participants. Voir Sturges et Whitfield [149].

Niveau	[1]	[2]	[3]
Défi	< 1, 0 >	< 1, 1 >	< 2, 1 >
< nom_{col}, nom_{app} >			

Tableau B.4: Défis encodés sous la forme de vecteurs de dimension deux qui correspondent au nombre maximum de couleurs et de modificateurs d'appartenance exprimés. La valeur de niveau désigne la complexité d'un défi, qui est calculée sur la base de son nombre d'opérations cognitives.

de couleurs créés par la combinaison de deux couleurs focales selon des pourcentages différents : 25%, 45%, 55% et 75%, respectivement.

Le contexte de l'interaction se compose d'un sous-ensemble des couleurs présents dans le monde. Le contexte est déterminé à chaque interaction par le locuteur, qui choisit sa taille et les couleurs qui en font partie en fonction de son défi actuel. Dans le défi de couleur basique, le contexte est créé en choisissant aléatoirement trois couleurs focales de l'anglais. Dans le défi d'appartenance graduée, le locuteur choisit cinq échantillons au hasard parmi les échantillons consensuels pour l'anglais. Enfin, dans le défi de combinaison de catégories graduées, le locuteur choisit six échantillons de couleurs qui correspondent à la combinaison de deux couleurs focales pour l'anglais.

Les résultats de l'expérience sont présentés sur la figure B.11. Les agents commencent par aborder le premier défi, pour lequel la population doit créer et coordonner son lexique pour (a) les termes pour désigner les couleurs et (b) un ensemble de prototypes de couleur. Les agents acquièrent rapidement confiance pour ce défi, car le succès communicationnel et la valeur de confiance augmentent rapidement. Autour de l'interaction 2000, une diminution brusque de ces deux mesures est observée lorsque certains agents atteignent une valeur de confiance maximale. Dans cette situation les agents entrent dans un état interne d'*ennui*, passent à la phase de réorganisation et abordent le deuxième défi. Dans ce défi, les agents doivent développer des constructions lexicales et des prototypes pour pouvoir indiquer la distance entre les échantillons et les prototypes de couleur.

Autour de l'interaction 5000, une fraction de la population a déjà atteint une valeur de confiance maximale pour le second défi, ce qui entraîne un chevauchement du défi actuel dans la population. À ce point, certains agents sont de nouveau dans un état interne d'*ennui* et sont motivés pour tenter le troisième défi alors que l'autre partie de la population s'occupe encore du second défi. Finalement, tous les agents passent au troisième défi et le pourcentage de succès communicationnel s'améliore graduellement à mesure que la population parvient à aligner son lexique et ses prototypes. En conséquence, le taux de réussite de l'alignement, qui mesure la cohésion moyenne du lexique dans la population, augmente également jusqu'à atteindre le même pourcentage que le taux de réussite de la communication. À la fin des simulations, tous les agents de la population ont une valeur de confiance maximale pour les trois défis et une valeur de succès communicative constante supérieure à 90%.

Le fait qu'un pourcentage de succès communicationnel de 100% n'est pas obtenu même quand tous les agents ont des valeurs de confiance maximales pour

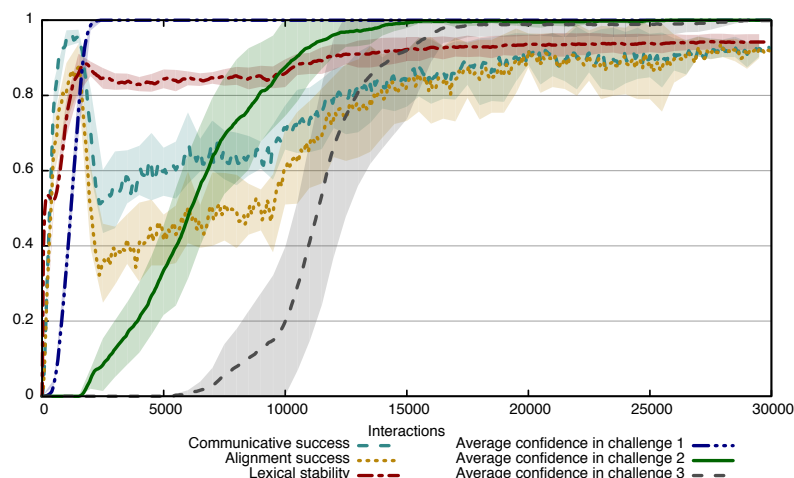


Figure B.11: Résultats de l'expérience pour une population de 10 agents sur une moyenne de 10 répétitions de 30000 interactions chacune.

tous les défis est attribuable au fait que les stratégies linguistiques disponibles (*couleur basique*, *d'appartenance graduée* et *combinaison de catégories graduées*) ne sont pas assez discriminatoires dans certaines situations. Cela est dû au fait que le locuteur ne peut pas conceptualiser une couleur dans des contextes particuliers. Cet effet est également observé par Bleys [21] quand il examine chaque stratégie linguistique indépendamment.

Gestion de défis multiples

Pour cette expérience, nous voulons tester l'utilité du système de motivation pour choisir non seulement *quand* les agents changent leur défi actuel, mais aussi *quel* défi un agent doit aborder ensuite, en fonction des défis qu'il connaît et des caractéristiques des contextes où se déroulent les interactions. Dans ce but, nous avons intégré les stratégies de couleur basique, de combinaison de couleurs, de luminosité et chromaticité dans une seule expérience.

Comme dans l'expérience précédente, les agents participent à un *guessing game* dans lequel ils communiquent sur des échantillons de couleurs. La population commence les simulations avec un inventaire de constructions vide (pas de constructions lexicales ou grammaticales) et pas de catégories de couleurs, de luminosité ou de chromaticité. Les agents augmentent leur inventaire de constructions et de catégories en utilisant plusieurs stratégies linguistiques, identifiées comme des défis communicationnels d'une complexité croissante.

La complexité d'une tâche de communication est de nouveau calculée en fonction du nombre d'opérations cognitives nécessaires pour la réaliser. Ce nombre diffère selon les différents défis et est utilisé pour déterminer le *niveau* de la tâche en question. Les agents commencent par tenter le défi le moins complexe, et utilisent la valeur du niveau associée aux tâches pour se déplacer entre elles, en fonction de l'état interne (*ennui*, *anxiété* ou *flow*) associé par chaque agent à

Niveau	[1]	[2]	[3]
Défi	$\langle 1, 0, 0 \rangle$	$\langle 1, 1, 0 \rangle$	$\langle 2, 1, 0 \rangle$
$\langle nom_{col}, nom_{lum}, nom_{chrom} \rangle$		$\langle 1, 0, 1 \rangle$	$\langle 2, 0, 1 \rangle$
		$\langle 2, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$

Tableau B.5: Les défis sont définis comme un vecteur de trois dimensions qui correspondent au nombre maximum de prototypes de couleur, de luminosité et de chromaticité exprimés.

chaque défi. Néanmoins, dans cette expérience, certaines stratégies linguistiques ont le même nombre d'opérations cognitives et sont donc assignées au même niveau de complexité. Cela implique que les agents peuvent affronter plusieurs défis en même temps et, afin de pouvoir développer un langage commun, ils doivent être capables de gérer simultanément diverses tâches de communication.

Comme dans l'expérience précédente, le premier niveau de complexité correspond à la stratégie de couleur basique, car elle a le plus petit réseau de contraintes sémantiques. Le deuxième niveau de complexité correspond à trois stratégies différentes : celles de *combinaison de couleurs*, de *luminosité* et de *chromaticité*, parce que ces trois stratégies nécessitent le même nombre de contraintes sémantiques. Finalement, le troisième niveau de complexité correspond aux stratégies formées à partir d'une combinaison de deux stratégies linguistiques du deuxième niveau. Une fois qu'un agent a atteint une valeur de confiance maximale pour un défi de niveau 2, il génère deux nouveaux défis en mélangeant une autre stratégie linguistique du même niveau à ce défi. Le tableau B.5 présente les différents défis communicationnels sous la forme d'un vecteur de trois dimensions.

Le monde est composé de 335 échantillons de couleurs dans l'espace de couleur CIE 1967 $L^*a^*b^*$. Le monde contient les couleurs focales pour l'anglais et les couleurs obtenus (a) en modifiant la valeur L^* de ces couleurs focales (en l'augmentant ou en la diminuant de 20.0 dans ses limites :[0,0,100,0]), (b) en modifiant la valeur chromatique C_{ab}^* des couleurs focales (en l'augmentant ou en la diminuant de 0.2), (c) en combinant deux couleurs focales pour l'anglais en pourcentages différents (25%, 45%, 55% et 75% respectivement) et (d) en combinant deux des modifications précédentes ((a), (b) et (c)). Les échantillons de couleurs de (d) sont créés en effectuant soit les modifications de luminosité et de chromaticité aux couleurs focales pour l'anglais, soit en appliquant des modifications de luminosité ou de chromaticité aux échantillons de couleurs de (c).

Le locuteur sélectionne le contexte, qui est un sous-ensemble des couleurs présents dans le monde, au début de chaque interaction. L'agent choisit la taille du contexte et les différents échantillons de couleurs qui en font partie. Ce choix dépend du défi actuel du locuteur. Pour le défi de niveau 1, le contexte est créé en choisissant aléatoirement trois couleurs focales pour l'anglais. Quand il aborde les défis de niveau 2, le locuteur compose des contextes différents de trois échantillons en fonction du défi qu'il veut tenter. Pour la *stratégie de luminosité*, l'agent sélectionne des couleurs qui varient dans leur valeur associée à la dimension L^* , pour la *stratégie de chromaticité* le locuteur sélectionne des échantillons

de couleur avec valeurs de h^*_{ab} similaires et pour la *stratégie de combinaison de couleurs* le locuteur choisit au hasard trois échantillons de couleur parmi les couleurs résultantes d'une combinaison de deux couleurs focales.

Finalement, pour les défis du niveau 3 le locuteur confectionne un contexte avec quatre échantillons de couleurs. Pour la *stratégie de combinaison de couleurs-luminosité*, ces échantillons sont choisis parmi les couleurs résultantes d'une modification de luminosité sur une combinaison de deux couleurs focales. Pour la *stratégie de combinaison de couleurs-chromaticité* sur des combinaisons de deux couleurs focales après une modification chromatique et pour la *stratégie de luminosité-chromaticité* sur des couleurs focales auxquelles une modification chromatique et de luminosité a été effectué.

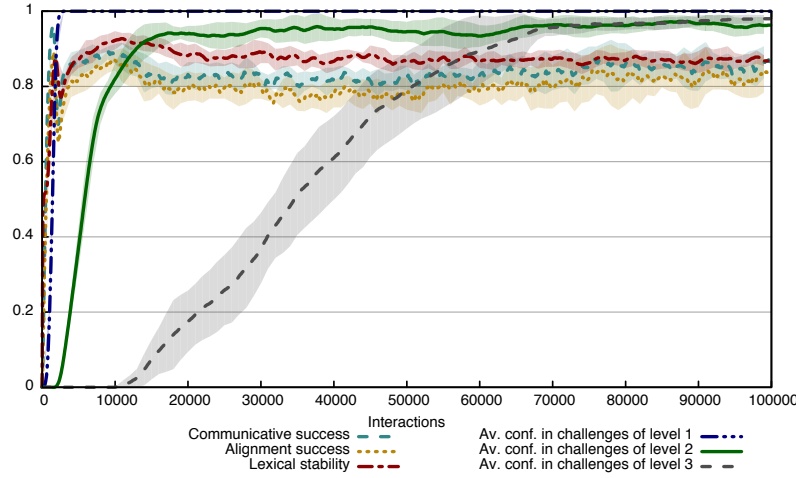
Les résultats de l'expérience sont présentés sur la figure B.12. Les agents commencent par aborder le défi de niveau 1, pour lequel ils doivent créer et coordonner un ensemble de prototypes et de mots pour se référer aux couleurs. Comme dans l'expérience précédente, la population accomplit cette tâche assez rapidement. Ils ont besoin d'environ 500 interactions pour communiquer avec un pourcentage de succès du 80%, et autour de l'interaction 1000 (environ 200 interactions par agent) ce pourcentage a augmenté à plus du 95%. En conséquence, la valeur de confiance moyenne du premier défi augmente rapidement et, peu après l'interaction 2000, tous les agents de la population ont une valeur de confiance maximale pour cette tâche.

Simultanément, on observe une baisse du succès communicationnel et de l'alignement. C'est une conséquence de l'ennui interne des agents, car le défi de niveau 1 n'est plus stimulant. Cela provoque l'entrée des agents dans la phase de réorganisation et cause la génération de trois tâches communicationnelles plus exigeantes (celles associées à un niveau 2 de complexité), qui correspondent aux stratégies de combinaison des couleurs, de luminosité et de chromaticité.

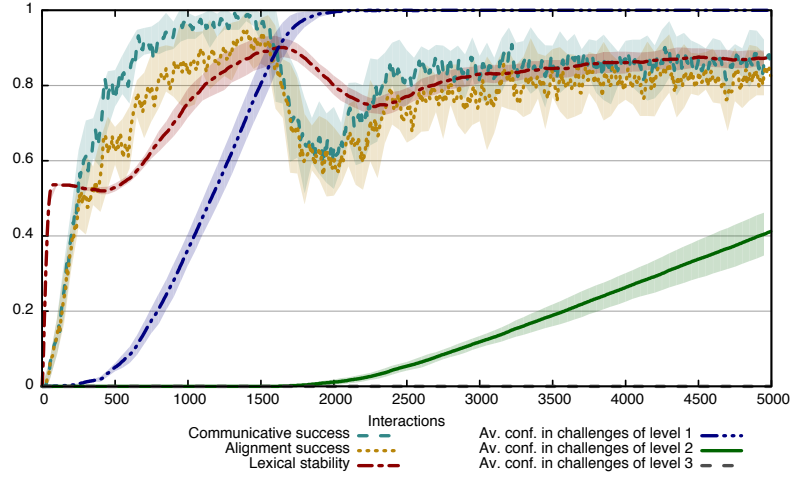
Dans le but de se remettre de la réduction du succès communicationnel et de pouvoir faire face aux nouveaux défis, la population crée des prototypes de luminosité et des prototypes chromatiques afin de distinguer les échantillons de couleurs dans les nouveaux contextes. Cependant, la confrontation avec trois défis communicationnels à la fois ralentit l'alignement de ces prototypes et des constructions lexicales qui leur sont associés. Cela est dû en partie aux problèmes que les auditeurs rencontrent lorsqu'ils doivent estimer le défi tenté par le locuteur. Néanmoins, la population parvient à augmenter le succès communicationnel au-delà du 80% une fois qu'elle commence à aligner ces prototypes.

Quand un agent arrive à une valeur de confiance maximale pour un défi de niveau 2, il éprouve un état d'ennui pour ce défi et l'utilise pour générer de nouveaux défis d'une complexité supérieure. Cependant, il choisit de se concentrer sur les défis restants du niveau 2 pour lesquels il n'a pas encore acquis une confiance maximale au lieu de s'attaquer aux défis récemment créés du niveau 3. Ce n'est qu'autour de l'interaction 15000 que certains agents de la population ont déjà acquis une confiance maximale pour les trois défis du deuxième niveau de complexité et commencent à affronter les défis du troisième niveau.

Cette situation entraîne un chevauchement des défis de niveaux 2 et 3 que la population tente de maîtriser. Il faut noter que ce n'est qu'à la fin des simulations



(a) Résultats de l'expérience pour une population de 10 agents sur une moyenne de 10 répétitions de 100.000 interactions chacune.



(b) Détail des 5.000 premières interactions de la population.

Figure B.12: Résultats de l'expérience pour une population de 10 agents sur une moyenne de 10 répétitions de 100.000 interactions chacune (figure B.12a) et détail des 5.000 premières interactions (figure B.12b).

que la population atteint les valeurs de confiance les plus élevées pour les défis de ces deux niveaux. Afin de comprendre ce comportement, nous devons examiner l'évolution de la confiance moyenne pour chaque tâche de communication (figure B.13).

Il apparaît que les stratégies qui impliquent des catégories chromatiques sont plus difficiles pour les agents que les autres tâches de communication. Le cas de la stratégie de chromaticité (qui correspond à la paramétrisation $\langle 1, 0, 1 \rangle$) est particulièrement intéressant, car la population n'atteint pas une valeur de confiance maximale pour ce défi mais se stabilise à une valeur d'environ 90%. Ce résultat est inattendu, mais s'explique par la combinaison de deux facteurs.

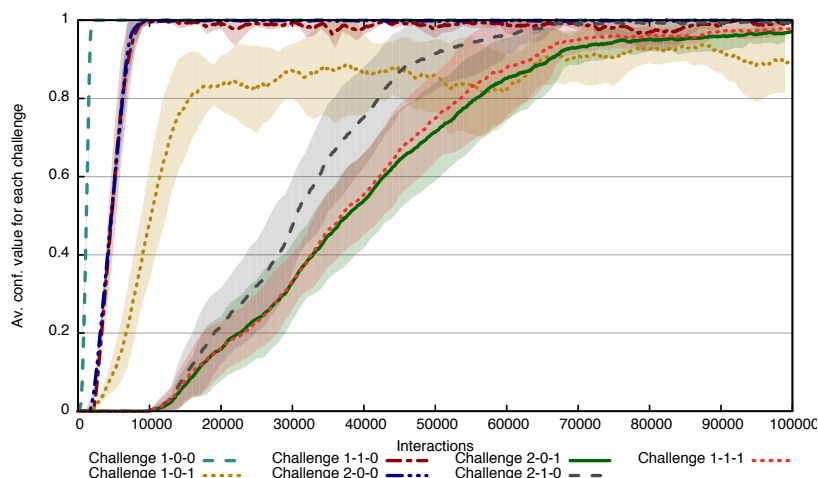


Figure B.13: Évolution de la valeur de confiance moyenne dans la population pour chaque défi.

Premièrement, l'ensemble d'échantillons de couleur que les locuteurs choisissent lorsqu'ils abordent ce défi résulte dans certains cas en contextes trop difficiles, dans la mesure où les couleurs focales anglaises "orange" et "marron" ont des valeurs de h^*_{ab} similaires. Deuxièmement, la mauvaise estimation de la tentative de défi du locuteur par l'auditeur, car la plupart des interactions dans lesquelles les locuteurs ont tenté les défis de type $\langle 2, 0, 1 \rangle$ et $\langle 1, 1, 1 \rangle$ et qui ont pour résultat un échec communicationnel ont été perçus par les auditeurs comme un défi de type $\langle 1, 0, 1 \rangle$.

De plus, le fait de devoir faire face à plusieurs tâches communicationnelles rend plus difficile pour l'auditeur la décision visant à déterminer quel défi doit être mis à jour à la fin d'une interaction, ce qui conduit à un plus grand nombre de mauvaises assignations. Ces erreurs ont également un impact sur la valeur de confiance associée aux défis de niveau 2, en ralentissant le gain de confiance de la population et même en le diminuant de façon récurrente sur de courtes périodes, comme on peut le constater pour la stratégie de luminosité (paramétrisation $\langle 1, 1, 0 \rangle$). Finalement, la population surmonte ces difficultés et termine les simulations par des valeurs de confiance élevées pour tous les défis.

Comme on l'a déjà constaté dans l'expérience précédente, à la fin des simulations la langue résultante ne permet pas à la population d'atteindre un taux de réussite communicative de 100% dans ses interactions linguistiques, mais atteint une valeur stable légèrement inférieure à 90% (plus précisément 88,40%). De la même façon, le succès de l'alignement se stabilise à une valeur légèrement inférieure au succès de la communication (86.00%). Le fait que les deux mesures ne s'ajustent pas au même pourcentage signifie qu'à la fin des simulations la population présente de légères variations dans son lexique et ses prototypes. Ceci est dû au fait que les agents conservent des constructions lexicales et des prototypes qui ne sont presque jamais utilisés. Ces différences sont présentées dans le

Constructions	Valeur	SD (σ)	Prot.	Valeur	SD (σ)
Cxns de couleur	11.56	0.91	Prot. de couleur	11.56	0.91
Cxns chromatiques	4.59	1.09	Prot. de chromaticité	3.51	0.44
Cxns de luminosité	8.52	1.65	Prot. de luminosité	5.95	0.54
Cxns lexicales	24.67	2.42	Prot.	21.02	1.34
Cxns grammaticales	6.96	0.07			

Table B.6: Résultats pour différentes mesures relatives aux constructions et prototypes pour la deuxième expérience dans le domaine des couleurs.

tableau B.6.

Bien que les constructions de couleur et leurs prototypes concordent parfaitement, les résultats montrent un nombre moyen de constructions de luminosité et chromatiques supérieur à celui des prototypes associés. Ce décalage est particulièrement prononcé dans le cas des constructions de luminosité, car la population a en moyenne environ 2.5 constructions de plus par rapport aux prototypes. Cette différence suppose que la langue résultante n'est pas entièrement alignée à la fin des interactions.

Conclusion

Les différentes expériences ont montré que la notion d'auto-motivation est un mécanisme utile pour le développement de systèmes de communication par une population d'agents artificiels. Les expériences menées dans la thèse ont montré que le système de motivation est capable de gérer des tâches de communication de différents niveaux de complexité dans des environnements discrets et continus. Nous avons également montré que ce mécanisme est intéressant pour coordonner plusieurs stratégies linguistiques de même niveau de complexité. En conséquence, le langage commun résultant de chaque expérience permet à la population de communiquer avec succès sur les objets présents dans son environnement.

Bibliography

- [1] David H. Ackley and Michael L. Littman. Altruism in the evolution of communication. In *Artificial life IV*, pages 40–48, 1994.
- [2] Ti Alkire and Carol Rosen. *Romance languages: a historical introduction*. Cambridge University Press, 2010.
- [3] Pierre Andry, Philippe Gaussier, Sorin Moga, Jean-Paul Banquet, and Jacqueline Nadel. Learning and communication via imitation: An autonomous robot perspective. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 31(5):431–442, 2001.
- [4] Pablo Arbelaez, Michael Maire, Charles Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.
- [5] Annalisa Baicchi. Complex adaptive systems: The case of language. In *Construction Learning as a Complex Adaptive System*, pages 9–31. Springer, 2015.
- [6] Gianluca Baldassarre and Marco Mirolli. Deciding which skill to learn when: temporal-difference competence-based intrinsic motivation (td-cb-im). In *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 257–278. Springer Berlin Heidelberg, 2013.
- [7] Gianluca Baldassarre and Marco Mirolli. *Intrinsically motivated learning in natural and artificial systems*. Springer, 2013.
- [8] Adrien Baranès and Pierre-Yves Oudeyer. R-iac: Robust intrinsically motivated exploration and active learning. *IEEE Transactions on Autonomous Mental Development*, 1(3):155–169, 2009.
- [9] Adrien Baranès and Pierre-Yves Oudeyer. Intrinsically motivated goal exploration for active motor learning in robots: a case study. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, Taipei, Taiwan, 2010.
- [10] Adrien Baranès and Pierre-Yves Oudeyer. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73, 2013.

- [11] Andrea Baronchelli, Tao Gong, Andrea Puglisi, and Vittorio Loreto. Modeling the emergence of universality in color naming patterns. *Proceedings of the National Academy of Sciences*, 107(6):2403–2407, 2010.
- [12] Andrea Baronchelli, Vittorio Loreto, and Andrea Puglisi. Individual biases, cultural evolution, and the statistical nature of language universals: The case of colour naming systems. *PLoS ONE*, 10(5):1–19, 05 2015.
- [13] Andrew G. Barto, Satinder Singh, and Nuttapon Chentanez. Intrinsically motivated learning of hierarchical collections of skills. In *Proc. 3rd Int. Conf. Development Learn*, pages 112–119, 2004.
- [14] Clay Beckner, Richard Blythe, Joan Bybee, Morten H. Christiansen, William Croft, Nick C. Ellis, John Holland, Jinyun Ke, Diane Larsen-Freeman, and Tom Schoenemann. Language is a complex adaptive system: Position paper. *Language learning*, 59(s1):1–26, 2009.
- [15] Tony Belpaeme and Joris Bleys. Language, perceptual categories and their interaction: Insights from computational modelling. In *Emergence of Communication and Language*, pages 339–353. Springer, 2007.
- [16] Brent Berlin and Paul Kay. *Basic color terms: Their universality and evolution*. University of California Press, Berkeley, 1969.
- [17] Daniel E. Berlyne. A theory of human curiosity. *British Journal of Psychology. General Section*, 45(3):180–191, 1954.
- [18] Katrien Beuls. *Towards an agent-based tutoring system for Spanish verb conjugation*. PhD thesis, Vrije Universiteit Brussel, 2013.
- [19] Katrien Beuls and Luc Steels. Agent-based models of strategies for the emergence and evolution of grammatical agreement. *PLOS ONE*, 8(3):e58960, 2013.
- [20] Mark H. Bickhard. Scaffolding and self-scaffolding: Central aspects of development. *Children’s development within social context, Vol. 1. Metatheory and theory; Vol. 2. Research and methodology*, 1–2:33–52, 1992.
- [21] Joris Bleys. *Language Strategies for the Domain of Colour*. PhD thesis, Vrije Universiteit Brussel, 2010.
- [22] Joris Bleys. Language strategies for color. In Luc Steels, editor, *Experiments in Cultural Language Evolution*, pages 61 – 85. John Benjamins, 2012.
- [23] Joris Bleys. *Language strategies for the domain of colour*. Language Science Press, 2015.
- [24] Joris Bleys, Martin Loetzsch, Michael Spranger, and Luc Steels. The Grounded Color Naming Game. In *Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication (Roman 2009)*, 2009.

- [25] Joris Bleys and Luc Steels. Linguistic selection of language strategies. In *European Conference on Artificial Life*, pages 150–157. Springer, 2009.
- [26] Edward J. Briscoe. Language as a complex adaptive system: co-evolution of language and of the language acquisition device. In *Proceedings of Eighth Computational Linguistics in the Netherlands Conference*. Citeseer, 1998.
- [27] Ted Briscoe. Co-evolution of language and of the language acquisition device. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 418–427. Association for Computational Linguistics, 1997.
- [28] Richard de Charms. *Personal Causation: The Internal Affective Determinants of Behavior*. New York: Academic Press, 1968.
- [29] Noam Chomsky. *Aspects of the Theory of Syntax*. The MIT Press, Cambridge, 1965.
- [30] Silvia Coradeschi and Alessandro Saffiotti. An introduction to the anchoring problem. *Robotics and Autonomous Systems*, 43(2):85–96, 2003.
- [31] Miquel Cornudella, Thierry Poibeau, and Remi van Trijp. The role of intrinsic motivation in artificial language emergence: a case study on colour. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1646–1656. The COLING 2016 Organizing Committee, 2016.
- [32] Miquel Cornudella, Paul Van Eecke, and Remi van Trijp. How intrinsic motivation can speed up language emergence. In *Proceedings of the European Conference on Artificial Life*, pages 571–578, 2015.
- [33] Leo P. Crespi. Amount of reinforcement and level of performance. *Psychological Review*, 51(6):341, 1944.
- [34] William Croft and D. Alan Cruse. *Cognitive linguistics*. Cambridge University Press, 2004.
- [35] Mihaly Csíkszentmihályi. *Beyond Boredom and Anxiety*. San Francisco: Jossey-Bass, 1975.
- [36] Mihaly Csíkszentmihályi. *Flow: The psychology of optimal experience*. Harper and Row, New York, 1990.
- [37] Joachim De Beule. The emergence of compositionality, hierarchy and recursion in peer-to-peer interactions. In *Proceedings of the 7th international conference on the evolution of language*, pages 75–82. Citeseer, 2008.
- [38] Joachim De Beule, Bart De Vylder, and Tony Belpaeme. A cross-situational learning algorithm for damping homonymy in the guessing game. *Artificial Life X*, pages 466–472, 2006.

- [39] Edward L. Deci and Richard M. Ryan. *Intrinsic Motivation and Self-Determination in Human Behavior*. Springer US, 1985.
- [40] Edward L. Deci and Richard M. Ryan. Motivation, personality, and development within embedded social contexts: An overview of self-determination theory. *The Oxford handbook of human motivation*, pages 85–107, 2012.
- [41] Diane M. Early, Juliette K. Berg, Stacey Alicea, Yajuan Si, J. Lawrence Aber, Richard M. Ryan, and Edward L. Deci. The impact of every classroom, every day on high school student achievement: Results from a school-randomized trial. *Journal of Research on Educational Effectiveness*, 9(1):3–29, 2016.
- [42] Nick C. Ellis. The emergence of language as a complex adaptive system. *The Routledge handbook of applied linguistics*, 666679, 2011.
- [43] Leon Festinger. *A theory of cognitive dissonance*, volume 2. Stanford university press, 1962.
- [44] Charles J. Fillmore. The mechanisms of "construction grammar". In *Annual Meeting of the Berkeley Linguistics Society*, volume 14, pages 35–55, 1988.
- [45] Jingyi Gao and Urmas Sutrop. The basic color terms of mandarin chinese: A theory-driven experimental study. *Studies in Language. International Journal sponsored by the Foundation "Foundations of Language"*, 38(2):335–359, 2014.
- [46] Emilia Garcia Casademont. Insight grammar learning. *Journal of Cognitive Science*, 17(1):27–62, 2016.
- [47] Simon Garrod and Martin J. Pickering. Why is conversation so easy? *Trends in cognitive sciences*, 8(1):8–11, 2004.
- [48] Kateryna Gerasymova and Michael Spranger. An experiment in temporal language learning. *Language Grounding in Robots*, pages 237–254, 2012.
- [49] Adele E. Goldberg. *Constructions: A construction grammar approach to argument structure*. University of Chicago Press, 1995.
- [50] Jacqueline Gottlieb, Manuel Lopes, and Pierre-Yves Oudeyer. Motivated cognition: Neural and computational mechanisms of curiosity, attention, and intrinsic motivation. In *Recent Developments in Neuroscience Research on Human Motivation*, pages 149–172. Emerald Group Publishing Limited, 2016.
- [51] David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jerome Monceaux, Pascal Lafourcade, Brice Marnier, Julien Serre, and Bruno Maisonnier. The nao humanoid: a combination of performance and affordability. *CoRR abs/0807.3223*, 2008.

- [52] Sandra Graham. Theories and principles of motivation. *Handbook of educational psychology*, 4:63–84, 1996.
- [53] Richard Hall. Psychology world — cognitive dissonance. https://web.mst.edu/~psyworld/cognitive_dissonance.htm, 1998. Online; accessed 25-09-2016.
- [54] Harry F. Harlow, Margaret Kuenne Harlow, and Donald R. Meyer. Learning motivated by a manipulation drive. *Journal of Experimental Psychology*, 40(2):228, 1950.
- [55] Greenberg Joseph Harold. *Essays in linguistics*. University of chicago Press, 1957.
- [56] Donald O. Hebb. Drives and the CNS (conceptual nervous system). *Psychological review*, 62(4):243, 1955.
- [57] Fabien Hervouet. *Exploration et structuration intrinsèquement motivées d’espaces d’apprentissage sensorimoteur: contributions théoriques, plate-forme et expérimentations*. PhD thesis, Université Montpellier 2, 2014.
- [58] Fabien Hervouet and Eric Bourreau. Fimo: Framework for intrinsic motivation. In *ECAL: European Conference on Artificial Life*, pages 997–1004, 2013.
- [59] Manfred Hild, Torsten Siedel, Christian Benckendorff, Christian Thiele, and Michael Spranger. Myon, a new humanoid. In *Language grounding in robots*, pages 25–44. Springer, 2012.
- [60] John H. Holland. Echoing emergence: Objectives, rough definitions, and speculations for echo-class models. In *Complexity*, pages 309–342. Perseus Books, 1999.
- [61] Paul Hopper. Emergent grammar. In *Annual Meeting of the Berkeley Linguistics Society*, volume 13, pages 139–157, 1987.
- [62] Clark L. Hull. *Principles of behavior: an introduction to behavior theory*. Appleton-Century, 1943.
- [63] Joseph M. Hunt. Intrinsic motivation and its role in psychological development. In *Nebraska symposium on motivation*, volume 13, pages 189–282, 1965.
- [64] Toru Ishida. Development of a small biped entertainment robot qrio. In *Micro-Nanomechatronics and Human Science, 2004 and The Fourth Symposium Micro-Nanomechatronics for Information-Based Society, 2004. Proceedings of the 2004 International Symposium on*, pages 23–28. IEEE, 2004.
- [65] Eva Jablonka, Marion J. Lamb, and Anna Zeligowski. *Evolution in four dimensions, revised edition: Genetic, epigenetic, behavioral, and symbolic variation in the history of life*. MIT press, 2014.

- [66] Ray Jackendoff. Possible stages in the evolution of the language capacity. *Trends in cognitive sciences*, 3(7):272–279, 1999.
- [67] Kimberly A. Jameson and Nancy Alvarado. Differences in color naming and color salience in vietnamese and english. *Color Research & Application*, 28(2):113–138, 2003.
- [68] Zhang Jin-Yu, Chen Yan, and Huang Xian-Xiang. Edge detection of images based on improved sobel operator and genetic algorithms. In *Image Analysis and Signal Processing, 2009. IASP 2009. International Conference on*, pages 31–35. IEEE, 2009.
- [69] Frédéric Kaplan and Pierre-Yves Oudeyer. The progress-drive hypothesis: an interpretation of early imitation. *Models and mechanisms of imitation and social learning: Behavioural, social and communication dimensions*, pages 361–377, 2007.
- [70] Paul Kay. Construction grammar. *Words and the Grammar of Context*, pages 123–131, 1997.
- [71] Mehdi Keramati and Boris Gutkin. Collecting reward to defend homeostasis: A homeostatic reinforcement learning theory. *bioRxiv*, page 005140, 2014.
- [72] Celeste Kidd, Steven T. Piantadosi, and Richard N. Aslin. The Goldilocks effect: Human infants allocate attention to visual sequences that are neither too simple nor too complex. *PLOS ONE*, 7(5):e36399, 2012.
- [73] Ekkehard Konig and Johan Van der Auwera. *The germanic languages*. Routledge, 2013.
- [74] Johan M. G. Lammens. *A computational model of color perception and color naming*. PhD thesis, State University of New York, 1994.
- [75] Stephen C. Levinson. Spatial language. *Encyclopedia of cognitive science*, 1(1):131–137, 2003.
- [76] Helen Lin, M. Ronnier Luo, Lindsay W. MacDonald, and Arthur W.S. Tarrant. A cross-cultural colour-naming study. part i: Using an unconstrained method. *Color Research & Application*, 26(1):40–60, 2001.
- [77] Martin Loetzsch, Pieter Wellens, Joachim De Beule, Joris Bleys, and Remi Van Trijp. The babel2 manual. Technical report, Tech. Rep. AI-Memo 01-08, AI-Lab VUB, Brussels, 2008.
- [78] Raman Maini and Himanshu Aggarwal. Study and comparison of various image edge detection techniques. *International journal of image processing (IJIP)*, 3(1):1–11, 2009.
- [79] Kathryn Merrick and Mary Lou Maher. Motivated learning from interesting events: adaptive, multitask learning agents for complex environments. *Adaptive Behavior*, 17(1):7–27, 2009.

- [80] Kathryn E. Merrick. *Computational Models of Motivation for Game-Playing Agents*. Springer, 2016.
- [81] Jaime L. Milford, Julia L. Austin, and Jane Ellen Smith. Community reinforcement and the dissemination of evidence-based practice: Implications for public policy. *International Journal of Behavioral Consultation and Therapy*, 3(1):77, 2007.
- [82] Marco Mirolli and Gianluca Baldassarre. Functions and mechanisms of intrinsic motivations. In *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 49–72. Springer, 2013.
- [83] Aleksandra Mojsilovic. A computational model for color naming and describing color composition of images. *IEEE Transactions on Image Processing*, 14(5):690–699, 2005.
- [84] Kay C. Montgomery. The role of the exploratory drive in learning. *Journal of Comparative and Physiological Psychology*, 47(1):60, 1954.
- [85] Omar Khayyam Moore and Alan Ross Anderson. Some principles for the design of clarifying educational environments. *Handbook of socialization theory and research*, 1968.
- [86] Orval Hobart Mowrer and Ray R. Lamoreaux. Avoidance conditioning and signal duration—a study of secondary motivation and reward. *Psychological Monographs*, 54(5):i, 1942.
- [87] Henry W. Nissen. A study of exploratory behavior in the white rat by means of the obstruction method. *The Pedagogical Seminary and Journal of Genetic Psychology*, 37(3):361–376, 1930.
- [88] Stefano Nolfi and Marco Mirolli. *Evolution of Communication and Language in Embodied Agents*. Springer-Verlag Berlin Heidelberg, 1st edition, 2010.
- [89] Pierre-Yves Oudeyer and Frédéric Kaplan. Discovering communication. *Connection Science*, 18(2):189–206, 2006.
- [90] Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:6, 2009.
- [91] Pierre-Yves Oudeyer, Frédéric Kaplan, and Verena V. Hafner. Intrinsic motivation systems for autonomous mental development. *Evolutionary Computation, IEEE Transactions on*, 11(2):265–286, 2007.
- [92] Simon Pauw. *Size matters: Grounding quantifiers in spatial perception*. PhD thesis, Institute for Logic, Language and Computation, 2013.
- [93] Ivan Petrovich Pavlov. *Conditional reflexes: An investigation of the physiological activity of the cerebral cortex*. H. Milford, 1927.

- [94] Roxana Rădulescu. Simulating the shift towards semantic gender in dutch. Master's thesis, Vrije Universiteit Brussel, 2015.
- [95] Debi Roberson, Jules Davidoff, Ian R.L. Davies, and Laura R. Shapiro. The development of color categories in two languages: a longitudinal study. *Journal of Experimental Psychology: General*, 133(4):554, 2004.
- [96] Alan R. Robertson. The cie 1976 color-difference formulae. *Color Research & Application*, 2(1):7–11, 1977.
- [97] Softbank Robotics. Pepper. *Softbank Robotics*, 2016.
- [98] Eleanor H. Rosch. Natural categories. *Cognitive psychology*, 4(3):328–350, 1973.
- [99] Richard M. Ryan and Edward L. Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1):54–67, 2000.
- [100] Richard M. Ryan and Edward L. Deci. A self-determination theory approach to psychotherapy: The motivational basis for effective change. *Canadian Psychology/Psychologie canadienne*, 49(3):186, 2008.
- [101] Olga V. Safuanova and Nina N. Korzh. Mapping into a perceptual color space. *Anthropology of Color: Interdisciplinary multilevel modeling*, page 55, 2007.
- [102] Massimiliano Schembri, Marco Mirolli, and Gianluca Baldassarre. Evolution and learning in an intrinsically motivated reinforcement learning robot. *Advances in Artificial Life*, pages 294–303, 2007.
- [103] Massimiliano Schembri, Marco Mirolli, and Gianluca Baldassarre. Evolving internal reinforcers for an intrinsically motivated reinforcement-learning robot. In *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*, pages 282–287. IEEE, 2007.
- [104] Jürgen Schmidhuber. Adaptive curiosity and adaptive confidence. Technical report, Technical Report FKI-149-91, Institut für Informatik, Technische Universität München, 1991.
- [105] Jürgen Schmidhuber. Curious model-building control systems. In *Neural Networks, 1991. 1991 IEEE International Joint Conference on*, pages 1458–1463. IEEE, 1991.
- [106] Jürgen Schmidhuber. Driven by compression progress: A simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. *CoRR*, abs/0812.4360, 2008.
- [107] Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

- [108] William Schueller and Pierre-Yves Oudeyer. Active control of complexity growth in naming games: Hearer’s choice. In *EVOLANG 2016*, 2016.
- [109] Marcel Schwob. *Études sur l’argot français*. Editions Allia, 1999.
- [110] Jean Simpson and Arthur W.S. Tarrant. Sex-and age-related differences in colour vocabulary. *Language and speech*, 34(1):57–62, 1991.
- [111] Satinder Singh, Andrew G. Barto, and Nuttapong Chentanez. Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 1281–1288, 2005.
- [112] Satinder Singh, Richard L. Lewis, Andrew G. Barto, and Jonathan Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development*, 2(2):70–82, 2010.
- [113] Kirk Skaugen. The game changer. <https://blogs.intel.com/technology/2015/08/the-game-changer/>, 2015. Accessed: 29-08-2017.
- [114] Burrhus F. Skinner. *The Behavior of Organisms: An Experimental Analysis*. The Century Psychology Series. Appleton-Century-Crofts, 1938.
- [115] Burrhus F. Skinner. *Science and human behavior*. Free Pr. [u.a.], New York, first paperback edition edition, 1953.
- [116] Kenny Smith, Simon Kirby, and Henry Brighton. Iterated learning: A framework for the emergence of language. *Artificial Life*, 9(4):371–386, 2003.
- [117] Linda B. Smith, Susan S. Jones, and Barbara Landau. Naming in young children: A dumb attentional mechanism? *Cognition*, 60(2):143–171, 1996.
- [118] C. Richard Spates, Sherry L. Pagoto, and Alyssa Kalata. A qualitative and quantitative review of behavioral activation treatment of major depressive disorder. *The Behavior Analyst Today*, 7(4):508, 2006.
- [119] Kenneth W. Spence. *Behavior theory and conditioning*. Yale University Press, 1956.
- [120] Michael Spranger. Evolving grounded spatial language strategies. *KI-Künstliche Intelligenz*, 27(2):97–106, 2013.
- [121] Michael Spranger. *The evolution of grounded spatial language*. Language Science Press, 2016.
- [122] Michael Spranger, Martin Loetzsch, and Simon Pauw. Open-ended grounded semantics. In *ECAI: European Conference on Artificial Intelligence*, pages 929–934, 2010.

- [123] Michael Spranger, Simon Pauw, Martin Loetzsch, and Luc Steels. Open-ended procedural semantics. In *Language grounding in robots*, pages 153–172. Springer, 2012.
- [124] Michael Spranger, Jakob Suchan, Mehul Bhatt, and Manfred Eppe. Grounding dynamic spatial relations for embodied (robot) interaction. In *Pacific Rim International Conference on Artificial Intelligence*, pages 958–971. Springer International Publishing, 2014.
- [125] Luc Steels. A self-organizing spatial vocabulary. *Artificial life*, 2(3):319–332, 1995.
- [126] Luc Steels. Self-organizing vocabularies. In *Proceedings of Artificial Life V, Nara, Japan*, pages 179–184, Nara, Japan, 1996.
- [127] Luc Steels. *The Talking Heads Experiment. Volume 1. Words and Meanings*. Laboratorium, Antwerpen, 1999.
- [128] Luc Steels. Language as a complex adaptive system. In *International Conference on Parallel Problem Solving from Nature*, pages 17–26. Springer Berlin Heidelberg, 2000.
- [129] Luc Steels. Evolving grounded communication for robots. *Trends in cognitive sciences*, 7(7):308–312, 2003.
- [130] Luc Steels. Language re-entrance and the ‘inner voice’. *Journal of Consciousness Studies*, 10(4-5):173–185, 2003.
- [131] Luc Steels. The autotelic principle. In I. Fumiya, R. Pfeifer, L. Steels, and K. Kunyoshi, editors, *Embodied Artificial Intelligence*, volume 3139 of *Lecture Notes in AI*, pages 231–242. Springer Verlag, Berlin, 2004.
- [132] Luc Steels. The emergence and evolution of linguistic structure: from lexical to grammatical communication systems. *Connection science*, 17(3-4):213–230, 2005.
- [133] Luc Steels. Regulating the growth of complexity in developmental systems, October 21 2008. US Patent 7,440,927.
- [134] Luc Steels. *Design patterns in fluid construction grammar*, volume 11. John Benjamins Publishing, 2011.
- [135] Luc Steels. Computational issues in fluid construction grammar. *Lecture Notes in Computer Science*, 7249, 2012.
- [136] Luc Steels. *Experiments in Cultural Language Evolution*. Advances in interaction studies. John Benjamins Publishing Company, 2012.
- [137] Luc Steels. Self-organization and selection in cultural language evolution. *Experiments in Cultural Language Evolution*. Amsterdam: John Benjamins, pages 1–40, 2012.

- [138] Luc Steels. Agent-based models for the emergence and evolution of grammar. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 371(1701), 2016.
- [139] Luc Steels. Basics of fluid construction grammar. *Constructions and frames*, 2016.
- [140] Luc Steels and Tony Belpaeme. Coordinating perceptually grounded categories through language: A case study for colour. *Behavioral and brain sciences*, 28(4):469–488, 2005.
- [141] Luc Steels and Joris Bleys. Planning what to say: Second order semantics for fluid construction grammars. In A. Bugarin Diz and J.S. Reyes, editors, *Proceedings of caepia '05. Lecture notes in AI*. Berlin: Springer Verlag, 2005.
- [142] Luc Steels and Emília Garcia Casademont. How to play the syntax game. In *Proceedings of the European Conference on Artificial Life*, pages 479–486, 2015.
- [143] Luc Steels and Martin Loetzsch. Babel: A tool for running experiments on the evolution of language. In *Evolution of Communication and Language in Embodied Agents*, pages 307–313. Springer, 2010.
- [144] Luc Steels and Angus McIntyre. Spatially distributed naming games. *Advances in complex systems*, 1(04):301–323, 1998.
- [145] Luc Steels and Pieter Wellens. Scaffolding language emergence using the autotelic principle. In *Artificial Life, 2007. ALIFE'07. IEEE Symposium on*, pages 325–332. IEEE, 2007.
- [146] Jan Storck, Sepp Hochreiter, and Jürgen Schmidhuber. Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the international conference on artificial neural networks, Paris*, volume 2, pages 159–164, 1995.
- [147] Andrew Stout and Andrew G. Barto. Competence progress intrinsic motivation. In *2010 IEEE 9th International Conference on Development and Learning*, pages 257–262. IEEE, 2010.
- [148] Andrew Stout, George D. Konidaris, and Andrew G. Barto. Intrinsically motivated reinforcement learning: A promising framework for developmental robot learning. Technical report, DTIC Document, 2005.
- [149] Julia Sturges and T.W. Allan Whitfield. Locating basic colours in the munsell space. *Color Research & Application*, 20(6):364–376, 1995.
- [150] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

- [151] Jillian C. Sweeney, Dave Webb, Tim Mazzarol, and Geoffrey N. Soutar. Self-determination theory and word of mouth about energy-saving behaviors: An online experiment. *Psychology & Marketing*, 31(9):698–716, 2014.
- [152] Wouter Van den Broeck. Constraint-based compositional semantics. In *Proceedings of the 7th International Conference on the Evolution of Language (EVLANG 7)*, pages 338–345, 2008.
- [153] Remi Van Trijp. The emergence of semantic roles in fluid construction grammar. In *The evolution of language. proceedings of the 7th international conference (evolang 7)*, pages 346–353. Citeseer, 2008.
- [154] Remi van Trijp. Fitness landscapes in cultural language evolution: a case study on german definite articles. In *Evolution of Language (Evolang-X)*. World Scientific, 2014.
- [155] Bart De Vylder and Karl Tuyls. How to reach linguistic consensus: A proof of convergence for the naming game. *Journal of Theoretical Biology*, 242(4):818 – 831, 2006.
- [156] Pieter Wellens. *Adaptive Strategies in the Emergence of Lexical Systems*. phd, Vrije Universiteit Brussel, Brussels, 2012.
- [157] Pieter Wellens, Martin Loetzsch, and Luc Steels. Flexible word meaning in embodied agents. *Connection Science*, 20(2):173–191, 2008.
- [158] Robert W. White. Motivation reconsidered: the concept of competence. *Psychological review*, 66(5):297, 1959.
- [159] William C. Wimsatt and James R. Griesemer. Reproducing entrenchments to scaffold culture: The central role of development in cultural evolution. *Integrating evolution and development: From theory to practice*, pages 227–323, 2007.
- [160] Ludwig Wittgenstein. *Philosophical Investigations*. Basil Blackwell, Oxford, 1953.
- [161] Peter Zackariasson and Timothy L. Wilson. Paradigm shifts in the video game industry. *Competitiveness Review: An International Business Journal*, 20(2):139–151, 2010.
- [162] Rolf A. Zwaan and Gabriel A. Radvansky. Situation models in language comprehension and memory. *Psychological bulletin*, 123(2):162, 1998.

Résumé

Dans cette thèse nous étudions comment des populations d'agents artificiels peuvent utiliser un système de motivation computationnel particulier, appelé l'autotelic principle, pour réguler leur développement linguistique et les dynamiques qui en résultent au niveau de la population.

Nous proposons d'abord une mise en œuvre concrète de l'autotelic principle. Le noyau de ce système repose sur l'équilibre des défis, des tâches à accomplir afin d'atteindre un objectif, et des compétences, les capacités que le système peut utiliser pour accomplir les différentes tâches. La relation entre les deux éléments n'est pas stable mais se déstabilise régulièrement lorsque de nouvelles compétences sont acquises, ce qui permet au système de tenter des défis de plus grande complexité. Ensuite, nous testons l'utilité de ce système de motivation dans une série d'expériences sur l'évolution du langage.

Nous avons démontré à travers de notre travail que l'autotelic principle peut être utilisé comme un mécanisme général pour réguler la complexité du langage développé de manière autonome en domaines discrets et continus.

Mots Clés

Évolution du langage, modélisation multi-agents, intelligence artificielle, systèmes de motivation.

Abstract

This thesis explores how populations of agents can use a particular motivation system called autotelic principle to regulate their language development and the resulting dynamics at the population level.

We first propose a concrete implementation of the autotelic principle. The core of this system is based on the balance between challenges, tasks to be done to achieve a goal, and skills, the abilities the system can employ to accomplish the different tasks. The relation between the two elements is not steady but regularly becomes destabilised when new skills are learned, which allows the system to attempt challenges of increasing complexity. Then, we test the usefulness of the autotelic principle in a series of language evolution experiments.

We have shown through our work that the autotelic principle can be used as a general mechanism to regulate complexity in language emergence in an autonomous way for discrete and continuous domains.

Keywords

Language Evolution, agent-based modelling, artificial intelligence, motivation systems.