Creativity and AI: A Conceptual Blending approach

Francisco Câmara Pereira

To Ana

Abstract

The main motivating factor behind this book is the appealing, yet often controversial, goal of computational modelling of creativity. A cross-disciplinary study of creativity is a mandatory first step and this will help us construct a synthesis from an AI perspective. From this, we focus on a Model of Concept Invention and, more practically, on a computational system, Divago.

The Model of Concept Invention is built over the expected principles for a creative system. It is formalized, although in a computationally idealistic manner, i.e. its implied complexity prevents it from a feasible implementation. Divago is the partial instantiation of this abstract model and comprises the main technical substance of the book. This system is centered on an implementation of the cognitive linguistics framework of Conceptual Blending, as well as on a mapping algorithm based on Metaphor work.

Divago was subject to experimentation in a range of applications and analyzed according to methodologies that have been proposed with the area of Creativity and AI. Other validation procedures are followed, namely in the comparison to other works and to Conceptual Blending literature.

Acknowledgements

In parallel to the work behind this book, a whole novel could have been written, with many more actors than appears in the final document. The truth is that, without these actors, it would certainly lose much of its value.

My supervisor, Prof. Dr. Amílcar Cardoso, has been a key actor from the first to the last chapter, sometimes as a patient advisor and listener, other times as a friend.

Some colleagues (and friends) had the leading role on occasion, even if they were unaware of it. In fact, a substantial part of this book was a result of talks, collaborations and constructive arguments with Tony Veale, Pablo Gervás, João Leite and Geraint Wiggins.

Also, without the rest of the AILab crew (before and after ECOS), a lot of science and friendship chapters would certainly be irreparably lost. These were Penousal, Paulo, Grilo, Xico, Leonor, Prof. Carlos Bento, Prof. Ernesto Costa, Jorge Tavares, André, Miguel Ferrand and Ana Alves.

Special thanks must be made to Paulo Ribeiro, João Pedro Martins, Bruno Leitão and Bruno Marques who have contributed with work that is also strongly linked to the thesis that originated the book.

A large part of this novel was done within the environs of the Departament de Informatics Engineering of the University of Coimbra, and I must not forget to thank my colleagues there and particularly the staff, Isabel, Paula, Manuela and Márcia, for their support.

The other side of the story, with different scenery and people, starts at home. Ana has given the strongest support and love as only she can give and has frequently drawn from a limitless reservoir of patience. Kali and Sansao have also been patient and supportive, in the way that only cats and dogs can be. Kali has left us during the last phase, and we feel a deep sorrow, for this and future novels would never be the same without her.

Finally, my parents and sister deserve a special mention for the many times that I missed them, especially when they needed me. And for the support they have given

from the beginning.

I must also kindly thank the reviewers of this (and previous) publications for the several constructive and incisive comments that definitely helped improve its potential as a contribution to science.

Table of Contents

Table of Contents **v**

1 Introduction **2**
1.1 Motivations . . . 2
1.2 Overview . . . 3
1.3 Contributions . . . 5
1.4 Structure . . . 6

2 Creativity **8**
2.1 Creativity theories . . . 8
2.1.1 Divergent Production . . . 9
2.1.2 Bisociation . . . 15
2.1.3 The Systems Model . . . 19
2.1.4 Boden's taxonomies . . . 23
2.1.5 Others . . . 25
2.1.6 Synthesis . . . 28
2.2 Computational Creativity . . . 30
2.2.1 Two accounts for characterizing creativity . . . 30
2.2.2 Creative systems . . . 36

3 Working with Concepts **45**
3.1 What is a Concept? . . . 45
3.2 Building concepts . . . 47
3.2.1 Concept Formation . . . 47
3.2.2 Concept Invention . . . 48
3.3 Mixing concepts . . . 49
3.3.1 Conceptual Combination . . . 49
3.3.2 Conceptual Blending . . . 51
3.4 Metaphor and Analogy . . . 65
3.4.1 Structure Mapping Engine . . . 66
3.4.2 Conceptual Scaffolding and Sapper . . . 68
3.4.3 Others . . . 76

4 A Model of Concept Invention **80**

A Top-Down approach 80
4.1 A Creative General Problem Solver 80
4.2 Description of the Model 84
4.3 Discussion 91

5 Divago **95**
5.1 Overview of the Architecture 95
5.2 Knowledge Base 97
5.3 Mapper 108
5.4 Blender 111
5.5 Factory 115
5.6 Constraints 119
5.6.1 Integration 119
5.6.2 Topology 123
5.6.3 Pattern Completion 124
5.6.4 Maximization/Intensification of Vital Relations 126
5.6.5 Unpacking 127
5.6.6 Web 129
5.6.7 Relevance 130
5.7 Elaboration 131
5.8 Divago as a creative system 134

6 Experiments **136**
6.1 The Boat-House 138
6.2 The Horse-Bird 145
6.2.1 Evaluating the Optimality Pressures 147
6.2.2 Finding the pegasus 150
6.3 Noun-noun combinations 158
6.3.1 Tuning 162
6.3.2 Free generation 165
6.3.3 Comparison to $\mathcal{C}^3$ 170
6.4 The creature generation experiment 172
6.4.1 Tuning 172
6.4.2 Free generation 176
6.5 The established Blending examples 183
6.5.1 Experiments with isolated principles 187
6.5.2 Experiments with combination of principles 193
6.5.3 Some considerations 197
6.6 Discussion 197

7 Conclusions and Future Directions **204**

A The effect of input knowledge **211**

B Established examples of Conceptual Blending **214**

C The Generalized Upper Model Hierarchy **223**

D Programming the frames **226**
D.1 Syntax and Overview 226
D.2 Programming of the frame 227
D.3 Examples 229

E Instances in the house-boat experiment **235**

F Experiments, Databases and other documents **238**

Bibliography **240**

f

Chapter 1

Introduction

> It is obvious that invention or discovery, be it in mathematics or anywhere else, takes place by combining ideas... (Hadamard)

> Pour inventer il faut penser à côté (Souriau)

> The useful combinations [of ideas] are precisely the most beautiful. (Poincaré)

1.1 Motivations

Right from the start, the main focus in AI research has always been with the issue of problem solving. Seen from this point of view, intelligence corresponds to the ability to solve (complex) problems, from the accurate autonomous movement of a robot arm to the understanding of a natural language sentence. The *classical* setting is that of a search in a *space of solutions* for the problem, where an *intelligent agent* looks for the best choices. A commonly used analogy is of this agent travelling in a *search space* with mountains (highly valued solutions), valleys (bad solutions), islands (mountains surrounded by valleys) and plains (areas where quality of solution hardly changes)[1].

One of the most common criticisms made of Artificial Intelligence methods of problem solving is their limited ability to deal with situations not predicted in the specification. The search space is normally strictly defined, however flexible, complex and adaptable the system seems to be. When facing a problem with no satisfactory

[1]Technically, the solution space can be seen simply as the set of solutions to the problem. The search space corresponds to the *ordered* set of solutions. This ordering can be given by heuristics, distance to the initial situation, or any other criterion that can drive the search.

solution in its search space, an AI system simply returns, at best, the least unsuccessful result that exists in that search space- even when the solution is achievable via the simplest operation of changing perspective, relaxing a constraint or adding a new symbol. In other words, such systems are hardly capable of performing what we normally call *creative behavior*, a fundamental aspect of intelligence.

However, the recognition that there is a deficit of creativity within AI systems does not by itself bring new solutions any more than it reasserts that computers, as we know them, are formal machines that are limited to their closed worlds. The question therefore arises about what can be done to make them more creative or even if, with the current computational architectures, that is possible at all. To some extent, some of the current state-of-the-art paradigms (such as Evolutionary Computation, Multi-agent Systems or Case Based Reasoning) have been responsible for many of the developments regarding the first part of the question. Indeed, we have been developing less rigid systems in previous years and sometimes even producing striking results. Nevertheless, when any of these systems finds a situation for which it was *a priori* not specified to solve, it is definitely not able to cope with it.

The second half of the question concerns primarily what the essential components of a creativity model could be and whether these can be present in a formal machine. There is no definitive answer for this, yet we can allow ourselves cross-fertilization from other areas, such as Psychology, behavior Linguistics, behavior Science and Philosophy, in the speculation and building of a possible solution.

The relationship between Intelligence and Creativity poses further questions. Are these two independent properties of cognition or, on the contrary, are they interrelated and inseparable? More specifically, if taking a traditional AI perspective: isn't creativity about search? Is it a different approach to intelligence?

These questions are present throughout this book, which is an attempt to approach them according to a perspective that, while centered on Computer Science and AI, also lifts contributions from other areas.

1.2 Overview

There is general agreement that the ability to find relations between apparently unrelated knowledge is a creative activity. As can be found in many studies from the area of behavior psychology, the creative faculties of the human mind are hugely dependent on the ability to search through spaces or "viewpoints" that are different from the ones immediately or more obviously involved. For example, according to [Marin et al., 1991], our capacities of abstraction, symbolic analysis, of finding less-obvious relations, among others, are associated to creative production. Indeed, many

important discoveries, great music compositions or paintings were reportedly achieved at times of wandering in domains not directly related to the actual problem (e.g. the famous dream of Kekulé, the discoverer of the structure of the Benzene molecule, who was dozing by the fire and dreaming of self-biting snakes when he made his major discovery [Boden, 1990]). One of these psychology theories [Guilford, 1967] concentrates on the idea of *divergent thinking*. Arthur Koestler [Koestler, 1964] also wrote about a related phenomenon, naming it *bisociation*. From the computer science point of view, the modelling of divergent thinking and bisociation seems extremely difficult mainly because it is far from formally specified and, even if it was, it would certainly demand behavior capacities that are still not achievable by computers. Yet, this does not mean that it is impossible to build models, perhaps less ambitious ones, that are capable of achieving a smaller degree of divergence, in which a computer is able to reason in a multi-domain knowledge base, eventually solving problems via transferring knowledge from different domains. Since different domains will contain different knowledge and possibly different symbols and representations, a model for reasoning in a multi-domain environment must have *translation* mechanisms, so that the transferred knowledge will still have meaning in the new context. There are well known behavior mechanisms that establish cross-domain relationships, namely Metaphor and Analogy, which have been studied to some depth within AI, and which are certainly good candidates for plausible cross-domain transfer.

A perfect cross-domain transfer mechanism will be futile if the new knowledge is not integrated into its novel context in a meaningful way. This integration demands processes and principles able to generate knowledge structures that can be considered as a whole rather than the sum of its parts. In other words, the transfer of new knowledge should not be condemned to result in a pastiche or a concatenation of the parts, instead an emergence of new structure, function or behavior is to be favoured. Two research trends from behavior Science aim to solve this problem, namely Conceptual Combination and Conceptual Blending (also known as Conceptual Integration). The former traditionally deals with the understanding of linguistic combinations (such as "pet fish" or "mountain stream") while the latter is conceived as a process that can apply across the behavior spectrum in general. Despite their differences, they both share the intent of understanding the behavior ability of integrating knowledge from distinct sources. Both have already been subject to computational modelling.

Finally, the unavoidable question of evaluation could justify a research programme on its own, with worries regarding expertise, intentionality, complexity, aesthetic judgement, constraint satisfaction and novelty, to name only a few topics. In the current context, the evaluation should be primarily concerned with whether the just created knowledge structures are worth considering for further use and treatment within the domain it was designed for. In other words, if it is both *novel* and *useful*

within this domain. The computational approach to novelty assessment has been based on similarity metrics or clustering techniques while determining usefulness is normally done via application of rules or checking constraint satisfaction. Conceptual Blending proposes a set of generic *Optimality Constraints* that aim to govern the generation of a blend. However, these are not explained formally, raising the challenge of their computational modelling.

We have just summarized some of the components for a *Model of Concept Invention* from cross-domain transfer. By concept invention, we mean the generation and addition of a new concept (or knowledge structure) to a domain in which this new concept could not be obtained by any internal means (e.g. deduction) and which can still be accepted as a valid concept for the domain. For example, before the invention of the airplane, the domain of "transport means" did not have the entire knowledge to lead to it. It was necessary to observe physical laws that were not taken into account for any other previous means of transport (even the balloon) in order to create different concepts of propulsion and sustaining structure.

1.3 Contributions

The main expected contributions of this book are:

- A reflection, overview and state-of-the-art survey about creativity research, according to different perspectives such as Philosophy, Psychology, behavior Science and Computer Science;
- A formally specified Model of Concept Invention, based on processes and principles that are coherent with the current research on creativity;
- An implemented system, Divago, which partially instantiates the Model of Concept Invention. Divago was applied to different domains and demonstrated to be capable of generating results that pass the criteria of creativity assessment used.
- A Computational Model of Conceptual Blending, which will become integrated within Divago. This is the first computational approach to Conceptual Blending [Fauconnier and Turner, 1998] that includes all the fundamental aspects of this framework.
- An assessment of the creativity of the results and of the system. We analyze the creativity of Divago with the frameworks suggested by Ritchie [Ritchie, 2001], Wiggins [Wiggins, 2001, Wiggins, 2003] and Colton et al [Colton et al., 2001].

1.4 Structure

The remainder of this book is structured as follows:

- **Chapter 2** is about research on creativity. It provides the necessary background regarding theories of creativity, computational approaches to creativity and frameworks for creativity assessment. In this chapter, the reader will also find the generic guidelines that underlie the rest of the book, namely at the level of the Model of Concept Invention (chapter 4) and of assessment of the results of Divago (chapter 6).

- **Chapter 3** starts by defining what a *concept* is in the context of our work. It also defines *concept invention* and compares it with *concept formation*, two kinds of concept building processes. Working with concepts is fundamental for this book, and specifically the framework of Conceptual Blending, which is also presented in this chapter. Conceptual Combination, a related area, is also presented, with particular focus to C^3, a system that will later on (in chapter 6) be compared to Divago. The chapter ends with an overview of computational approaches to Metaphor and Analogy, which deal with concept networks and from which we developed a part of Divago (the Mapper). After this chapter, the reader will have obtained a first insight on the practical aspects involved in this work (in chapter 5) and a clearer impression of the necessary notions regarding concepts.

- **Chapter 4** is dedicated to the description and formalization of the Model of Concept Invention. There, the reader will find a theoretical model, in the sense that it has not been totally implemented or specified up to the detail of computational implementation. This model provides a set of modules that we argue should be present in a system that is meant to invent concepts.

- **Chapter 5** describes Divago in detail, a system that partially implements the model presented in chapter 4. This description will take into account the modules of that model (with redefinition of the formalizations when necessary) and the framework of Conceptual Blending, which is the basis for the *bisociation mechanism* of Divago. After finishing this chapter, the reader will know Divago in depth, namely its knowledge representation, search mechanism and blending model.

- **Chapter 6** is dedicated to the experiments made with Divago. We show the five different experiments made: house-boat, for analysis of the search space; horse-bird, for the study of behavior of Divago with regard to the *Optimality*

Constraints; noun-noun, for the generation of noun noun combinations and comparison to C^3; creatures generation, for the testing of Divago as an engine for generating concepts in a game environment; and established blending examples, for the validation of the Blending model implemented in Divago. The reader will get an idea of the behavior of the system within these different situations, namely with attention to the *novelty* and *usefulness* of the results. Throughout this chapter, we will analyze the system according to the frameworks of Ritchie [Ritchie, 2001] and Colton et al [Colton et al., 2001].

- **Chapter 7** is dedicated to the final conclusions and discussion of future directions to take. There, the interested reader will find a multitude of research directions, some related to the generic aspects of creativity, concept invention and Blending, some more specifically directed towards the further developments of Divago.

Chapter 2

Creativity

In the first half of this chapter, we present approaches to creativity within Psychology, Philosophy, Cognitive Science and AI. The second half is specifically dedicated to the area of computational creativity, where we will show the state-of-the-art both at the level of the theoretical foundations and at the level of implementations.

2.1 Creativity theories

Creativity has been the motivation for many lines of writing throughout human history, for it is such an appealing and mysterious aspect of our existence. However, it is also noticeable that its study, from a scientific perspective, has been neglected until the second half of the twentieth century [Albert and Runco, 1999]. The early twentieth century scientific schools of psychology, such as structuralism, functionalism and behaviorism, devoted practically no resources at all to the study of creativity [Sternberg and Lubart, 1999]. The oft cited foremost turning point was when Joy Paul Guilford, in his *American Psychological Association* presidential address, challenged psychologists to pay attention to what he found to be a neglected but extremely important attribute, namely, creativity [Guilford, 1950]. The so called first *golden age* of creativity then took place, with many newly founded research institutions. However this revolution did not last for long. In fact, from 1975 to 1994, only about 0.5% of the articles indexed in *Psychological Abstracts* concerned creativity [Sternberg and Lubart, 1999]. Today, it seems that the subject has gained another burst of output(the second *golden age*). Indeed, unprecedented resources are being directed towards creativity research in many areas.

In the following sections, the reader will be introduced to some of the works on creativity that influenced this book. These works come from the areas of Psychology

(section 2.1.1), Philosophy (section 2.1.2) and Cognitive Science (2.1.4). Without having a direct influence on our own work, the contribution of Csikszentmihalyi is also presented in section 2.1.3 for three special reasons: it is often cited in works of Creativity and AI[1]; it is also a respected and referential work within the area of Psychology; it reasserts some of the conclusions given by the previous sections, attesting their current acceptance. In section 2.1.5, we will complete the state-of-the-art of creativity with an overview of other works. Finally, a synthesis will be made in section 2.1.6, with particular emphasis on the aspects relevant to this book.

2.1.1 Divergent Production

Until J. P. Guilford introduced the operation of *divergent production* in his *Structure of Intellect*(SOI), creativity was generally considered a phenomenon separated from intelligence, a *state of mind* that was common for those considered gifted and a blessing for those we perceive of as lucky. Until Guilford, the prominent works could be roughly summarized to three: Catherine Cox ([Cox, 1926]), who argued that creativity was a complex, multivariate behavior (as opposed to a single ability or trait)[2]; Helmholtz [Helmholtz, 1896] and Wallas [Wallas, 1926], the latter two being the creators of the four steps model[3] (preparation, incubation, illumination and verification) that became the classical stance, within Psychology, of what a creativity model should involve. This model is not contradicted by Guilford and still concurs with many current views of the subject. In section 2.1.3, we will take a closer look to this model. For the moment, we are interested in giving the reader a short overview of SOI, with particular attention to divergent production, the operation most linked with creative production.

The major aim of SOI was to give to "the concept of 'intelligence' a firm, comprehensive, and systematic theoretical foundation" [Guilford, 1967]. This very ambitious goal must be viewed from a historical perspective: during the first part of the twentieth century, many measuring tests of mental ability appeared, often motivated by the need to quantify "intelligence". This need was increased by the advents of the first and second world wars, when fast and effective processes of selection were fundamental for recruitment (mainly in areas such as the air force or intelligence services). The overstated relevance of testing the concept of intelligence justified the sarcastic sentence of E. G. Boring: "... intelligence as a measurable capacity must at the start

[1]Particularly in multi-agent systems approaches.
[2]As cited in [Martindale, 1999]
[3]As cited in [Albert and Runco, 1999]

be defined as the capacity to do well in an intelligence test" [Boring, 1923]. Guilford himself, who also made significant contributions to this area of psychometry, pointed out the lack of a coherent psychological theory behind tests in general, this becoming the general motivation for the SOI. His more specific intentions were to provide SOI as a frame of reference for the study of the "intellectual factors". An intellectual factor corresponds to an aspect of intelligence represented by a triple *operation/product/content* (see figure 2.1).

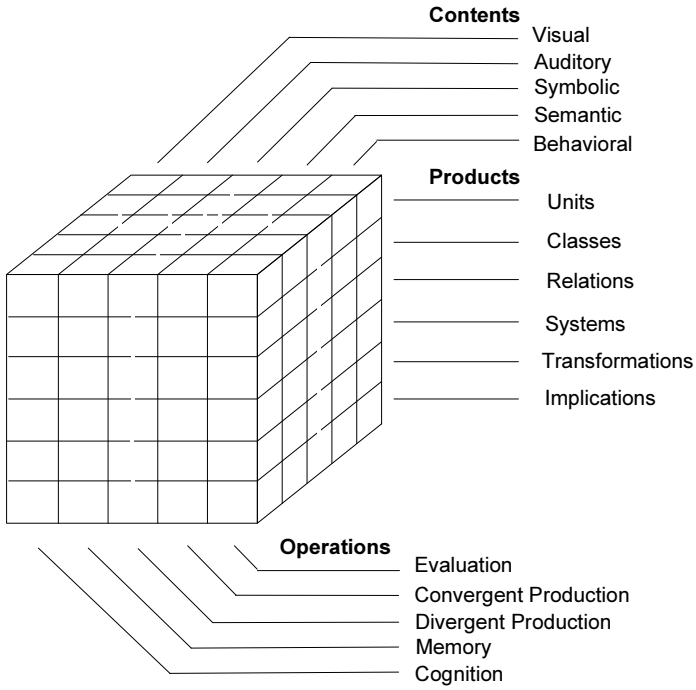

Figure 2.1: The Structure of Intellect (SOI) (from [Guilford, 1967]).

For each factor, Guilford proposes tests, the majority of them implying correlations of many factors. For example, for cognition of symbolic classes (CSC), he suggests tests like:

> Which pair of numbers does not belong with the others?
>
> A. 1-5
>
> B. 2-6
>
> C. 5-8
>
> D. 3-7
>
> *answer:C* (in all other cases the difference is 4)

This test has correlations with CSR (cognition of symbolic relations) and CMS (cognition of semantic systems). With SOI, there would be no single value to represent

the intelligence of a subject, instead a set of values would represent his/her main intellectual qualities and defects.

Perhaps the major contribution of SOI to the area of psychology and, more specifically, to the notion of intelligence has to do with the demonstration, supported by a large variety and quantity of empirical work, that intelligence is not monolithic: there is a multitude of factors to take into account and one cannot find a unique and absolute measure (as Catherine Cox had claimed before). However uncontroversial and obvious this may seem today, the fact is that only during the last decades of the twentieth century was there general acceptance of the idea that single measures like the IQ are very fragile indicators for peoples's behavior and abilities. Another important contribution, the one that most interests us, has to do with the inclusion of creativity as a fundamental aspect of intelligence. More specifically, Guilford considers *creative production* as a general ability that humans have, and which depends on many different intellectual factors, but, most of all, on an operation: that of *divergent production* (DP). His formal definition of divergent production reads: "generation of information from given information, where the emphasis is upon variety and quantity of output from the same source; likely to involve transfer." [Guilford, 1967]. DP composes four fundamental abilities:

- fluency - generation of a large number of solutions for a problem
- flexibility - generation of varied solutions
- originality - generation of solutions that are: rare within the population; remotely related; *clever* responses
- elaboration - ability to think of details

As with the rest of SOI, Guilford proposes a series of tests. In DP-tests, subjects are asked to exhibit evidence of divergent production in several areas, including that of semantic units (e.g. listing consequences of people no longer needing to sleep), of figural classes (finding as many classifications of sets of figures as is possible), and of figural units (taking a simple shape such as a circle and elaborating upon it as much as possible). For example, the following test should measure divergent production of semantic classes (DMC):

From the list of words to the left, make some small subclasses of objects:

1.	arrow	
		alternate classes
2.	bee	
		1,2,5,7 (found in the air)
3.	crocodile	
		3,4,6 (found in the water)
4.	fish	
		2,3,4,7 (animals)
5.	kite	
		3,4,5,7 (have tails)
6.	sailboat	
		etc.
7.	sparrow	

From these tests and reflections on the whole model, Guilford also proposes another concept as fundamental to creativity, that of *transfer recall*: "Things are recalled in connection with cues with which they were not experienced before. Transfer recall is retrieval of information instigated by cues in connection with which the information was not committed to memory storage. "[Guilford, 1967]. In other words, transfer recall is the operation that allows knowledge in memory, however semantically distant and apparently unrelated to the problem at hand, to be brought and applied to a current situation. This is what we call *cross-domain transfer* throughout this book.

To summarize, the operation of divergent production is the very basis for the set of phenomena that are commonly associated with creativity in people, although, as Guilford himself points out,

> '...creative potential is not a single variable, any more than intelligence. Creative performances in daily life are enormously varied in the demands that they make on intellectual resources. The performances singled out for their more obvious signs of creativity - novelty, ingenuity, inventiveness - probably involve one or more divergent production abilities as key aspects, or transformation abilities, outside the DP-operation category as well as within it.'

Although creativity is normally linked more to free-association, unconstrained reasoning or unexpectedness than to method, constraint satisfaction or inference, it has been clear from reading many studies (many described or referred to within this document) that a great deal of mastery of knowledge, expertise within a domain and focus is fundamental. Thus, although not so much emphasized by Guilford, the converse operation of DP, *convergent production* (CP), is also fundamental (as pointed out by Csikszentmihalyi, in section 2.1.3), since it provides deductive reasoning or compelling inferences. "Convergent production rather than divergent production is the prevailing function when the input information is sufficient to determine a unique

answer." SOI tests for evaluating convergent production essentially measure the ability to solve puzzles, equations, classification tasks and problems in general that yield a logically sound unique solution.

Guilford makes a thorough comparison between DP and CP:

> '[In DP], the problem itself may be loose and broad in the requirements for solutions; or the problem, if properly structured, may call for a unique solution, but the individual may have an incomplete grasp of it; or he may comprehend the problem fully, but he is unable to find the unique answer immediately.(..) In CP, an answer can be rigorously structured and is so structured and an answer is forthcoming without much hesitation. In the former, restrictions are few; in the latter they are many; in the former, the search is broad; in the latter it is narrow. In the former, criteria for success are vague and somewhat lax and may, indeed, stress variety and quantity; in the latter, criteria are sharper, more rigorous, and demanding.'

Thus, according to Guilford, CP and DP are two complementary facets of our productive capacity. This capacity, along with cognition (which he considers a more specific operation: that of comprehension and understanding), memory and evaluation, make part of a model of *problem solving* and *creative production* that the author proposes as an operational integration of all the aspects of SOI. Although this model is essentially a speculation, it is interesting to reproduce the original diagram to the reader (figure 2.2).

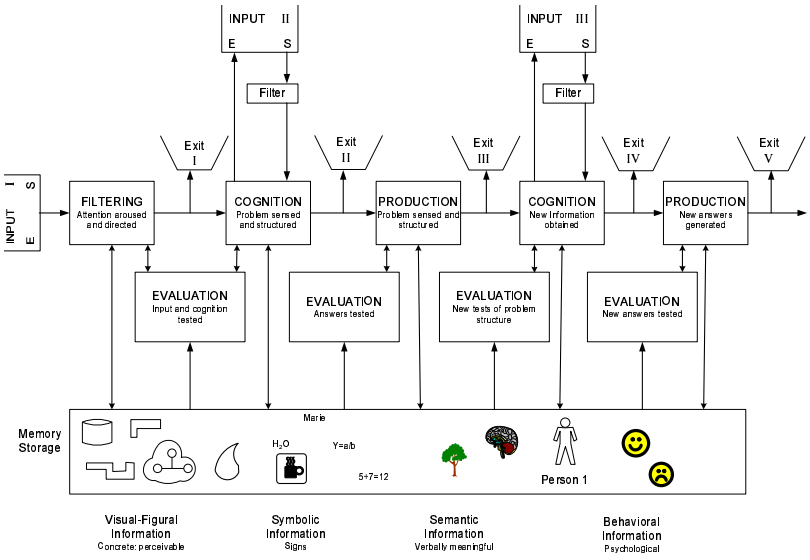

Figure 2.2: Guilford's model of *problem solving* and *creative production* (from [Guilford, 1967], pp 315).

It is far outside the scope of this book to present this model in detail. Since Guilford did not explore it deeply himself, it is an abstract suggestion for how things should be when solving problems. Two aspects, however, should be retained: Guilford argued for problem solving and creative production being the same, thus building a common model for both; he considers a heterogenous memory with many kinds of representation, perspectives, domains, all cohabiting together, in an organized whole. This is what we call *multi-domain environment* throughout this book.

We would like to finish this section with some thoughts about Guilford's work, taking into account, obviously, that this is a work that is almost 40 years old. The first issue is the supremacy of verbal versus non-verbal representation, which sometimes seems to imply that thought is defined by (verbal) language and not the opposite. Allowing some speculation from our side, we believe this is due to the dualist tendency of the time, where mind is detached from matter, as opposed to the current view, of embodiment, where some cognitive scientists see consciousness and mind as a result of the interaction of cognition with the whole physical experience. In other words, today, Guilford's symbolic and semantic contents would be connected as much with verbal symbols as with any other kinds of symbols, such as social or behavioral symbol systems (e.g. politeness rules, traffic lights).

Another criticism relates to the vagueness of some definitions, such as *cleverness* or *generation of logical possibilities*. These being related to divergent production, it is very important to clarify them. Here too, we must allow ourselves some speculation bounded by context. By *cleverness*, or clever solutions to a problem, the author means solutions that both respond more effectively to a problem (than the usual, convergent, ones) and are rare to find in a search space. When defining divergent production (as opposed to convergent) as being with the ability to generate logical possibilities, Guilford meant the creation of unsound logical facts (or rules) that, however, are not revealed to be inconsistent with the rest of the knowledge (e.g. facts that are not deducible, but do not contradict existing knowledge). Here, too, we would like to add that sometimes inconsistencies do arise and creativity comes out of the confrontation between the inconsistencies and the theory (e.g. Kepler's discoveries about elliptic versus circular orbits[4]). Guilford would certainly agree with this opinion.

In conclusion, the Structure of Intellect is now certainly outdated, and its contribution is now seen from a historical perspective. However, with regard to the psychology of creativity, Guilford's legacy about divergent and convergent production is still a constant reference. For us, it has become a modelling inspiration.

[4]This example will be further explained in the next section.

2.1.2 Bisociation

Before going into further detail about Arthur Koestler's work, we would like to add that *The Act of Creation* is a rich philosophical and psychological reference that encompasses ideas that are still currently accepted and explored, as we can see in current trends such as *embodiment* or *conceptual blending*. In many ways, bisociation prefigures the blending framework. Indeed, we might even ask in what way is blending merely "parameterized bisociation", that is, bisociation with more elaborate principles and constraints. This will be discussed in due course.

Written during the early sixties, when behaviorism was the dominating trend within psychology, this book takes the opposite position (which is close to a structuralist view) and aims to explain the act of creativity, tackling this challenge from several different perspectives. While it certainly misses many aspects and perhaps fails in depth to favor the breadth, it proposes a set of ideas that we will try to synthesize here and which will be taken by us for the sake of argument of some of our options.

In *The Act of Creation*, Arthur Koestler presents a theory that unifies three sides of human behavior commonly deemed creative: humor, science and the arts. According to him, the underlying processes are the same, but applied in different contexts, subject to different intentions and perspectives. In order to support his theory, Koestler proposes a set of definitions regarding knowledge and problem solving, namely *matrices of thought*, *codes of rules* and *strategies*.

A matrix of thought (or simply, a matrix) is "any ability, habit, or skill, any pattern of ordered behavior governed by a 'code' of fixed rules" [Koestler, 1964]. In his example of chess playing (as an ability), he proposes a matrix as being "the pattern before you, representing the ensemble of permissible moves. The code which governs the matrix can be put into simple mechanical equations which contain the essence of the pattern in a compressed, 'coded' form". A code of rules is then what defines the matrix, which means that both represent the same entity from different perspectives, one intensional, the other extensional. A strategy corresponds to the selection of elements within the matrix in order to achieve a goal or pattern of behavior. In the case of chess, this would be the choice of the "next move".

We find several obvious correlations with AI problem solving, namely a *matrix* corresponds to the set of all possible solutions for a given problem (the solution space), defined by the *code of rules*, a set of constraints that define what a valid solution must be like. The strategy is then the search procedure, the method used to choose solutions.

According to Koestler, the creative process is connected to what he terms *bisociation* of matrices, a phenomenon that occurs when two (or more) matrices become intersected: a reasoning is being followed in one of the matrices and, for some reason (e.g. external stimulus, need, dream, or trance-like state), a "clash" happens with another matrix and there is a leap to an alternate reality.

Humor

In humor, for example, bisociations introduce the sudden association to the unexpected, the illogical that triggers laughter, sometimes via double-meaning, phonetics, caricature, satire, to name but a few. Let us quote a short example:

> Two women meet while shopping at the supermarket in the Bronx. One looks cheerful, the other depressed. The cheerful one inquires:
> 'What's eating you?'
> 'Nothing's eating me'
> 'Death in the family?'
> 'No, God forbid!'
> 'Worried about money?'
> 'No...nothing like that.'
> 'Trouble with the kids?'
> 'Well, if you must know, it's my little Jimmy.'
> 'What's wrong with him, then?'
> 'Nothing is wrong. His teacher said he must see a psychiatrist.'
> Pause. 'Well, well, what's wrong with seeing a psychiatrist?'
> 'Nothing is wrong. The psychiatrist said he's got an Oedipus complex.'
> Pause. 'Well, well, Oedipus or Shmoedipus, I wouldn't worry so long as he's a good boy and loves his mamma.'

Here, we see a clash between the matrices of Freudian psychiatry pathologies and the logic of common sense: if Jimmy is a good boy and loves his mamma, there can't be much wrong. Koestler even arrives to an extreme claim that "any two matrices can be made to yield a comic effect of sorts, by finding an appropriate link between them and infusing a drop of adrenalin" [Koestler, 1964].

Science

In his extensive analysis of scientific discovery throughout the history of science, Koestler points out several observations that converge to the idea that "each basic

advance was effected by a more or less abrupt and dramatic change; the breaking down of frontiers between related territories, the amalgamation of previously separated frames of reference or experimental techniques (..) All decisive advances in the history of scientific thought can be described in terms of mental cross-fertilization between different disciplines" [Koestler, 1964]. In other words, all the major advances observed resulted from bisociative thinking and these include examples like the discoveries from Archimedes, Copernicus, Kepler, Galileo, Darwin, Poincaré, Kekulé, Einstein, to name a few. In summary, in each of these situations, the knowledge has so far proved inapplicable; none of the various ways of exercising a skill, however flexible and adaptable, has led to the desired goal. The solution came out of a new synthesis of previously unconnected matrices of thought; a synthesis arrived at by "thinking aside". For example, Kepler's laws of planetary motion represent the first synthesis of astronomy and physics which, during the preceding two thousand years, had developed along separate lines. Kepler served his apprenticeship under Tycho de Brahe, who had improved the astronomy observation instruments and methodology, thereby allowing hitherto unequalled abundance and precision. Given the new data, there were clear inconsistencies in the traditional astronomy predictions, mainly because they were based on very entrenched dogmas such as that "all heavenly motion must be uniform and in perfect circles". By retaining much of the metaphysical and theological basis that Kepler himself believed in, he was able to postulate the existence of a physical force acting between the sun and the planets, thus leading to a revolution in astronomy. Planets no longer move in circles, but in elliptic orbits.

Koestler also points out the fundamental aspects of the *ripeness* of the discoverer, i.e. he must be prepared, predisposed to the discovery[5], and have the ability to find hidden analogies (i.e. to find relations where no one has found them before) within different matrices. Another interesting observation is that "verbal thinking plays only a subordinate part in the decisive phase of the creative act (..) as the creative process of discovery depends on unconscious resources and presupposes a regression to modes of ideation which are indifferent to the rules of verbal logic". The conclusion is that words are essential, but sometimes become snares, decoys, or strait-jackets.

An interesting quotation comes from the mathematician Henri Poincaré:

> Among chosen combinations the most fertile will often be those formed of elements drawn from domains which are far apart... Most combinations so formed would be entirely sterile; but certain among them, very rare, are the most fruitful of all.

[5]There are countless examples in which the solution appeared but the scientist wasn't able to understand it or to see it as such.

Arts

The third perspective that Koestler analyzes is that of the artist. As with humor, he starts by focussing on the physical manifestations connected to it, the many sensuous phenomena we feel in moments of self-transcendence, from goose bumps to weeping, thereby arriving at the *emotive potential* of a matrix, with its capacity to generate and satisfy participatory emotions (e.g. by identification, aggression, tension, relaxation). Perceiving a form of art, of deluding oneself without losing track of reality, means exploring this matrix with higher emotive potential, moving from the trivial present to a plane remote from self-interest while forgetting current preoccupations and anxieties: "The capacity to regress, more or less at will, to the games of the underground, without losing contact with surface, seems to be the essence of the poetic, and of any other form of creativity". The author thus proposes the act of interpretation as being also bisociative, and thus creative from the point of view of the *recipient*.

As in scientific discovery, metaphor and imagery also come into existence by a process of seeing an analogy where no one saw one before, its aesthetic satisfaction depending on the emotive potential of the matrices involved. According to Koestler, discoveries of art derive from "the sudden transfer of attention from one matrix to another with a higher emotive potential". In other words, as with science, the greatness of an artist rests in creating a new idiom - a novel code which deviates from the conventional rules. The key turning points result from a new departure along a new line, where we can find bisociations in the grand style - cross-fertilization between different periods, cultures, and provinces of knowledge. Once a new idiom is established, "a whole host of pupils and imitators can operate it with varying degrees of strategic skill". Here, Koestler clearly shows his view on "true creativity - the invention of a new recipe" as opposed to "the skilled routine of providing variations for it". This dichotomy also corresponds to the *transformational* and *exploratory* creativity that Margaret Boden discusses, which will be approached in section 2.1.4.

In conclusion, Koestler argues that bisociation is active in those three aspects of human creativity. In humour, by the collision of matrices; in science by their integration; and in arts by their juxtaposition.

Although rarely showing it in a formal or algorithmic fashion, Koestler provides an insight to one of the most definite phenomena behind the creative process, that of combining knowledge from different domains. More than describing in detail what happens in cognition, he identifies the consistent existence of what he calls *bisociation* within a very wide range of situations commonly deemed creative, and while it may be arguable that not every creative act complies with this description, it is certainly

true that many of them result from the association of apparently semantically distant sources, which, for some reason, combine in such a way that novel and useful knowledge emerge so naturally that those sources no longer seem so distant. From the perspective of Guilford's theory, the notion of bisociation takes many intersections with that of transfer recall, although the latter seems reduced to the act of retrieving unexpected elements from memory, and the former to the actual processes of combination involved.

2.1.3 The Systems Model

The work of Mihaly Csikszentmihalyi on Creativity [Csikszentmihalyi, 1996] is the most recent of the ones reviewed in detail here. This does not mean, however, that it has a radically different perspective as we will see. Indeed one of the purposes of looking at this widely referred study is to demonstrate the current validity and acceptance of the works of Koestler and Guilford, as well as to provide some other comments that we believe are of importance to the present work.

From a study that took several years, in which the author and colleagues interviewed and analyzed ninety one people widely deemed creative, Csikszentmihalyi proposes a description of how creativity works and how culture evolves as domains are transformed by the curiosity and dedication of a few individuals. He proposes a *systems model of creativity* that takes into account the interaction of three elements: the *domain*, the *person* and the *field*. The *domain* consists of a set of "symbolic rules and procedures. Mathematics is a domain, or at a finer resolution algebra and number theory can be seen as domains. Domains are in turn nested in what we usually call culture, or the symbolic knowledge shared by a particular society, or by humanity as a whole." [Csikszentmihalyi, 1996] Csikszentmihalyi defines the creative *person* as designating individuals who, like Leonardo, Edison, Einstein or Mozart have changed our culture in some important respect. The *field* includes all the individuals and institutions that act as gatekeepers to the domain, the peers that will judge the person and the ideas. It is the field that selects what new works of art, objects or theories deserve to be recognized, preserved and remembered. Each of these three elements is necessary for a creative idea, product, or discovery to take place.

Thus the author proposes a very strong definition of creativity. Let us call it creativity with a big C: "Creativity is any act, idea, or product that changes an existing domain, or that transforms an existing domain into a new one (..) A creative person is someone whose thoughts or actions change a domain, or establish a new domain (..) So, in a sense, the most momentous creative events are those in which entire new symbolic systems are created." [Csikszentmihalyi, 1996]. In order to understand the

process behind these transformations that shape civilization, Csikszentmihalyi analyzed and interviewed individuals who were able to create or drastically change one domain, at least once in their lives. The list included renowned artists, politicians, economists and scientists, some having been awarded Nobel prizes or other important awards in their field. Although presumably not providing statistical validity (e.g. there is no control group or objective measures in general), the author identified a set of traits that were common among the individuals [Csikszentmihalyi, 1996]:

1. Creative individuals have a great deal of physical energy, but they also know how to be quiet and at rest.

2. Creative individuals tend to be smart, yet also naive at the same time.

3. A third paradoxical trait refers to the related combination of playfulness and discipline, or responsibility and irresponsibility

4. Creative individuals alternate between imagination and fantasy at one end, and a rooted sense of reality at the other. Both are needed to break away from the present without losing touch with the past.

5. Creative people seem to harbor opposite tendencies on the continuum between extroversion and introversion.

6. Creative individuals are also remarkably humble and proud at the same time.

7. In all cultures, men are brought up to be masculine and to disregard and repress those aspects of their temperament that the culture regards as feminine, whereas women are expected to do the opposite. Creative individuals to a certain extent escape this rigid gender role stereotyping.

8. Generally, creative people are thought to be rebellious and independent. Yet it is impossible to be creative without having first internalized a domain of culture (...) hence it is difficult to see how a person can be creative without being both traditional and conservative and at the same time rebellious and iconoclastic.

9. Most creative persons are very passionate about their work, yet they can be extremely objective about it as well.

10. Finally, the openness and sensitivity of creative individuals often exposes them to suffering and pain yet also a to great deal of enjoyment.

On his explorations concerning the creative process itself, Csikszentmihalyi follows the traditional description of five steps (that descends from the four step model of

Wallas, with Verification split into Evaluation and Elaboration): Preparation, Incubation, Insight, Evaluation and Elaboration. He then frames these steps within the systems model of domain, person and field.

As confirmed in each of the individuals studied, there is the need for a tremendous amount of information about the domain and the field, normally achieved after years of hard work. This *preparation* depends as much on external factors (e.g. family, education, socioeconomic factors, political issues within a field) as on internal factors (e.g. curiosity, persistence, talent). The *incubation* is generally described as the process of solving a problem in the *underground of cognition*, after the creative person feels that they have been blocked: "Cognitive theorists believe that ideas, when deprived of conscious direction, follow simple laws of association. They combine more or less randomly, although seemingly irrelevant associations between ideas may occur as a result of a prior connection"[Csikszentmihalyi, 1996]. This mysterious and often controversial step was also confirmed by the individuals studied, who often reported finding the solution to a problem in unpredictable or consciously unprepared situations, sometimes after years of working on the problem (or even having left it alone). As the author points out, this *insight*, also found recurrently in the literature of creativity, is only possible when the person (in fact the whole system) is prepared to identify it. This corresponds to the idea of ripeness, as described by Koestler. For this to happen, the person must be in the right place at the right time, with a significant amount of confidence, knowledge and luck, as frequently confirmed by the interviewed individuals.

Evaluation and *elaboration* are steps that gradually become more dependent on the whole system and less on the individual, as the act, idea, or product is confronted with the domain and the field, although the person also becomes part of this evaluation, particularly in less objective domains.

Two important aspects which raise from Csikszentmihalyi's observations are the duality of divergent/convergent thinking and integration across and within domains, both of which are consistently reported and analyzed. From the observations, creative people are able to perform well (and with constant switches) in both the opposite ways of thinking reported in section 2.1.1: divergent and convergent.

> People who bring about an acceptable novelty in a domain seem able to use well two opposite ways of thinking: the convergent and divergent(..). Divergent thinking is not much use without the ability to tell a good idea from a bad one - and this selectivity involves convergent thinking.

Divergent thinking is extremely important for the phases of Preparation and Incubation, since these phases are characterized by curiosity and searching. Convergent thinking is a determinant for the Elaboration, Evaluation and also Insight, since it brings about the ability to tell a good idea from a bad one, to follow the established rules of the domain and confront them with new knowledge. However, the boundaries must not be so strict. Indeed, the creative process intertwines frequently between divergent and convergent thinking, as well as any of the five steps just described.

In the same way that Guilford's contribution (divergent and convergent thinking) has been accepted and backed up by the work of Csikszentmihalyi, so Koestler's bisociation also appears as "the norm rather than the exception". Although rarely identifying the phenomenon with the name of bisociation, the author repeatedly provides examples and reports of situations that involve the cross-domain transfer of ideas, the bringing together of domains that appear to have nothing in common and integration or synthesis both across and within domains. As he points out, "creativity generally involves crossing the boundaries of domains, so that, for instance, a chemist who adopts quantum mechanics from physics and applies it to molecular bonds can make a more substantive contribution to chemistry than one who stays exclusively within the bounds of chemistry." [Csikszentmihalyi, 1996]

The work just covered suggests many important issues for the modelling of creativity and of creativity supporting tools. From a system's perspective, one should not simply focus on one single element (person, domain or field), for it is from their interaction that creativity emerges. It also raises many traits and conditions that need to be present for the creation of novel and useful ideas. However we must point out that this study lacks many of the scientific bases necessary to assert with much confidence some of these more abstract conclusions, namely because the individuals come from a very specific class: successful, recognized, well established people, in general happy with their own accomplishments, and normally in an advanced phase of their lives (over sixty years old, in general). This means that many creative and non-creative people have been left out, some of whom are unsuccessful so far, struggling to be recognized, yet still be extremely creative. In other words, the set of individuals corresponds to the class of creative persons that the system (person, domain and field) has brought to the world, but the system is too dynamic not to confront data with other systems (e.g. a control group) or other states of the system (e.g. different ages).

Another aspect that Csikszentmihalyi himself has raised is that, according to these criteria, children can be talented but never really creative, because "creativity involves changing a way of doing things, or a way of thinking, and that in turn requires having mastered the old ways of doing or thinking." He thus leaves space

for two other classifications: *personal creativity*, experiencing the world in novel and original ways; and *talent* or *brilliance*, the ability to express unusual thoughts which are interesting and stimulating. Any of these three classifications associated with the word "creativity" (true creativity, personal creativity and talent) provides interesting challenges to computational modelling and also leads us to Boden's taxonomy (h-creativity and p-creativity), thus providing similar analyses with regard to AI and computational creativity.

2.1.4 Boden's taxonomies

Of the works and authors in this section, there is no doubt that Margaret Boden [Boden, 1990] is the most read and cited within the field of AI and Creativity. The simple reason for this is the fact that she pioneered the effort of analyzing some of the work that had been done (in AI) from a perspective that takes into account those philosophical and psychological issues which are traditionally deemed as creative. In so doing, she proposes a set of classifications for analyzing a program (as much as a human). These classifications themselves have raised much debate, some of which we will cover here.

The first classification proposed by Boden concerns the fact that there is novelty in an idea or discovery: Whether it is novel for a single person or for the whole of human history. In principle, every new and useful idea or discovery is creative for its producer. This is called *psychological* creativity (or p-creativity). The other reference is history. When an idea is novel and useful for the whole of human history, then we are faced with *historical* creativity (or h-creativity). This is perhaps the less controversial classification, although many authors argue that true creativity cannot exist without an external evaluation (e.g. [Csikszentmihalyi, 1996], [Lubart, 1999]). Another aspect is that one can never determine h-creativity in absolute terms because an idea can be h-creative and not be seen as such for years. And the reverse also happen, when an idea and an author are regarded as h-creative by the society, but the original idea should actually have been credited to a preceding author. There are countless examples of these misjudgments in human history. This view does not remove the validity of h-creativity as presented, but it testifies to how complex the problem can be. It is also important to mention that p-creativity is the main focus of Boden's analysis, as she is mainly concerned with the personal perspective.

The other classification brought by Boden [Boden, 1990] pertains to the process needed to produce the novel idea or discovery. She thus presents two kinds: *combinatorial* creativity and *exploratory-transformational* creativity. The combinatorial

creativity results from "unusual combination of, or association between, familiar ideas. Poetic imagery, metaphor and analogy fall into this class." [Boden, 1999]. Alternatively, exploratory-transformational creativity (ET-creativity) is about how a subject deals with a *conceptual space*. Her definition of *conceptual space* is: "an accepted style of thinking in a particular domain - for instance, in mathematics or biology, in various kinds of literature, or in the visual or performing arts..." [Boden, 1999]. There is a clear similarity with Koestler's matrices throughout the many descriptions that Boden provides for conceptual space, although neither provide a formal definition. ET-creativity further subdivides into two distinct categories: *exploratory*(e-) and *transformational* (t-). E-creativity deals with the exploration of the conceptual space without jumping out of its boundaries, without breaking strong constraints, and it is normally based on the mere "tweaking" of the most superficial dimensions. Sometimes it is capable of achieving p-creative or even h-creative outcomes, but still without changing the (well defined) conceptual space. T-creativity involves some transformation "of one or more of the (relatively fundamental) dimensions defining the conceptual space concerned" [Boden, 1999]. In other words, it demands changes in the conceptual space, such as re-representation, change in the evaluation or integration of new concepts. Boden also sees this kind of creativity as *impossibilist*, "in that ideas may be generated which - with respect to the particular conceptual space concerned - could not have been generated before (they are made possible by some transformation of the space)." [Boden, 1990]:519-520. Thus, there is a clear opposition between e- and t- creativity, although for some authors, it is about the level of abstraction. Indeed this taxonomy has raised many points of debate.

The first point is that transformational creativity is also exploratory at a meta-level [Wiggins, 2001, Colton, 2001, Ram et al., 1995]. In other words, given the nature of t-creativity, the only possible way to transform a conceptual space is to change its own defining rules. This would involve being aware of its own defining rules, in other words, being able to do meta-level reasoning. Following this argument, it leads us to the conclusion that this change of meta-level rules would necessarily be (at some point, even if at a meta-meta-level, and so on) exploratory. This argument has been formalized in [Wiggins, 2001].

Another criticism concerns the vagueness of the definition of conceptual space ([Wiggins, 2001], [Ritchie, 2001]). Although she provides many examples, such as from within the broad areas of expertise like music, writing or physics, it is never sufficiently clear from a computational modelling perspective what it actually comprises. More specifically, should it correspond to a *solution set*, i.e. the set of solutions to a problem? Is there any ordering, so it becomes then a *search space*? This issue may become important when considering the computational modelling (and analysis) of

t- and e-creativity. For example, changing the ordering of concepts would correspond to a transformation of the search space, but not of the solution set. Since in the latter case there is no introduction of new concepts, one cannot say it could not have been generated before (therefore it should be e-creativity). On the other hand, many discoveries and art revolutions (i.e. h-creative events) may have been based more on this kind of restructuring the space than on the generation of *impossible* concepts, which would mean that t-creativity is not necessarily a *superior* kind of creativity. We think that these issues could be better clarified with a more precise definition of *conceptual space*.

The final point to present here regards the distinction between combinatorial creativity and ET-creativity [Ritchie, 2001]. Although not having raised as much debate as the previous issues, essentially because Boden herself dropped this differentiation, this one has particular interest for our work. If one sees the problem of combinatorial creativity as the generation of a new concept from the association of previous ones (as the definition says), one can also accept a conceptual space containing all the possible combinations. By doing so, there is no difference between the act of exploring the conceptual space of possible combinations, and the act of generating a combination (which would exist in that conceptual space). Similarly, if, with the novel association, a novel concept emerges that could not have been generated before, then we could have achieved t-creativity. In other words, although combinatorial creativity may be regarded as a particular kind of creativity (which is also the one approached in this book), it should also be included in and not distinguished from the set of ET-creativity phenomena. This is, as far as we know, a common and uncontroversial perspective towards ET- and combinatorial creativity.

It is without doubt that Margaret Boden produced a comprehensive analysis of creativity and AI that has been used and applied in many works. We too will use some of the ideas described. Above all, she contributed successfully to the provocative question about whether computers can be creative. Indeed they can be, although perhaps at a very limited level in comparison to human creativity, at the least at a level that does not demand self-awareness. This may be a very mechanistic, *unromantic*, level, but it is nonetheless clearly able to surprise humans with outcomes that we ourselves would normally have no problem in considering as the result of creative behavior. In section 2.2, we will give an overview of some of these systems.

2.1.5 Others

Given the recent increase in creativity research, it is not a simple task to provide a meticulous overview without leaving out any fundamental works. There is a great

variety of approaches and therefore we will provide a summary of the most cited works from those approaches that are most prominent and field-covering: cognitive psychology (Finke, Ward and Smith); confluence theories[6] (Sternberg and Lubart); neuroscience (Martindale and Greenough); motivation and intention (Amabile) and biographical case studies (Weisberg).

Finke and his colleagues have proposed what they call the Geneplore model, according to which there are two main phases in creative thought: generative and exploratory [Finke et al., 1992]. Many potential ideas or solutions are created, followed by their extensive exploration. From laboratory experiments, these researchers concluded that subjects generate a set of "*preinventive* structures, in the sense that they are not complete plans for some new product, tested solutions to vexing problems, or accurate answers to difficult puzzles" [Ward et al., 1999]. From these partial structures, a phase of exploration and interpretation takes place that attempts to construct a feasible solution to a problem, by focusing and expanding these structures. Constraints can be imposed at any time during the generative or exploratory phase. This model "acknowledges that a range of factors other than cognitive processes contribute to the likelihood of any individual generating a tangible product that would be judged to be 'creative' " [Ward et al., 1999]. From a broad perspective, the Geneplore model falls into the class of divergent-convergent models, as proposed by Guilford and agreed by Csikszentmihalyi.

The *investment theory* of Sternberg and Lubart falls into the category of confluence theories (theories that offer the possibility of accounting for diverse aspects of creativity). It suggests that "creative people are ones who are willing and able to 'buy low and sell high' in the realm of ideas. Buying low means pursuing ideas that are unknown or out of favor but that have growth potential. (..) The person persists in the face of this resistance and eventually sells high"[Sternberg and Lubart, 1996]. From extensive experimentation, Sternberg and Lubart developed a model that presupposes the interaction of six distinct but interrelated resources: intellectual abilities, knowledge, styles of thinking, personality, motivation, and environment.

At a different level of research, Martindale and Greenough studied the variability of level of arousal and attention in the performance of creativity tests (e.g. Remote Associations Test, Similarities Test of divergent thinking), by observing galvanic skin response fluctuations, heart rate variability, cortical activation, as well as other biometrical measures [Martindale and Greenough, 1974], [Martindale, 1999]. One interesting conclusion was that creative individuals have a highly variable level of arousal, rather than a basal (i.e. *stable* in this context) level of arousal, which means that

[6]Csikszentmihalyi's systems model also falls into this category

"creative inspiration occurs in a mental state where attention is defocused, thought is associative, and a large number of mental representations are simultaneously activated" [Martindale, 1999]. Moreover, the authors also associate their work with the primary-secondary process thesis [Kris, 1952], which says that creative individuals have a greater ability to switch between two modes of thought (primary and secondary) than less creative individuals. Primary process thought is found in states such as dreaming and reverie (as well as in psychosis and hypnosis), it is autistic, free-associative, analogical. Secondary process is the abstract, logical, reality-oriented thought. Martindale found supportive evidence confirming Kris's proposal, meaning that creativity is not just based on primary process thought, but on its systematic intertwining with the secondary process. Again, we find a clear similarity to the ideas of divergent thinking (the primary process) and convergent thinking (the secondary process), as discussed before.

The work of Teresa Amabile on motivation and intention is also often cited in literature. She proposes a two-pronged hypothesis about how motivation affects creativity: "The intrinsically motivated state is conducive to creativity, whereas the extrinsically motivated state is detrimental"[Amabile, 1983]. Intrinsic motivation is associated with the enjoyment of the work in itself, while extrinsic relates to engaging in an activity in order to meet some goals external to the work. She also proposes a confluence model with intrinsic motivations, domain-relevant knowledge and abilities, and creativity-relevant skills. These include: " a) cognitive style that involves coping with complexities and breaking one's mental set during problem solving; b) knowledge of heuristics for generating novel ideas, such as trying a counterintuitive approach, and c) a work style characterized by concentrated effort, an ability to set aside problems, and high energy (..)".

The issue of re-representation is emphasized by some researchers (e.g. [Karmiloff-Smith, 1993], [Oxman, 1997]), who propose that a process of representational re-description precedes creative domain exploration, in which previously implicit knowledge is more clearly perceived. New patterns thus emerge and novel inter-domain connections may be made.

Finally, in a totally different direction (essentially based on biographical case studies), Robert Weisberg challenges many of the works just described above by arguing against what he calls the *tension view* of the relationship between creativity and knowledge [Weisberg, 1999], which says that: since knowledge about a problem is not complete, and in face of a blockage, the person is left to its abilities to discover new solutions via free-association, divergent thinking, etc. This discontinuous view of knowledge evolution is a "dominant one in modern theory" [Weisberg, 1999]. Weisberg proposes a continuous view, by arguing that new discoveries and revolutionary

artworks are the result of a state of maturity and knowledge richness. This *foundation view* thus concludes that creativity and knowledge are positively related: "The reason that one person produced some innovation, while another person did not, may be due to nothing more than the fact that the former knew something that the latter did not."[Weisberg, 1999]

As pointed out at the beginning, there are certainly important works that have been left out. Yet, for the present book it is more important to provide the reader with its fundamental bases and their synthesis than with giving exhaustive knowledge about lateral issues. The idea here was two-fold: to suggest that, from a computational modelling perspective, many different approaches can be taken; but, nonetheless, there are common aspects across almost all of the approaches. We will attempt to include these aspects in the next section.

2.1.6 Synthesis

Amongst the works mentioned heree, there is an undisputed agreement that creativity involves the creation of a *novel* and *useful* product. We may find different words for creation (generation, production), novelty (originality), usefulness (value, appropriateness, utility, significance, adaptability) and product (idea, concept, solution to a problem), but there is no doubt that these are either synonymous or different perspectives on the same subject. Even though not as uncontroversial, the majority of the works also agree that the cognitive processes that bring about a novel and useful product consist of a pair of opposite *styles* of thinking: divergent and convergent. The former is characterized by allowing ideas that defy logical reasoning (e.g. unsound conclusions, contradictory associations, inconsistent sets of facts) or that correspond to unprecedented associations[7]. In opposition, convergent thinking corresponds to logical reasoning, which follows well-defined constraints and is normally associated to methodic, purposeful thought[8]. We have already presented these two concepts in section 2.1.1, but, as often equalled, they have been associated, when not synonymous, to primary/secondary-process [Kris, 1952] and generative/exploratory

[7]Here we emphasize the originality aspect of divergent thinking, as it was defined by Guilford, giving flexibility, fluency and elaboration a secondary role. In our opinion, flexibility would solely depend on originality, for we can only get varied solutions if each one is sufficiently different from the others, i.e. original. On the other hand, we see fluency as a characteristic of the thinker, not of the thought itself (a thought can be original, but never fluent). Finally, analyzing past work, elaboration has been considered belonging to the convergent side.

[8]This definition of logical reasoning comes from a psychology perspective, therefore it may be incomplete from an AI logician point of view. However, we cannot describe these concepts more formally than allowed by the literature itself.

[Finke et al., 1992] (also, to a lesser degree, to bisociation [Koestler, 1964], transformational/exploratory [Boden, 1990] and incubation/elaboration [Wallas, 1926]), and therefore needed to be put in a more actual context. The other motivation for this redefinition is to claim that, if a creative product must be novel and useful and its creation involves the process of divergence and convergence, then one cannot build a creative system without modelling both divergent and convergent processes. Moreover, divergence would be primarily responsible for granting novelty, and convergence for usefulness. We will come back to these issues later.

It has also been repeatedly emphasized that knowledge is of central importance in creativity. Virtually every author in this study argued that no important discovery or major artwork is likely to transpire without its creator having acquired deep knowledge about the domain in question. Furthermore, some also claim that having broad knowledge is also fundamental, in order to potentiate transfer across domains and awareness of the environment. Here we have described two perspectives: in-depth and in-breadth. Again, we are tempted to associate these two categories with convergence and divergence, respectively. This means that knowledge from the domain in question would in turn benefit the convergence towards useful solutions, helping to discern the good from the bad, while knowledge from a variety of domains would promote divergence in problem solving.

The existence of different levels of creativity has also been claimed by many authors. Some arguing for a continuum (e.g. [Koestler, 1964]), some for a clear distinction (e.g. [Boden, 1990]): On one extreme, we have the creativity with a big C, transformational creativity or true creativity. On the other side, we have the personal, exploratory, or mundane creativity. We do not intend to say that these are synonymous concepts, but to emphasize the bipolarity of the analyses made.

A final and more controversial issue pertains to the role of society in creativity. Some authors (e.g. [Csikszentmihalyi, 1996]) argue that there can be no true creativity[9] without the society, i.e., something does not exist as creative unless it is externally judged as so. It is a dynamic ascription that depends on the interaction of several entities (in the case of Csikszentmihalyi, the person, the domain and the field). Others (e.g. [Finke et al., 1992]) argue for the existence of creativity for its own sake, i.e. an individual can generate a creative idea, without having feedback from the society. These two points of view correspond to what Boden called h- and p-creativity, respectively, and thus reflect more a difference of perspective than of the essence of creativity.

[9] We remind that Csikszentmihalyi allows a kind of personal creativity, which is though secondary for the system and for what he calls *true* creativity.

2.2 Computational Creativity

In this section, we introduce the area of computational creativity, which aims to build computational systems that are able to exhibit creative behavior. Since this is not a universally accepted definition (how can we know, after all, when the behavior is creative?), some works carry on to the field of Artificial Intelligence the debate that we have presented in the previous section. One of the immediate effects of this is the need for formal accounts for creativity, while the other is the experimental implementation of such systems.

2.2.1 Two accounts for characterizing creativity

The legacy left by Boden's descriptive hierarchy sparked off the attention towards analyzing AI systems from a creativity perspective (e.g. [Ram et al., 1995], [Bentley, 1999]). Notwithstanding the many fragilities, some of which have already been named, the point was made of the need of such analyses or, at the least, for the consideration of Creativity within AI. However, the lack of formal approaches and of stable epistemological and methodological settlements condemns this research to endlessly recycle unsolvable problems. While it seems currently impossible to say that a system is creative, or even intelligent, without any controversy, it may be possible to classify it according to criteria that are based both on formal computational accounts and on theories such as Boden's.

We now present the two approaches towards characterizing the creativity of AI systems. The first one derives from an assumed attempt of formalizing Boden's theory and is centered on the process, while the second one deals with evaluation and focusses on the product. We will apply these works when analyzing our system in chapters 4 and 6. In order to keep this section comprehensible for a general audience, we tried to reduce the formal expositions to the minimum necessary. On some parts, these become indispensible to clarify concepts for those interested in technical detail, but they are still accompanied with an informal description.

Characterizing Creativity in AI

Wiggins [Wiggins, 2001, Wiggins, 2003] views exploratory creativity as a search for concepts in the space of all possible concepts of a domain (e.g. in the music domain, it would be the set of all possible sequences of sounds). This search is *a priori* constrained by a set of rules that define "acceptable" elements (Boden's conceptual space), $\mathcal{R}$. This set could consist, for example, of rules of style. This could comprise

both creative and non-creative elements, but all correct according to those rules of style. This set of elements is also known (in [Wiggins, 2001, Wiggins, 2003]) as the set $\mathcal{C}$.

Another set of rules that Wiggins proposes would be the set of "valued" elements, those that, regardless of being "correct" or "according to style", become successful in solving the problem or achieve the goal that is being sought (e.g. pieces of music that, although breaking stylistic rules, become for some reason a good choice). This set, $\mathcal{E}$, is much harder to define either formally or informally, for often it is dependent on dynamic aspects (e.g. aesthetic change, specific goals, specific context, personal mood, etc.).

Finally, the third set of rules proposed is the one that defines the *strategy* followed when exploring the conceptual space, $\mathcal{T}$. This represents the choices, be they intentional or unconscious, that the creative agent (human or not) makes to *travel* in the space of possible elements. For example, some people prefer to work "top-down", i.e. to define the whole structure, then proceed reifying the ideas, others "bottom-up". Others rely on ad-hoc methodologies, even randomness. It is not guaranteed, though, that all elements found are acceptable (belong to $\mathcal{C}$). In other words, the application of $\mathcal{T}$ may produce non-acceptable outcomes. This aspect is extremely important in terms of creativity, as it opens a door to re-thinking the whole system.

According to Wiggins [Wiggins, 2001], it is thus from the interaction of these three sets, $\mathcal{R}$, $\mathcal{E}$ and $\mathcal{T}$ that exploration of the universe of concepts can be analyzed in the light of exploratory creativity. Each of these sets can change with time, in the same way that science and arts evolve by changing their rules, goals or methodologies. It is this constant change that drove Wiggins (and others [Wiggins, 2001, Colton, 2001, Ram et al., 1995]) to a conclusion that the process of exploratory creativity (the three sets) is also exploratory in and of itself. In other words, using the same formalization as the one referred to above, Wiggins proposes the demonstration that transformational creativity is equivalent to exploration at the meta-level [Wiggins, 2001]. The idea is that transformational creativity can only happen with changes in (at least) one of the three sets and, if we jump one level up and consider an exploration in the space of possible rule sets (e.g. the space of style rule sets, space of "value judgments", space of strategies), then the same analysis can be made. Of course, the question arises: when to stop this recursion?; or even, is this conclusive about the act of creativity? Aren't those transformations driven bottom up or emergent (e.g. serendipity, empirical observation, sensorial evolution), rather than a meta-level activity? Of course, Wiggins setting is more a proposal for the analysis and discussion of creativity within AI than an actual statement of how things work cognitively and, in that sense, it has been an interesting base to apply.

Creativity Assessment

While Wiggins centered his formalization on several aspects of the process (the constraints, the strategy, the conceptual space), giving less attention to the product, Ritchie proposes a set of *features*[10] for assessing creativity on the basis of the results of the system (i.e. the product), its initial data and the items that gave rise to its construction (the *inspiring set*) [Ritchie, 2001].

Prior to describing the features, we have to give a set of definitions. The first one regards the notion of *basic item*, an entity that a program produces. "This is *not* a definition of what would count as successful or valid output for the program, merely a statement of the data type it produces". Ritchie proposes two *rating schemes* to analyze the produced items: *typicality ratings* (*typ*) and *value ratings* (*val*). There is also the notion of *inspiring set*, I, the set of basic items that, explicitly or implicitly, lay behind a generative system. For example, for a rule-based system, the elements implicitly defined by those rules would be in the inspiring set; for a case-based system, the initially stored cases would also be part of the inspiring set; for an example-based, learning system, the examples given in the training phase would belong to the inspiring set. The definition of the inspiring set is fundamental to measure how successful a system is in obtaining novel ideas, but it is often hard to find, since the designer of the system is rarely conscious of all the influences beneath their own choice. For example, a system for composing a certain style of music may have been designed via analysis of a set of pieces (which would be part of its inspiring set), while if it were made from musicology theories, defining that set would be a harder task.

Finally, we need to define four notation conventions. We assume that X and Y are two sets of items, and that F can be a function such as *typ* or *val*:

[10] Although Ritchie has named them "criteria", in practice they do not behave as such: none is an explicit condition for (or even against) creativity. In order to avoid these criteria to be seen as *desired* characteristics of creative artifacts, we simply name them as *features*. The assumption is that they are simply measurable features that can help characterize the creativity of a system.

$T_{\alpha,\beta}(X) \stackrel{def}{=} \{x \in X \mid \alpha \leq typ(x) \leq \beta\}$:	The subset of X falling in a given range of normality.
$V_{\alpha,\beta}(X) \stackrel{def}{=} \{x \in X \mid \alpha \leq val(x) \leq \beta\}$:	The subset of X falling in a given range of quality.
$AV(F,X) \stackrel{def}{=} (\sum_{x \in X} F(x)/ \mid X \mid)$:	The average value of a function F across finite set X.
$ratio(X,Y) \stackrel{def}{=} \mid X \mid / \mid Y \mid$:	The relative sizes of two finite sets X, Y.

Ritchie proposes fourteen features to assess the creativity of a system's output, R. Although it is assumed that R corresponds to the result(s) of a single run, it is also suggested the generalization of these features to a set of runs, in order to cover the general behavior of the system. Thus, in this case, we invite the reader to consider R as the set of results that a system has been able to produce until a given point in time. The features are intended to measure the behavior of the system in terms of average quality of results, their typicality and of their ratios with regard to R and to the set of typical and valued items. In general, isolated features will not say anything regarding creativity. Each one contributes with its own part to the overall picture of the analysis of the output. Some of them (the later ones), however, are combinations of the others and allow us to conjecture about creativity potential.

feature 1 $AV(typ, R) > \theta$, for suitable θ.

The first feature compares the average of typicality of items with a value, θ. The following feature studies to what extent typical items form a significant proportion of the results:

feature 2 $ratio(T_{\alpha,1}(R), R) > \theta$, for suitable α, θ.

feature 3 $AV(val, R) > \theta$, for suitable θ.

feature 4 $ratio(V_{\gamma,1}(R), R) > \theta$, for suitable $\gamma\theta$.

Features 3 and 4 follow the same reasoning as the first two, but applied to value (*val*). Alternatively, the fifth feature classifies the success of a system as its ability to obtain a high proportion of highly valued items, within the set of the typical ones:

feature 5 $ratio(V_{\gamma,1}(R) \cap T_{\alpha,1}(R), T_{\alpha,1}(R)) > \theta$, for suitable α, γ, θ

The following three features compare the set of highly valued, yet untypical results, to the output, to the whole set of untypical results and to the set of typical highly valued outcomes.

feature 6 $ratio(V_{\gamma,1}(R) \cap T_{0,\beta}(R), R) > \theta$, for suitable β, γ, θ.

feature 7 $ratio(V_{\gamma,1}(R) \cap T_{0,\beta}(R), T_{0,\beta}(R)) > \theta$, for suitable β, γ, θ.

feature 8 $ratio(V_{\gamma,1}(R) \cap T_{0,\beta}(R), V_{\gamma,1}(R) \cap T_{\alpha,1}(R)) > \theta$, for suitable β, γ, θ.

A program might be replicating its entire inspiring set, in which case it might be said it is not creative since (at least some of) the outputs are not novel:

feature 9 $ratio(\mathrm{I} \cap R, \mathrm{I}) > \theta$, for suitable θ.

Thus, a high value in this feature is something we *don't* want if looking for creativity in a system. Conversely, the system may produce outcomes that do not belong to the inspiring set. Thus, in feature 10, we calculate the ratio of all the items generated with respect to the ones (from that set of generated ones) that also belong to the inspiring set:

feature 10 $ratio(R, \mathrm{I} \cap R) > \theta$, for suitable θ.

Features 11 and 12 propose some possible perspectives on the generated items not belonging to the inspiring set: proportion of those that are typical (feature 11); proportion of those that are valuable (feature 12). The former stresses the capability of the system to produce results that are not replications, yet still fitting the "norms" (i.e. still being typical). The latter estimates how well the system is able to produce valued items that are not replications.

feaure 11 $AV(typ, (R - \mathrm{I})) > \theta$, for suitable θ.

feature 12 $AV(val, (R - \mathrm{I})) > \theta$, for suitable θ.

Finally, features 13 and 14 give an estimate about the proportion of highly typical and valued novel results. The latter justly fits the view of creativity as the generation of "novel and valued" products.

feature 13 $ratio(T_{\alpha,1}(R - \mathrm{I}), R) > \theta$, for suitable α, θ.

feature 14 $ratio(V_{\gamma,1}(R - \mathrm{I}), R) > \theta$, for suitable γ, θ.

These features pose two obvious problems for their application. The first one has to do with the rating schemes *val* and *typ*, namely the former would demand a compromise that is rarely explicitly made in everyday observation of creativity, of what a *valuable* outcome is composed of exactly. The second problem regards the variables involved (α, β, γ and θ). Finding acceptable values will depend on experimentation in different contexts. Yet, until now, there has been no application of these. Furthermore, their scales will differ among features (e.g. features 4 till 9 yield values in the interval [0, 1], feature 8 can give any positive real number, feature 10 always results in values higher than 1). In this work, we assume α, β and γ to be

0.5.

Another issue is that Ritchie considers typicality and value, rather than novelty and usefulness. While usefulness and value are often meant as synonymous (in the sense that something is valued when it accomplishes a purpose), typicality runs opposite to novelty. Assuming the risk of oversimplifying these notions, we consider, in this work, that typicality is converse to novelty (i.e. $novelty(x) = 1 - typ(x)$) and value equals usefulness. This is important for the analyses made in chapter 6.

In terms of direct application to computational systems, there has been almost no examples, and therefore empirical values and considerations for the features are not available. There may be three reasons for this happening: the frameworks are still immature and therefore demand further work; there has been no practical need for systems to measure their creative potential and to compare with their peers, a fact that contrasts with the claim that some make of being creative systems; they may be considered wrong or useless for analyzing computational creativity. Regarding this latter possibility, it is clear that these are the only accounts so far for the formal analysis of the creativity of a computational system (except for even less developed formalizations, such as the serendipity equations of Figueiredo and Campos [Figueiredo and Campos, 2001] or work on measuring surprise [Macedo and Cardoso, 2001]). Since it is imperative to determine, even if not in an absolute fashion, the creativity of the model we propose, we will apply these ideas to analyze our system.

It is also patent that these two approaches have a lot in common. For example, Wiggins $\mathcal{R}$ and $\mathcal{E}$ may correspond to Ritchie's *typ* and *val* respectively. There is however a fundamental difference: while Ritchie looks *inside* the system's results and their evaluative schemes and is dependent on them to define every feature, henceforth considering only exploratory creativity[11], Wiggins' formalism confronts the system with the universe ($\mathcal{U}$) of possible items, thus allowing for meta-level analysis and therefore transformational creativity. In figure 2.3, we summarize a classification of concepts within the framework of Wiggins, also taking into account the inspiring set, the notions of typicality and value and features presented. This diagram is based on the interaction between three sets: $\ll \mathcal{R}, \mathcal{T}, \mathcal{E} \gg$, the set of concepts that can be obtained by the strategy $\mathcal{T}$; $[\![\mathcal{R}]\!]$, the set of items defined by the rules $\mathcal{R}$; $[\![\mathcal{E}]\!]$, the set of valued items, acording to the rules $\mathcal{E}$. Transformational creativity would thus have the effect of changing the set of reachable concepts. The set "reachable creative concepts"has been named after the description of the features 5, 10, 11, 12, 13 and

[11] He proposes t-creativity as the situation described by feature 6 (high *val*, low *typ*), but for which there is a (yet to be defined) rating scheme typ' that highly classifies the items.

14, but it can be argued that the label "creative"should be given to the set of "valued untypical concepts", since the latter favours novelty and value.

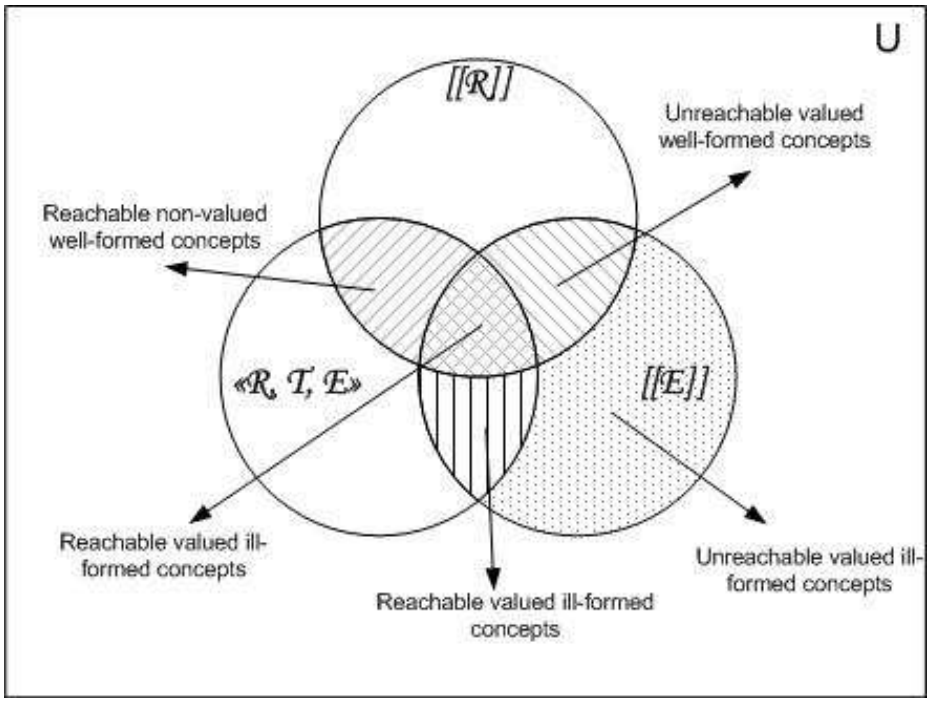

Figure 2.3: The classification of concepts within the universe $\mathcal{U}$ according to the sets $\ll \mathcal{R}, \mathcal{T}, \mathcal{E} \gg$, $[\![\mathcal{R}]\!]$ and $[\![\mathcal{E}]\!]$. Remember that $\mathcal{C} = [\![\mathcal{R}]\!]$. Inside parenthesis are the features associated with each set.

2.2.2 Creative systems

What is a creative system?

Before proceeding, we must define the necessary (but not sufficient) conditions for a computational system to be included in the list we study here as being creative:

- It should aim to produce solutions that are not replications of previous solutions (known to it).
- It should aim to produce solutions that are acceptable for the task it proposes.

Of course, these correspond to the classical definition of novelty and usefulness, from a perspective of p-creativity, in Boden's terms. Thus, we are allowing for the *classical* extremes: a random process that can find good items; a system that explores the whole search space with a brute-force blind method; a system that generates different outcomes every time, but which all look very similar to humans. All these

are creative systems, according to this classification, but we intend primarily to rule out the (much larger) set of AI systems that focus solely on the second condition: to accomplish a well defined goal. As with intelligence, there is a continuum of creativity *degree* in such systems, from the utterly non-creative to the undoubtedly creative, and some dimensions can be inspected in order to sort out further classifications:

- The complexity it is able to treat, without breaking the conditions above - creative abilities are often more salient (or vital) in problems with high complexity. A program able to satisfy our criteria in highly complex settings is definitely on the (highly) creative side of the continuum.

- Its ability to reason at different levels of abstraction - A system that can deal with different levels of abstraction for the same problem would certainly be closer to the goal of meta-level reasoning (and t-creativity) than a system without such ability.

- Its ability to process different sorts of representation - It has been often referred in Creativity literature (e.g. [Karmiloff-Smith, 1993]) that re-representation potentiates creative outcomes.

- Its ability to work in more than one domain - Creativity cannot be regarded in isolation. A system that is able to deal with more than one domain without structural changes should be considered more creative than one that needs reprogramming to work with different domains. The former should be closer to the ability of cross-domain transfer.

- Its ability to evaluate its own productions - Self-assessment or self-criticism has also been mentioned as central to creative production. A system that can evaluate its own productions would tend to be on the creative side of the continuum.

Some systems and models

Of all the surveys presented in this book, the overview of the field of creative systems is the most difficult to make for two particular reasons: there has been a surprisingly high proliferation of such systems during the past five years, spreading across a variety of areas and approaches; only a few identify themselves as "creative systems", preferring different classifications such as "cyber art", "generative systems" or "creative design". Moreover, from these, only a few consciously follow an underlying "computational model of creativity". Thus, in order to analyze the models that have been used so far, it is necessary to abstract them from the existent implementations.

The list of systems chosen for the overview obeys three conditions. Each system should have its implementation description available somewhere (ruling out many commercial implementations), it should also have been published recently[12] (except for some *classical* examples), and it must satisfy, even if only assumed informally, the definition above for creative systems.

The approaches presented range all over the traditional AI paradigm classification spectrum of Symbolic, Sub-symbolic and Hybrid systems. This division could therefore be a starting point to structure this overview. Nevertheless, we prefer to organize it according to issues that have been discussed so far regarding the theme of creativity.

The first issue regards the opposition of perspectives *cognition/society*, which were also observed in section 2.1. Some works follow the *systems perspective* of Csikszentmihalyi, via multi-agent environments in which creativity emerges as a result of multiple interaction. Examples of this approach are the Hybrid Society [Cardalda, 1999], The Digital Clockwork Muse (TDCM) [Saunders and Gero, 2001], Design Situations (DS) [Sosa and Gero, 2003] and SC-EUNE [Macedo and Cardoso, 2001]. On the other side, there are the systems that follow approaches based on a cognitive perspective (i.e. the machine is one single agent) by applying domain-dependent rules (e.g. Lothe's Mozart Minuets [Lothe, 2000], Aaron [Cohen, 1981]), statistical models (e.g. Craft and Cross's fugal exposition generation [Craft and Cross, 2003], the Postmodernist generator [Bulhak, 2000]), reusing past experience (e.g. the Case-Based Reasoning approaches of ReBuilder [Gomes et al., 2002] and ASPERA [Gervás, 2000a]), evolutionary computation (e.g. NevAr [Machado and Cardoso, 2002] and Poevolve [Levy, 2001]), modelling specific cognitive phenomena like Metaphor (e.g. Sapper [Veale, 1995, Veale and Keane, 1997, Veale and O'Donoghue, 2000]), Analogy (e.g. Copycat, [Hofstadter and Mitchell, 1988]) or Conceptual Combination (e.g. C^3 [Costello, 1997]). Of course, individual agents in multi-agent systems need to have individual processes, and therefore these two perspectives are not totally incompatible.

A second issue regards evaluation: is the system performing self-evaluation by any means in order to obtain the final product, or is there the participation of a user, the generation of the product being a result of the interaction (in this case, the general system - human plus machine - could be seen as a two agent system)? In general, multi-agent systems presuppose a built-in evaluation strategy, normally becoming part of the interaction between agents (agents reward other agents for positive evaluation, as happens in TDCM), the Hybrid Society being an exception (humans

[12]We conventionalize "recently" to comprehend the past four years, for it is since 2000 that events exclusively dedicated to the subject have been held yearly.

can be part of the egalitarian multi-agent system and produce and/or evaluate). In single-agent architectures, some kind of self-assessment is also built-in, either via probability, rules, or pre-trained mechanisms (like a Neural Network). These self-assessments rely more on the appropriateness of the product according to a style or goal than on aesthetic judgment. Yet, one should not read this too strictly since we can find a variety of self-evaluation methods that consider aesthetics (e.g. NevAr allows the application of aesthetic principles based on perception; HR contains interestingness heuristics). Some systems rely on the active participation of the user in the generation process. A typical example is the interactive genetic algorithm (IGA), where the algorithm generates items that are evaluated by the user (e.g. NevAr and [Sims, 1991][13]), and this evaluation is used to produce the subsequent items. Max [Campos and Figueiredo, 2001] is an agent that searches the Web for interesting pages (according to a user profile) in order to trigger serendipitous insights for the user, who has to give Max appropriate feedback to continue the cycle. MuzaCazUza and ReBuilder are two Case-based reasoning systems on music and software reuse (respectively), which rely on the user for the adaptation of cases (in the case of ReBuilder, it also provides an analogy mapping method of adaptation).

A third issue has to do with the use of memory. Does the system keep track of past runs, and therefore is it able to profit from past experience? This can be seen as an instantiation of the *preparation* phase discussed earlier. A few systems have this property, namely SC-EUNE, NevAr, ReBuilder, Max, Metacat and Sapper.

Being able to do meta-level reasoning, i.e. to reason about the *method of reasoning*, could only be found in three systems: HR, Metacat and Sapper. The former [Colton et al., 1999], named after Hardy and Ramanujam, was designed by Simon Colton to carry out discovery in pure mathematics. It performs a complete cycle of mathematics, including building its own mathematical concepts, making conjectures about the concepts, attempting to prove the true conjectures and finding counterexamples for the false ones. It builds new concepts according to seven heuristics and nine production rules. All these are considered the most generic possibilities across the field of mathematics. It has a meta-level reasoning version [Colton, 2001], in which it builds a high-level theory that contains concepts and conjectures about the concepts and conjectures of the lower-level theory. For example, it is able to form the high-level concept of "function" (there is a unique second object for each first object found in the pairs which make up the examples for these concepts). Examples of low-level functions are definitions for prime numbers, perfect numbers, pairs and so on. The HR project has also a multi-agent version with four HR-based agents with different strategies running in parallel [Colton et al., 2000], which cooperate by

[13]These two cases are more specifically *interactive genetic programming* (IGP) systems

exchanging new concepts and conjectures.

Metacat [Marshall, 2002] is the latest evolution of Copycat [Hofstadter and Mitchell, 1988], which is a system for solving puzzle analogies (such as "abc→abd::xyz→?") that applies a bottom-up parallel strategy to find mappings between the source and the target, as well as explanatory relations within them, and to associate these mappings and relations with concepts in a *Slipnet*. This Slipnet is a semantic network with variable distance between concepts (examples of concepts are the letters of the alphabet, relations like *opposite* or *predecessor*, and attributes like *rightmost* or *first letter of the alphabet*) where a spreading activation algorithm is used to determine the plausible mappings (e.g. "c" corresponds to "z" - both are rightmost - and "c" is predecessor of "d" - in the alphabet). When a set of concepts is activated, a candidate solution is projected. If it fails, Copycat will *slip* to concepts that have short distance to the activated ones (e.g. *rightmost* to *leftmost*) and try other activation patterns. Metacat is a sibling of Copycat and it adds a set of meta-level features. It keeps traces of the run and creates abstract *themes*, which consist of pairs of Slipnet concepts. For example, a theme for representing the idea of alphabetic-position symmetry between "a" and "z" would have the Slipnet concepts *alphabetic-position* and *opposite*. It keeps this new information in an episodic memory that it uses to compare analogies between different runs (it outputs texts like "this reminds me of the problem X..."). It is also able to justify the reason behind a puzzle solution by the analysis of its trace. However, it uses this meta-level knowledge (traces and themes) superficially in the sense that, unlike HR, it only uses it for communicative purposes (with a user) and does not improve or change its own internal knowledge, or even use previous solutions to solve present problems.

Sapper (which will be presented in detail in section 3.4.2) is also said to be capable of meta-level modification, by altering the "construction space" that guides the mapping process ([Veale and O'Donoghue, 2000]). With this capability, the system should be able to propose different metaphor interpretations for the same input data.

In spite of recurrently being asserted throughout the creativity theories in section 2.1, the ability to do cross-domain transfer of ideas is absent in the majority of the systems. Even worse, the majority is tailored to work with its own single domain. In this matter, again and surprisingly, HR presents itself as the only one able to do cross-domain transfer [Steel, 1999] and work with different domains. However, these two capabilities rarely come together. The cross-domain version of HR was built with special attention to concepts from mathematical fields[14], while its application to domains separate from mathematics (e.g. animal classification [Colton et al., 2000])

[14]For example, the use of triangular number (from number theory) to predict the order of the duplicated node (in graph theory).

has been made in an isolated manner. Two other systems are capable of cross-domain transfer, normally at an abstract level. Sapper is a metaphor interpretation system that will be thoroughly presented in section 3.4.2, since it is important for our work, while Drama will be briefly described in section 3.4.3. Both find cross-space mappings between semantic networks, which can be used to transfer knowledge, i.e. problem solving by analogy.

From this analysis, we may now abstract a taxonomy for computational models of creativity:

- *Systems model* (SM). Creativity results from the interaction of a society of agents. Each agent may have a similar or different role, and be implemented according to the methods below, but its interaction with others is essential to find creative items.
- *Evolutionary model* (EM). Creativity emerges as a result of evolution of the artifact. This evolution is made in parallel with concurrent streams of candidate artifacts that eventually converge to a maximum. The judgments (fitness functions) are either given by a user (IGA's) or via algorithmic methods such as neural networks or heuristic rules.
- *Domain-centered model.* Creativity results from expertise on a specific domain. Different domains invite different specific methods or knowledge structures (even if the general approach remains the same). This model can be divided into three sub-types:
  - *Expert Systems model* (ESM). Items are generated by following well established constraints and methodological rules of the domain of application. Creativity is stimulated by allowing randomness in well-bounded decision making points.
  - *Case-Based Reasoning model* (CBRM). Creativity is the result of reuse and adaptation of past experience with attention to the present context. According to [Ram et al., 1995], this is achieved in five steps: problem interpretation, problem reformulation, case and model retrieval, elaboration and adaptation and evaluation.
  - *Statistics model* (STM). Creative items are generated from non-deterministic automata that result from analysis on selected data[15]. When it is well-trained, issues like evaluation or memory are embedded in the automaton,

[15]It should be said that these systems are many times built with an analytical intention (e.g. for prediction or classification) rather than for generative purposes, although becoming able to be used as both.

which rarely produce *wrong* outcomes or outcomes that differ considerably from the initial data.

- *Cognition-centered model* (CCM). Creativity results from mental processes that can be computationally modelled. It is domain-independent and therefore items are represented at a conceptual level that needs to be reified at application level. This reification may be made externally, but it must be consistent with the concept description.

In tables 2.1 and 2.2, we give a summary of the characteristics of the systems analyzed. Some systems are not described in detail here because their description would not add pertinent facts for this book. Many more systems were left out (never mind the commercial ones), so this is a very small sample of the state-of-the-art which hopes to cover the wide breadth of the approaches.

Perhaps due to the youth of the area of creative systems or to the different purposes of each system, few provide formal analysis of the creativity involved. The formalisms of Wiggins, Ritchie and the others have scarcely been applied, which is understandable given the problems which arise when determining the inspiring set or the value for the many variables involved. Only WASP [Gervás, 2002] and Dupond [Mendes, 2004] have been analyzed so far with those formalisms, and it is still complicated to compare them with other systems (as emphasized in [Pereira et al., 2005]). In this book, we give another contribution to this fundamental aspect of evaluation of creative systems.

We would like to conclude this section by noting that, in spite of the current proliferation of creative systems, the large majority is exploratory. One can say that transformational creativity has so far been achieved by systems such as HR and Metacat, although only at an elementary level. These systems deal with meta-knowledge but are still far from actually *transforming* their conceptual space, strategy, knowledge representation or evaluation function.

Name	Theme	Reasoning Paradigm	Domain
HR	Concept Formation	Rule-based	Many (essentially Maths)
SC-EUNE	Surprise Modelling	Rule-based	Architecture
TDCM	Aesthetic evolution	IGA+NN	Artistic images
Design Situations	Design	Hybrid	Design
Hybrid Society	Aesthetic evolution	IGA+NN	Music
NevAr	Image generation	IGP	Artistic images
MuzaCazUza	Melody generation	CBR	Music
ReBuilder	Software Reuse	CBR+Analogy	Software Engeneering
Max	Serendipity	Rule-based	Web searching
Metacat	Analogy	Hybrid	Puzzles
ASPERA	Style imitation in poetry	CBR	Poetry
[Conklin and Witten, 1995]	Music prediction	Statistical	Music
[Craft and Cross, 2003]	Style imitation in music	Statistical model (n-gram)	Music
[Sims, 1991]	Image generation	IGP	Artistic images
Copycat	Analogy	Hybrid	Puzzles
[Lothe, 2000]	Style imitation in music	Rule-based	Music (Mozart Minuets)
EMI	Style imitation in music	Statistical model (n-gram)	Music
[Manurung et al., 2000]	Poem generation	Hybrid (rule-nased+GA)	Poetry
Poevolve	Poem generation	IGA+NN	Poetry
WASP	Poem generation	Rule-based	Poetry
JAPE	Pun generation	Rule-based	Humour
Postmodernist generator	Thesis generation	Statistical	Literature/humor
MAKEBELIEVE	Text Generation	Rule-based	Stories
Dupond	Text Generation	Rule-based	Conversation
Brutus	Text Generation	Rule-based	Stories
Aaron	Image generation	Rule-based	Artistic images
HAHAcronym	Acronym generation	Rule-based	Humour
Sapper	Metaphor	Hybrid	Many
Drama	Analogy	Hybrid	Many
C^3	Conceptual Combination	Rule-based	Noun-noun compound

Table 2.1: Generic description of the analyzed creative systems (part I)

Name	Architecture	Model	Episodic Memory	Evaluation	Reference
HR	Allows both	CCM, ESM	No	Built-in	[Colton et al., 1999]
SC-EUNE	Multi-agent	CCM, SM	Yes	Built-in	[Macedo and Cardoso, 2001]
TDCM	Multi-agent	SM, EM	No	Built-in	[Saunders and Gero, 2001]
Design Situations	Multi-agent	SM	No	Built-in	[Sosa and Gero, 2003]
Hybrid Society	Multi-agent	SM	No	Interactive	[Cardalda, 1999]
NevAr	Single agent	EM	Yes	Inter./Built-in	[Machado and Cardoso, 2002]
MuzaCazUza	Single agent	CBRM	No	Interactive	[Ribeiro et al., 2001]
ReBuilder	Single agent	CBRM, CCM	Yes	Interactive	[Gomes et al., 2002]
Max	Single agent	CCM, ESM	Yes	Interactive	[Campos and Figueiredo, 2001]
Metacat	Single agent	CCM	Yes	Built-in	[Marshall, 2002]
ASPERA	Single agent	CBRM	No	Built-in	[Gervás, 2000a]
[Conklin and Witten, 1995]	Single agent	STM	No	Built-in	
[Craft and Cross, 2003]	Single agent	STM	No	Built-in	
[Sims, 1991]	Single agent	EM	No	Interactive	
Copycat	Single agent	CCM	No	Built-in	[Hofstadter and Mitchell, 1988]
[Lothe, 2000]	Single agent	ESM	No	Built-in	
EMI	Single agent	STM	No	Built-in	[Cope, 1991]
[Manurung et al., 2000]	Single agent	EM,ESM	No	Built-in	
Poevolve	Single agent	EM	No	Built-in	[Levy, 2001]
WASP	Single agent	ESM	No	Built-in	[Gervás, 2000b]
JAPE	Single agent	ESM	No	Built-in	[Binsted, 1996]
Postmodernist generator	Single agent	STM	No	Built-in	[Bulhak, 2000]
MAKEBELIEVE	Single agent	ESM	No	Built-in	[Liu and Singh, 2002]
Dupond	Single agent	ESM	No	Built-in	[Mendes, 2004]
Brutus	Single agent	ESM	No	Built-in	[Bringsjord and Ferrucci, 2000]
Aaron	Single agent	ESM	No	Built-in	[Cohen, 1981]
HAHAcronym	Single agent	ESM	No	Built-in	[Stock and Strapparava, 2003]
Sapper	Single agent	CCM	Yes	Built-in	[Veale and Keane, 1997]
Drama	Single agent	CCM	No	Built-in	[Eliasmith and Thagard, 2001]
C^3	Single agent	CCM	No	Built-in	[Costello, 1997]

Table 2.2: Generic description of the analyzed creative systems (part II)

Chapter 3

Working with Concepts

Since in this book we propose a model of concept invention, it is therefore imperative to define what exactly is meant by a *concept* and by *concept invention*. Furthermore, we have to present and explain in detail the cognitive and computational basis applied at the level of working with those concepts. Thus, in this chapter, we define Concept and Concept Invention (and oppose it to *concept formation*). We also introduce Conceptual Combination, Conceptual Blending, Metaphor and Analogy, which occur in different parts of this book. All these work at the level of concepts or networks of concepts.

3.1 What is a Concept?

Perhaps the most specific definition we can give is that a concept is an abstraction that refers to ideas, objects or actions. Concepts can be dynamic entities, i.e. they can change with time (e.g. the concept of "phone" has evolved along with its technology), person (e.g. for some people a "crocodile" is a "pet", while for others it is not) or context (e.g. the concept of "giant" will differ radically when comparing an "elephant" with a "human" and with a "dinosaur"). In some domains, normally scientific, they can also be formal and static (e.g. the concept of "prime number" is not supposed to change). More than about the definition, much debate has been about how concepts are represented in cognition. There are three main views:

- Prototype view [Rosch, 1975]. Concepts are represented in the mind by *prototypes*, rather than by explicit definitions, which can be used to differentiate when an instance is or is not an example of the concept. Concepts are represented by an "idealized" prototype, which has the "average" characteristics of the concept (e.g. the prototype of "bird" would have "has wings", "has feathers", etc.) or

by a "paradigmatic" prototype (e.g. a coffee cup for a cup or wooden spoon for large spoon). Of course, this view raises problems because concepts are not necessarily static entities, definable with a fixed set of properties.

- Exemplar view [Medin and Schaffer, 1978]. Concepts are represented by their most common exemplars. Therefore, classifying an instance consists in determining which remembered exemplars are the most similar. This view implies that we organize experience in an episodic memory. If considered in isolation, the exemplar view fails in many aspects. Although it is agreed that knowledge is dependent on individual experiences, the ability to do abstraction, to generalize from experience, is fundamental, otherwise memory would be insufficient for reasoning.

- Theory view [Murphy and Medin, 1985]. The representation of concepts is based on *micro-theories*. A micro-theory describes the concept with facts about the concept (or related concepts) and causal connections between them. For example, the concept "bird" would have the facts that "it flies", "it has wings", etc., but also rules that explain causality (e.g. *Why do birds fly? Why do they nest in trees?*). Thus, a micro-theory can be seen as comprising a concept network (with causal links) and rules about the concept. This view also poses some problems such as these two: what should the limits of a micro-theory be (e.g. should we explain flight by physical rules, or with common sense and to which level of detail)? Since concepts can be dynamic, representing them with a theory would raise all sorts of problems of non-monotonic reasoning (how to represent change? how to maintain consistency and tractability?).

In AI, these three views have been applied. To name a few examples: the prototype view is common in systems that represent concepts as attribute value sets, such as in some machine learning systems (e.g. *version space* learning [Mitchell, 1978], *decision trees*); the exemplar view is typical in Case-Based Reasoning systems, where episodic memory is used to compare old to new problems; the Theory view is common in Logics (for example in Inductive Logic Programming) and in systems that use semantic networks, such as Sapper and Copycat (presented in sections 3.4.2 and 2.2.2).

Throughout this book, whenever we refer to concepts, we assume the Theory view, both in relation to our work and to the work of others, except when explicitly stating an alternative. It is also important to state the relationship between concept and category. In our work, a category is itself also a concept, but viewed from the perspective of membership (e.g. the concepts dog and wolf belong to the canine category, while the concepts canine and feline belong to the mammal category).

3.2 Building concepts

Throughout the literature, there seems to be some confusion with the notions of concept discovery, formation, invention, generation, design and creation. Sometimes they are synonymous to each other, sometimes they are considered different. We propose a distinction between two ways of building concepts: concept formation and concept invention. We provide a consensual definition for concept formation, from Psychology, which coincides (also consensually) with concept discovery. The definition for concept invention (or generation or creation) may be less agreed upon since it is based on less formal principles.

3.2.1 Concept Formation

In Psychology, Concept Formation (also known as concept learning or concept discovery) is associated with the development of the ability to respond to common features of categories of objects or events. In forming a concept, the subject must focus on the relevant features and ignore those that are irrelevant. In AI, this task is normally taken by machine learning, in which patterns are abstracted from analysis of data. In fact, the goal of machine learning systems is to form concepts. In this sense, if these systems happen to favor deliberately the formation of novel concepts (as opposed to systems built for well-defined goals, such as a pattern detection Artificial Neural Network, or a decision tree for classification), they can be classified as creative systems. A good example of this is the scientific discovery field. Again, we refer to HR [Colton et al., 1999] since we give for it is the most recent one of a series of systems centered on mathematics concept formation (e.g. AM and EURISKO [Lenat, 1984]) that use machine learning.

We can see from these definitions that concept formation is more concerned with analysis than with synthesis. In other words, works about concept formation deal more with abstraction of data regularities than with the invention of novel concepts[1]. There is a fundamental difference between these two processes: the former is based on finding sound explanations for data regularities, while the latter on producing concepts without concerns of soundness. However, this does not necessarily mean that concept formation is not creative. Quite the opposite, the capacity to perceive

[1]Thus the name "conceptual invention", given in some contexts (e.g. the Learning Cycle of Lawson-Abraham [Lawson et al., 1989] contains a step in which "the students and/or teacher derive the concept from the data, usually a classroom discussion (the CONCEPTUAL INVENTION phase)") seems now unfortunate. Words such as "discovery" or "formation" are clearly less ambiguous.

regularities and associations that no one has found before is definitely behind many of the major achievements of humanity.

3.2.2 Concept Invention

A concept has been invented (as opposed to formed) when it cannot be deduced from its generative process and when it did not exist before, intensionally or extensionally. In other words, the process of concept invention is unsound (e.g. abduction[2]). This definition covers a very broad range of possibilities, from randomness to heuristics-based search.

In concept invention, evaluation becomes a fundamental issue. Since there is no *a priori* notion of validity, criteria must be met for the assessment of the generated concepts. These criteria can coincide with those discussed for creativity (i.e. novelty and usefulness) or to problem solving (i.e. satisfying a goal). Since these are, again, difficult criteria, concept invention is normally applied as a generative phase to feed other sound procedures, which can guarantee validity. For example, in scientific discovery systems (e.g. HR, AM, EURISKO), conjectures are generated from the application of heuristics; in conceptual design the process of concept invention (or generation) commences by establishing structural relationships and searching for regularities and combining them into concept variants [Reffat, 2002]. In AI systems in general, concept invention has been implemented based on heuristics (e.g. in HR), parallel processes (e.g. in Copycat), evolutionary techniques (e.g. in NevAr), to name a few. The main argument here is that in neither case is the novel concept the *logical conclusion* from data analysis, but merely a bounded guess to be explored later.

To conclude, we must stress that, in practice, there is not such a strict separation between formation and invention (rather, there is a continuum). Every discovery involves conjecturing (i.e. inventing - or speculating about - new concepts yet to be proven), a process that has a great deal of its power in unsound processes, like aesthetics, intuition and free-association. In the systems referred to above (HR, AM and EURISKO), the conjecture generation step is fundamental and it is achieved with the application of production rules and heuristics to evaluate how interesting yet-to-be-proven concepts are.

The distinction between formation and invention could be reduced to a problem of constraint satisfaction: formation has stronger constraints to satisfy than invention,

[2]Abduction can be roughly described as the assertion of premises, given the truth of the conclusion. For example, with the rule $A \wedge B \rightarrow C$ and facts B and C, one can *abduce* (or assert the truth of the fact) A, which is not necessarily true.

which is ill-defined. However, a clearer distinction is needed since these correspond to two distinct, yet inter-dependent, steps of creativity: rationality and imagination. Once again, convergence and divergence. While rationality is more constrained, thus more limited but *computationally* implementable, imagination allows a world of possibilities that for a formal machine are hardly feasible, or even possible at all. Indeed it can be argued that for all AI modelling may become more undetermined and randomised, it is ultimately formal and deterministic, so a paradox arises here: shouldn't concept formation be everything when dealing with machines? We propose that, at least philosophically, we should consider these two forms of working with concepts. And, even if at the end we are doomed to determinism and formalism, we must not ignore imagination when attempting to model computational creativity.

3.3 Mixing concepts

3.3.1 Conceptual Combination

Conceptual combination is the process of combining two or more concepts together, often resulting in a novel concept with an emergent structure of its own. Although regarded as a universal cognitive process, the main motif of study in the conceptual combination community is language compositionality, more specifically interpretation of noun noun (e.g. "pet fish") and adjective noun (e.g. "blue cup") combinations. In this context, the first word ("pet", "blue") is called *modifier*, while the second is the *head* ("fish", "cup"). Four types of combination are proposed (see e.g. [Hampton, 1997], [Keane and Costello, 2001]): relational, property mapping, conjunctive and known-concept. Relational combinations establish some relationship between the modifier and the head (e.g. in "bed pencil", "a pencil that you *put beside* your bed for writing some messages" [Keane and Costello, 2001]); property mapping involves a property of one concept being asserted to the other (e.g. in "bed pencil", "a pencil *shaped* like a bed" [Keane and Costello, 2001]); conjunctive combinations conjoin both concepts in some way, the interpretation being both the modifier-concept and the head-concept (e.g. in "bed pencil", "a big, flat pencil that is a bed for a doll"); known-concepts or lexicalized compounds are those that are commonly used and established in communication (e.g. "pencil case"), sometimes effectively forming a single lexical unit (e.g. "railway" or "lipstick").

There are four main theories for conceptual combination: Abstract relations [Gagné and Shoben, 1997]; Dual-Process [Wisniewski, 1997]; Composite Prototype [Hampton, 1987]; and Constraints [Costello, 1997]. The abstract relations theory says that only a limited number of predicates can relate the modifier with the head

noun: CAUSE, HAS, MAKES, MADE OF, FOR, IS, USES, LOCATED, DERIVED FROM, ABOUT, DURING, and BY. The dual-process theory proposes two kinds of processes for conceptual combination, structural alignment and scenario construction. Structural alignment explains property and conjunctive interpretations (there is an alignment of attributes of both concepts), while scenario construction explains relational interpretations (e.g., "a night flight is a flight taken at night").

The other two theories are important for our work, so we will pay them more attention. The composite prototype model of James Hampton focusses on conjunctive combinations (e.g. "pet bird") and proposes that, when forming a concept such as "pets" that are also "birds", people take their prototype representations of "pet" and "bird" and combine the prototypes into a composite to represent the conjunction. This new concept then inherits its own attribute values from one or the other constituent parent according to certain principles. For example, the `location` slot for "pet" has the value *in the home*, while the same slot for "bird" has the value *in the wild*. "Pet bird" inherits the value from "pet" rather than from "bird". On other attributes, the opposite might happen (e.g. the slot `covering`: *feathered* - is inherited from "bird" rather than from "pet", where its most common value is *furry*) [Hampton, 1997]. Two influences on attribute inheritance are: the centrality of the attribute (e.g. the `location` of "pet", as staying at home, is central, while `color` is not); its possibility in the composite (e.g. the value *migrates* could not be possible in a "pet bird"). Another important aspect of the composite prototype model is *emergence*: attributes which are considered true of the conjunction, but *not* true of either constituent. It appears that a major source of emergent properties is simply knowledge of the world - or "extensional feedback" [Hampton, 1987]. Although the author argues that "we can not expect any model of conceptual combination to account directly for such effects" [Hampton, 1997], he presents two specific sources of emergence: exemplar-based, in which typicality of items in conjunctive categories can vary as a function of the kinds of exemplar found in those categories (e.g. a "small spoon" is typically made of metal, while a "large spoon" is not); theory-based, a background theory is applied to infer emergent attributes (e.g. a "beach bicycle" must have particularly wide tyres). All these ideas from James Hampton are also explored in Conceptual Blending, presented in section 3.3.2.

The Constraints theory of Fintan Costello and Mark Keane [Costello, 1997] describes conceptual combination as a process which constructs representations that satisfy the three constraints of diagnosticity, plausibility and informativeness. Diagnosticity requires the presence of diagnostic properties from each of the concepts being combined. The diagnostic properties of a concept are those which occur often in instances of that concept and rarely in instances of other concepts (similar

to salience [Milosavljevic and Dale, 1996]). The plausibility constraint requires the preference to semantic elements which are already known to co-occur on the basis of past experience. This constraint would predict that the interpretation "an angel pig is a pig with wings on its torso" would be preferable to "an angel pig is a pig with wings on its tail". Informativeness requires an interpretation to convey a requisite amount of new information. Informativeness excludes feasible interpretations that do not communicate anything new relative to either constituent concept; for example, "a pencil bed is a bed made of wood" [Keane and Costello, 2001].

Costello and Keane implemented a computational model of their theory. The system is named Constraints on Conceptual Combination (or C^3) and will be subject to a comparison with Divago in chapter 5.

As could be seen, conceptual combination is viewed as a cognitive process, although its analysis is usually constrained to a particular language, primarily to English. There are, however, aspects which are specific to some language or family of languages. Indeed, while Dutch and German also allow the same kinds of combinations, other languages such as Portuguese or French don't. In Portuguese, the use of prepositions (e.g. *de, para, etc.*) in combinations guarantees non-ambiguity. In principle, two consecutive nouns in Portuguese correspond to a conjunctive interpretation (and, rarely, to a property interpretation). Of the four theories presented, only the composite prototype model seems to be totally language independent, perhaps because it is directed to conjunctive combinations. This does not mean that the other theories are less valid or unrelated to cognition or creativity, rather we say that they favor problems at the level of language rather than at the level of concepts. In this sense, they are models of interpretation and natural language understanding, without paying attention to other domains like visual arts, music or scientific discovery. In these domains too, conceptual combination is constantly present, and is often shared across different fields and media (e.g. in the *Baroque* style, abstract concepts such as *ornamentation* or *luxury*, travel across the several fields). Thus, we conclude that these models are *a priori* limited as models of creativity (except for Hampton's model, since it focusses on generic concepts and considers emergence).

3.3.2 Conceptual Blending

The framework of Conceptual Blending (CB), also known as Conceptual Integration, was developed by Gilles Fauconnier and Mark Turner, and was initially motivated towards specific cognitive phenomena such as Metaphor, Metonymy and Counterfactual Reasoning (Fauconnier and Turner, 98). Blending is generally described as involving

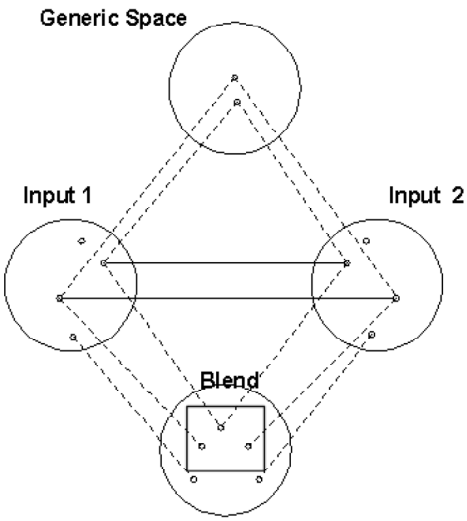

Figure 3.1: The standard, four-space model, of Conceptual Blending.

two input knowledge structures (the *mental spaces*) that, according to a given mapping, will generate a third one, called Blend. This new domain will maintain partial structure from the inputs and add an emergent structure of its own.

More recently Fauconnier and Turner propose CB as an explanation for various cognitive phenomena. This claim has been taken even further in the book "The way we think"[Fauconnier and Turner, 2002], in which the authors suggest their framework for explaining "the nature and origin of cognitively modern human beings". These bold claims incurred the voice of the critics, which we will summarise once we have explained CB in some detail. As the reader will see, we agree with these critics but we think that some of the essential components of Conceptual Blending should be given merit. Most of all, we think that CB as a framework, and in the way it is presented (mainly before [Fauconnier and Turner, 2002]), allows it to be considered by AI, and more specifically by Computational Creativity. The question now arises of whether it could be considered *per se* a model of creativity. According to our views given above (in chapter 2), we can hardly say that CB, alone, is a model of creativity: indeed, there is nothing deliberately dedicated to novelty in the framework; there is no proposed method for distinguishing whether a blend is or is not creative (e.g. "red pencil" is considered to have complex structure - does this make "red pencil" fcreative?); there is no clue for how or why a pair of inputs should be chosen to potentiate creativity. On the other hand, it proposes a many-step method for bisociation (that indeed follows a conceptual combination philosophy, such as proposed by James Hampton, briefly described above [Hampton, 1987]), which we consider as vital for the Creativity Model presented in this book. And, although formally undefined in so many issues, it is more detailed and explained than the other models (of combination)

studied. Thus, it is the combination of CB with an AI search method, combined with novelty and usefulness heuristics, that we will propose as a creative system.

As just shown, Conceptual Blending is not formally or algorithmically described in its fundamental details by Fauconnier and Turner. As far as we know, there are only a few formal accounts of this subject, apart from our own [Goguen, 1999, Veale and O'Donoghue, 2000, Lee and Barnden, 2001] and its formalization is clearly not given priority within the mainstream CB community (see discussion about computational modelling in [Fauconnier and Turner, 2002, p. 110]).

The computational realization of this model is definitely a big challenge since Conceptual Blending has many particularities that vary according to the situation, complex components like intuition, social behavior, expectation or common sense. In other words, there are several issues that are clearly hard to model. Yet, the intersection of AI and CB may bring, if not the computational model of the framework, at least methods or algorithms that may bring important contributions for the field of Computational Creativity, as we hope to demonstrate.

The framework

A blend is a concept or web of concepts whose existence and identity, although attached to the pieces of knowledge that participated in its generation (the inputs), acquires gradual independence through use. We often find a blend as being a concept that has the structure of other concepts, yet also having its own (emergent) structure. We find examples of blends in many sorts of situations. People have been making blends from at least the times of Greek mythology (e.g. Pegasus) till today (e.g. the Pokemon creatures). They are present throughout our daily communication (e.g. "John digested the book"), technological evolution (e.g. "Computer virus", "Computer desktop"), arts (e.g. Mussorgsky's "Pictures at an exhibition"; Kandynsky's "Improvisations"), advertising (e.g. Swatch is a blend of "swiss" and "watch"). The works of [Mandelblit, 1997], [Sweetser and Dancygier, 1999], [Coulson, 2000] and [Veale and O'Donoghue, 2000] are examples of how CB can contribute to Linguistics, Creative Cognition, Analogy and Metaphor.

The first fundamental element of Conceptual Blending is the mental space. A mental space is "a partial and temporary representational structure which speakers construct when thinking or talking about a perceived, imagined, past, present or future situation" [Grady et al., 1999]. "Mental spaces are small conceptual packets constructed as we think and talk, for purposes of local understanding and action. (..) [they] are very partial. They contain elements and are typically structured by frames. They are interconnected, and can be modified as thought and discourse

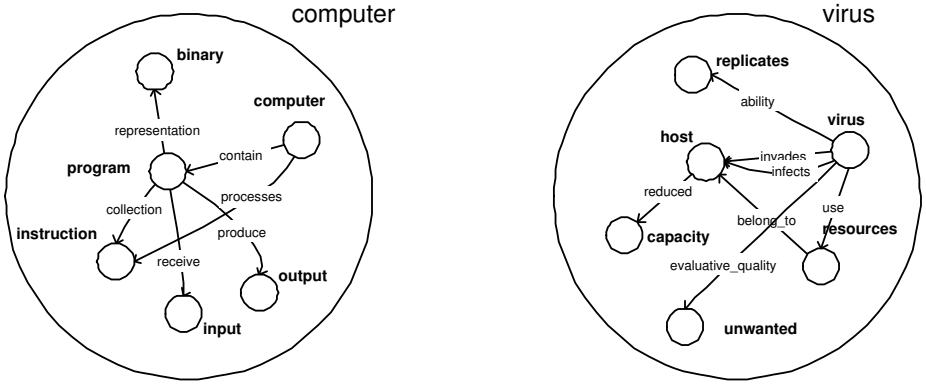

Figure 3.2: Two simple mental spaces for computer program and for virus.

unfold. Mental spaces can be used generally to model dynamic mappings in thought and language" [Fauconnier and Turner, 2002, 40]. From a symbolic AI perspective, a mental space could be represented as a semantic network, a graph in which we have nodes identifying concepts[3] (corresponding to the *elements* of a mental space) interconnected by relations. The definitions of mental space still allow many other representations (e.g. cases in Case-Based Reasoning, memes in Memetics or even the activation pattern of a Neural Network in a given moment) but these would certainly demand more complex computational treatment, especially with regard to the mapping. In figure 3.2, we show two possible mental space representations for "computer" and "virus".

In order to generate a blend, we must find mappings between the two mental spaces. We call these cross-space (or cross-domain) mappings. They connect elements of one mental space to others, in another mental space. A mapping may be achieved through different processes (e.g. identity, structure alignment, slot-filling, analogy) and doesn't have to be 1-to-1, i.e., an element may have more than one counterpart or it can have no counterparts at all. A possible mapping for the "computer virus" blend is shown in figure 3.3.

Another important notion is that of frames. When "elements and relations are organized as a package that we already know about, we say that the mental space is framed and we call that organization a frame" [Fauconnier and Turner, 2002, 102]. A frame is therefore a kind of abstract prototype of entities, actions or reasonings. For example, the mental space of "bus" could be organized according to the frame "transport means", while the mental space of "Mary's wedding" could be organized by the "marriage" frame. Of course, this reminds us of the frames and scripts from

[3]In this context, a concept is identified by a node, but its definition comes from its relationships with the other concepts, which is consistent with the Theory view given at the beginning of this chapter.

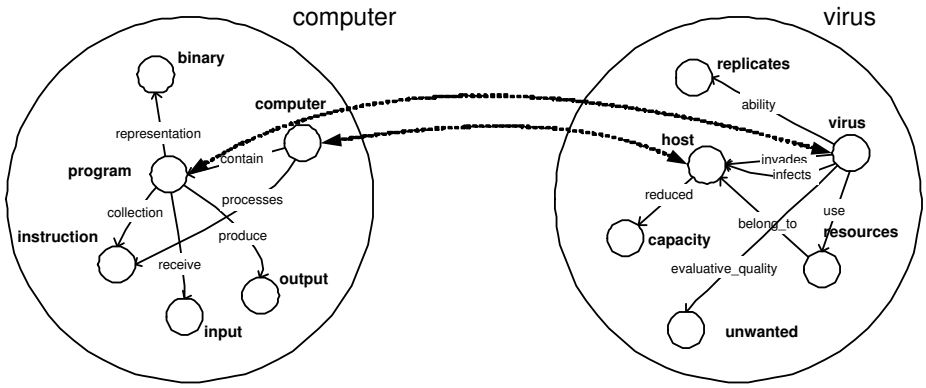

Figure 3.3: Cross-space mapping between the mental spaces of computer and virus.

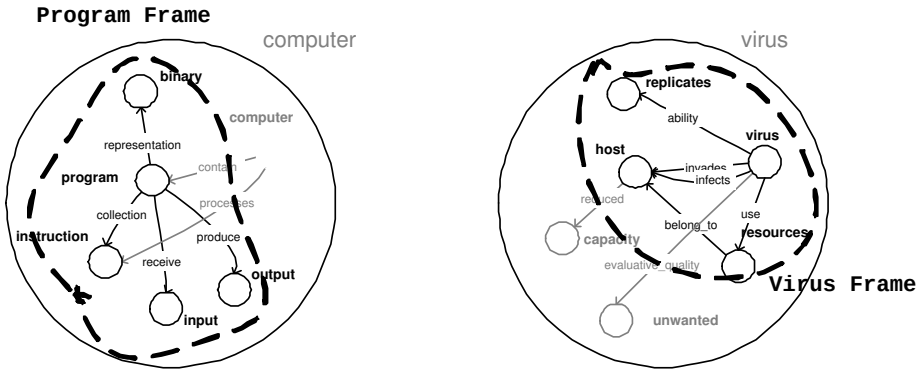

Figure 3.4: The organizing frames of the mental spaces of computer and virus.

early AI research but, in this case, these frames may be dynamic (they can change with time, individual and context, as with concepts in general) as well as compositional (there are many layers of abstraction for frames). We call the principal frame underlying a mental space *organizing* (e.g. "transport means" is the organizing frame of "bus", while "container" is not). In figure 3.4, we present two organizing frames of the "computer" and "virus" mental spaces (the `program frame` and the `virus frame`).

The frames preserve order in the blend, in the sense that they guide the process of blend construction to recognizable wholes. This does not mean, however, that the blend will integrate one single frame. Sometimes, as in the example of "computer virus" (in figure 3.5), the blend will inherit structure from both frames. As we can see in this example, some elements may not be projected. In this case, the "input" and "output" elements of computer viruses are normally a lot more subtle (or hidden) than in the usual programs.

In CB, the generation of a blend takes three (not necessarily sequential) steps:

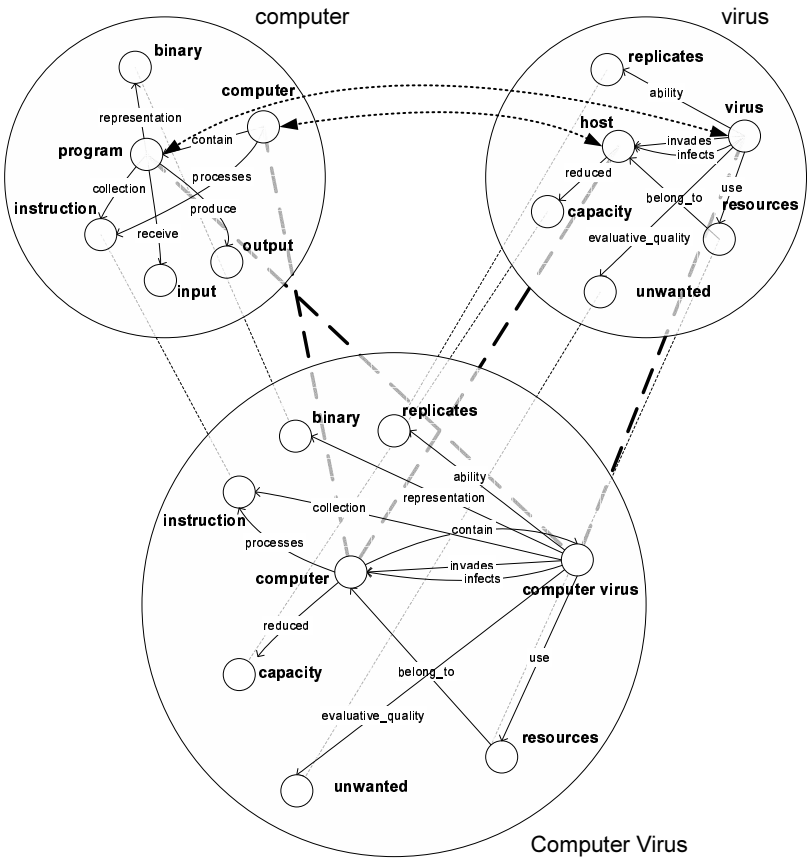

Figure 3.5: The computer virus blend.

- Composition.

  "Projection of content from each of the inputs into the blended space. Sometimes this process involves the 'fusion' of elements from the inputs (...)" [Grady et al., 1999]. Taken together, the projections from the inputs make available new relations that did not exist in the separate inputs. The paired elements are projected onto the blend as well as other surrounding elements and relations. This is a *selective projection*, i.e., some elements get projected to the blend, some don't.

- Completion. "The filling out of a pattern in the blend, evoked when structure projected from the input spaces matches information in long-term memory. In this way, the completion process is often a source of emergent content in the blend" [Grady et al., 1999]. Knowledge of background frames, cognitive and cultural models, allows the composite structure projected into the blend from the inputs to be viewed as part of a larger self-contained structure in the blend. The pattern in the blend triggered by the inherited structure is "completed" into the larger, emergent structure.

- Elaboration. "The simulated mental performance of the event in the blend, which we may continue indefinitely" [Grady et al., 1999]. The structure in the blend can then be elaborated. This is called "running the blend". It consists of cognitive work performed within the blend, according to its own emergent logic.

We illustrate the process of blending construction with another classic example, the "Riddle of the Buddhist Monk", which comes from Arthur Koestler's *The Act of Creation*:

> A Buddhist Monk begins at dawn one day walking up a mountain, reaches the top at sunset, meditates at the top for several days until one dawn when he begins to walk back to the foot of the mountain, which he reaches at sunset. Make no assumptions about his starting or stopping or about his pace during the trips. Riddle: Is there a place on the path that the monk occupies at the same hour of the day on the two separate journeys?

Following Koestler's solution to the riddle, Fauconnier and Turner suggest that it involves blending two different input spaces, one concerning the upward trip (day 'd_1') and another one concerning the downward trip (day 'd_2'). A generic space holds the commonalities between the two input spaces (a moving individual, his position, a path linking foot and summit of the mountain, a day of travel, and motion in an unspecified direction) [Fauconnier and Turner, 2002, p. 45]. The reasoning thus goes: first, composition of elements from the inputs makes relations available in the blend that do not exist in separate inputs. Only in the blend we have two individuals instead of one. Second, completion brings additional structure to the blend (e.g. the *frame* of "two people starting a journey at the same time from opposite ends of a path"). At this point, the blend is integrated: it is an instantiation of a familiar frame ("two people starting a journey..."). By virtue of this frame, we can now *run the blend*, i.e., elaborate it. In this case, it coincides to applying intuitive movement laws in opposite directions, which will make the two imagined monks, a_1 and a_2, eventually meet each other at some point, thus answering the riddle. This is what Fauconnier and Turner call *emergent behavior*. This reasoning is schematized in figure 3.6 [Fauconnier and Turner, 2002, p. 43]. This example also leads us back to the discussion of Conceptual Blending and Creativity above. CB does not actually *tells* us to blend the input spaces so that the monk performs both journeys on the same day. This is a crucial step to distiguish a creative from a non-creative solution (or a non-solution) to the puzzle. Rather, once we make this decision, CB allows us to perform the blend and determine the results (i.e., that a point does exist). In

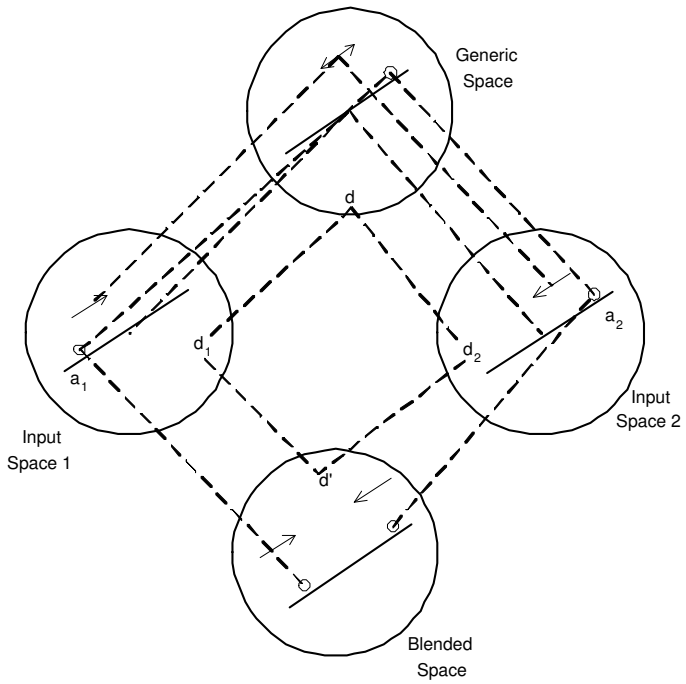

Figure 3.6: Blended spaces for the Riddle of the Buddhist Monk

fact, there is an initial step of high complexity (which inputs to choose to solve a problem?) that is so far not accounted by the CB framework.

In general, the projection of data to the blend is not exhaustive or predetermined in any way. Each element may be projected "untouched" to the blend, it may be "fused" with other concepts, it may be projected to another element (usually the projection of its counterpart) and it may not even be projected. This selective projection brings considerable complexity to the blending process because it raises the number of possible combinations to an extremely large number. This space of potential blends will certainly not be completely traversed for a new blend construction. In a quite different manner, projections are selected through a constraint-guided process of accommodation towards satisfying a set of *Optimality Principles* [Fauconnier and Turner, 2002]:

- Integration - The blend must constitute a tightly integrated scene that can be manipulated as a unit.
- Pattern Completion - Other things being equal, complete elements in the blend by using existing integrated patterns as additional inputs.
- Topology - Other things being equal, for any input space and any element in that space projected into the blend, it should be optimal for the relations of the element in the blend to match the relations of its counterparts.

- Maximization of Vital Relations - Other things being equal, maximize the vital relations in the network. Turner and Fauconnier identify 15 different vital relations: change, identity, time, space, cause-effect, part-whole, representation, role, analogy, disanalogy, property, similarity, category, intentionality and uniqueness.

- Intensification of Vital Relations - Other things being equal, intensify vital relations.

- Web - Manipulating the blend as a unit must maintain the web of appropriate connections to the input spaces easily and without additional surveillance or computation.

- Unpacking - The blend alone must enable the perceiver to unpack the blend to reconstruct the inputs, the cross-space mapping, the generic space, and the network of connections between all these spaces

- Relevance - Other things being equal, an element in the blend should have relevance, including relevance for establishing links to other spaces and for running the blend. I.e. it should have a *good reason* to exist.

These constraints work as *competing pressures* and their individual influence in the process should vary according to the situation; when the value of one grows, others may decrease. As far as we know, there is no work yet towards an objective study of the optimality pressures, measuring examples of blends or formally specifying these principles. This, we believe, inhibits considerably the appreciation and application of Conceptual Blending in scientific research, thus a lateral motivation for the work presented here becomes that of testing and specifying a formal proposal of these optimality pressures.

Among the many possible classifications of blending networks, Fauconnier and Turner particularly stress three kinds: mirror networks, single-scope and double-scope networks. A mirror network is one in which both input spaces share the same organizing frame, and so does the blend. In single- and double-scope networks, input spaces have different organizing frames. In a single-scope network, the blend has the organizing frame of only one of the input spaces, while, in a double-scope network, the organizing frame of the blend results from a combination of the inputs. The latter, as argued by [Fauconnier and Turner, 2002], is fundamental for human modern cognition and is deemed more creative. In the examples above, "computer virus" can be considered a double-scope blend, since its structure comes from a combination of the input's structures, while the "Buddhist monk" is a mirror network because both

input spaces share the exact same structure (both have a path, a monk, a direction, a mountain, etc.), the only difference being the direction.

CB is a promising contribution to modelling creativity since it consists of the generation of new concepts from integration of previous knowledge, with its own emergent structure (the whole is bigger than the sum of its parts); it has a domain independent evaluative methodology (e.g. the optimality constraints); it is consistent with processes often associated with creativity such as Metaphor and Conceptual Combination [Fauconnier and Turner, 1998, Veale and O'Donoghue, 2000, Coulson, 2000]; and, finally, it is not, by itself, a deterministic process, rather it is a context-sensitive process that considers "in parallel" a set of constraints that may interact and yield a (potentially large) varied space of equally valid solutions. In other words, with the same starting conditions, we may get many different results, all with the same overall value.

Since, in this book, we propose a computational model of Conceptual Blending, we will also apply some *established* examples. We call these *established* because they have often been cited in various articles or represent classic situations approached in Blending literature. They were taken from the main literature references (namely from [Fauconnier and Turner, 2002] and [Coulson, 2000]). In section 6, we will apply them for purposes of validation of our blending module. All the examples are described in Appendix B.

So far, there is little work on computational Conceptual Blending. We can only name three examples: Joseph Goguen's Algebraic Semiotics' approach [Goguen, 1999]; Veale and O'Donoghue's extension of Sapper to account for Blending [Veale and O'Donoghue, 2000]; and Barnden and Lee's approach to Conceptual Blending with counterfactuals [Lee and Barnden, 2001]. Joseph Goguen proposes to describe blending using the *algebraic semiotics* formalism. Algebraic semiotics is a formal theory of complex signs addressing interface issues, in a general sense of "interface" that includes user interface design, natural language and art. In the context of Semiotics, there are two views of what a *sign* is, from Saussure [Saussure, 1983] and Peirce [Peirce, 1958]. Goguen follows mainly Peirce's definition[4]:

> A sign (..) [in the form of a *representamen*] is something which stands to somebody in some respect or capacity. It addresses somebody, that is, creates in the mind of that person an equivalent sign, or perhaps a more developed sign. That sign which it creates I call the *interpretant* of the first sign. The sign stands for something, its *object*. It stands for that

[4]As cited in [Chandler, 2002]

> object, not in all respects, but in reference to a sort of idea, which I have sometimes called the *ground* of the representamen. [Peirce, 1958, p. 2228]

According to this model of sign, the traffic sign for "stop" would consist of: the red light facing traffic (the representamen); vehicles halting (the object) and the idea that a red light indicates that vehicles must stop (the interpretant). In the work of Goguen, a complex sign (or a sign system) is a sign that may have several levels of subsigns with an internal structure. He thus developed Algebraic Semiotics as being a computational treatment of sign systems. "Building on an insight from computer science, that discrete structures can be described by algebraic theories, sign systems are defined to be algebraic theories with extra structure, and semiotic morphisms are defined to be mappings of algebraic theories that (to some extent) preserve the extra structure" [Goguen, 1999]. Describing blends as being semiotic morphisms of sign systems (the input spaces), Goguen thus applies Algebraic Semiotics as a way of formalizing Conceptual Blending. More specifically, he argues that two category theory constructions, $\frac{3}{2}$pushouts and $\frac{3}{2}$colimits, give blends that are "best possible" in a sense that involves ordering semiotic morphisms by quality. Some examples of how this quality can be measured are:

- The most important subsigns of a sign should map to correspondingly important subsigns of its representation (more technically, this calls for preserving important sorts and constructors).
- It is better to preserve form (i.e., structure) than content, if something is to be sacrificed.
- The most important axioms about signs should also be satisfied by their representations.

We presented Goguens' work rather briefly and informally for much more space would be needed and the reader would have to become acquainted with algebraic semiotics and their application to Blending and we think that this is an unnecessary effort with regard to understanding our work. However, when presenting our model of blending, we will return to Goguen's formalization whenever there are similarities and discrepancies. There, we hope, the reader will also gain a better insight into Goguen's approach. As a final remark, we have to say that although this formalization should be credited as the first attempt to clarify some of CB's aspects with as much accuracy as possible, it leaves out some important issues, such as the optimality constraints, the actual processes of construction (composition, completion and elaboration) or selective projection. Some of these are theoretically accounted for (or indirectly implied by the formalization, such as the quality constraints example above, which can

be seen as optimality constraints), but they are not realized in any specific way (e.g. with specific processes for generating mappings and doing selective projection), which would certainly unravel many more problems. Finally it is noticeable that, in spite of efforts in this direction, the formal notation for blending given by Goguen has not been applied by the cognitive linguistics community, perhaps because it is too complex, or, more probably, because the community still finds it ineffective in studying blends.

Veale and O'Donoghue [Veale and O'Donoghue, 2000] present a computational model which relies on the metaphor interpretation system, Sapper (presented in the next section), to establish a dynamic blend between two domains. This blend, rather than being an independent new domain, corresponds to a unifying set of correspondences of concepts from both domains, built according to a constructor space. Therefore, although the authors argue to the contrary, this work misses the actual creation of the fourth space, the blend, which should have the same sort of structures as the inputs. It ends up being the set of *awakened dormant bridges* that are raised during the process (this will be thoroughly explained in section 3.4.2) which correspond to a mapping rather than being an independent new domain. The assignment of pairs of mapped elements doesn't necessarily imply a specific blend, as some elements may get projected, others get fused or be absent in the blend (selective projection). In our opinion, Sapper generates what we later will call a *blendoid*. As for emergent structure, it relies on "candidate inference" (transfer of structure from source to target), which is sufficient for laying the basis for novelty, but not for exploring it (i.e. there is no "running of the blend"). The authors also address the optimality pressures [Veale and O'Donoghue, 2000] as being by-products of the pressure to find a good isomorphism, but we think these are a set of varied perspectives, hardly reducible to a single measure[5]. Perhaps the weakness of their approach to blending is that the authors did not elaborate this idea more than in [Veale and O'Donoghue, 2000], nor give any practical results or detailed demonstrations, for then we would be able to contrast our proposals to theirs. As we will see, Sapper, more specifically its cross-space mapping mechanism, will also be useful for the work presented here.

Lee and Barnden [Lee and Barnden, 2001] focus the problem of reasoning with counterfactuals from the point of view of their ATT-Meta system [Barnden, 1998] (which will be briefly described in section 3.4.3), further analysing it from a perspective of Conceptual Blending. A counterfactual is the reasoning associated to expressions of the form "If S1 then S2" (e.g. "If John had sold his shares then he would have made a profit.") and it implies *a priori* contradictions when making

[5]As the reader will see in the conclusions, we also consider the reduction of the number of principles to three: Relevance, Topology and Integration.

straight truth assertions of the constituents S1 and S2 (it must be true that "John did not sell his shares", otherwise the "if" condition would not be necessary). Thus, to check the truth value of a counterfactual, one has to reason by hypotheses (for which ATT-Meta is well suited). Starting by a simple example, such as the one just given, and explaining how it could be processed via ATT-Meta's mechanisms, the authors then proceed to more complex situations that would imply a conceptual blending, such as "If Julius Ceasar was in charge of the Korean War then he'd have used the atom bomb". This would imply the blending of "Roman Empire" and "Korean War" domain knowledge into the same space prior to analysing the counterfactual in the same fashion as the simpler examples. The authors do not explore in depth any of the mechanisms of Conceptual Blending described here, as it is not the focus of their work. Conversely, we will not focus on counterfactual reasoning throughout this book since it would imply a deviation from its main objectives. However, there is no reason to believe that its results should not be extended towards this type of reasoning.

Some criticisms

We will now give special attention to the weaknesses and problems of the Conceptual Blending framework. It is noticeable that the Conceptual Blending framework is still ongoing research, possibly in its early versions. Naturally, it has been subject to some criticisms and its evolution to the next stages of the cycle of research certainly depends on searching for valid answers to these concerns .

The first and most obvious weakness of Conceptual Blending is its vagueness and lack of formality across its many aspects. Starting from the notion of mental space, it is unclear what it is exactly, to which extent it is cognitively plausible (it should be plausible, given the claims of CB as fundamental to cognition). Indeed, the reader might have felt some discomfort with the definition we gave, and which we tried to clarify with AI examples. This problem of definition of mental spaces becomes clear when discussing domains and frames. In some examples, we see the blending of domains, in some others (as happens in Seana Coulson's book [Coulson, 2000]) of frames, without understanding why these are not just named as "mental spaces". If there were a more clear notion of mental spaces, perhaps the Optimality Principles would become less vague (see, for example the explanation for "Intensification of Vital Relations"[6] above). This obscurity is carried over to the whole methodology that the authors use to deconstruct blends. There seems to be no specific set of rules for analyzing a blend other than intuition. A clear sign of this can be observed in

[6]Of course, in [Fauconnier and Turner, 2002], this constraint is more explained, but from the beginning to the end we keep confused with the lack of clear difference to Maximization of Vital Relations.

the generic space shown in our examples (see Appendix B), which sometimes is a set of generalizations, sometimes it represents specific knowledge, and other times it is just absent. Each new example may yield different analysis from different people, which compromises the predictability of this framework. This, of course, shows more in complex blends, which we tried to reduce to a minimum in the examples given (in fact, being subjective as little as possible was one of the main restrictions for selecting examples for experiments). In simpler blends, when it is clear that we are *indeed* faced with a blend (e.g. "computer virus", "Riddle of the Buddhist Monk", Pegasus, Dracula), the framework does not seem to be so controversial. Nevertheless, it is still not clear how to distinguish a simple from a complex blend and, even worse, it is not clear how to distinguish a blend from a non-blend. Ultimately, it seems that everything that has a symbolic meaning is a blend (e.g. sign language, money, machine dials, etc.), which of course leads to very extreme claims. This reasoning falls into the same category as the claim that "every language is metaphoric", also a relativist perspective that, although maybe philosophically interesting, only risks to sterilize its development unless a valid paradigm shift is made (i.e. "since every language is metaphoric, let us develop a different theory of language, and demonstrate its validity").

All these problems lead to the issue of falsifiability. Since the Conceptual Blending framework does not predict the more complex blends and the distinction between what is and what is not a blend is obscured, it is in principle not falsifiable, and therefore not a theory, in a modern science sense[7]. Assuming the different perspective of research programs from Lakatos[8], the framework of Conceptual Blending could be considered a research program, although its belt of auxiliary hypotheses needs to be more formally defined. In other words, these auxiliary hypotheses still need to be falsifiable.

These criticisms are intended to encourage work that, from our point of view, is fundamental and also motivates us. We do not promise to give any perfect blending machine or even to demonstrate that ours is the formalization of the whole Conceptual Blending framework. To do so, it would be necessary to solve the problems described above (what is a mental space, a frame or a projection; what should the optimality

[7] A fact or a theory being *falsifiable* means that it can, in principle, be proven false. According to Karl Popper [Popper, 1959], a theory that explains a phenomenon must be falsifiable through an experimental result that is implied by the theory. One can justify a theory rationally if it is not (yet) falsified.

[8] A research program consists of a theoretical core which is protected by a *belt* of auxiliary hypotheses. When the theory is falsified, an auxiliary hypothesis should be reconsidered, not the theoretical hard core. A revolution is explained as a change in the theoretical hard core [Lakatos, 1978]. Lakatos elucidated the activity of science not as the project of trying to refute one theory but as investigating empirical phenomena within the theoretical frame of a research program (in [van den Bosch, 2001]).

principles and the selective projections be about) prior to dedicating an entire thesis to it. Here, we will propose a computational level answer to some of them.

In spite of all these criticisms, it has been claimed that Conceptual Combination theories (such as presented earlier in section 3.3.1) cannot predict more than Conceptual Blending does, i.e., the same level of predictions with noun noun combinations can be done with CB (see [Coulson, 2000]). Indeed, as said by James Hampton, only a small set of emergent features can be predicted by theories, which limits predictability to the more constrained and closed world situations (i.e. in an ideal, yet unrealistic, scenario, we have two concepts defined with a universally accepted and stable representation). As an analytical model, it can become productive, as the examples of [Mandelblit, 1997], [Sweetser and Dancygier, 1999], [Coulson, 2000] and [Veale and O'Donoghue, 2000] show how CB can contribute to Linguistics, Creative Cognition, Analogy and Metaphor.

To conclude, Conceptual Blending is as an elaboration of other works related to creativity, namely Bisociation (in section 2.1.2), Metaphor (in the following section) and Conceptual Combination[9]. As such, it attracts the attention of computational creativity modelers and, regardless of how Fauconnier and Turner describe its processes and principles, it is unquestionable that there is some kind of blending happening in the creative mind.

3.4 Metaphor and Analogy

Metaphor and analogy are two cognitive mechanisms that have been recognized as underlying the reasoning across different domains[10]. Because of this, they play an indomitable role in creativity and must be discussed here. Although no consensus has been reached in the current literature regarding a clear distinction between metaphor and analogy, it is clear that their mechanics share many commonalities. It is widely accepted in analogy research that many of the problems of metaphor interpretation can be handled using established analogical models, such as the structure mapping approach [Gentner, 1983]. Thus, we present a set of works that involve mapping across distinct domains, namely SME (section 3.4.1) and Sapper (section 3.4.2). Although only the latter has been of direct influence to our work, SME deserves particular

[9]Another common criticism is that the authors of Conceptual Blending have not given fair credit to preceding works, such as James Hampton's Composite Prototype Model [Hampton, 1987], which, as can now be understood, stated the same general lines followed ten years later by Conceptual Blending, although it is also fair to acknowledge that Fauconnier and Turner have made much deeper explorations into the same issues.

[10]This claim is nowadays widely agreed, as metaphor is seen as a cognitive rather than a linguistic device. For an extensive *figurative versus literalist* analysis, we redirect the reader to [Veale, 1995]

attention for it has been the main reference in Analogy in the past few years and was the starting point and the benchmark for other systems, which will also be covered in an overview (section 3.4.3).

3.4.1 Structure Mapping Engine

The Structure Mapping Engine (SME) of Dedre Gentner, Kenneth Forbus and Brian Falkenhainer [Falkenhainer et al., 1989] was initially built as a computational implementation of the Structure Mapping Theory (SMT) of Dedre Gentner [Gentner, 1983][11]. In this theory (as generally accepted in the field), analogy consists of a mapping of knowledge from one domain (the base) into another (the target) and may be used to guide reasoning, to generate conjectures about an unfamiliar domain, or to generalize several experiences into an abstract schema. Moreover, SMT is based on the intuition that analogies are supported on relations: "No matter what kind of knowledge (causal models, plans, stories, etc.), it is the structural properties (i.e., the interrelationships between the facts) that determine the content of an analogy". Thus, analogical processing is decomposed into three stages [Falkenhainer et al., 1989]:

1. Access: given a current target situation, retrieve from long-term memory another description, the base, which is analogous or similar to the target.

2. Mapping and Inference: construct a mapping consisting of correspondences between the base and target.

3. Evaluation and Use: estimate the 'quality' of the match. Three kinds of criteria are involved: the structural criteria include the number of similarities and differences; the second criteria concerns the validity of the match; the third criteria is relevance, i.e., whether or not the analogy is useful to the reasoner's current purposes.

SME deals only with the Mapping and Inference stage (although also providing a domain-independent structural evaluation). In terms of knowledge representation, it differentiates between *entities*, *predicates* and *dgroups*. Entities correspond to the lower level objects or constants; predicates are higher-level primitives of three sorts (functions, attributes and relations); and dgroups correspond to a collection of entities and predicates about them. Below, we give an example of a dgroup named *simple-heat-flow*.

[11] It is important to give reference here to the works of P. Winston[Winston, 1980] and T. Evans [Evans, 1968] which laid the foundations for the study of Analogy in AI, particulary the Structure Mapping approaches.

```
(defDescription simple-heat-flow
    entities (coffee ice-cube bar heat)
    expressions (((flow coffee ice-cube heat bar) :name hflow)
                ((temperature coffee) :name temp-coffee)
                ((temperature ice-cube) :name temp-ice-cube)
                ((greater temp-coffee temp-ice-cube) :name >temperature)
                (flat-top coffee)
                (liquid coffee)))
```

To clarify, we give an example of each kind of representation: *coffee* is an entity, *flow* is a relation, *temperature* is a function and *liquid* is an attribute. It is clear that an attribute can be represented as a relation (e.g. (property coffee liquid) is the same as (liquid coffee)).

SME establishes potential cross-domain linkages via *match hypothesis construction rules* (MHC). These rules are programmable externally and specify the conditions that must be met in order to create a cross-domain linkage (known as *match hypothesis*(MH)).

SME constructs the cross-domain mapping by calculating the largest, maximal collection of MH's. "A collection is maximal if adding any additional match hypothesis would render the collection structurally inconsistent"[Falkenhainer et al., 1989]. Being structurally consistent means that (i) the MH's do not assign the same base concept to multiple target concepts; (ii) if a match hypothesis MH is in the collection, then so are MH's which pair up all of the arguments of MH's base and target concepts[12]. Collections are called *gmaps*, each containing, apart from a set of MH's, a set of candidate inferences (new relations that will be projected to the target, if the gmap is chosen) and an evaluation score.

SME was able to solve many *classic* analogy problems, such as the Solar-System - Rutherford Atom analogy or the Heat-Water flow analogy. In figure 3.7, we show its interpretation for the Heat-Water flow analogy. It can be seen that Gmap#1 produced the correct inference that *temperature causes heat flow*.

The major problem with SME is its intractability with unstructured representations as pointed out in [Veale and Keane, 1997]. As the authors of SME acknowledge, "worst-case performance occurs when the description language is flat (i.e., no higher-order structure) and the same predicate occurs many times in both the base and the target" [Falkenhainer et al., 1989]. There are two major reasons for this last problem:

[12]I.e. if we have MH(A,B) and a relation r, with (r A C) and (r B D) then the collection must also contain MH(C,D)

1. Run MHC rules to construct match hypotheses.
2. Calculate the Conflicting set for each match hypothesis.
3. Calculate the EMaps (collections of entity matches) and NoGood sets (collections of conflicting entity matches) for each match hypothesis by upward propagation from entity mappings.
4. Merge match hypotheses into gmaps.
   (a) Interconnected and consistent.
   (b) Consistent members of same base structure.
   (c) Any further consistent combinations.
5. Calculate the candidate inferences for each gmap.
6. Score the matches
   (a) Local match scores.
   (b) Global structural evaluation scores.

Table 3.1: Overview of the SME algorithm

the knowledge representation is predicate centered and therefore, when predicates are not hierarchically structured, it becomes flat, even when entities are themselves structured (e.g. a description of a multi-part object); SME makes an exhaustive search to obtain the largest mapping. In an attempt to overcome these problems, Forbus and Oblinger proposed a sub-optimal Greedy version of SME [Forbus and Oblinger, 1990], which applies their *greedy merge* algorithm. Roughly speaking, this algorithm iteratively cuts the search tree whenever it finds *high quality* local maxima. This quality is given by the pmap's "structural evaluation score". The authors considerably optimized SME's algorithm, sacrificing its optimal results. However, again, in comparison with Sapper [Veale and Keane, 1998], the algorithm efficiency remains far from being computationally feasible.

3.4.2 Conceptual Scaffolding and Sapper

Tony Veale developed a model of metaphor interpretation centered on the *contemporary theory* [Lakoff and Johnson, 1980], which proposes to explain metaphors via a process of structuring one concept (the tenor) with knowledge from another concept (the vehicle), with the purpose of (i) emphasizing certain associations of the tenor over others ("my dentist is a barbarian'"); (ii) enriching the conceptual structure of the tenor by analogy with another domain ("the CPU is the brain of the computer"); (iii) conveying some aspect of the tenor which defies conventional lexicalization ("the leg of the chair", "the neck of the bottle")[Veale, 1995]. The most revolutionary assumption from the contemporary metaphor theory is that metaphors belong to conceptual classes that are deeply entrenched in our world experience. Examples of metaphorical concepts include 'ARGUMENT is WAR' ("I will defend myself against his claims"),

```
Rule File:  literal-similarity.rules Number of Match Hypotheses:  14

Match Hypotheses:
  (0.6500 0.0000)   (>PRESSURE >TEMP)
  (0.7120 0.0000)   (PRESS-BEAKER TEMP-COFFEE)
  (0.7120 0.0000)   (PRESS-VIAL TEMP-ICE-CUBE)
  (0.9318 0.0000)   (BEAKER-6 COFFEE-1)
  (0.6320 0.0000)   (PIPE-8 BAR-3)
  o o               o
  o o               o

GlobalMappings:
  Gmap#1:  (>PRESSURE >TEMPERATURE) (PRESSURE-BEAKER TEMP-COFFEE)
           (PRESSURE-VIAL TEMP-ICE-CUBE) (WFLOW HFLOW)
      Emaps:  (beaker coffee) (vial ice-cube) (water heat) (pipe bar)
      Weight:  5.99
      Candidate Inferences:  (CAUSE >TEMPERATURE HFLOW)

  Gmap #2: (>DIAMETER >TEMPERATURE) (DIAMETER-1 TEMP-COFFEE)
           (DIAMETER-2 TEMP-ICE-CUBE)
      Emaps:  (beaker coffee) (vial ice-cube)
      Weight:  3.94
      Candidate Inferences:

  Gmap #3: (LIQUID-3 LIQUID-5) (FLAT-TOP-4 FLAT-TOP-6)
      Emaps:  (water coffee)
      Weight:  2.44
      Candidate Inferences:
```

Figure 3.7: Complete SME interpretation of Heat-Water flow analogy(from [Falkenhainer et al., 1989])

'TIME is MONEY' ("She wasted hours in solving it"), or 'SAD is DOWN' ("Don't let yourself down"). Moreover, most non-trivial metaphors can be reduced to complexes of simpler core metaphors, grounded in our spatial understanding of the world (e.g. 'ABSTRACT STATES as LOCATIONS': "Bill went mad", "Suzie went to sleep"; 'TIME as a LOCATION': "at 5 o'clock", "in March").

In his model, Veale proposes two different, subsequent, steps for metaphor interpretation: extraction of a *conceptual scaffolding* between the ideas evoked by a metaphoric utterance, by identification of underlying core metaphors; establishment of relations between those ideas for extensive explanation of the metaphor, involving cross-domain transference (*Sapper*). The former works top-down, while the latter, bottom-up.

The essential role of conceptual scaffolding is to build a skeleton that will be the basis for Sapper in the search for plausible relations using domain knowledge. Therefore, the relations built during scaffolding are not intended to capture all the subtleties and nuances of meaning in a metaphor, rather, they are generic guidelines for constraining the interpretation. The scaffolding model takes two distinct

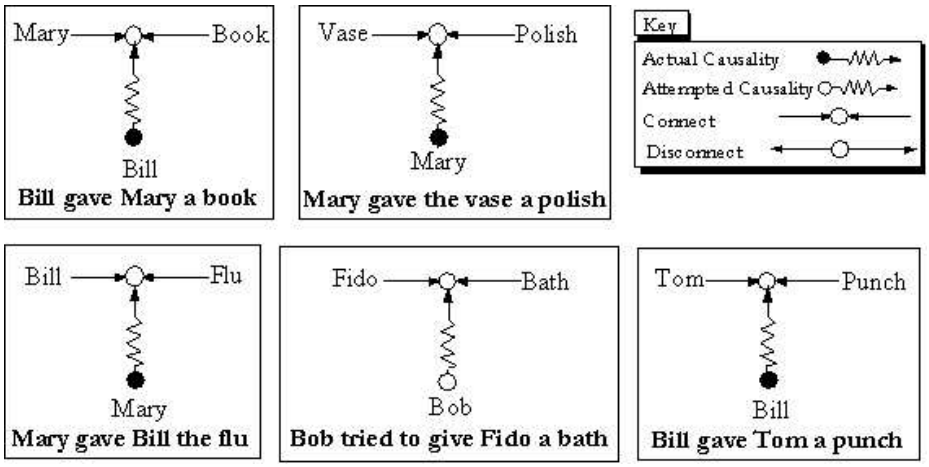

Figure 3.8: Example of the *scaffolding construction* with four primitives (*Actual* and *Attempted Causality*, *Connect* and *Disconnect*)

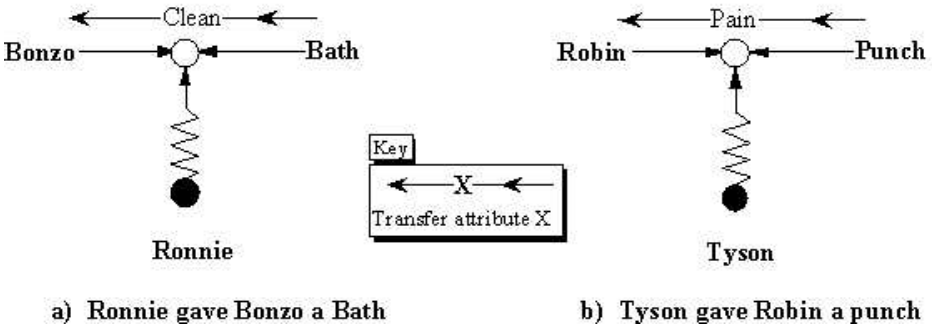

Figure 3.9: Example of the *scaffolding elaboration*

phases: scaffolding construction followed by scaffolding elaboration. Scaffolding construction is spatial in nature, interrelating the elements of the utterance through the use of primitive building blocks (see figure 3.8). The composition of this structure is obtained from the verbs involved, but other elements, such as adjectives (e.g., Big versus Small, Lightweight versus Heavy), and explicit conceptual relations (e.g., Father, Partner and Manufacturer) also contribute to the scaffolding.

Scaffolding elaboration labels the associations just described with particular inter-concept relation, such as *Colour* for *connect(Porsche, Black)* and *Manufacturer* for *connect(Macintosh, Apple-inc)*. These relationships are derived from the interaction of the concepts governed by the scaffolding structure (see figure 3.9). The appropriate relationship or case is thus dependant upon the nature of the concepts involved.

Thus, conceptual scaffolding builds a network of relations and concepts that correspond to the skeletal interpretation of an utterance. The metaphor interpretation is only completed after the creation of cross-domain links between the concepts, which

will allow the transfer of knowledge and finally, the explication of the metaphor. This is done by Sapper.

Sapper works with a semantic network that contains the information brought from the conceptual scaffolding, also enriched with background domain knowledge. It applies a spreading activation based process in order to determine novel cross-domain relations, thus reproducing much of the connectionist philosophy in a symbolic framework. These are normally called localist networks, since a distinct unit, or fixed cluster of units, is assigned to each concept, and an activation-carrying inter-unit linkage is assigned to each inter-concept relation (of the appropriate conductivity to capture the salience of the relation). There are two generic aspects regarding Sapper memory that are the most important for our purposes:

- The representation of all knowledge is equal. If not given a specific context, all concepts and relations are equally relevant (or irrelevant), i.e. there is no built-in hierarchy or ordering to organize the memory.
- Activation flow is entirely opportunistic. The most activated concepts will be those that happen to be in the spots where the activation waves are higher and in larger number, independently of what the concept actually is or means. In other words, again, there is no *a priori* preference for concepts or relations.

Since concepts are intrinsically dynamic, their representation should also be dynamic and impartial. In other words, one concept can play a central role and have a particular meaning in one context and be lateral and have a different meaning in another context, thus it is the situation that shapes it, not its representation. This is a well-known problem in AI, and it is clear that Sapper does not solve it except at the very specific level of cross-space mappings, as we will see below.

Sapper has two modes of processing, which interchange constantly as the cross-domain *bridges* are established:

- Structural inference is performed in a symbolic mode of processing,
- Opportunistic activation flow occurs in the connectionist mode of processing, in which particular relations (the *dormant bridges*, as laid down in the symbolic mode) are recognized to represent domain crossover points between the tenor and vehicle schemata, and are thus *awakened*.

The structural inference is based on two rules[13]: *triangulation* and *squaring*. The triangulation rule states that: "whenever two concepts share an association with a

[13]There is also the *incremental* rule, which is only applied in successive runs of Sapper, taking advantage of previous traces, but this is only a lateral subject here.

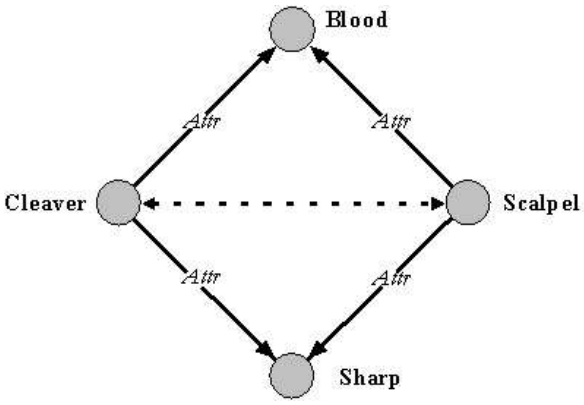

Figure 3.10: Example of the triangulation rule. Dashed arrow represents a *dormant bridge* (from [Veale, 1995])

third concept (the associations may be of different strengths), this association provides evidence for a plausible (i.e. dormant) bridge between both schemata"[Veale, 1995]. In figure 3.10, we see a double application of this rule: *cleaver* and *scalpel* share an association with both *blood* and *sharp*, leading to the establishment of a dormant bridge.

The squaring rule states that, when two concepts A and B linked by a cross-domain bridge (i.e. a dormant bridge which has already been awakened) share the same association with two different concepts C and D (resp.), then this association provides evidence for a plausible (i.e., dormant) bridge between C and D. This rule depends on the bridges having already been awakened (under the connectionist mode), thus it is considered *second order*. In figure 3.11, a dormant bridge is created between *general* and *brain surgeon* by the application of the squaring rule, since there is a cross-domain bridge between *command centre* and *brain*.

In the Sapper model, activation travels in waves, each wave-form having an amplitude (encoding the activation energy, or zorch) and a unique signature frequency. Each localist concept node is considered to possess a unique prime resonant frequency, which is used to modulate any activation waves that pass through this node (see fig. 3.12).

This propagation strategy therefore allows activation waves to be *deconstructed* via prime factorization. This deconstruction reveals not only the original source node of the wave (representing either the tenor or vehicle of the metaphor), but also the path through conceptual space travelled by the wave. When two activation waves cross over at the same inter-concept bridge, Sapper is thus in a position to determine whether the waves originate at different source nodes in the network. If this is indeed the case, Sapper awakens this bridge as possibly constituting a valid domain crossover

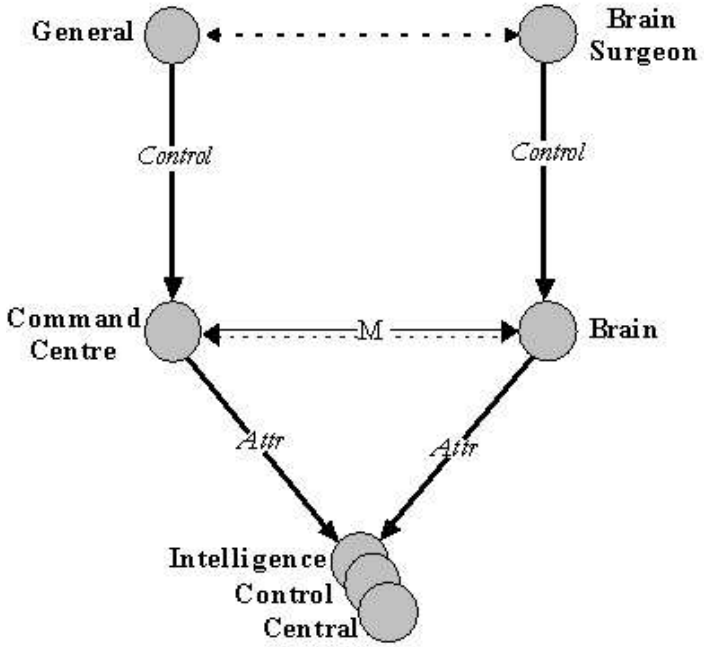

Figure 3.11: Example of the squaring rule. The bidirectional arrow with 'M' label represents a *cross-domain bridge* (from [Veale, 1995])

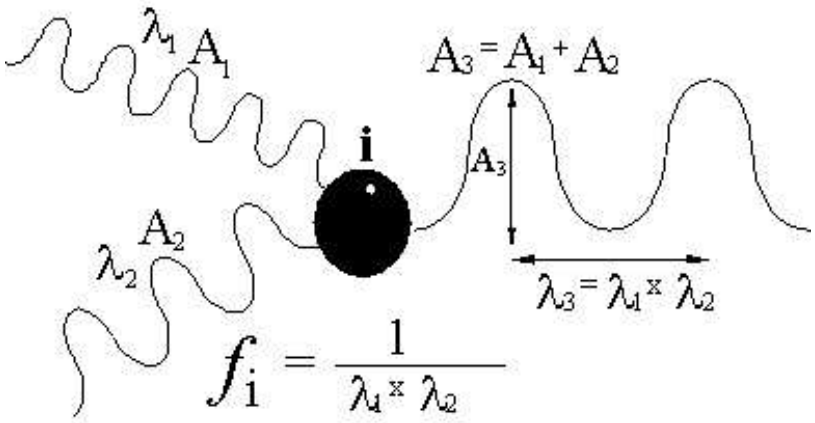

Figure 3.12: Activation Waves in Sapper possess both amplitude (or Zorch) and frequency. This signature frequency of an activation wave is the product of the resonant frequencies of those nodes encountered by the wave in the conceptual space. (from [Veale, 1995])

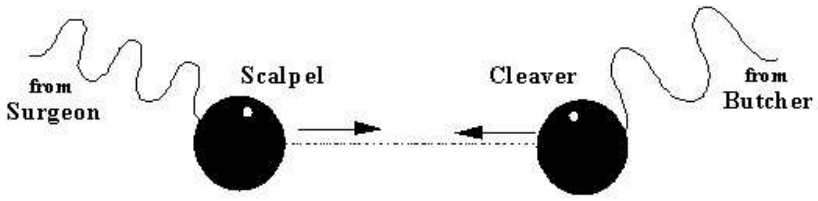

Figure 3.13: A conceptual linkage is deemed to provide a plausible match hypothesis when it becomes a cross-over path for competing activation waves from the tenor and vehicle concept nodes. (from [Veale, 1995])

0.98	If General is like Surgeon
0.76	Then 18th-Century-General is like Saw-Bones-Doctor
0.95	and Soldier is like Patient
0.81	and Casualty is like Corpse
0.91	and Bombing-Raid is like Surgery
0.93	and Atomic-Bomb is like Radiation-Therapy
0.75	and Nerve-Gas is like Disinfectant
0.88	and Enemy-Army is like Cancer
0.78	and Enemy-Soldier is like Cancer-Cell
0.5	and On-Target is like Precise
0.7	and Snub-Fighter is like Scalpel
0.5	and Military-School is like Medical-School
0.17	and Battlefield is like Operating-Theatre
0.17	and Military-Uniform is like White-Smock

Table 3.2: Output from Sapper for 'GENERALS are SURGEONS'

point (see fig. 3.13). [Veale, 1995, Veale and O'Donoghue, 2000]

There is one final point to add to this description of the Sapper algorithm. Inter-concept linkages also exhibit a certain resistance (the inverse of conductivity) to the flow of activation energy. This provides an attenuation effect in the amplitude of the waves at each node they encounter. When this amplitude drops below a predetermined threshold, it ceases to propagate. Dormant bridges have the highest (infinite) resistance, until they are awakened and attributed a resistance consistent with the structural evidence brought by the wave.

Finally, we give an example of the interpretation of "My surgeon is a butcher" generated by Sapper (figure 3.14) and the returned output for "The General is a Surgeon" (in table 3.2, figures on the left represent the conductivity of the cross-domain linkages).

To conclude, while Conceptual Scaffolding is knowledge dependent and needs detailed specification and coding of the core metaphors it analyzes, Sapper is an algorithm that finds a 1-to-1 mapping between two domains which, although not

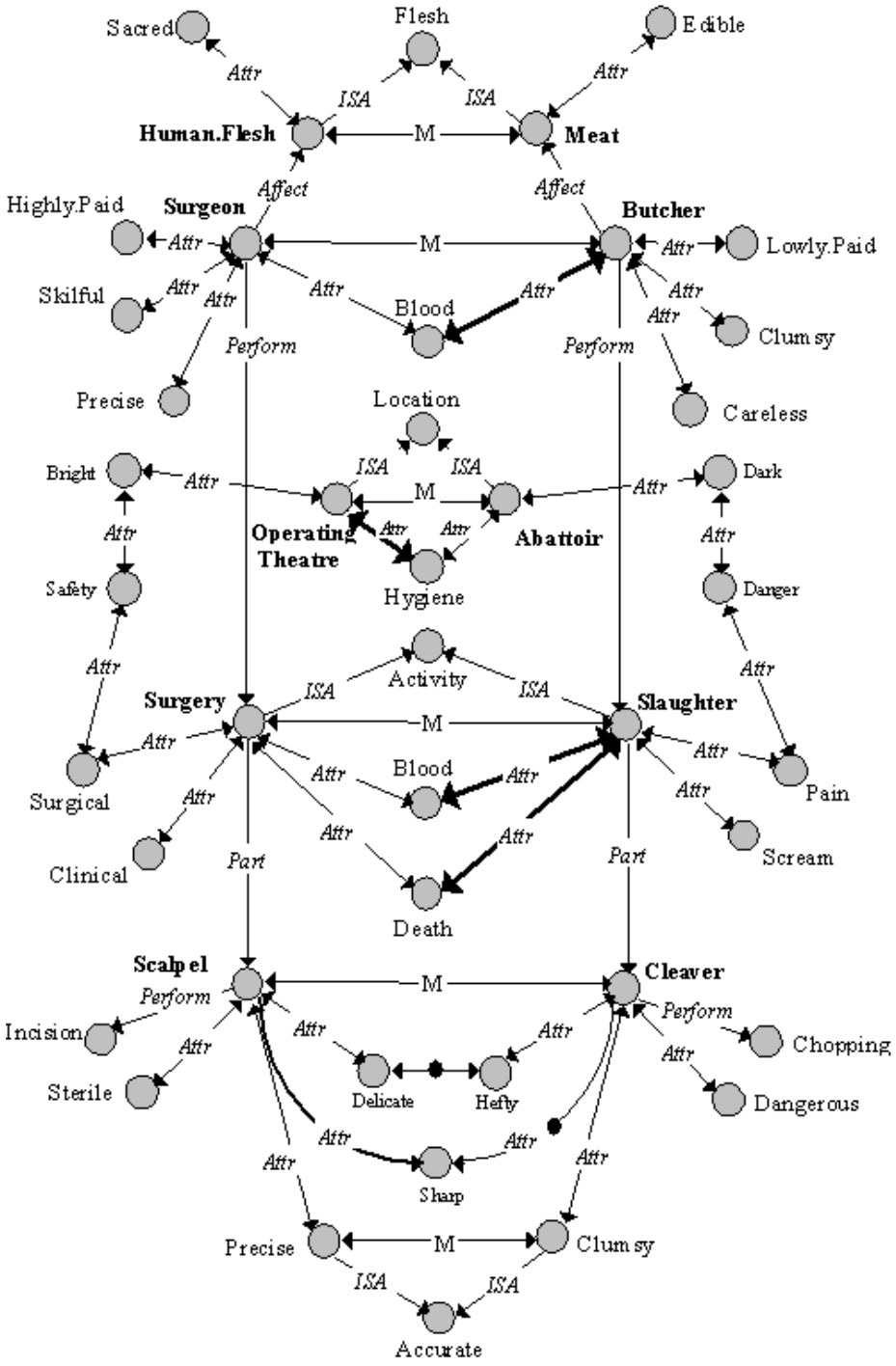

Figure 3.14: Interpretation for 'SURGEONS are BUTCHERS' (from [Veale, 1995])

guaranteeing the optimal solution (whatever the criteria chosen), is computationally tractable and does not demand big compromises regarding knowledge representation or configuration.

Veale and Keane [Veale and Keane, 1997] presented a very interesting comparison of Sapper with SME (Greedy and non-greedy versions) and ACME [Holyoak and Thagard, 1989] (described later). This analysis vividly reports how well Sapper behaves in comparison with its peers, namely concerning tractability issues. Apart from this paper, the interested reader should also inspect Ferguson et al's reply [Ferguson et al., 1997].

3.4.3 Others

As with other surveys (e.g. [French, 2002]), we classify systems on metaphor and analogy into three types: symbolic (ATT-Meta, SME and MIDAS), connectionist (LISA) and hybrid (Drama, Copycat and Sapper). Once again, this overview undoubtedly leaves out many systems. Our purpose is to provide a broad picture of the field, and eventually lead to a synthesis of the features and problems that characterize it.

ATT-Meta [Barnden, 1998] is a rule-based system for metaphorical reasoning initially projected to process mental states in discourse. It therefore focuses on specific types of metaphor schemata, such as 'MIND PARTS as PERSONS', and applies built-in commonsense models in order to interpret sentences like "One part of John was insisting that Sally was right". ATT-Meta then triggers rules that propose possible interpretations for the metaphor, according to different *pretences*[14]. It isolates pretences (thus avoiding logical inconsistencies) within *cocoons*, enabling the simultaneous consideration of several different, possibly conflicting, hypotheses. This system allows representation of uncertainty in its knowledge, which will then serve to evaluate the truth probability of a pretence (and propagating this probability to other, dependent, pretence cocoons) and propose a plausible interpretation. ATT-Meta does not itself deal with natural language input directly. Therefore, a user supplies hand coded logic formulae that are intended to express the literal meaning of small discourse chunks (two or three sentences)[Barnden, 1999].

Another work on metaphor reasoning is MIDAS [Martin, 1990]. As with Conceptual Scaffolding, MIDAS approaches interpretation with the assumption that there is a set of *core metaphors*. These are stored in a knowledge base that is continually augmented with extended metaphors, which derive from the core metaphors. Each metaphor is represented by a structure called *metaphor-sense*, which contains

[14]The system *pretends* that a given interpretation, however ridiculous (e.g. John having *literally* one person in each part of the mind), can be real. In doing so, the system is said to be "semantically agnostic".

a source, a target, and a set of associations. An association is represented by a *metaphor-map*, which links a source concept to a target concept. MIDAS interprets a metaphoric utterance by retrieving the most similar metaphor and adapting it to the current situation. In this sense, it works as a Case-Based Reasoning system, whose learning ability relies on the storing of newly adapted cases.

John Hummel and Keith Holyoak proposed an artificial neural-network model of relational reasoning, LISA (Learning and Inference with Schemas and Analogies) [Hummel and Holyoak, 1997], which uses synchrony of firing to bind distributed representations of relational roles (e.g., the roles of opposite-of(X, Y)) to distributed representations of their fillers (e.g., black and white). Thus, a proposition corresponds to a pattern of activation. LISA has a Working Memory (WM) containing the target (and the source, after retrieval) being investigated; and a Long Term Memory (LTM), which holds the candidate sources. When a target is specified in WM, its pattern of activation triggers the retrieval of the appropriate source proposition from LTM, which is the one that is better synchronized with the pattern of activation of that target. These two memories have distinct representations. WM comprises a distributed representation (as is traditional on pure connectionist system), while the LTM is localist (as with Sapper). For example, if the WM contains the target proposition "Beth sells her futon to Peter", then it may retrieve an analogous source proposition (e.g. "Bill sells his car to Mary (and so Mary owns the car)"). When two analogous propositions are met in LISA's WM, their co-mapped constituents are co-activated in synchrony (Bill to Beth, car to futon, etc.) and it is possible to transfer inference from the source to the target (i.e. "Peter owns the futon"), following the same activation procedure ("Peter" is co-activated with "Mary", who "owns" a "car", which is co-activated with "futon"). LISA's main limitations concern the WM memory sizes and LTM representation issues. Indeed, the WM can only store one proposition at a time, which forbids solving complex analogies. Moreover, the built-in representation of LTM makes LISA an uncreative system with low flexibility, since it demands the explicit coding of each proposition.

Drama is a system that aims to integrate semantic and structural information in analogy making [Eliasmith and Thagard, 2001]. It has a set of particularities that make it unique among its peers. The foremost is its application of *holographic reduced representations* (HRRs) [Plate, 1994] memories, which allow the distributed, vector-based, representation of concepts and relations in Drama. The storage operation of a vector in a HRR is called *convolution*, while the retrieval operation is called *correlation*. HRRs allow the convolution of large amounts of information in the same memory space, but the more they store, the lower reliability they will provide in correlation. It is then necessary an *error-cleaning* mechanism. It is claimed that

HRRs are cognitively plausible models of memory [Eliasmith and Thagard, 2001]. Other systems also apply distributed representations such as neural networks (e.g. LISA), so a thorough comparison should be made to understand which one is better in analogy contexts. In Drama, each ground concept is attributed a random vector that is then stored in a HRR, along with its semantic information (properties and ISA relations, each defined as an independent vector). Domain structure (relations between different concepts) is also stored in the HRR in the form of vectors. Given the source and a target proposition vectors, Drama starts the analogy-mapping by obtaining their similarity (via vector dot product). When they are sufficiently similar (higher than a threshold), it then proceeds to the constituents. For each pair of similar constituents, Drama builds a node in a network (the *mapping* network), and establishes links between nodes that participate in the same relation. This latter process is the same as ACME's [Holyoak and Thagard, 1989] algorithm for analogy mapping: in ACME, the algorithm starts by establishing a network of mapping pairs, each node containing a pair, each pair linked to other pairs. Using the LISA example above, ACME (or Drama) would initially build nodes for "sells" with "sells" then for "Beth" with "Bill", "car" with "futon", etc. each of these nodes being co-activated. It could also generate competing mapping nodes in the network (e.g. "Beth" with "Mary" - both are women) which would have little co-activation with the former nodes. Then, with a spreading activation process (as in Sapper), it would select the mapping sets that best satisfy the constraints of *similarity*, *structure* and *purpose*, as defined in [Eliasmith and Thagard, 2001] and [Holyoak and Thagard, 1989]. In theory, Drama can integrate both structure and meaning, which would be a major breakthrough in analogy research, but, since the ground concepts are given random vectors, the *meaning* is entirely dependent on the property and ISA relations, which end up as being structural knowledge as any other relation. Indeed, although the authors treat these relations differently, they do not correspond to the notion of *meaning* that they advocate. More specifically, the problem lies in the randomness of encoding the ground concepts and their resulting similarity (e.g. in one run, "dog" and "cat" can be more similar than "dog" and "freedom"; while in the following one, the opposite can happen without any particular reason). The authors claim that this is coherent with the psychological differences between people, but randomness does not seem to be a good model for it. A proper solution would be to learn the meanings, as also suggested in [Eliasmith and Thagard, 2001]. However, this learning algorithm is by itself a challenge.

Also unique in many aspects, Copycat [Hofstadter and Mitchell, 1988] is a system for solving puzzle analogies (such as "abc→abd::xyz→?") as already presented in section 2.2.2. This system has many nuances and has been deemed an example of

computational creativity, as well as its related family: Tabletop, Letter-Spirit and Metacat [Hofstadter, 1995]. Nevertheless, it has been criticized as only being able to work on a very specific, exhaustively defined domain. In fact, while an omnicompetent Slipnet is theoretically plausible, in practice, serious resources are necessary even for simple domains.

The issue of knowledge representation has been evoked constantly as a central problem in any of these systems. Some approach it by focusing on specific domains (ATT-Meta, Copycat), some try to cover generic knowledge (Sapper, SME, MIDAS, LISA, Drama). Some rely on structure mapping (SME, Sapper), some on axiomatic inference (ATT-Meta, MIDAS), some try integrating both (Drama, Copycat). Nevertheless, it is clear that each one is ultimately dependent on built-in domain representations (the exception being Drama, which solved the problem with randomness). The path now taken in the area seems to lead to hybrid approaches, both in terms of paradigm (symbolic and connectionist) and in terms of inference mechanism (axiomatic and structure mapping).

The majority of the works (with the exception of Copycat) allow only 1-to-1 structure alignment between domains, but it has been pointed out that many-to-one mappings may be useful, both in metaphor and in analogy [Falkenhainer et al., 1989]. Even more, structural alignment easily falls prey of the representation of domains. For example, having a source with "isa(dog, pet)" and "isa(pet, animal)", and a target with "isa(cat, animal)" and "isa(animal, entity)", it would not yield the analogy of "dog" with "cat", rather "cat" would be mapped to "pet", and a possible analogical inference could be "isa(dog, cat)". This once again raises the problem of representation. As we will see in chapters 5 and 6, the same questions can be raised for this work.

To conclude, there is general agreement that metaphor and analogy are cognitive mechanisms that can uncover aspects unforeseen, by bringing knowledge from one source to a target and thus making predictions, solving a problem or even by expressing concepts that have yet no conventional meaning. So far, the computational approaches to these mechanisms have essentially relied on cross-domain mappings and structure alignment, which is not to say that other approaches aren't worth of attention.

Chapter 4

A Model of Concept Invention

A Top-Down approach

In this chapter, we will take a top-down approach, presenting and discussing the requirements for an abstract creativity model, and then progressing towards a formal model of concept invention that will be the subject of an implementation presented in the next chapter. The reader should understand that, on the way to, and particularly when arriving at the actual implementations, many choices have to be made, both in terms of what aspects to focus on (e.g. bisociation *vs* re-representation) and in terms of practical decisions (e.g. implementing algorithms, choosing representations). We will try to justify each decision whenever any of these choice points arise.

4.1 A Creative General Problem Solver

In chapter 2, questions regarding creativity were raised. By doing so, we intended to provide the reader with the set of principles followed in the construction of the model for computational creativity proposed here:

- Knowledge. It has been emphasized that there is rarely creativity without knowledge and that both quantity and quality should be treated as equally important. A model for creativity should consider a heterogenous knowledge base, in the sense that it should not include solely the *typical* knowledge for solving a specific problem, but instead many different domains and perspectives towards more than one problem [Weisberg, 1999].

- Re-representation. It is also important to be able to understand the existing range of knowledge according to different points of view. A model for creativity

should be able to change the representation of a concept without losing its meaning [Karmiloff-Smith, 1993].

- Bisociation. The notion of bisociation is connected with cross-domain transfer, to the ability to find unprecedented associations. A model for creativity should be able to find and explore associations between distinct knowledge structures, namely structures that seem apparently distant and unrelated [Koestler, 1964].

- Meta-level reasoning. The ability to reason about reasoning is also a trace of creativity. This aspect is perhaps the most difficult to specify, but should also be taken into account. As well as being able to process the knowledge, a model of creativity should be able to process its own processes of processing knowledge, preferably without having to employ different techniques for each level of abstraction [Colton, 2001, Wiggins, 2001].

- Evaluation. An indisputable part of the creative process has to do with evaluation, both in terms of the self and of the society. A model for creativity should be able to do self-assessment and react to external evaluation [Csikszentmihalyi, 1996, Boden, 1990].

- Interaction with the environment. No model of creativity should be designed without taking into account the environment. Indeed, some researchers have emphasized that creativity can only be perceived against a context, which includes the individual (as a producer and recipient), the society, the History, the motivations, in other words a set of aspects that lie outside the scope of the new concept or idea being considered [Csikszentmihalyi, 1996].

- Purpose. There is always a purpose in any creation, even though it may be sometimes extremely subtle. We do not agree with the argument that a creative system does not have to be goal-oriented. Creativity happens as a necessity rather than as a purposeless activity, whether for satisfying some fuzzy aesthetic preferences or for solving a practical problem [Amabile, 1983].

- Divergence/convergence. One of the main conclusions taken from chapter 2 is the existence of two *modes of thinking*, the divergent and the convergent, both important to creativity in different aspects. Thus, a model of creativity should consider both divergent thinking, which is when free-association is sought, a less controlled search is allowed, constraints can be broken and inconsistencies may be generated; and convergent thinking, which is methodic and driven by rationality [Guilford, 1967].

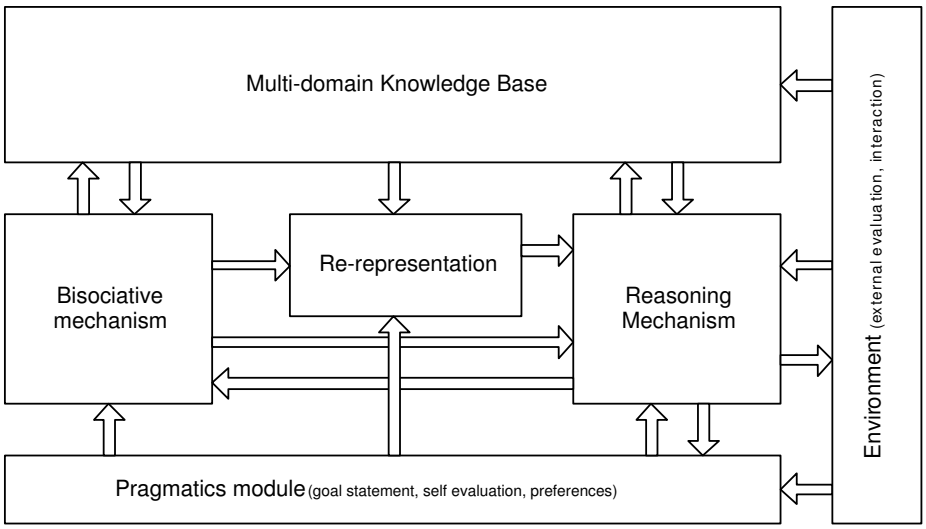

Figure 4.1: The Creative General Problem Solver

- Ordinary processes. A final aspect to raise is that there is no reason to believe that, underneath a creativity model, there need be processes that are special or fundamentally different from the ones applied in *non-creative* reasoning. Furthermore, there is no reason to argue that bisociation and divergent thinking are not grounded in the same cognitive processes as any other cognitive phenomenon or mode of thought. In other words, all these are manifestations of intelligence, with divergent thinking and bisociation being the ones that are more commonly identified with creativity [Guilford, 1967, Csikszentmihalyi, 1996, Koestler, 1964, Boden, 1990, Finke et al., 1992].

From these eight principles, we propose a Creative General Problem Solver (after an analogy to the General Problem Solver of Ernst and Newell [Ernst and Newell, 1969]). In doing this purely philosophical exercise, we have two intentions: to provide a model that summarizes all aspects and to focus on the relationship between Creativity and AI.

In figure 4.1, we show a model that considers the many aspects referred to above. The *reasoning mechanism*, perhaps the least obvious in terms of its internal workings, should be responsible for controlling the whole system and for doing the search according to the goal, preferences and evaluation given by the *pragmatics* and *environment* modules. The working mode of the system (either divergent or convergent) should depend on the use that the reasoning mechanism makes of the *bisociative mechanism* and *re-representation* modules. A purely convergent mode would not use any of these modules while a purely divergent mode would apply them for every step

in the search. The re-representation module provides different alternatives of representation of the knowledge in the *multi-domain knowledge base*, while the bisociative mechanism is expected to find and propose associations between any two distinct structures in that knowledge base. A particular aspect of the knowledge base (not explicit in the diagram) is that it should also contain the specification of every other module; in other words, the whole model is itself part of the knowledge base. This is definitely the most complex aspect and relates to meta-level reasoning. Indeed, if the system is to reason about its reasoning, its knowledge base must also consider, directly or indirectly, the representation of all its processes. The question is how this can be implemented and where this recursion ends.

Now, analyzing this model from an AI perspective, the basic question arises: Why do we call it creative? Is it not *yet another* AI model? Is it not *just* search over a complex space, which contains the solutions given by the *plain* knowledge base, added with its several different re-representations, the associations that result from the bisociation method, and the description of the processes themselves? From the (behaviorist) perspective of the product, one can call it creative *when*, from this search over a complex space, novel and useful ideas tend to emerge. From the (cognitive) perspective of the process, we can argue that, if the set of principles listed above are indeed connected to creativity, then such a model should produce more novel and useful solutions to a problem than if the same problem was given to a *classical* AI model (which would essentially have no bisociative mechanism, meta-level reasoning or re-representation module, and would have a much more focussed knowledge base). These proofs seem impossible to demonstrate formally, so we are left to explain exactly what such a model brings new to the area. Its possible contribution to AI is that it explicitly incorporates for the first time all those aspects mentioned, and, most of all, the assertion that creativity should be considered as one side of problem solving and, therefore, of intelligence. To clarify some more, such a model could solve a problem in a convergent manner, i.e. recurring to the knowledge specifically directed towards the problem (let us call it *problem-specific* knowledge), independently of how many times the same problem and solution had been brought up. It could also try to find a different solution, also with problem-specific knowledge, possibly seeming more creative. Solving in a divergent manner would imply the search for other associations and/or representing knowledge differently or even changing its own processes. In this case, when finding previously unseen solutions, the result may have a higher probability of being deemed creative by an observer. In any of these cases, we deal with a classical view of intelligence as problem solving, in some cases resulting in creative solutions.

The model just presented is very abstract and consists essentially of a set of guidelines to be explored further either by us (some are covered in more detail later in this book) or by others in future work.

In the next section, we descend one level more in our top-down approach to develop a creativity model. More specifically, we focus on a set of features already considered, namely bisociation, an heterogenous knowledge base, meta-level reasoning, convergent/divergent modes of thinking and purpose. These are the foundations for our model about concept invention.

4.2 Description of the Model

We now present our model of concept invention. It is focussed essentially on bisociation, an heterogenous knowledge base, meta-level reasoning, convergence/divergence and purpose. Therefore, we pay little attention to the aspects of interaction and evaluation and leave out re-representation. Interaction will be reduced to goal statement and configuration, and evaluation will be based on self-assessment. These choices result from a priority given to subject *versus* society, and therefore focussing on its inner processes. Because the issue of re-representation deserves an entire thesis to itself, we decided not to consider it further. Nevertheless, we will return to it when we think it relevant in this text.

Before giving details, we would like the reader to imagine an ideal scenario where a system has been given a goal to reach. This goal could be something like "the specification of a flying transportation object". The system has not enough knowledge of airplanes, or physics, or something that could lead by sound processes to reach directly this goal. Or it has indeed the necessary knowledge, but the complexity of the search space is too big to reach the goal in a reasonable amount of time. It could then enter a mode of divergence, in which combinations between concepts in memory would be made, always checking if something similar to the goal is achieved. After reaching the most promising idea (say, after spending a while in divergence, it had found "a bird with a box connected"), and if still not achieving the goal (e.g. "the box is too heavy"), this system would then return to a convergent mode, in order to elaborate the idea to reach a satisfactory solution. If, in the end, a good solution had still not been found, the system could also try to invent new ways of combining concepts, of elaborating, and of searching i.e. it would try to improve its own processes. Our model of concept invention is concerned with the divergent part

of this scenario. In figure 4.2, we show its diagram. It has six modules:

- Multi-domain knowledge base. The knowledge base follows the exact same principles as described for the Creative General Problem Solver.
- Bisociative mechanism. The bisociative mechanism starts by finding mappings between concepts and creates a new concept. Then, from these mappings, it transfers knowledge from each of the co-mapped concepts to the new, bisociative, concept. The mappings do not have to rely on similarity: they can represent conflicts that are striking, surprising or even incongruous (as suggested by Koestler [Koestler, 1964]).
- Reasoning mechanism. The reasoning mechanism applies two strategies. The divergent strategy makes use of the bisociative mechanism to generate new concepts and picks the ones that get a better evaluation. The convergent strategy makes use of the elaboration mechanism to generate better concepts from the ones resulting from the divergent strategy
- Evaluation. The Evaluation module returns the measure in which, according to a goal, a given concept satisfies the I of novelty and usefulness.
- Elaboration. Elaborating or adapting means reworking a concept to comply with context and domain-dependent constraints. In other words, the Elaboration module is concerned with eliminating inconsistencies and completing a concept with valid knowledge.
- Goal. The Goal should be given externally, it defines the purpose of the concept being sought.

This model was first sketched and presented, with slight but irrelevant differences, in two papers: "Modelling Divergent Production: a multi domain approach", presented at ECAI'98 [Pereira, 1998], and "Wondering in a Multi-Domain Environment with Dr. Divago", presented at CSNLP'99 [Pereira and Cardoso, 1999]. At neither time was it formalized in detail. This will be done in the following pages.

For those readers unaware or not interested in this formal specification, we will also describe each idea informally. In order to perceive the underlying ideas, it should not be necessary to understand every technical detail.

In this formalization, we will borrow some definitions from Geraint Wiggins [Wiggins, 2001] (see section 2.2.1), namely the universe, $\mathcal{U}$, of concepts, the language $\mathcal{L}$ and the traversal strategy, $\mathcal{T}$.

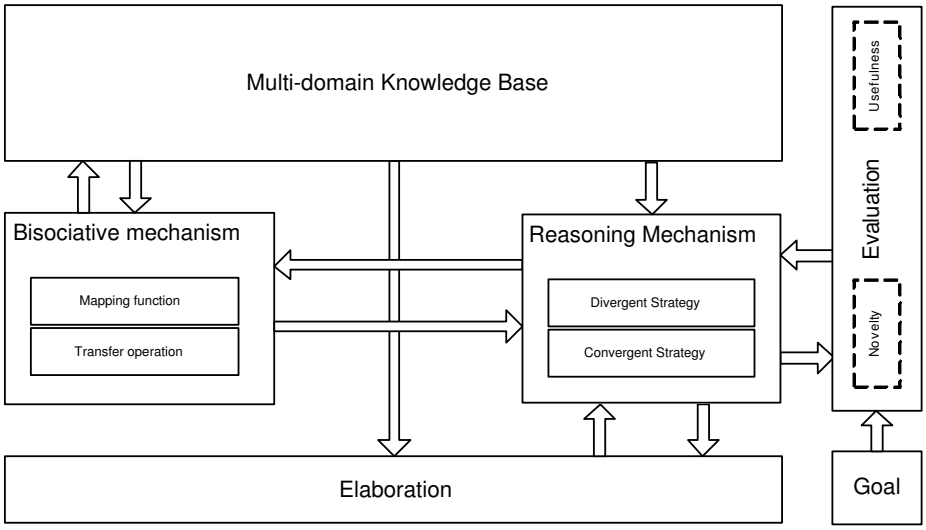

Figure 4.2: A Model of Concept Invention

Thus, let the alphabet $\mathcal{A}$ contain all possible *atomic* symbols (constants and variables) conceivable. Let us also define a process, p, by which we can compose elements of $\mathcal{A}$ in order to get higher order knowledge structures, the language $\mathcal{L}$, which may comprise predicates and functions (a predicate P would take the form $P(x_1, x_2, ...x_n)$, with $P, x_1, x_2, ...x_n$ being symbols of the alphabet $\mathcal{A}$). To simplify, we assume that these higher order knowledge structures have the form of facts and rules (as in a logic program, e.g. [Leite, 2003]) and that the process p is a generative grammar that allows the generation of all possible concepts (i.e. logic programs) of language $\mathcal{L}$. Therefore, this will be our universe $\mathcal{U}$, of concepts, which will contain, as well as other concepts, all the rule sets $\mathcal{T}$, $\mathcal{E}$ and $\mathcal{R}$, and their associated sets, $\ll \mathcal{R}, \mathcal{T}, \mathcal{E} \gg$, $[\![\mathcal{R}]\!]$ and $[\![\mathcal{E}]\!]$. Remember that, according to section 3.1, each concept is defined as a micro-theory (formally, a logic program). Informally, let us imagine the universe $\mathcal{U}$ of all possible concepts representable by a language $\mathcal{L}$.

Following the principles in section 4.1, our model must consider a knowledge-base with many different domains and, in order to allow meta-level reasoning, the description of its own controlling processes. To sum up, the knowledge base in our Model of Concept Invention should contain concepts from the domain for which we intend to invent concepts, from other domains, and the description of the model itself. The logic program (micro-theory) that describes a concept specifies its relationship with other concepts as well as its inner characteristics. The set of concepts contained (intensionally or extensionally) in a knowledge base KB is defined as $\mathcal{U}_{KB}$, with $\mathcal{U}_{KB} \subseteq \mathcal{U}$. A knowledge base corresponds to a set of concepts that cohabit the same physical or virtual memory space. It should be a model of an individual's own complete knowledge. A knowledge base is *multi-domain* if it contains concepts from

more than one domain.

Let a domain D correspond to a set of concepts from KB (so $D \subseteq \mathcal{U}_{KB}$) such that all of them relate to a unique, underlying concept. Of course, this invites the extremes that the KB itself is a domain ("everything is member of KB"); and that each concept is a domain ("Each piece of knowledge that describes the concept is related to that concept"). The notion we intend to bring is that a domain incorporates knowledge relating to the same subject. For example, a set of musical pieces, along with style rules corresponds to a domain of music, while a set of recipes with data describing available ingredients is a domain of cookery. As with our own knowledge, domains can encode different levels of expertise, detail, and so on. This notion of domain is rather imprecise and it is not conclusive for the model we are presenting. We introduced it mainly to assert that the knowledge base comprises potentially disparate kinds of data that are commonly associated to knowledge about different subjects.

According to Koestler, Guilford, Fauconnier and Turner, and many other theorists already referred to in this document, the associativity between concepts is fundamental in creativity. This takes us to the next definition, the *mapping function*:

$$\phi : \mathcal{U} \times \mathcal{U} \longrightarrow \{0, 1\}$$

This definition implies that any element from $\mathcal{U}$ can be mapped to an undetermined number of other elements (more specifically, whenver $\phi(x, y)$ is 1, this implies that x can be mapped to y). ϕ should obey the following axioms:

1. $\phi(x, y) = \phi(y, x) : x, y \in \mathcal{U}$, i.e. ϕ is symmetric.
2. $\phi(x, \emptyset) = 0 : x, \emptyset \in \mathcal{U}$, where $\emptyset$ is the empty concept.

The *empty* concept has no content (e.g. a logic program with no facts or rules) and can be represented by a constant (in Divago, this will be the atom *nill*).

We will call the function that maps elements from two domains a *cross-domain mapping function*:

$$\phi_{D_1,D_2} : D_1 \times D_2 \longrightarrow \{0, 1\}, \text{ where } D_1 \text{ and } D_2 \text{ are distinct domains.}$$

The act of bisociation is not completed until a novel concept emerges. In the many examples given (of bisociation, blending, conceptual combination), knowledge is transferred from each co-mapped element to the novel concept. This is where, in Koestler's model, the act of creation happens, and where, in Conceptual Blending and Conceptual Combination, *emergence* starts to happen:

$$\omega : \mathcal{U} \times \mathcal{U} \longrightarrow \mathcal{U}$$

ω, the *transfer operation*, can be defined as:

$$\omega(x,y) = \begin{cases} \emptyset & : \phi(x,y) = 0, \\ k & \text{otherwise, with some } k \in \mathcal{U}. \end{cases}$$

Informally, the transfer operation is an operation over two concepts (x and y) such that, if there is a mapping between them, a concept k (from $\mathcal{U}$) is created. There is no way to specify this operation in more detail, since there are many different accounts for how concepts are combined together. In the next chapters, we will propose a possible version whose results will also be demonstrated. ω is not necessarily deterministic, i.e. k may correspond to a set of probabilistic choices. Two other axioms for ω also follow:

1. ω is not symmetric. I.e., $\omega(x,y) = \omega(y,x)$ is not necessarily true.
2. $\omega(x,\emptyset) = \emptyset$

The set Ω contains all possible bisociations within $\mathcal{U}$. We call it the *bisociation set*:

$$\Omega = \{k : k = \omega(x,y), x, y \in \mathcal{U}\}$$

When ω is applied to two entire domains, we obtain the set Ω_{D_1,D_2}. We call it a *domain bisociation*:

$$\Omega_{D_1,D_2} = \{k : k = \omega(x,y), x \in D_1, y \in D_2\}$$

The bisociation set thus contains all possible bisociations for a knowledge base KB. In our model, this set contains the structures that result from what Guilford called divergent production, also to which [Finke et al., 1992] called the *pre-inventive structures*, in the Geneplore model.

The choice or ordering of the bisociation set can only be made if there is a goal to be reached. The agent should be looking for something, otherwise there would be no particular reason for picking one element from the bisociation set, i.e. to explore pre-inventive structures or to converge on something interesting. Thus we define the set $\mathcal{U}_g$ of all possible expressed goals, based on $\mathcal{L}_g$, such that $\mathcal{L}_g \subseteq \mathcal{L}$. A goal can range from very specific requirements for a problem (e.g. a set of design requirements) to abstract (e.g. achieve *balance* in a picture) and vague requirements (e.g. need for joy). Associated evaluation functions, which test whether a concept does or does not fulfil the I, and verify its novelty, must underpin any of these goals. Thus, we have two functions, *novelty* and *usefulness*:

$$nov : \mathcal{U} \times \mathcal{U}_g \longrightarrow [0,1]$$

The function *nov* returns the novelty of a concept w.r.t. a goal g. In the *typ* function of [Ritchie, 2001], which should be the inverse of *novelty*, the goal was implicitly considered, but we think goals should not be singular or encoded in functions, they should also be seen as concepts and be members of $\mathcal{U}$. This means, in the notion of concept followed here, that goals should also be expressed as micro-theories. Another implication is that goals themselves can be bisociated. The *usefulness* is given by the other function:

$$use : \mathcal{U} \times \mathcal{U}_g \longrightarrow [0, 1]$$

Again, the accomplishment of a goal is as fundamental as the concept itself in order to assess its usefulness. Something is only useful (or appropriate) if it is seen in context.

Now examining the bisociation set, its size may vastly increase and the novelty and usefulness of its members may vary extensively. In other words, the search space for getting good concepts can become extremely complex. Following [Wiggins, 2001], we also propose the traversal set, $\mathcal{T}$, which embeds the strategy used by an agent to traverse a search space. We assume that $\mathcal{T} \subseteq \mathcal{U}$, i.e. the strategy is also defined as a set of concepts from the universe $\mathcal{U}$, also implying that the same operations could be applied to the strategy itself. We propose no practical realization for this meta-level reasoning in this work, but we want to stress that it is a fundamental aspect if we want to reach the limits of computational creativity.

The bisociation set is traversed by a strategy $\mathcal{T}_d$. In the traversal of these pre-inventive structures, definitive values for novelty and usefulness are sometimes hard to assess, so priority to other measures of interest (not necessarily driven by the goal) may also be applied, such as diversity, simplicity, re-representation potential, etc. In other words, although we are only considering the functions of *novelty* and *usefulness* here, other factors may be of importance in the selection of elements from the bisociation set. $\mathcal{T}_d$ should thus be understood as the *divergence strategy*, which should cognitively correspond to the act of wandering for possible solutions, or inspirations, to a problem.

Also following Wiggins approach, we suggest the function operator, $\ll ., ., . \gg$, which selects elements of $\mathcal{U}$ from existing ones. Wiggins proposed an application of this operator as $\ll \mathcal{R}, \mathcal{T}, \mathcal{E} \gg$, under the assumption that a search strategy $\mathcal{T}$ would traverse a space (partially) ordered by the rule sets $\mathcal{R}$ and $\mathcal{E}$. In our case, we propose the traversal of the bisociation set, Ω, partially ordered by the functions of *novelty* and *usefulness*. The implied correspondences ($use \leftrightarrow \mathcal{R}$ and $nov \leftrightarrow \mathcal{E}$) are no coincidence, rather we can assume they are instantiations: from $[\![\mathcal{R}]\!]$, we select those that are *useful* for a goal; from $[\![\mathcal{E}]\!]$, we prefer those that are novel[1]. In so doing we

[1]The reader may have noticed that we have a clash here: in section 2.2, we associated $\mathcal{R}$ to

avoid some of the vagueness implied by the definitions of $\mathcal{R}$ and $\mathcal{E}$ and redefine the step of the space traversal to be:

$$x_{i+1} = \ll use, \mathcal{T}_d, nov \gg [x_i]$$

Informally, what we mean exactly here is that we expect our divergent search procedure to consider both *novelty* and *usefulness* in the traversal of the concept spaces. Moreover, this traversal is cumulative in the sense that it counts with the concepts already inspected (in the same way that a composer will not invent the same exact idea twice, since in the second time he will have the previous concepts in memory). The application of this operation repeatedly to achieve all possible concepts would lead us to the generation of the complete divergent strategy set. Formally (again following Wiggins notation), this set corresponds to:

$$\ll use, \mathcal{T}_d, nov \gg^{\diamond} [\Omega]$$

Notice that we apply the function directly to the set Ω, instead of giving the "starting symbol", the empty concept $\perp$. The intent is to explicitly assert that this search is made within the set Ω, instead of the whole universe $\mathcal{U}$. Also notice that the resulting set will be a subset of the set Ω itself. Informally, after defining the set of all possible bisociations (certainly a very large set), one needs to *choose* good ones. This can only be done via I such as novelty and usefulness in conjunction with a search strategy.

We have reached the point where a (set of) concept(s) is found, still in its "pre-inventive" state. In other words, it would be expected that further exploration is needed in order to arrive at a proper answer to the goal(s). This corresponds to the convergent phase, also previously referred. Let us now have the function θ, called *elaboration function*:

$$\theta : \mathcal{U} \times \mathcal{U} \longrightarrow \mathcal{U}$$

This first argument of this function is the concept to be elaborated, while the second corresponds to the method used. Below, there are three kinds of elaboration:

- $\theta_{\mathcal{R}} : C \times R \longrightarrow C'$, where $R \in \mathcal{R}$ is a rule. This is called *rule-based elaboration*, which happens when a rule or a set of rules (external to the concept) is applied for elaboration. These rules can be heuristics, causal rules, whatever kind of production rules available.

Ritchie's *typ* and $\mathcal{E}$ to *val*. This would imply the correspondences *use*↔*typ* and *nov*↔*val*. An alternative would be to map $\mathcal{R}$ to *nov* and $\mathcal{E}$ to *use*. This is incoherent and demonstrates the ambiguity in the meaning of $\mathcal{R}$ and $\mathcal{E}$. The problem seems to be that $\mathcal{E}$ should not be independent of $\mathcal{R}$, as much as it shouldn't be independent of purpose (or usefulness).

- $\theta_C : C \times C \longrightarrow C'$, where C is a concept. This *internal-logic elaboration* consists of applying reasoning methods exclusively taking into account the concept's micro-theory. Examples of these reasoning methods are deduction, induction and abduction within the micro-theory of C.
- $\theta_{\mathcal{CC}} : C \times C_1 \longrightarrow C', C \neq C_1$ such that C, C_1 are concepts. This is the *cross-concept based elaboration*, where the first concept is elaborated by association/comparison to other concepts in the knowledge base. Sometimes, new knowledge or structure is added or removed to a concept by comparison to other concepts. A special case of this is the *cross-domain based elaboration*, when C is elaborated by concept(s) from a domain that differs from that in which C is integrated. This kind of elaboration is often used in analogy, when knowledge from one domain gets projected onto the other.

We also add an axiom for θ:

1. θ is not symmetric. I.e. $\theta(D, K) = \theta(K, D)$ does not necessarily hold.

We can now define the set Θ of *elaborated concepts*:

$$\Theta = \{\theta(C, \mathcal{K}) : C \in \ll use, \mathcal{T}_d, nov \gg^{\diamond} [\Omega] \wedge K \in \mathcal{U}\}$$

As with the bisociation set, the set of elaborated concepts can be ordered by the functions of *novelty* and *usefulness*, yielding the search space that is to be traversed by $\mathcal{T}_c \subseteq \mathcal{U}$, the *convergent strategy*. The final result, the ordered set of *concept inventions*, is given by:

$$\ll use, \mathcal{T}_c, nov \gg^{\diamond} [\Theta]$$

4.3 Discussion

This model implies the interaction of three concept sets, all belonging to the universe $\mathcal{U}$: the set $\mathcal{U}_{KB}$, consisting of all the concepts implicitly or explicitly defined in the available knowledge base; the set Ω, which contains all the bisociations generated; the set Θ, comprising all the possible elaborations of the elements from Ω. Two more sets should be considered, regarding the definition of the search strategies, $\mathcal{U}_{\mathcal{T}}$ and the goals, $\mathcal{U}_g$. In figure 4.3, we depict all these sets as well as the directions taken by the search strategies. We should remember that, since this is a purely theoretical analysis, one should not read too much into the exact position and size of the sets. Our intention here is to relate our model to each of the sets and their intersections.

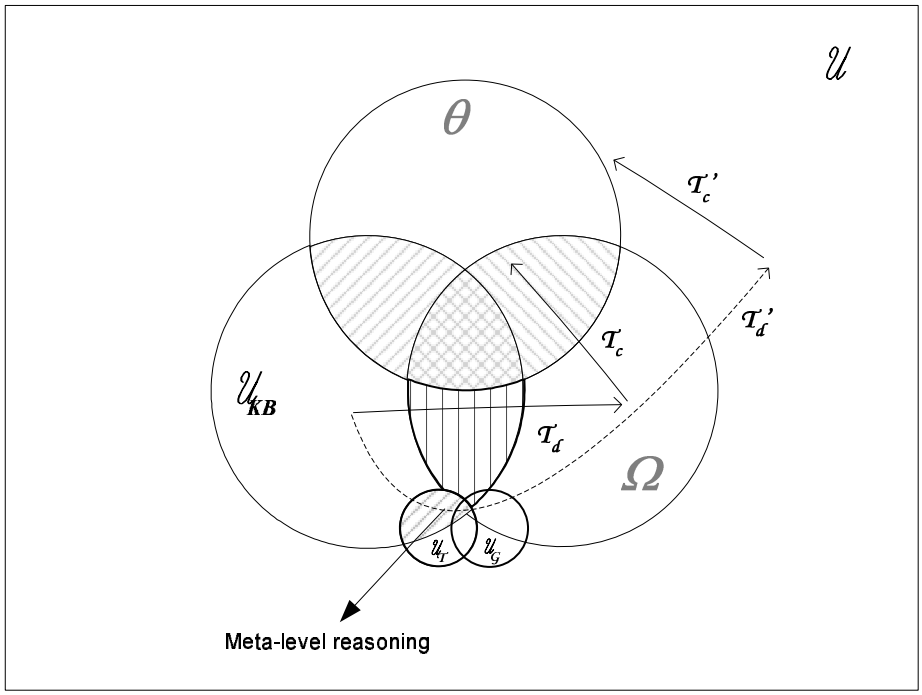

Figure 4.3: An analysis of the model presented

We defined two sorts of strategies, the "simple" strategies ($\mathcal{T}_c$ and $\mathcal{T}_d$) and the "ideal" strategies ($\mathcal{T}_c'$ and $\mathcal{T}_d'$). They should represent a continuum of possibilities. The "simple" (or rather computationally "realistic") strategies consider a clearly bounded concept space. The path depicted show what would be expected of a plain application of the model just described. It starts from the universe of known concepts, $\mathcal{U}_{KB}$, then applies bisociation, thus inspecting the set Ω with the divergent strategy. The convergent strategy will then take the search within the set Θ. The "ideal" strategies should be able to diverge more, considering the sets $\mathcal{U}_{\mathcal{T}}$ and $\mathcal{U}_g$, as well as having the freedom to jump off the bisociation space as it was defined (e.g. due to a change in its own goal and strategy). Again, this is more a dissertation than a practical proposal. Indeed, this leads us to consider the "simple" strategies as a realizable step towards that ideal. The intuition behind the two sides of this continuum is the range between day-by-day creativity and the revolutionary creativity, or the big "C". We suggest that divergence as well as convergence are constantly present in daily problem solving, although only in special situations does it become necessary to diverge considerably, i.e. to apply a $\mathcal{T}_d'$ kind of strategy.

A somewhat different analysis can be made regarding the expected weight of novelty and usefulness in the application of the strategies (figure 4.4). Here, we have a closer match with the diagram proposed for Wiggins' formalization, in section 2.2.

We have placed the universe of concepts defined by the knowledge base, $\mathcal{U}_{KB}$, in

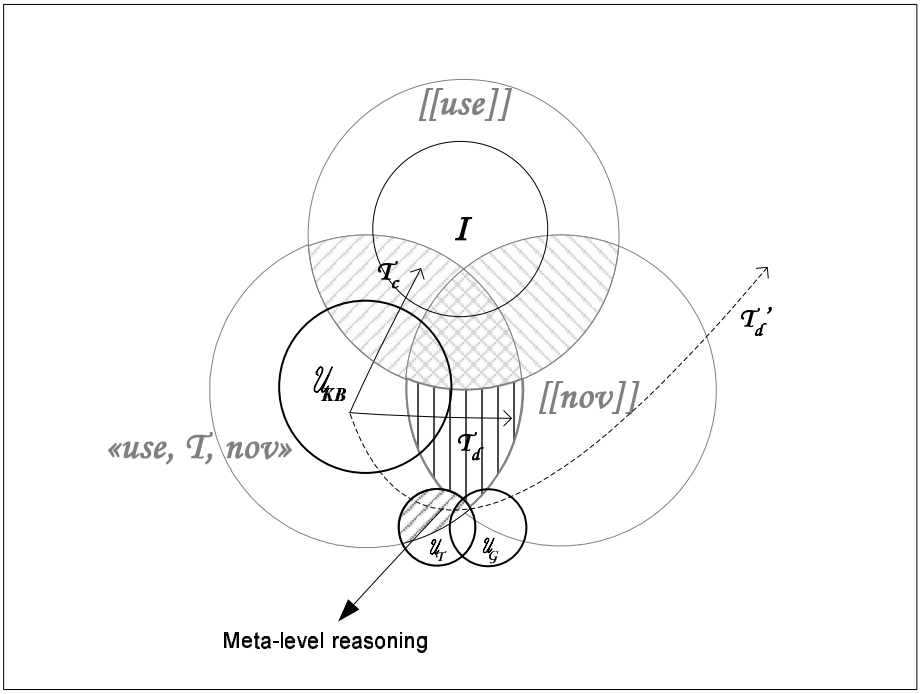

Figure 4.4: An analysis of the model presented by applying the framework of Wiggins, modified by instantiating $\mathcal{R}$ with *use* and $\mathcal{E}$ with *nov*.

the center of $\ll use, \mathcal{T}, nov \gg$. In fact, as well as being totally available for the search method, $\mathcal{U}_{KB}$ should contain a variety of concepts, ranging from the useful to the non-useful, from the novel to the typical[2]. We also assume that $\mathcal{T}_c$ would be directed towards usefulness and $\mathcal{T}_d$ towards novelty, although there is no formal indication for that on the Model. We are intuitively led to this suggestion, and empirical results (in chapter 6) will indeed confirm this reasoning.

In the imagined scenario given in section 4.2, we arrived at a point where, when the system did not have the sufficient knowledge or the search space was too big to reach a goal in a reasonable amount of time, it would enter a divergence mode. However, it is clear that the complexity of this mode could (and possibly would) be higher than the convergent one. In other words, in the case of our model, if there was a mapping between each pair of the n concepts from KB, we would have $n \times (n-1)$ bisociations, if the transfer operation, ω, only produced one new concept for each pair. Since we are assuming a large and varied knowledge base, this value would be extremely big, even with an optimistic perspective. Indeed, if we also consider the possibility of changing $\mathcal{T}$ and $\mathcal{G}$, then we soon reach the conclusion that it is

[2]We could consider $\mathcal{U}_{KB}$ as the inspiring set, since it is given a priori. However, applying an element from the inspiring set (i.e. a known concept) in an untypical way (e.g. using an apple as a weapon, instead of food) would thus be a reinvention by some features of Ritchie. Moreover, it is common to have elements in the inspiring set that are not part of $\mathcal{U}_{KB}$, as well as conversely.

unrealistic to search but only a little portion of this (new) search space. Thus, the purpose of presenting and discussing this model is not to implement it entirely, but to state our position on what modelling concept invention with bisociation is about and to lay the foundations for practical implementations such as the one we will describe in the next chapter. A question thus arises: Why is this concept invention?

As we can observe, even if an outcome given by this model is logically deducible from KB, its generation is not based on soundness[3], in the same way that someone solves a problem without having followed a conscious sequence of steps. This does not mean that this is the way unconsciousness works, or even how humans invent concepts, rather it is a model for how it can be computationally simulated. Thus, we call it a model of concept invention because it produces new (and potentially useful) concepts from an unsound process, which agrees with the definition we gave in section 3.2.2.

Again, we would like to raise the question about search. After all, isn't this just search in a (complex) space? What more do we have to offer than any other AI model? The answer is simply yes, it is search. And this is an AI model which, as many other AI models, aims to simulate a specific kind of human behavior that has been rarely approached before. In this case, creativity, more specifically concept invention. Would this mean that creativity is part of (or is a kind of) intelligence? The answer we give is that they are definitely related and, in order to invent a concept, rationality (an indisputable component of intelligence) is necessary.

Apart from raising philosophical questions, what else is this model for? In other words, what could be its applications and what is the degree of *implementability* of its components? Such a model could be applied in situations where the generation of new concepts is important, such as in design, architecture or games, to name a few. Ideally, it could be applied as a meta-level reasoning engine to help with situations where a *lower-level* system, dedicated to a specific domain, could not find a solution, as suggested in the imagined scenario described earlier. To some extent any of its modules can be implemented, however the capability of meta-level reasoning is, perhaps, the hardest to construct, since it demands self organization and assessment, two capacities that machines can hardly achieve. For this reason, meta-level reasoning has not been implemented in our system, Divago, which will be described next. In this system, we will provide some suggestions for how other aspects of this model can be implemented, namely cross-space mapping, bisociation, the knowledge base, the reasoning engine, the evaluation and the elaboration.

[3]Of course, this depends on the mapping functions and transfer operations used, but we are considering any kind of functions or operations, even randomness, which would be likely to produce many inconsistencies.

Chapter 5

Divago

We will now present our system. It is called Divago, after the Portuguese expression "(Eu) divago", which means "I wander". In the previous chapter, we explained our Model of Concept Invention, which comprehends the main theoretical substance of this book. In this chapter, we seek to provide a practical instantiation of its modules. The construction of Divago demanded many compromises between the overall goal of instantiating the model of concept invention and the specificities that appeared during the development and reflection upon each of the modules. For this reason, there are some points of conflict between them, namely in the implementation of the search strategies and in the choice of constraints. Where the reading becomes harder to follow (due to formalization or algorithms), we will try to synthesize the message in a manner as fluent as possible.

5.1 Overview of the Architecture

In figure 5.1, we show the architecture of Divago. Before entering into details, we prefer to give a superficial overview of how it works, with attention to the role that each module takes and to the data flow (represented by arrows in the diagram).

The knowledge base contains a set of concepts, each one defined according to several different kinds of representations (concept maps, rules, frames, integrity constraints, instances). The concept maps, rules, frames and integrity constraints follow the Micro-theory view, while the instances agree with the Exemplar view.

The first step for the invention of a new concept is the choice of the input knowledge, in this case a pair of concepts. Since, in Divago, we are focusing on the mechanisms of divergence and bisociation, we provide no specific algorithm for this selection. This choice is either given by a user or randomly generated. After being given a pair

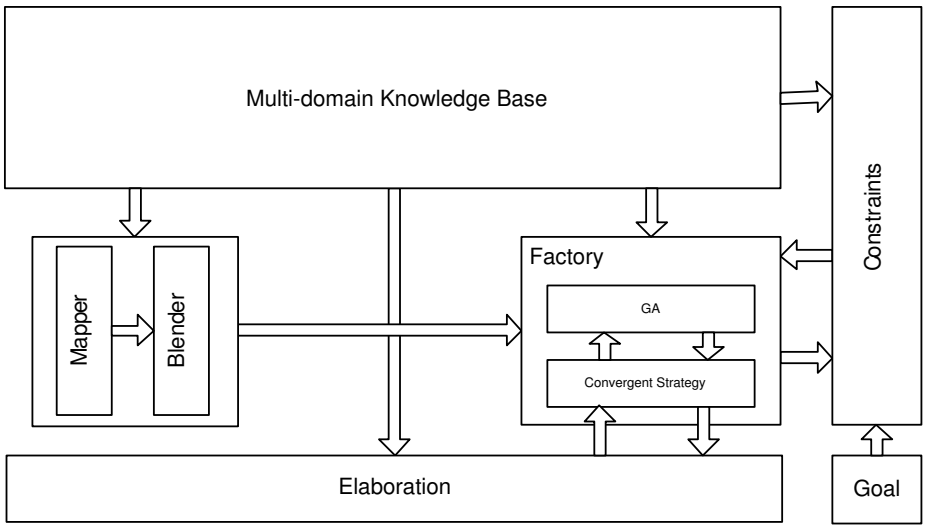

Figure 5.1: The architecture of Divago

of concepts, the Mapper builds a structural alignment between (the definitions of) them. It then passes the resulting mapping to the Blender, which then produces a set of *projections* that implicitly define the set of all possible blends. This will be the search space for the reasoning mechanism, the Factory.

The Factory is based on a parallel search engine, a *genetic algorithm* (GA), which searches for the blend that best complies with the evaluation given by the Constraints module. Prior to sending each blend to this module, the Factory sends it to the Elaboration module, where it is subject to the application of domain or context-dependent knowledge. The GA thus interacts both with the Constraints and Elaboration modules during search.

The evaluation of a blend given by the Constraints module is based on an implementation of the Optimality Principles (in section 3.3.2). Apart from the blend itself, our implementation of these principles also takes into account knowledge that comes from the Knowledge Base (namely integrity constraints and frames), as well as the accomplishment of a goal that comes in the form of a query. Any of these issues will be described shortly. The Elaboration module essentially applies internal-logic elaboration and rule-based elaboration. The rules involved are also part of the knowledge base.

After reaching a satisfactory solution or a specified number of iterations, the Factory stops the GA and returns its best solution. In some cases, this result is also the input of an Interpretation module, which produces an interpretation of the new concept. In collaboration with other researchers, we developed Interpretation modules that generate 2D images (in the house-boat experiment), textual descriptions

(horse-bird experiments) and 3D images (creatures experiment).

Both the Mapper and the Elaboration modules are optional, for different reasons. The mappings provided by the Mapper are essentially based on the Analogy and Metaphor works presented in section 3.4. However, in some situations, these mappings are very restrictive. Thus, without having implemented alternative procedures, we allow an externally defined mapping (which, in some experiments, is user-defined). The Elaboration can also be bypassed for experimentation reasons. When analyzing results, the elaboration can hide the *real* results, i.e. it can fix problems by itself that we may need to watch in order to assess the functioning of the system.

In comparison with the model presented in the previous chapter, a difference immediately arises that the mechanisms of divergent and convergent search ($\mathcal{T}_d$ and $\mathcal{T}_c$, respectively) are not separated in Divago. On the contrary, they work intertwined: the method for divergence (the GA) uses the method of convergence (which applies the Elaboration) once for every blend found.

Another difference is that Divago is not processing its own internal specifications. In other words, we leave meta-level reasoning, which might support transformational creativity, for future developments. As discussed before, this is an extremely complex task *per se*.

We will now describe in greater detail each of the six modules: the Knowledge Base, the Mapper, the Blender, the Factory, the Constraints and the Elaboration.

5.2 Knowledge Base

All representation in the Knowledge base follows a symbolic approach (as opposed to sub-symbolic ones, such as neural networks). Nevertheless, we see no reason to doubt that the same mechanisms could also be applicable with other representation paradigms. There are many different kinds of structures in the knowledge base of Divago, namely the *concept maps*, the *rules*, the *frames*, the *integrity constraints* and the *instances*. The syntax used in the knowledge base (and in the whole system) is the same as in the *Prolog* language.

We call *concept maps* to the semantic networks that are used in Divago to describe a concept or a domain. A Concept Map is a graph in which nodes represent *concepts* and arcs represent *relations*. A concept is thus defined in association to other concepts, which will therefore also intervene within the concept's definition. For this reason, some confusion may arise so we ought to clarify now the notions of domain, concept

and element[1]. Whenever we have a concept map in which there is no central concept, rather many different concepts participate with the same degree of importance, we call it a *domain*. Examples of domains could be "biology", "computers", "music". In any of these, we focus on many concepts rather than only one (e.g. in "music", we have "harmony", "melody", "rhythm", etc.). When the concept map focusses on a single concept x, we say it is the concept map of the *concept* x. For example, the concept map of the concept "horse" will have associations to "nose", "legs", "mane", etc. in order to define what a horse is. Of course, each of these are also concepts, but as they get farther away from the *main concept* (the one about which the concept map is built - in the example, "horse"), they get less specified (e.g. the concept "human", from the concept map of "horse" has no associations to "intelligence", "face", "society", etc. and therefore it is only superficially identified, possibly with relations such as "owner-of" or "rider"). In order to avoid confusion, we adopt the convention that each of these nodes of a concept map will be named *element*[2], instead of *concept*. Thus, a *Concept* will be made up of the concept maps, rules, frames, etc. as proposed in the previous chapters, and following essentially the micro-theory view.

The difference between a domain and a concept is subjective and depends on the level of granularity. Every domain is by itself a concept and every concept can be seen also as a domain (even if it is a micro-domain). Throughout this book, unless explicitly stated to the contrary, we assume that a concept map is defining a single concept, rather than a domain. To sum up, Divago follows the micro-theory view of concepts (presented in section 3.1), in which a concept is defined by facts and rules. We will see that Divago also allows the use of *instances*, which agrees with the exemplar view.

The choice of symbols for elements and relations in our concept maps is arbitrary, yet, mainly after the horse-bird experiment, we followed two normalization principles. The first one is that relations must belong to (or descend from) the Generalized Upper Model hierarchy (GUM) [Bateman et al., 1995], a general task and domain independent *linguistically motivated ontology* that intends to significantly simplify the interface between domain-specific knowledge and general linguistic resources. GUM occupies a level of abstraction midway between surface linguistic realizations and *conceptual* or *contextual* representations. Being split into two hierarchies, one containing all the concepts and the other all the roles, GUM gives us a large set of primitive

[1]This classification is being used for this book, as the use of the names "domain", "concept" and "concept map" has raised some ambiguity in preceding publications. We will follow the present definition throughout this document and future publications. It is also important to inform that the notions and underlying rationales maintain the same.

[2]The name *element* is also used by Fauconnier and Turner to refer to the same entities inside mental spaces.

relations to standardize our choices in the concept map. It is important to notice that, in our maps, the members of the concept hierarchy of GUM (e.g. "color", "ability", etc.) are used as relations (e.g. "color(mane, black)", "ability(horse, run)"). In Appendix C, we reproduce a copy of the whole GUM hierarchy used.

The second principle that we follow in the construction of the concept maps is that elements in our knowledge base may only be represented as nouns, adjectives, preferably in the singular form, or numerals (in the particular case of numbers) in English language. As we said, these are only normalization principles for the construction of the concept maps, so, in theory, the model itself doesn't take into account the lexical categories of the words used, following only the principle that "the same word corresponds to the same element".

Given the importance that the concept maps have for Divago, we developed another system, Clouds (which was the subject of an MSc. thesis), with the goal of helping a user build her own concept maps and avoiding bias as much as possible. The system led the user towards different areas of the concept map and, as it gets expanded, she can no longer keep track of the whole, leaving to Clouds the task of leading the construction via questioning the user. Clouds was used to generate the maps for some of the experiments presented here. It is far from the theme of this book, so we direct the interested reader to [Pereira et al., 2000, Pereira and Cardoso, 2000, Pereira, 2000]).

Let us now define formally a concept map. We will use the same definitions and symbols given in the previous chapter.

Let $\mathcal{AE} \subseteq \mathcal{A}$ be a set of symbols, which we will call the *elements* and let $\mathcal{AR} \subseteq \mathcal{A}$ be another set of symbols, which we will call the *relations*. A *Concept Map* CM is a set of binary predicates with the form:

$$X(Y, Z),\ X \in \mathcal{AR}, Y, Z \in \mathcal{AE}$$

We also define the *exhaustive closure* CM^+ as the concept map with all elements $\mathcal{AE}$ and relations $\mathcal{AR}$ between them, i.e.

$$CM^+ = \{X(Y, Z) : X \in \mathcal{AR}, Y, Z \in \mathcal{AE}\}$$

Therefore, $CM \subseteq CM^+$.

In tables 5.1 and 5.2, we show examples of concept maps for "horse" and "bird" (made with Clouds). These maps are necessarily arbitrary in the sense that each person would draw her own maps, a result of the different conceptualization and individual points of view one can take. Some relations such as "pw" (part-whole), "member_of" (category inclusion relation), "isa", "sound" are either shorter words for the same relations of GUM or extensions made to this hierarchy.

isa(horse,equinae)	pw(leg, horse)	purpose(horse, food)
isa(equinae,mammal)	purpose(leg, stand)	sound(horse, neigh)
existence(horse, farm)	pw(hoof, leg)	purpose(mouth, eat)
existence(horse, wilderness)	purpose(horse, traction)	purpose(ear, hear)
pw(nose, horse)	eat(horse, grass)	color(mane, dark)
pw(mane, horse)	ability(horse, run)	size(mane, long)
pw(tail, horse)	carrier(horse, human)	material(mane, hair)
quantity(hoof, 4)	quantity(leg, 4)	purpose(horse, cargo)
pw(eye, nose)	quantity(eye, 2)	member_of(horse, ruminant)
pw(ear, nose)	quantity(ear, 2)	ride(human, horse)
pw(mouth,nose)	purpose(eye, see)	motion_process(horse,walk)
isa(farm, human_setting)		

Table 5.1: The concept map of *horse*

isa(bird, aves)	existence(bird, house)	isa(aves,oviparous)
lay(oviparous, egg)	existence(bird,wilderness)	purpose(bird, pet)
purpose(bird, food)	purpose(eye, see)	smaller_than(bird, human)
pw(lung, bird)	motion_process(bird, fly)	purpose(beak, chirp)
purpose(lung, breathe)	quantity(eye, 2)	quantity(wing, 2)
isa(owl, bird)	isa(paradise_bird, bird)	quantity(claw, 2)
ability(bird, fly)	pw(wing, bird)	conditional(wing, fly)
pw(feathers, bird)	pw(beak, bird)	purpose(wing, fly)
purpose(beak, eat)	purpose(claw, catch)	sound(bird, chirp)
isa(parrot, bird)	ability(parrot, speak)	pw(straw, nest)
pw(eye, bird)	pw(leg, bird)	purpose(leg, stand)
pw(claw, leg)	role_playing(bird, freedom)	quantity(leg, 2)
isa(nest, container)	isa(house, human_setting)	

Table 5.2: The concept map of *bird*

We often represent concept maps graphically, in which the relations are arcs and elements are nodes. Figure 5.2 shows an excerpt of the concept map for "horse".

The concept map corresponds to the factual part of the micro-theory of the concept. The inferential part comprises *rules* that explain the inherent causality, *frames* that have the role of providing a language for abstract or composite concepts and *integrity constraints*, particular rules that serve to assess the consistency of the concept.

Rules have the form:

$$A_0 \wedge A_1 \wedge ... \wedge A_i \longleftarrow B_0 \wedge B_1 \wedge ... \wedge B_j,\ A_i \in K$$

$$with\ K = CM^+ \cup \{not\ R : R \in CM^+\}$$

This allows for the use of negation as well as a conjunctive set of atoms (A_i) in the

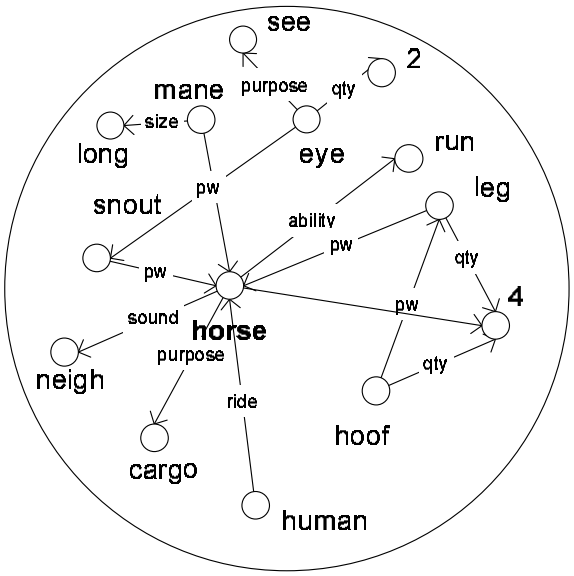

Figure 5.2: Concept map of horse (an excerpt).

head, although with specific constraints: $A_i \in K$ and B_j can be any atom or logical expression (e.g. a comparison). For example, a rule for inferring that something is at the gaseous state (and not at the solid or liquid state) could be:

$$state(X, gaseous) \wedge not\ state(X, solid) \wedge not\ state(X, liquid) \longleftarrow ebulition\_point(X, N) \wedge temperature(X, T) \wedge T > N$$

The syntax followed by Divago for rules is defined by a predicate rule/6:

$$rule(Domain, Name, PosConds, NegConds, AddList, DelList).$$

where *Domain* corresponds to the domain or concept with which the rule is related, *PosConds* and *NegConds* correspond to the (positive and negative, resp.) sets of conditions of the rule and *AddList* and *DelList* correspond to the (positive and negative, resp.) sets of conclusions of the rule.

The frames have the role of describing abstract concepts, situations or idiosyncracies. A frame consists of a set of conditions that the concept map must satisfy. When a concept c satisfies all the conditions of a frame f, we say that c *integrates* f. Frames are formally very similar to rules but they are applied differently in the process:

$$frame(Name):\ A_0 \wedge A_1 \wedge ... \wedge A_i \longleftarrow B_0 \wedge B_1 \wedge ... \wedge B_j$$

where $Name$ is an identifier of the frame. A frame should be a meta-level concept that is tightly integrated according to a situation, structure, cause-effect or any other

relation that ties a set of elements onto one, abstract or broad, concept. For example, the frame of "transport means" (below) corresponds to a set of elements and relations that, when connected together, represent something that has a container and a subpart (e.g. an engine) that serves for locomotion.

$$
\begin{aligned}
& frame(transport\_means(X)) : \\
& carrier(X, people) \longleftarrow have(X, container) \wedge have(X, Y) \wedge \\
& \qquad\qquad purpose(Y, locomotion) \wedge drive(\_, X)
\end{aligned}
$$

When a concept map *integrates* the "transport means" frame, then we can either say that it is itself a "transport means" or one of its constituents is a "transport means". For example, the concept map of school bus would integrate this frame, while the concept map of classroom wouldn't.

The syntax for representing a frame is the following:

$$frame(Domain, Name, PosConds, NegConds, AddList, DelList).$$

An extremely important aspect to remember about frames is that they allow the inclusion of *Prolog* scripts inside any of the sets *PosConds, NegConds, AddList* and *DelList*. This offers great power to frames since these scripts will be *run* whenever the frame is *inspected* and *integrated*. In other words, in order to check whether a concept map integrates a frame, it will execute the scripts included in the sets *PosConds* and *NegConds* and, during the elaboration phase, the frames that are integrated will have their sets *AddList* and *DelList* also executed. We will clarify each of these mechanisms during this chapter. The frames can thus become externally defined scripts or programs, executed whenever their conditions apply [3]. We propose to consider several types of frames according to their degree of abstraction and functional aspects. A very specific frame comprehends a well defined set of relations and elements, such as in "transport means". A highly abstract frame is one that considers many different types of elements and relations. For example, when a concept map satisfies a "noun-noun combination" frame, it means that it consists of the concept that results from a combination of two nouns - each one an independent concept in itself (e.g. "pet fish", "gun wound").

In terms of their function, frames can be classified as *organizing*, *pattern identifying* or *transforming*. An organizing frame is a frame that determines the general structure of a concept map. For example, in the concept map for school bus, "transport means"

[3]Rules can also contain *Prolog* scripts. While in frames, we use complex scripts (e.g. for specifying what kind of projections to apply in a blend), for rules, we normally use simpler ones (e.g. for calculating the meeting point of two moving objects, as needed in the Buddhist monk example.).

could be an organizing frame. A frame is pattern identifying when it matches a pattern within a larger concept map. For example, in a school trip concept map (which could be a large concept map with details about many concepts such as bus, teacher, driver, study material, theme, route, etc.), one could find a "transport means" frame (focussing on the school bus part), this time becoming pattern identifying. Normally, organizing frames are a lot more abstract than the pattern identifying ones, although sometimes the same frame can take both functions. Transforming frames may only make sense within a bisociation context (and thus will get clearer as we progress through this book). A transforming frame identifies a transformation that occurs during the blending of two input concepts. For example, if the new concept map integrates a "new ability" frame, it means that there were new "ability" elements and relations transferred from one of the input concepts to the context of the other (e.g. in Pegasus, wings are transferred from bird to horse, giving it the ability to fly). The specification for "new ability" could be something like:

$$
\begin{aligned}
&frame(new\_ability(d1)):\\
&\quad new\_ability(X,A) \longleftarrow ability(X,A) \wedge not\, rel(d1, ability(X,A)) \wedge\\
&\qquad\qquad purpose(P,A) \wedge pw(P,X) \wedge\\
&\qquad\qquad projection(blend, d1, X, X) \wedge\\
&\qquad\qquad projection(blend, d2, A, A)
\end{aligned}
$$

Reading this informally, it says that if some element X has, in the concept map of the *blend* (that is to say, the new concept map), the ability A, which was not present in X's input space, d1, then we are in face of a "new ability" given to X. It also says that this "new ability" should have a minimal justification, i.e. there must be a subpart P of X whose purpose is to provide ability A (e.g. if something flies, it should have wings). Furthermore, we can also require that X and A be projected from different inputs ($d1$ and $d2$, resp.) to the *blend*.

Thus, a frame will serve, in the process of concept invention, as a tool for pattern identification, for providing directives for the construction of concepts, and for elaboration. As we will see, frames are essential to control the system. In Appendix D, we provide a thorough description of these knowledge structures, their specific keywords and some examples.

Although rules and frames are formally very similar, we should now clear out their distinction and underlying rationales. Rules take an important role in the definition of a micro-theory of a concept, however in the blend construction, they end up having a passive part (they can be applied to the input domains before the process, or to the blend, after it is generated). Frames, on the other hand, can take an active part

on the blend construction: they can be specified as goals by a user and they help structuring the blend, often becoming the scaffolding around which new blends are constructed. In fact, as the experiments will show, they are fundamental to help Divago achive meaningful results.

As the reader may have noticed, our notion of frame departed from modelling Fauconnier and Turner's "tightly integrated structures" to operational scripts which actually govern transformations during blend construction. In cognitive linguistics terms, we believe this partially falls into the image schema realm (although the definition itself of image schemas in CL has not reached a consensus, e.g. [Johnson, 1987, Neisser, 1976, Thomas, 1999]) in the sense that our frames can be patterns that recurrently provide structured understanding of various concepts. Taking an example from Lakoff [Lakoff, 1987] of a *part-whole* schema (in Prolog form):

$$frame(generic, pwSchema, [findall(X, pw(X, Y), L), op(exists(L))], [], [], []).$$

This frame covers every part-whole relation in a concept map (the larger it becomes, the more "part-whole" based is the concept - the more the schema becomes meaningful for the concept). It can be said that these frames are too strict for the "metaphorical" potential of image schemas - the part-wholeness of the schema should not have to be processed literally (in the same way that an individual is part of the society, as opposed to a wheel being *literally* part of an engine), but it is also true that the set of concepts it can match depends on the abstractness of the frame. For example, the relation "pw" above could be replaced by a more general one (e.g. "belonging") or it could be defined via an algorithm.

The same reasoning could be taken for any other image schema (once we find the relationships and their arguments), but it becomes difficult to argue that frames *are* images schemata, because we didn't explore the relationships any further, namely regarding the cognitive basis, the role of perception, learning or any other aspect regarding the *mentality* of image schemata. What we are trying to say is that, with our frames, some of the properties of the image schemata can be *simulated*, namely being a generic structure, applicable to different concepts, thus attributing to these concepts a particular association (to the image schema - e.g. "part-wholeness"), and triggering associated inferences. Frames could be seen as a strictly symbolic - computational - version of image schemata. Again, we redirect the interested reader to Appendix D, which can clarify further the scope of these structures.

The *integrity constraints* serve to specify logical impossibilities. Each integrity constraint consists of a set of propositions that should not be simultaneously true.

Let A_i be any atom or logical expression, an integrity constraint ic is defined as:

$$false \leftarrow A_1 \wedge A_2 \wedge ...not\ A_{n-1} \wedge not\ A_n \wedge ...$$

Two examples of integrity constraints could be for specifying that something cannot be dead and alive at the same time and for avoiding part-whole recursion, i.e. something cannot have a part-whole relation (pw) with itself:

$$false \leftarrow state(X, dead) \wedge state(X, alive)$$

$$false \leftarrow pw(X, X)$$

In the Prolog syntax that we use, an integrity constraint is represented by the predicate integrity/3:

$$integrity(Domain, Pos, Neg)$$

where Domain is the concept or Domain to which the integrity constraint belongs and Pos and Negs are the positive and negated conditions of the constraint.

Finally, we can define a *Concept Micro-Theory* (or a Domain theory), CT, as being a tuple (CM, R, F, IC), where CM is a Concept Map, R is a set of rules, F a set of frames and IC a set of integrity constraints.

The micro-theories may be compared to Joseph Goguen's *sign systems* [Goguen, 1999]. In a sign system, we have sign and data sorts, partial orderings on each of these, relations and functions, constructors to build upper level signs, priorities on these, and axioms. In our micro-theories, we have no formal distinction between sign and data sort. In principle, every element is equal, thus its classification and partial ordering can only make sense in a concept map (e.g. an isa ontology with animal classification would correspond to a sign sort ordering, while another one with colors or numbers would be a data sort ordering), which also contains the relations and functions. Frames are our constructors, but there is no ordering or priority over them. Only goal frames, used in the Constraints module, have priority over the other frames. Finally, rules and integrity constraints can be seen as Goguen's axioms.

Another level of representation allowed in Divago is that of the instances. Along with the micro-theory, one can also add instances to the concept definition (this corresponds to the Exemplar view, as in section 3.1). Instances are represented as structures of knowledge that apply (some of) the elements present in the *micro-theory*.

Let $\mathcal{AA} \subseteq \mathcal{A}$ be a set of symbols (the arguments) and let $\mathcal{AF} \subseteq \mathcal{A}$ be a set of relations (the functors). Let $\mathcal{LC}$ be a set of compositions such that:

$$c \in \mathcal{LC} \ : \ c = \begin{cases} x \in \mathcal{AA} \\ I(y_1, y_2, ..., y_n), \quad y_i \in \mathcal{LC}, I \in \mathcal{AF} \end{cases}$$

Figure 5.3: An instance of a house

Thus an *Instance* can be represented in the form $I(x_1, x_2, ..., x_n)$, where $I \in \mathcal{LF}$, and $x_i \in \mathcal{LC}$. x_i is compositional, i.e. it can be used as an argument to another x_j. An example of an instance of the concept "house", as used in the "house-boat" experiment, could be:

```
case(1,0, house, [sons=2, size=small, type=simple, son_name=roof, son_name=body]).
case(1,0:0, roof, [shape=triangle(30)]).
case(1,0:1, body, [sons=3, in=[left/90,off/25, right/90],son_name=structure,
son_name=window, son_name=door]).
case(1,0:1:0, structure, [shape=square]).
case(1,0:1:1, window, [shape=square(5), in=[off/20, right/90, off/15, left/90]]).
case(1,0:1:2, door, [shape=rectangle(4, 10), in=[off/3]]).
```

Using the functors 'case', ':', '=' and '/', this instance describes the several parts of a house, starting from its top-level element ("house") to the smallest constituents ("door"). It associates each of the elements (that are also part of the theory) to a 2D drawing language (*Logo*). Its interpretation generates the image in figure 5.3.

Since the details of these instances and their syntax are not central for this book, we redirect the description of the language used to the Appendix E.

Instances are useful for interpreting a new concept in the sense that they can attach a semantics to a concept and its constituents. For example, with a visual instance of a house, one sees an example of what a door can look like. This will be observed in the house-boat and creatures experiments.

Finally, in Divago, a concept is defined by the pair (CT, I), where CT is the theory and I is a set of instances. The Knowledge Base can simultaneously have many different concepts, from different domains. However, during concept invention, Divago only considers a pair of concepts (or domains) and a special domain, the generic domain.

The generic domain contains all knowledge that is applicable to all concepts and to the process of concept invention. It has encoded the hierarchy of GUM in a predicate arc/5, which can be used to generalize/specialize the relations found in a concept map. This facility is used by some frames. The generic domain also contains an *isa* ontology, which is used mostly by the Mapper (shown in next section) to build

correspondences between elements of different concepts. The majority of the frames used by Divago also belong to the generic domain. In table 5.3, we show some of the frames from the generic domain that were used in the experiments.

Frame name	**Conditions**
aframe	The blend contains identical structure from input 1
aprojection	The blend contains the same elements of input 1
bframe	The blend contains identical structure from input 2
bprojection	The blend contains the same elements of input 2
new_ability	An element has an ability relation not existent in any of the inputs
function_transfer	An element in the blend has a function that was not present in its input.
analogy_transfer	Transfer all neighbor elements and relations of an element of one input to the projection of its mapping counterpart from the other input.

Table 5.3: Some frames of the generic space

In the generic domain, we find also integrity constraints and rules. An entire copy of the generic domain is given in Appendix F. In fact, the generic domain holds most of the knowledge for Divago, while each specific concept has only encoded the essential, usually consisting solely of the concept map.

To summarize, Divago employs a large variety of knowledge structures, each one with its own role: concept maps for structural relationships within a concept (or domain); rules for defining procedures specific to a concept; frames as abstract concepts that allow the system to identify patterns in concepts and to infer further knowledge (running the blend); integrity constraints to establish limits for reasoning; and instances, as examples of the concept (or domain). As we have said, these do not all have to cohabit simultaneously in the system. Indeed, with concept maps, frames and integrity constraints, the system is able to give results as shown in chapter 6. The need of each representation type will depend on the problem at hand: if working with concepts that need reasoning that depends on "hidden" inferences (e.g. temporal reasoning - story plot blending; spatial reasoning - scenario blending), we will need rules; if we intend to design specific objects (e.g. 3D objects), then instances may be necessary to provide a *real* output; if making experiments at the conceptual level (e.g. testing examples of Conceptual Blending), then maybe frames and concept maps will be enough. The integrity constraints are present in almost every application, as they become essential to advise Divago not to follow *wrong* paths[4].

[4]They are not mandatory, their role is to identify logically incongruous or inconsistent situations. But if there is a *good reason* for their maintenance, they will prevail (as happens in some creativity situations, as mentioned in section 2.1.2).

5.3 Mapper

The Mapper defines the mapping function ϕ of the model presented in this book. In the definition we gave, this function is oversimplified, since it only provides a binary association between pairs of concepts. In Divago, a concept is itself a structure with many different sub-structures, thus the mapping becomes somewhat more complex. This justifies a revised version of the mapping operation, ϕ:

$$\phi : \mathcal{U} \times \mathcal{U} \longrightarrow \mathcal{M}$$

where $\mathcal{M}$ is the powerset (set of all sets) of all possible pairs of sub-structures from concepts of $\mathcal{U}$. In this module, we propose to use exclusively the concept maps. To state this more clearly, for any pair, CM_1 and CM_2, of concept maps, the Mapper will find a set of mappings between their elements, each pair having one element from CM_1 and one from CM_2.

In his formalization of Conceptual Blending, Goguen introduces the *semiotic morphisms*. A semiotic morphism is a structure preserving mapping, as it should map sorts to sorts, subsorts to subsorts, data sorts to data sorts, constructors to constructors, etc. [Goguen, 1999]. However it is assumed that these should only be partial maps. As far as we are aware, Goguen does not suggest any specific algorithm for building semiotic morphisms. We believe that an algorithm such as the one implemented for the Mapper could be a viable solution.

The Mapper uses a spreading activation algorithm to look for the largest isomorphic pair of subgraphs (contained in the concept maps). In this context, two graphs are considered isomorphic when they have the same relational (arcs) structure, independently of the elements (nodes). There is potentially more than one structure matching between any pair of concept maps and this complexity grows worse than exponential with the number of elements (nodes) [5] However, since it only allows alignment when it finds equal relations in both graphs, the number of possible solutions can be drastically reduced, while still demanding that Mapper makes the search in a huge space. Furthermore, the algorithm starts with a randomly selected pair of elements, so the "perfect choice" (or even the same choice) is not guaranteed every time we run it.

The Mapper uses an algorithm of structure matching inspired by Tony Veale's Sapper framework [Veale, 1995]. We have already presented Sapper and therefore

[5] Assuming n as the number of elements of the largest (in number of elements) of the two concept maps, we will have a search space of at most $n!$ possible mappings. So, with an exponential k^n, as n approaches infinity, $\frac{k^n}{n!}$ will approach 0, meaning that the search space will expand more than exponentially as the number of elements grows. In little-o notation, we have that k^n is $o(n!)$.

the differences will now be further enhanced. While Sapper needs two cycles to obtain the mapping (one for laying down dormant bridges with the triangulation rule and one for finding the mapping)[6], our Mapper uses three cycles: one for laying down dormant bridges (with both triangulation and squaring rules); another one for spreading activation (in our case, a flood fill algorithm); and a final cycle for finding the mapping. In the first cycle, Mapper builds new dormant bridges whenever two elements from the two input concept maps share the same relation to the same element (the triangulation rule). Here, the generic domain (and particularly the isa-ontology) is extremely important because it is a source of shared knowledge. The Mapper also adds a dormant bridge between every two elements that share the same relation to two different elements that are connected by a dormant bridge (the squaring rule). Thus, while Sapper adds dormant bridges as the mapping is found, the Mapper creates all possible dormant bridges in the first cycle. The second cycle in Mapper spreads activation throughout the concept maps. This is different to Sapper where this activation has no prime factorization or wave. It has only an activation value that decays as it passes by elements. This activation starts at 100 and is reinforced when passing near a dormant bridge. Below a threshold (the default value is 20), it stops spreading. After this second cycle, the network will have a set of sub-graphs with activated elements, centered in the dormant bridges. The final cycle starts with the random (or user-given) choice of one of the dormant bridges, the *seed* mapping. This dormant bridge is *awakened*, and thus becomes the first mapping. Then it progresses in parallel in both concept maps, so that each new pair of elements to be mapped (i.e. each dormant bridge visited) is connected by a pair of equivalent relations to a previously awakened dormant bridge.

As a result of the algorithm, the Mapper returns a set of mappings between the two concept maps. This module was born out of an idea to implement a version of Sapper that would not worry about returning the best (widest?) mapping or would bias the mapping towards the highly activated nodes (for instance, in Sapper, the choice "Scalpel: Snub-Fighter" beats out "Scalpel: B-52" [Veale, 1995, chapt. 6] due to higher activation; in Mapper, any could be selected). The principle was that, if it clearly became less effective (slower, with smaller mappings, etc.) than Sapper, then we would directly use Veale's algorithm. However, we gradually found that the Mapper had limitations that would not be resolved by changing to Sapper. As we will see in the experiments, restricting mappings to structure alignment narrows the potential of the system, thus we gave the Mapper a secondary (i.e. optional) role in

[6]Actually, the latest version of Sapper, which has no "prime factorization or wave" employs only one phase [Veale and O'Donoghue, 2000]. But we based ourselves in Veale's PhD thesis work, thus we will focus on that version. Furthermore, as said before, the Sapper project has much more testing and documentation for the first version than for the latter one.

			vegetable_food	↔	vegetable
			food	↔	food
ear	↔	wing	horse	↔	bird
nose	↔	bird	equinae	↔	aves
eye	↔	lung	animal	↔	animal
mouth	↔	feathers	human_setting	↔	house
2	↔	2	wilderness	↔	wilderness
hear	↔	fly	ruminant	↔	oviparous
			run	↔	fly
	1		cargo	↔	pet
			neigh	↔	chirp
			nose	↔	lung
			mane	↔	feathers
			tail	↔	beak
mouth	↔	beak	leg	↔	eye
nose	↔	bird	hoof	↔	wing
eye	↔	lung	4	↔	2
ear	↔	feathers	eye	↔	leg
eat	↔	eat	ear	↔	claw
			hear	↔	catch
	2		grass	↔	grass
				3	

Figure 5.4: The three mappings

Divago. On the other hand, the behavior of the module was sufficiently satisfactory to be retained in some situations. In spite of the complexity involved, this module is fast in returning a mapping and it achieves the same results as Sapper in the majority of the time .

As an example, we show in figure 5.4 the three different mappings produced for the concept maps of horse and bird (from tables 5.1 and 5.2). It is important to understand that every relation has the same weight in the graph and there is no domain knowledge or special heuristics considered in the mapping construction. This means that the results may contain non-intuitive associations (e.g. “4” associated with “2”; “nose” with “bird”).

5.4 Blender

In Goguen's algebraic semiotics Blending formalization [Goguen, 1999], a blend is some *sign system* that results from the semiotic morphisms from the input and generic spaces (the *injections*). These morphisms should be mutually consistent. The "best blend" (to what he calls 3/2 *pushout*) would thus result in an ordering of semiotic morphisms by quality, e.g. they should be as defined as possible, should preserve as many axioms as possible, and should be as inclusive as possible (i.e. contain the maximum number of mappings between concepts). Although this author considers this "best blend" as the best result over a conjunction of compromises between criteria, again it is not clear what exactly these criteria are, apart from structure mapping. Generically, we follow some of the same ideas as Goguen, namely the application of criteria for ordering a set of candidate blends, and having in structure a fundamental index for quality. The Blender is the first part of this approach (which will be completed with the Factory, the Constraints and the Elaboration module) and focuses on calculating the set of all possible blends.

Assuming a mapping m, generated by the mapping operation ϕ, as defined by the Mapper or by an external source, we must specify the *transfer operation*, ω, which will transfer knowledge from two concepts into one (as in chapter 4). As with the mapping function, so the transfer operation works with the concept maps.

First, we have to define what a *blending projection* is. A blending projection of an element x from concept map CM is a non-deterministic operation that maps x to another element (in the *blend*) which is either x, $\varnothing$, $x|y$ or y (y is the counterpart of x, i.e. $(x, y) \in m$). The symbol $x|y$ is called a *compound* and can be read as being *both x and y at the same time*. In order to consider this symbol, we must have the alphabet $\mathcal{A}_B$, which contains the alphabet $\mathcal{A}$ plus every combination of pairs $x|y$ that are possible to obtain from symbols of $\mathcal{A}$. Thus, given $\mathcal{A}$ and $\mathcal{A}_B$, two concept maps CM_1 and CM_2 (the two input concepts), a mapping m (given by ϕ), a *blending projection* γ is the operation γ: $\mathcal{A} \longrightarrow \mathcal{A}_B$, such that:

$$\gamma(x) = \begin{cases} x \vee x|y \vee y \vee \varnothing & \text{if} \quad x \in CM_1, \exists y \in CM_2 : (x, y) \in m \\ x \vee y|x \vee y \vee \varnothing & \text{if} \quad x \in CM_2, \exists y \in CM_1 : (y, x) \in m \\ x \vee \varnothing & \text{if} \quad (x \in CM_1, \nexists y \in CM_2 : (x, y) \in m) \ \textit{or} \\ & \quad\ \ (x \in CM_2, \nexists y \in CM_1 : (y, x) \in m) \end{cases}$$

Informally, a blending projection determines, for each element of a concept map, what its correspondent will be in the blend. When such an element (x) has a counterpart in the mapping, then it can be *projected* as a copy (x), as a compound with

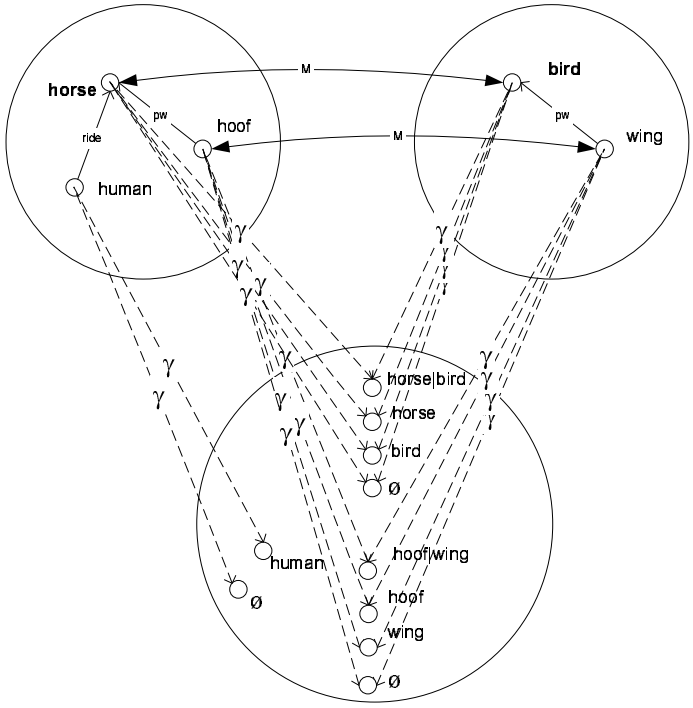

Figure 5.5: The blending projection applied to two small concept maps.

the counterpart $(x|y)$, directly as its counterpart (y) or have no projection at all. For example, from the third mapping in figure 5.4, "wing" could be projected to "wing", "hoof|wing", "hoof" or be absent in the blend. In figure 5.5, we sketch all possible projections from two little concept maps. Notice that "human" has no mapping counterpart, therefore it can only map to its copy or to $\varnothing$.

A blend is defined by the blending projections. The transfer operation, ω, is defined by an algorithm that composes the blend by transferring knowledge from the inputs to the blend, according to the projections. It corresponds to the step of *Composition* of the Conceptual Blending framework. The algorithm follows:

```
   Input:
Two input concepts, C1 and C2, defined by the pairs (CT1, I1) and (CT2, I2),
respectively, with CT1=(CM1, R1, F1, IC1) and CT2 =(CM2, R2, F2, IC2).


Algorithm:
Let Blend ⟵ {}
For i=1,2 do
  For each relation r(a,b) in concept map CMi do
      Add relation to Blend with the form
```

```
        r(γ(a), γ(b)), iff γ(a) and γ(b) are not ∅
  EndDo
  For each rule r from R_i, in the form
        r = c_1(x_1, y_1) ∨ c_2(x_2, y_2) ∨ ... ∨ c_m(x_m, y_m) ∨ {Code_c} ⟵ p_1(z_1, t_1)
        ∧p_2(z_2, t_2) ∧ ... ∧ p_n(z_n, t_n) ∧ {Code_p}, do
      Add new rule to Blend such that each c_m(x_m, y_m) is substituted by
      c_m(γ(x_m), γ(y_m)) (when γ(x_m), γ(y_m) ≠ ∅) and p_m(z_m, t_m)
      is substituted by p_m(γ(z_m), γ(t_m)) (when γ(x_m), γ(y_m) ≠ ∅)
      Copy Code_p and Code_c (the scripts) directly to the new rule.
  EndDo
  For each frame f from F_i do
      Apply the same process as with rules
  EndDo
  For each integrity constraint ic from IC_i do
      Apply the same process as with rules
  EndDo
  For each instance s from I_i, in the form
        s = I(x_1, x_2, ..., x_n), do
        Add new instance to Blend with the functor I and apply the same process as
        with relations (but with arity n and recursively)
  EndDo
EndDo
```

This algorithm basically creates a blend by applying the projections to all the constituents of the input concepts. As the projection operation, γ, is non-deterministic, when this algorithm is applied without selecting specific projections (i.e. without restricting to only one projection for each element of the concept maps, as will be done by the Factory module), it does not produce a single blend, rather it generates what we call a *blendoid*. The blendoid contains all possible constituents (relations, rules, frames, instances and integrity constraints) that can be present in any blend of two specific input concepts. In other words, it implicitly includes all the search space of blends (see example in figure 5.6).

Taking a close look over the search space of blends, we notice that, for an *input* concept 1 with a concept map with m different elements and an *input* concept 2

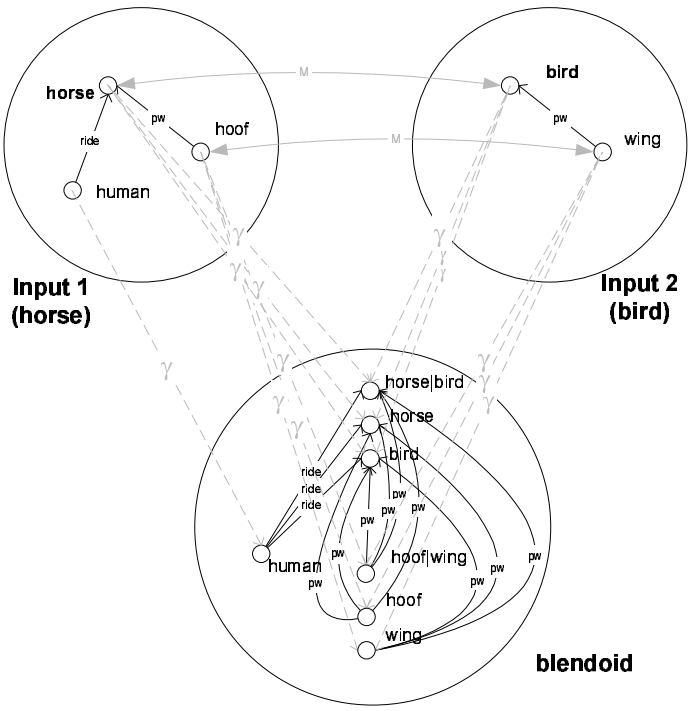

Figure 5.6: A blendoid.

with n elements, we may have the maximum of $\sum_{i=1}^{k}[\binom{m}{i} \cdot \binom{n}{i} \cdot i!]$ different mappings (if we use the isomorphic mappings, as in the Mapper), with the largest mapping having a size $k = min(m, n)$. To understand this formula, let us consider the extreme case in which all relations (in both concept maps) are equal. Counting all mapping sizes (from 1 to k), we have, for a mapping size[7] i, all combinations of m elements matching all arrangements of n elements. In reality, the number of mappings is much lower since there is a variety of different relations in both inputs. Furthermore, we may also assume that the Mapper will normally produce only the largest mappings (smaller mappings are generated only when the Mapper loses activation prematurely when doing the spreading activation process)[8].

Assuming each blending projection is independent, we will have a total of $l = m+n$ different projections in the definition of every blend. So, in the "least complexity scenario", the size of the mapping is 0, meaning that we only have two choices for

[7]A mapping of size i will have i correspondences between input concept 1 and input concept 2.

[8]This helps to understand why, in spite of this combinatorics analysis, the actual number of different mappings generated is rather small (e.g. for the concept maps in tables 5.2 and 5.1, the Mapper generates only three mappings, see fig. 5.4).

each of the l elements (either it gets projected to the blend or it is not projected), thus we have 2^l different blends. If the size of the mapping is k (the maximum possible), we have four choices for each of $2k$ elements (k elements in each of the domains) because each element x mapped to y can be projected either to x, y, $x|y$ or $\varnothing$. Apart from these $2k$ elements, the rest $(l - 2k)$ has only two possibilities. This leads us to the conclusion that, for a mapping of size s $(0 \leq s \leq k)$, we have $4^{2s} \times 2^{l-2s}$ different possible blends to choose, which is a very large search space. For example, for $m = n = 20$ (an "average" sized pair of networks), we have at least 2^{40} (if $s = 0$) and at most 4^{20} (if $s = 20$) different solutions. If we remember that the Optimality Principles are mutually competing pressures, then we may guess that this is a very complex search space. Obtaining a *good blend* is the main motivation for the Factory module, which will be the subject of the next section.

The Blender module provides two fundamental services to the Factory: it generates the blending projections (only once and before the Factory starts searching); it provides the transfer operation, which is used by the Factory each time it needs to create a blend.

5.5 Factory

The Factory is the processing core of Divago. It corresponds to the reasoning mechanism of our model of concept invention and is responsible for applying the divergent strategy, $\mathcal{T}_d$, which is, as we will see, encoded as a *genetic algorithm*.

In our context, the output of the Factory, i.e. the invented concept, will correspond to a blend. Since a blend is primarily defined by a string of projections, then searching for an invented concept becomes searching for the string of projections that originates the best blend. A string of projections that comprises one and only one projection for each element from the input concepts is called a *selective projection*. Therefore, as discussed in the previous section, for a pair of input concept maps, and a mapping of size s, we have a number of $4^{2s} \times 2^{l-2s}$ different possible selective projections.

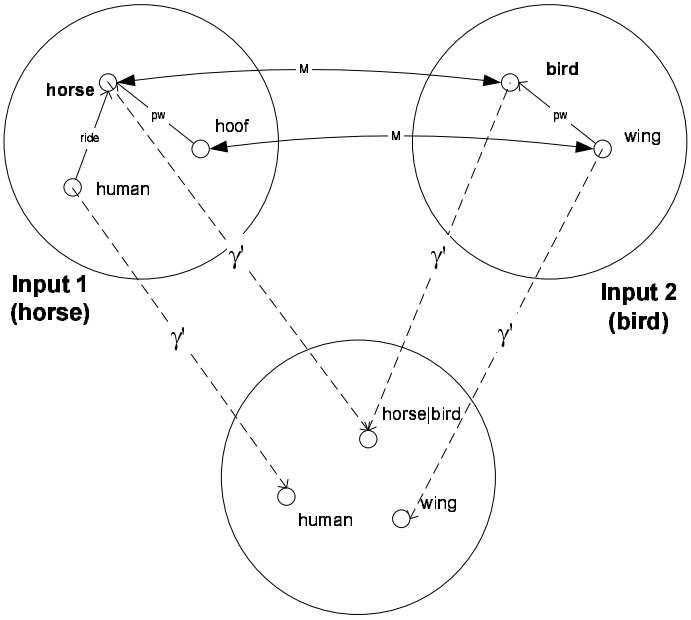

Figure 5.7: A possible selective projection from the example of figure 5.5.

Assuming $\gamma'(x)$ as an algorithmic function that returns each time one and only one of the possible projections allowed by $\gamma(x)$, and the pair of concepts C_1 and C_2 (containing concept maps CM_1 and CM_2, respectively), a *selective projection*, λ can be defined as:

$$\begin{aligned}\lambda(C_1, C_2) = \quad & \{\gamma'(x_1), \gamma'(x_2), \ldots, \gamma'(x_m), \gamma'(y_1), \gamma'(y_2), \ldots, \gamma'(y_n)\}, \\ & \text{with } x_m \in \mathcal{AE}_{CM_1} \wedge y_n \in \mathcal{AE}_{CM_2}\end{aligned}$$

with $\mathcal{AE}_{CM_i}$ corresponding to the set of elements that are present in CM_i.

Given the complexity of the search space, and for computational reasons, we decided to implement a parallel search algorithm, a genetic algorithm (GA): a framework inspired by evolutionary theories in which we have a sequence of *populations* of *individuals*, each *individual* having a *fitness* value that represents its *survival* and *reproduction* possibilities. Each individual has a *genotype*, its birth given genetic code, and a *phenotype*, the actual interpretation of the genotype (in nature, the animal itself). A genetic algorithm works as follows:

1. N individual genotypes are randomly created (sometimes the researcher might have an idea as to what is a good genotype and would direct the creation of the

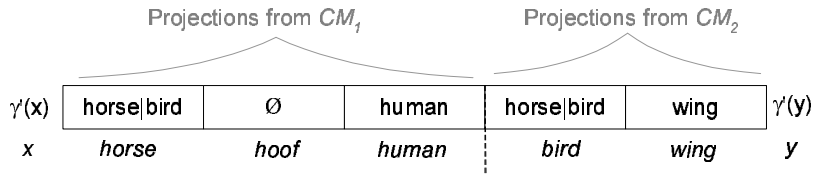

Figure 5.8: The genotype of the individual corresponding to the selective projection of figure 5.7.

initial population). These individuals are the initial population.

2. Each individual in the population is evaluated by a *fitness* function. This evaluation is based on the phenotype.

3. The best individuals are chosen for reproduction. This choice can be based on aspects other than fitness value (e.g. biodiversity).

4. The genotypes of the chosen individuals reproduce using the methods of crossover and mutation and a new population is formed. Other operators are also used. In our case, we also use asexual reproduction (direct copy of the genotype)

5. Steps 2 - 4 are repeated for a set amount of times or until a halt condition is met.

This well-known framework has had much success in problems with a search space with the complexity characteristics that we described. Further explanation of GA's is far outside of the scope of this book, so we direct the interested reader to [Goldberg, 1989].

In our GA, the genotype corresponds to a "selective projection". The *individual* is thus an ordered sequence of projections (the *genes*), each one with one of the allowed values (from the set x, y, $x|y$ and $\varnothing$). The phenotype is constructed with the transfer operation, as given by the Blender module, and elaborated by the Elaboration module. In figures 5.8 and 5.9, we show examples of the genotype and the phenotype (generated with the transfer operation).

The initial population has 100 individuals selected randomly. The evaluation of an individual is made by the application of the Optimality Principles, which then

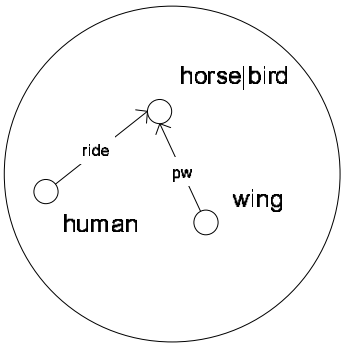

Figure 5.9: The phenotype - a *horse|bird* has wings and is ridden by humans.

participate in a weighted sum, yielding the fitness value. This work is performed in the Constraints module. The selection of the individuals is of the *roulette-wheel* type, i.e. the ones with higher fitness have a greater probability of being chosen. Our algorithm uses 3 reproduction operations: asexual reproduction (the individual is copied to the next population); crossover (two individuals exchange part of their list of projections) and mutation (random changes in the projections). It also allows the random generation of a new individual, which can be useful when the current population has low biodiversity. The system stops when a predefined number of iterations of this process has been done, when it stabilized around an acceptable maximum for more than a predefined number of iterations or when an individual was found that has a satisfactory value.

As we have said before, the choice for a GA is connected to computational preferences, thus we do not make a claim for any cognitive implications in this matter. Other choices could be followed (e.g. simulated annealing), but GAs are highly versatile (with a simple tweak in the process, it can be changed into a simulated annealing) and offer a bulk of experimental and theoretical bases from which to apply methodologies or choose parameters. With this GA, Divago is able to search in a huge space of blends according to the preferences of the user. The best solution is not guaranteed, but it is reasonable to expect that the higher the number of iterations, the more likely it is to find a good blend, if one exists in the search space.

5.6 Constraints

The Constraints module implements the Optimality Principles. It makes a preprocessing of each blend (checking frame satisfaction and completion, integrity constraint violation, vital relation projection, etc.) and then obtains a value for each of the eight measures. These values then participate in a weighted sum, which yields the *value* of the blend (normalized to fall into the [0,1] interval) which is returned to the Factory. The weight attributed to each optimality pressure is defined by the user. Our proposal for a computational realization of the eight Optimality Principles concerns solely the representation and scope of this model[9]. This doesn't mean that this proposal should not be verified or tested with regard to cognition and the blending phenomena in general, rather we state that we didn't base our measures on cognitive experiments, but mainly tried to follow the philosophy behind the description given by Fauconnier and Turner in the context of our formal model.

While these constraints consider *usefulness*, as well as many other aspects, it is clear that they lack any explicit concern to *novelty*. Instead of adding an extra constraint for novelty, we decided to face it as an effect rather than as a construction principle. In other words, we intend to verify if, with the present architecture and constraints, Divago can produce novelty. In the experiments, we will be able to see that it is capable of some degree of novelty.

We will now present our computational version of the eight optimality principles.

5.6.1 Integration

Frames gather knowledge around abstractions, tightening the links between elements. They organize a concept into a more understandable whole. For example, in figure 5.10, two specific frames integrate the blend into a more broad concept of "flying equinae".

[9] A first approach was published in [Pereira and Cardoso, 2003b], followed by [Pereira and Cardoso, 2003c]. The current version is a revision with differences in Topology,

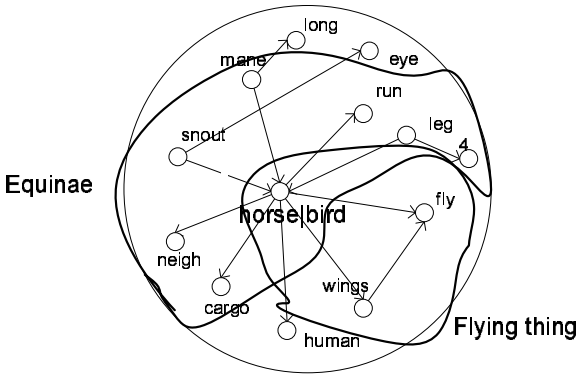

Figure 5.10: The blend satisfies (or *accomplishes*) the frames "Equinae" and "Flying thing"

Assuming the set F of frames that are satisfied in a blend, we define the *frame coverage* of a blend to be the set of relations from its concept map that belong to the set of conditions of one or more frames in F. The larger the frame coverage of the blend, the higher the integration value should be. Yet, a blend that is covered by many frames should be less integrated than a blend with the same coverage, but with less frames. In other words, if a single frame covers all the relations of a blend, it should be valued with the maximal integration, whereas if it has different frames being satisfied and covering different sets of relations, it should be considered less integrated. The intuition behind this is that the unity around an integrating concept (the frame) reflects the unity of the blend. The Integration measure that we propose varies according to this idea. It also takes integrity constraints into account so that, when a frame violates such a constraint, it is subject to penalty.

Definition 5.6.1 (*SingleFrameIntegration*). For a frame f with a set C of conditions, a blend b, with a concept map CM_b, its blendoid with a concept map, CM_{B^+}, and VI, the set of integrity constraints that are violated in the frame, the *integration* value, I_f is defined by:

$$I_f = \frac{\#C}{\#CM_b} \times (1-\iota)^{\#VI} \times (1 + \frac{\#CM_b}{\#CM_{B^+}})/2$$

being ι a penalty factor between 0 and 1, a value that penalizes a frame for each violation of integrity constraints. An integrity constraint is violated if its premises are

true. In the context of the integration measure of frame f above, f violates integrity ic if the conditions C_{ic} of ic are verified and $C_{ic} \bigcap C \neq \emptyset$. In other words, f needs to violate ic in order to be integrated.

We would like to further clarify the above formula: the first factor represents the ratio of coverage of b w.r.t. f; the second factor means that each integrity constraint violation implies an exponential discount; the third factor serves the purpose of maximizing the size of the blend (if two frames have the same ratio of coverage, the one that contains more relations should have higher integration); the division by 2 aims to normalize the result between 0 and 1.

While the value for a single frame integration is described above, the integration measure of a blend w.r.t. a set of frames is not necessarily straightforward. At first sight, it is appealing to just add the values of integration of all frames, or of the union of their condition sets. Or even their intersection. But this would lead to the wrong results, because a set of frames can not be reduced to a single frame from the point of view of integration. In this measure, we want to stimulate unity, coverage and take into account the strength of each frame individually. In terms of unity, we argue that the set of relations that make the "core" of all the frames that are satisfied (i.e. the intersection of the sets C of conditions of all frames) should be highly valued. On the other side, the coverage of this "core" will be smaller than the overall coverage (or equal, if the frames have equivalent C sets), which leads us to take into account the disjoint sets of relations of the frames. Finally, the integration of each individual frame (as defined above) should also be present in the overall measure. These last two issues (the overall coverage and the integration of individual frames) are subject to a disintegration factor because they reflect the existence of different, not totally intersected, frames. We propose this factor, α, to be a configurable value from the interval $[0, 1]$. It is now time to present our proposal for the *Integration* measure of a blend:

Definition 5.6.2 (*Integration*). Let $F_b = \{f_1, f_2, ..., f_i\}$ be the set of the frames that have their conditions (C_i) satisfied in the blend b, α, the disintegration factor

(with $0 < \alpha < 1$), and I_{f_i}, the single frame integration value, as in 5.6.1.

$$Integration = I_{\bigcap_0^i C_i} + \alpha \times Uncoverage \times \sum_0^i I_{f_i}$$

The $Uncoverage$ value consists of the ratio of relations that do not belong to the intersection of all frames w.r.t. the total number of relations considered in the frames:

$$Uncoverage = \frac{\#\bigcup_0^i C_i - \#\bigcap_0^i C_i}{\#\bigcup_0^i C_i}$$

The integration measure is fundamental to the blending process. It leads the choice of the blend to something *recognizable* as a whole, fitting patterns that help to determine and understand what a *new* concept is.

In order to illustrate this reasoning, in figure 5.11, we show 4 blends and the respective frame coverage. Blend A clearly gets the highest Integration value (all the relations are covered by a single frame); B is also totally covered, but by two different frames; Blend C should get lower Integration value than B because it does not cover every relation (Uncoverage is higher than 0); finally, blend D would possibly get the lowest value (depending on the value of α) because, although covering every relation, there is a high dispersion of frames. To help make the calculations clearer, let us pick specifically the blends A and D and determine their Integration values. To simplify, let us assume that $CM_{B^+} = CM_b$ (the blendoid size is necessary only for normalization purposes, so any size would do for this example) and that there are no integrity constraint violations. Our blend A will have a single frame integration, I_f, of 1, because it contains all 12 relations of the blend. Since it is the only one and there are no uncovered relations ($Uncoverage$ is 0), the overall Integration will also be 1. Now, for the Integration of the blend D, we must repeat the process. There will be four frames (with I_f values of $\frac{2}{12}$, $\frac{2}{12}$, $\frac{2}{12}$ and $\frac{1}{12}$). There is no intersection between all frames, which implies that the first term of the sum will be 0. Furthermore, no two relations belong to different frames ($Uncoverage$ is maximum, i.e. 1) and the sum of our four single frame integrations will be $\frac{2}{12} + \frac{2}{12} + \frac{2}{12} + \frac{1}{12} = \frac{7}{12}$. This leads us to the second term of the $Integration$ measure: $\alpha \times 1 \times \frac{7}{12}$ (remember α is smaller

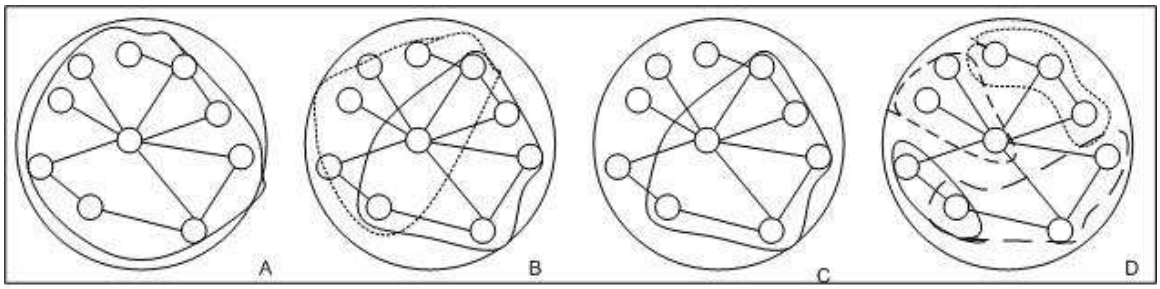

Figure 5.11: The role of frame coverage in Integration value

or equal to 1). This shows that the Integration value of blend D will be considerably smaller than that of the blend A, as intended.

5.6.2 Topology

The Topology optimality pressure brings *inertia* to the blending process. It is the constraint that drives against change in the concepts. This happens because, in order to maintain the same topological configuration as in the inputs, the blend needs to maintain exactly the same neighborhood relationships between every element, ending up being a projected copy of the inputs. In practice, this pressure is normally one that is disrespected without a big loss in the value of the blend. This is due to the *imagination* context that normally involves blends, i.e. novel associations are more tolerable. Of course, this still depends on the type of blend we are pursuing: if it regards an analogical or structure alignment construction (e.g. the Buddhist Monk), then Topology is vital; if it regards a free combination (e.g. a "horse-bird", an imaginary object with a goal), then Topology may become secondary.

In our Topology measure, we follow the principle that, if a pair of elements, x and y, is associated in the blend by a relation r, then the same relation must exist in one of the inputs. If so happens, we say that $r(x, y)$ is *topologically correct*. Thus, the value of Topology corresponds to the ratio of topologically correct relations in the concept map of the blend.

Definition 5.6.3 (*Topology*). For a set $TC \subseteq CM_b$ of *topologically correct* relations,

such that

$$TC = \{r(x, y) : r(x, y) \in CM_1 \cup CM_2)\}$$

where CM_1 and CM_2 correspond to the concept maps of inputs 1 and 2, respectively, the topology measure is calculated by the ratio:

$$Topology = \frac{\#TC}{\#CM_b}$$

Intuitively, this measure represents the amount of relations from the inputs that were projected unchanged to the blend. We are aware that this is stricter than the topology constraint in the blending framework, as it is based on identity rather than counterparts. As the reader will see, this role of analysing counterparts as maintaining original structure/neighborhood is left for the Unpacking constraint. At the moment, the only way to violate topology is by having a pair of concepts projected to the same one (e.g. "horse" and "bird" projected to "horse"), bringing a new relation that was exclusive to one of the domains (e.g. $ability(bird, fly)$ projects to $ability(horse, fly)$; $pw(wing, bird)$ projects to $pw(wing, horse)$).

5.6.3 Pattern Completion

The Pattern Completion pressure brings the influence of patterns either present in the *inputs* or coming from the *generic* space. Sometimes a concept may seem incomplete but makes sense when "matched against" a pattern.

At present, in the context of this work, a pattern is described by a frame, i.e. we don't distinguish between these two notions, and therefore pattern completion is basically frame completion. The act of completing a frame consists in asserting the truth of the ungrounded premises (which is done in the Elaboration module), a process that happens only after a sufficient number of premises is true. We call this the *completion threshold*, a value that is externally configured in Divago. To the measure regarding the conditions that are actually satisfied by a frame f in a blend b,

we call the *completion evidence* of f, $e(f_i, b)$. Therefore, completion can only happen when the completion evidence is higher than the completion threshold.

Definition 5.6.4 (*Completion Evidence*). The completion evidence e of a frame f_i with regard to a blend b is calculated according to the following.

$$e(f_i, b) = \frac{\#Sat_i}{\#C_i} \times (1 - \iota)^{\#VI}$$

where Sat_i contains the conditions of each f_i that are satisfied in b, ι is the integrity constraint violation factor and VI the set of violated integrity constraints.

As in the Integration constraint, we have the problem of taking into account multiple frames. This time, given that we are evaluating possible completion of subsets of relations, instead of sets of relations that are actually verified in the blend, it is difficult to find such a linear rationale (e.g. would two patterns each with an individual completion x value higher than three each having slightly less than x?). As a result, we propose to find the union of all the conditions contained within the patterns and then estimate its own completion evidence:

Definition 5.6.5 (*Pattern Completion*). The Pattern Completion measure of a blend b with regard to a set F with n frames is calculated by

$$PatternCompletion = e(\cup_0^n f_i, b)$$

This measure has a very important role in increasing the potential of the blend, for it brings the "seeds" that may be used in the Elaboration module. In figure 5.12, we illustrate Pattern Completion with two examples. Assuming a frame with three conditions ($pw(X, Y)$, $purpose(Y, fly)$ and $ability(X, fly)$), on the left it has a completion evidence of 66.6% (two relations out of three are already accomplished: $pw(Horse\_Bird, wings)$ and $purpose(wings, fly)$), whereas on the right the completion evidence is only 33.3% (only $pw(Horse\_Bird, wings)$ exists). For both, since we consider only one frame (i.e. one *pattern*), the value of Pattern Completion is the same as of the completion evidence.

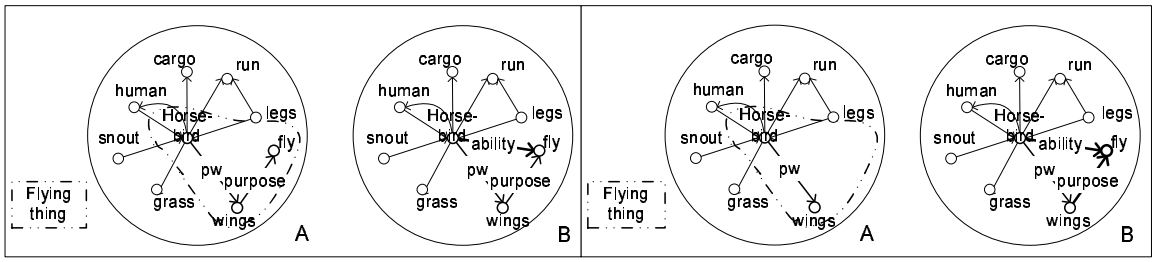

Figure 5.12: Pattern Completion examples

5.6.4 Maximization/Intensification of Vital Relations

Fauconnier and Turner propose a set of *vital relations* that should govern the blend creation [Fauconnier and Turner, 2002]. As already stated, our approach to CB relies essentially on earlier works from the same authors, thus we must point out that the approach to vital relations discussed here leaves more open questions that encountered solutions. We may say that we are facing vital relations as merely *salient* relations, without agonizing with their role in compression - which should be, one might say, their essential role. Compression is the phenomenon of bringing relations between concepts from different inputs (the *outer-space* relations) to the blend (i.e. they become *inner-space* relations in the blend)[10]. Given the fuzzy definition of Intensification of Vital Relations and the fact that we are not focussing on *compression* for this work, the distinction between Maximization and Intensification becomes yet another step towards subjectivity, which we intend to avoid. In practical terms, this means we only propose here a measure for Maximization and leave the discussion of the distinction and specific measure of Intensification for further research (possibly when a computational view of compression is explored in depth)[11].

By default, we allow the same vital relations[12] between and within two concept maps, some being only rarely used (e.g. change, disanalogy, intentionality). Divago also accepts the definition of other relations as being *vital*. For example, in inventing

[10]To know more about compression, please read [Turner, 2006].

[11]In fact, an attempt was made in [Pereira and Cardoso, 2003c, Pereira, 2005] to model Intensification of Vital Relations, yet this proposal, which assumes more than one type of mapping, could not be rigorously tested or further explored.

[12]Change, identity, time, space, cause-effect, part-whole, representation, role, analogy, disanalogy, property, similarity, category, intentionality and uniqueness.

concepts for a game, one may decide that the vital relations are "strength", "defense", "ability" and so on. The effect of this choice may be that, when giving more emphasis (higher weight in the fitness function) to Maximization of Vital Relations, the resulting blends will contain the maximum possible number of these relations.

For implementing this measure, we estimate the impact of the vital relations to the blend calculated by the ratio of vital relations w.r.t. the whole set of possible vital relations, contained within the *blendoid*.

Definition 5.6.6 ($Maximization\_VR$). Let Υ be a set of *vital relations*. From the concept map of the blend b, we may obtain the set of vital relations in b, B_{VR}:

$$B_{VR} = \{r(x, y) : r(x, y) \in CM_b \wedge r \in \Upsilon\}$$

From the blendoid (the union of all possible blend), B^+, we have B^+_{VR}:

$$B^+_{VR} = \{r(x, y) : r(x, y) \in CM^+_B \wedge r \in \Upsilon\}$$

Finally, the Maximization of Vital Relations measure is calculated by the ratio

$$Maximization\_VR = \frac{\#B_{VR}}{\#B^+_{VR}}$$

5.6.5 Unpacking

Unpacking is the ability to reconstruct the whole process starting from the blend. In our view, such achievement underlies the ability to reconstruct the input spaces. I.e. the reconstruction of the input spaces from the blend demands the assessment of the cross-space mappings, the generic space and other connections. Thus, what we are proposing is that Unpacking can be reduced to the ability to reconstruct the inputs. This is because there is no way to properly reconstruct the inputs without a reconstruction of the cross-space mappings, generic space and the connections between spaces.

Unpacking should take the point of view of the "blend reader", i.e. someone or something that is not aware of the process of generation, thus not having access to

the actual projections. Being such, this "reader" will look for patterns that point to the "original" concepts. Once again we use the idea of *frames*, more specifically the *defining frame* of an element, which comprises the immediate surrounding elements and relations. For example, if the element "wing" was projected onto x in the blend, the defining frame with regard to the "bird" concept map would consist of *purpose(x, fly), conditional(x, fly), quantity(x, 2)* and *pw(x, bird).* The more that these relations are found in the blend, the more likely it is that the "reader" will find it easy to understand the relationship between x and "wing".

Definition 5.6.7 ($DefiningFrame$)**.** Given a blend b and an input space d, the element x (which is the projection of the element x_d of input concept map d to b) has a defining frame $f_{x,d}$ consisting of

$$f_{x,d} = C_0, C_1 \ldots C_n \longrightarrow true$$

where $C_i \in \{r(x, y) : r(x_d, y) \in CM_d\}$. Assuming that k is the number of conditions (C_i) of $f_{x,d}$ that are satisfied in the blend, the unpacking value of x with regard to d (represented as $\xi(x, d)$) is

$$\xi(x, d) = \frac{k}{n'}$$

where n' is the number of elements to which x is connected. We calculate the *total estimated unpacking value* of x as being the average of the unpacking values with regard to the input spaces. Thus, having input concept maps 1 and 2, we have

$$\xi(x) = \frac{\xi(x, 1) + \xi(x, 2)}{2}$$

Definition 5.6.8 ($Unpacking$)**.** Let $\mathcal{X}$ be the set of m elements of the blend b, generated from input concept maps 1 and 2. The *Unpacking* value of b is calculated by

$$Unpacking = \frac{\sum_{i=0}^{m} \xi(x_i)}{m}, x_i \in \mathcal{X}$$

In Figure 5.13, we present the defining frame for "horse", in the "horse" concept map. In Blend 1, the element "horse|bird" (the projection of "horse") will have the

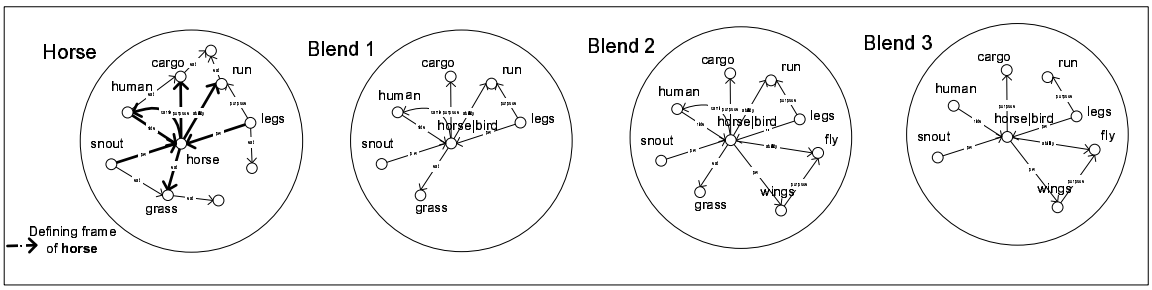

Figure 5.13: Unpacking examples

highest Unpacking value (w.r.t. "horse" concept map) because it fits precisely into its defining frame. In Blend 2, the value is lower because there are two new relations (with "fly" and "wings"), meaning that it is not exactly same element. Blend 3 will get the lowest Unpacking value of all three because it also lacks some relations (e.g. with "run" and "grass").

5.6.6 Web

The Web principle concerns being able to "run" the blend without cutting the connections to the inputs. In our opinion, this is not an independent principle, being co-related to those of Topology and Unpacking because the former brings a straightforward way to "maintain the web of appropriate connections to the input spaces easily and without additional surveillance or computation" and the latter measures exactly the work needed to reconstruct the inputs from the blend. This is not to say that Web is the same as Topology or Unpacking. Rather, on one side, Topology provides a pressure to maintain the most fundamental connection to the input: the same structure; on the other side, Unpacking evaluates the easiness of reestablishing the links to the inputs. These two values combined in a weighted sum yield, we propose, an estimation of the strength of the web of connections to the inputs:

Definition 5.6.9 (Web).

$$Web = \lambda \times Topology + \beta \times Unpacking$$

with $\lambda + \beta = 1$.

Since this is not an independent variable, making independent experiments with the Web measure would not add any valuable conclusion and thus it will not be covered in this book.

5.6.7 Relevance

The notion of "relevance" or "good reason" for a blend is tied to the *goal* of the blending generation. A blend, or a part of it, may be more or less relevant depending on what it is for. Once again, frames take a fundamental role as being "context" specifiers, (i.e. the set of constraints within a frame describe the context upon which the frame is accomplished). Therefore, having a set of *goal frames*, which can be selected from the ones available in the Knowledge Base or specified externally, a blend gets the maximum Relevance value if it is able to satisfy all of them.

An aspect of the goal frames is that they become queries. For example, if we want to find a concept that "flies", we could build a goal frame with the relation $ability(x, fly)$. The blends that satisfy this frame would be highly relevant.

Definition 5.6.10 (*Relevance*). Assuming a set of goal frames, F_g, the set F_b of the satisfied frames of blend b and the value PCN_F for the pattern completion of a set of frames F in blend b, we have

$$Relevance = \frac{\#(F_g \cap F_b) + \#F_u \times PCN_{F_u}}{\#F_g}$$

where F_u, the set of unsatisfied goal frames, consists of $F_u = F_g - F_b$. This formula gives the ratio of satisfied and partially satisfied goal frames w.r.t. the entire set, F_g of goal frames.

From the point of view of creativity, we propose the use of Relevance as a "usefulness" measure, an idea that will be applied in some experiments.

5.7 Elaboration

The Elaboration module is responsible for the application of several methods of elaboration and completion to the blend. It is where the rules and frame conclusions are triggered and where the completion of uncompleted frames (or patterns) is done. Hence, it encodes the elaboration function, θ, of the model presented in the previous chapter. It allows the three different kinds of elaboration explained: rule-based, internal logic and cross-concept based.

The rule-based elaboration, $\theta_{\mathcal{R}}$, is based on the application of rules from the generic domain. Whenever the premises of any of these rules are proven true, then their conclusions are inspected and the corresponding effects are processed to the blend. As an example, we have the rule below, where we say that, if an x lives in a *house*, and its *habitat* is *water*, then it must *live in* a *water tank*, placed *in* the *house*:

```
rule(generic, water_tank, [lives_in(X, water_tank), habitat(X, water)],
                          [lives_in(X, house)],
                          [lives_in(X, water_tank), in(water_tank, house)],
                          [lives_in(X, house)]).
```

If the premises are found to be true, then the Elaboration module will add the relations $lives\_in(X, water\_tank)$ and $in(water\_tank, house)$ to the concept map of the blend and also delete the relation $live\_in(X, house)$. Another, more complex example, is a rule that applies the movement laws to determine the meeting point of two objects $X1$ and $X2$ moving on the same line:

```
rule(generic, meeting_time, [starting_position(X1, P0X1), starting_position(X2, P0X2),
                             {X1\=X2}, speed(X1, SX1), speed(X2, SX2), day(X1, D),
                             day(X2, D), starting_time(X1, T0), starting_time(X2, T0)],
                             [],
                             [{Dif is SX2-SX1, Dif\=0,T is (P0X1-P0X2)/Dif}, meet(X1, X2),
                             time(X1, T), time(X2, T)],
                             []).
```

The internal-logic elaboration, θ_C, in this module inspects the frames that are accomplished by the blend and in turn applies their effects, as with the rules. In the special case of rules that are part of the blend (as a result of projection from one of the inputs), we can also consider internal-logic elaboration, although the procedure that controls their application is the same as for any other rules.

The cross-concept based elaboration is probably the least explored elaboration method used. It is based on pattern completion. When the completion evidence of a frame (as calculated in the Pattern Completion measure) surpasses a minimum specified value (the *completion threshold*), then the Elaboration module will add the missing relations that can be *fully defined*. For example, in figure 5.12 (left), if the completion threshold was below 66%, then the Elaboration module would add the relation $ability(horse|bird, fly)$. In this case, the relation is fully defined because both arguments ($horse|bird$ and fly) are known. If, on the contrary, the instantiation of the frame's premises had yielded $ability(horse|bird, \_)$ or $ability(horse|bird, \varnothing)$, then no new relation would be created. We call this cross-concept elaboration because it is based on the transfer of knowledge from an external concept (the uncompleted frames) to the concept map of the blend. We are aware that this may be both an unsafe and incorrect way of doing cross-concept based elaboration. It can be unsafe because, apart from the completion evidence, there is no other method for ensuring correctness or meaningfulness of the added knowledge. Only when dealing with goal frames (i.e. when there is an external motivation to accomplish the frame), does the completion have a meaningful potential consequence. It can be an incorrect perspective on cross-concept based elaboration because the source concept (the frame) is created from an analysis of internal logic (the Pattern Completion measure), rather than being another different concept that, for some reason, appears to be a good source of knowledge.

The rules and frames applied in this module may mutually influence each other. For example, the new knowledge added by a frame may in turn trigger a rule and so on. This means that the system is sensitive to order of application. To reduce this effect, the Elaboration module applies the rules iteratively until no change to the

concept map is made, i.e. until it stabilizes around a set of relations. The drawback of this approach is that it becomes sensitive to cycles. For example, suppose the following list of rules, with r_1, r_2 and r_3 being relations, and a blend containing the single relation r_1.

$$r_2 \vee not\ r_3 \longleftarrow r_1$$

$$r_3 \vee not\ r_1 \longleftarrow r_2$$

$$r_1 \vee not\ r_2 \longleftarrow r_3$$

After running the first rule, the Elaboration module would start by adding r_2, then would trigger the second rule, which would remove r_1 and trigger the third rule, which would return to the initial state, indefinitely. We are aware that this is far from an unknown problem in the area of Logic Programming, and thus we believe that good solutions may have already been found. However, for this book, and for the current version of Divago, the rules must be coded with attention to avoiding cycles, since we have not investigated farther in this subject.

This module is very useful when one is knowledgeable about the domain for which Divago is inventing concepts. For example, for the creatures experiment, designed for a game project, we added some specific rules and frames relating to solutions to problems (e.g. adding a wooden leg for a creature when there is one leg missing) and for calculating values (e.g. calculating a new *strength* value when there are conflicts between two possible candidates).

The Elaboration module could have been applied after the GA cycle, to the resulting *best blend* produced by the Factory, but this choice would imply that the generation of the *best blend* itself could not take into account the improvements from the Elaboration. In other words, the system would tend to avoid generating solutions where, in spite of having originated from a low valued non elaborated blend, the result after elaboration would compensate its previous imperfections. This was the main reason for integrating it within the GA of the Factory.

As with the Mapper, so the Elaboration module is optional for it can obscure the inner workings of the blending mechanism in hiding imperfections of the blend. Since

this is important for the validation of the system, we also allow the selection of the specific elaboration methods to apply.

Getting back to the model of concept invention that was the subject of the previous chapter, we said that the elaboration function, θ, should be used to define the space that would be traversed by the convergent strategy. Instead, the Elaboration module is being (optionally) applied by the genetic algorithm of the Factory, which we described as our divergent strategy. In this case, to be coherent with ourselves, we ought to acknowledge that, when the Elaboration module is used, we have a strategy that shares both the divergent and convergent perspectives of the model. Thus, we present here no *pure* convergent strategy. At first sight, this could be seen as a flaw in Divago and results from the deliberate compromise of focusing divergence and bisociation and disregarding other issues. However, and also for the same reasons, it became gradually clear that the flaw is in the model itself. In fact, it seems much more natural to consider convergence and divergence intermixed with each other, rather than having a strict separation. We recall the analyses made in section 2.1, in which we met the convergence/divergence dichotomy. Although this duality was salient, in nowhere it was proposed to exist a strict separation (rather, an interaction was often considered).

5.8 Divago as a creative system

To finalize this chapter, let us analyze Divago with the same criteria of section 2.2.2, where a classification of creative systems was proposed:

- Architecture. **Single agent**. As we have explained before, the approach that we are following is centered on a single isolated system. In future stages, a natural development would be to include it in a multi-agent environment.

- Model. Divago fits entirely the **Cognition Centered Model** (CCM) paradigm,

as it was developed from analyses of creativity from Cognitive Science, Psychology, Philosophy and AI, and most of all it partially integrates a computational implementation of a proposal for a cognitive mechanism, named Conceptual Blending.

- Episodic Memory. Divago has **NO** true mechanism of Episodic Memory. Although its implementation may possibly imply little more than a feedback loop (the outputs would become part of the KB), we are dedicated to a *feedforward* version, as it showed sufficiently complex by itself.

- Evaluation. Divago has a **built-in** evaluation made by the Constraints module. The only active participation of an external entity happens in the beginning of the process (by setting the goal and the weights to associate to each optimality constraint, and possibly also the input concepts and a mapping).

- Theme. The theme of this project is **Concept Invention**.

- Reasoning paradigm. Divago is clearly a **Hybrid** system, in that it makes use of rule-based reasoning, genetic algorithms and, to a much lesser degree, connectionism (in the Mapper module).

- Domain. As we will see in the next chapter, Divago has been applied to a **variety of domains**, namely 2D drawings, 3D creature design and linguistic creativity.

Chapter 6

Experiments

Ever since its first sketches, Divago has been subject to experiments in several domains. At the risk of making experimentation itself divergent, we decided on this in-breadth approach for two reasons: we have been arguing in this book for the consideration of multi-domain environments in computational models of creativity; with an in-depth approach, we would focus more gradually on specific domain issues than on Divago itself. Whatever the approach, validating this system with respect to creativity is a goal that we have been pursuing, without finding any definitive and uncontroversial solutions as yet. Because this is a fragile issue, we must follow the most solid principles we can find. First, we need to avoid building the input knowledge structures involved ourselves, the only exceptions being the first two experiments, the house-boat and the horse-bird. Second, we try to *read* as little as possible from the results except when there is a well-defined interpretation mechanism. In other words, we try to avoid putting our own point of view on to ambiguous events. Third, we seek to provide the statistically most significant data as possible to support the claims and conclusions achieved. We follow the Central Limit Theorem, which says that "the distribution of means from repeated samples from a population will be approximately normally distributed if sample size is large (> 30) and exactly normally distributed if the population itself is normally distributed". Therefore, without knowing in advance the distribution of the populations involved, we will rely on the condition that each sample must be large. For example, each of the optimality constraint weight configurations tested was subject to 30 runs with the same starting conditions. Fourth,

we estimate some of the features for creativity assessment, as presented in section 2.2, which will allow us to follow the same evaluation framework throughout the experiments, which will be useful for comparing the behavior of Divago within its own domains, as well as providing benchmarks for future comparisons with other systems.

The experiments are presented in chronological order and so the reader will also perceive the evolution of the system in terms of the modules used, the control over the results, the methodology and the interpretation of the results. In the boat-house, we generate the whole set of possible combinations (of boat|house drawings), using only the Mapper and the Blender. Only in the horse-bird experiments were we able to apply an objective analysis of the results, when we applied the Factory and the Constraints modules for the first time. There, we made the definition of novelty (the *nov* function) that was applied to the rest of the experiments. The noun-noun experiments were meant to test Divago with a large Knowledge Base and to compare it with C^3, a Concept Combination system (see chapter 3). The creature generation experiments are a study for the application of Divago to game environments and also the first experiments with the use of the Elaboration module. Finally, we apply Divago to some established blending examples in order to validate it as a Blending model.

It will be obvious that, as we experiment with the system, some Optimality Constraints are preferred over others, leading eventually to the elimination of some and to the conclusion that one could reduce the list to a subset of fundamental Optimality Constraints.

We will not hide the difficulties in analyzing Divago, namely in respect to the value (or quality or usefulness) of the results, the evaluation of the fine-tuning of the system or the individual effect of each of its components in the results. Nevertheless, we hope to provide a set of objective conclusions and benchmarks that may be useful for future comparisons.

6.1 The Boat-House

The first extensive experiment we made with Divago, published in [Pereira and Cardoso, 2002], had the goal of generating and analyzing the entire search space from two input concepts. The context then given would be that of a system with a specific goal (e.g. draw a "house"), but with a limited set of possibilities (e.g. only one drawing example available), that would *ask* Divago to extend its knowledge base of possible drawings.

In this experiment, the knowledge representation was restricted to concept maps (built using Clouds [Pereira and Cardoso, 2000]) and instances. Such a limited representation was chosen because, apart from practical reasons (this was the first experiment with Divago), an important decision was to follow whose view on concept representation would be the center of further developments (either micro-theory or exemplar view).

The choice for a "house" and a "boat" was made after some blending and concept combination works (e.g. [Goguen, 1999, Andersen, 1996]). We intended to blend these two concepts and interpret the newly generated instances according to an unambiguous process. In this case, we decided to define them according to a simple language (very similar to *Logo* [Abelsson and diSessa, 1981]), which enabled us to draw simple objects (a house and a boat) and see the generated space without heavy computational work. In this language, examples of commands are `on/5`, meaning "draw line for 5 pixels", `off/5` meaning "move 5 pixels without drawing" or left/45, meaning "turn left 45 degrees". This language as well as the syntax of the instances are described in Appendix E.

The tables 6.1 and 6.2 show the concept maps of "house" and "boat". A short interpretation of these concept maps tells us facts like "a boat has a sail, a hatch, a mast and a vessel, the vessel is the floating structure that serves as container" or "humans live in houses, that have many rooms, a roof, a window, a door and a

isa(house,physical_structure)	isa(human,mammals)
isa(physical_structure, physical_entity)	isa(night, time_object)
isa(time_object, information_entity)	isa(skyscraper, physical_structure)
isa(door, physical_object)	isa(window, physical_object)
isa(roof, physical_object)	isa(observation, task)
purpose(roof, protection)	isa(protection, task)
isa(body, physical_object)	isa(container, physical_object)
isa(room, house_part)	isa(house_part, space_location)
isa(day, time_object)	isa(water_proof, property)
isa(tree, vegetable)	isa(vegetable, living_entity)
live_in(human, house)	color(night, black)
have(house, door)	have(house, window)
have(house, roof)	have(house, body)
purpose(body, container)	purpose(window, observation)
purpose(door, entrance)	property(skyscraper, very_big)
purpose(body, container)	have_many(skyscraper, house)
have_many(house, room)	

Table 6.1: The house domain concept map

isa(boat, physical_structure)	isa(sailing_boat, boat)
isa(sail, physical_object)	isa(movement, task)
isa(triangle, geometric_form)	isa(geometric_form, information_entity)
isa(water_proof, property)	isa(hatch, physical_object)
isa(observation,task)	isa(mast, physical_object)
isa(vessel, physical_object)	shape(sail, triangle)
shape(hatch, circle)	have(sailing_boat, sail)
have(sailing_boat, hatch)	have(sailing_boat, mast)
have(sailing_boat, vessel)	have(vessel, floating_structure)
purpose(sail, movement)	purpose(hatch, observation)
purpose(mast, support)	purpose(vessel, container)
property(sailing_boat, slow)	property(hatch, tiny)
property(boat,water_proof)	place(sailing_boat, sea)
use(human, sailing_boat)	sail(human, sailing_boat)

Table 6.2: The boat domain concept map

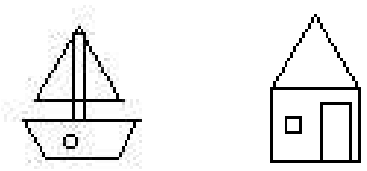

Figure 6.1: The boat and the house, as drawn from the instances

entrance	↔	movement	body	↔	sail
task	↔	task	container	↔	movement
protection	↔	support	door	↔	hatch
roof	↔	mast	entrance	↔	observation
door	↔	sail	house	↔	sailing_boat
house	↔	sailing_boat	physical_structure	↔	boat
physical_structure	↔	boat	window	↔	mast
window	↔	hatch	roof	↔	vessel
body	↔	vessel	water_proof	↔	slow
water_proof	↔	slow	protection	↔	container
container	↔	container	observation	↔	support
observation	↔	observation			
1			**2**		

Table 6.3: Two mappings for the house-boat experiment

body[1]".

The Mapper generated 4 different mappings. In table 6.3, we show the two mappings which most commonly appeared. While some concept mappings come naturally (like "window-hatch" or "body-vessel", in mapping 1), others, less intuitively acceptable, appear as a consequence of the exhaustiveness of the mapping function. For example, "water_proof-slow" appears because both can be "properties" of something (e.g. "physical_structure can be water_proof", and "boat can be slow").

In table 6.4, we can see an excerpt of the blendoid corresponding to mapping 1. According to it, possible relations in a blend could be that "a house|sailing boat has a window|hatch that serves for observation, a door|sail that serves for entrance|movement and has the shape of a triangle", etc. Notice the combinatorial explosion that results from the choices given by the blending projection operation.

Apart from definitions of shape that emerge in the blend (like, *shape*(*door*|*sail*,

[1]We are calling *body* to the four walls that hold the house

isa(entrance|movement, task)	purpose(door|sail, entrance|movement)
isa(entrance, task)	purpose(door, entrance|movement)
isa(movement, task)	purpose(sail, entrance|movement)
isa(roof|mast, physical_object)	purpose(door, entrance)
isa(roof, physical_object)	purpose(sail, entrance)
isa(mast, physical_object)	purpose(door, movement)
purpose(roof|mast, protection|support)	purpose(sail, movement)
purpose(roof, protection|support)	shape(door|sail, triangle)
purpose(mast, protection|support)	shape(door, triangle)
purpose(mast, protection)	shape(sail, triangle)
purpose(mast, support)	have(house|sailing_boat, body|vessel)
purpose(roof, protection)	have(house, body|vessel)
purpose(roof, support)	have(sailing_boat, body|vessel)
have_many(house|sailing_boat, room)	have(house, body)
live_in(human, house|sailing_boat)	have(sailing_boat, body)
live_in(human, house)	have(house, vessel)
live_in(human, sailing_boat)	have(sailing_boat, vessel)
have(house|sailing_boat, door|sail)	have_many(skyscraper, house|sailing_boat)
...	...

Table 6.4: The blendoid concept map for house and boat

triangle)), we don't know exactly how to produce the visual re-interpretation of objects (e.g., what is the visual shape of *door*|*sail*?). In other words, how can we *read* a blend? Since there is a (visual) precise semantics for some of the concepts involved (such as *roof*, *sail* or *door*), in the form of Logo procedures, we must decide how to use them to produce the new drawings. In the case where these concepts are found alone, their interpretation is straightforward (just read the corresponding Logo procedures in the potentially new context), but in the case of compounds (e.g. *door*|*sail*), the problem becomes difficult. The ideal solution would be to find a way of getting one degree of abstraction down and also blend the Logo procedures themselves. However, the explorations done in this direction were leading to a degree of complexity unjustifiable for the goals of the experiment. For this reason, the interpreter made for these drawings ignored the compounds and produced both alternatives (e.g. for *door*|*sail*, a drawing with *door* and another with *sail*), thus producing the same results as with separate projections. Perhaps the most important conclusion from this experiment was that the interpretation of a compound in itself opens up another

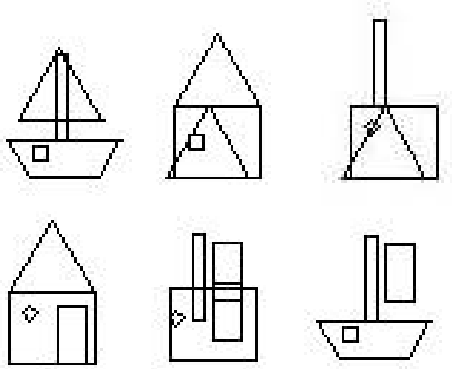

Figure 6.2: Images that result from mapping 1.

blending problem, recursively until a final and definitive answer is found (a possible solution in drawings could be to apply *visual morphing*). In spite of reducing the search space considerably by not proposing a different interpretation for compounds and only taking into account the mappings that would have visual effects, a large set of new drawings was produced. The mappings presented in table 6.3 generate respectively a set of 240 and 408 drawings with repetitions (giving approximately 80 and 100 different images, resp.). From mapping 1, we found drawings such as those shown in figure 6.2.

By analyzing these house|boats, we can see some subtle transfers (e.g. the square hatch in the first sail boat; the circular window in the house) and some blends that clearly share knowledge from both inputs, either visually fortunate (e.g. the boat with rectangular sail) or unfortunate (e.g. the house with the triangular door and a mast on top). It is also of relevance to say that these unfortunate instances appear as a consequence of not having specific domain-knowledge for generating a drawing or just because of unfortunate combinations (second, third and fifth images).

When applying mapping 2, the results are as shown in figure 6.3. Notice for example the different placement of the circle (and door). With the mapping 3 ($body \leftrightarrow vessel$, $door \leftrightarrow mast$ and $window \leftrightarrow sail$), Divago produced drawings such as in figure 6.4. Finally, the fourth mapping, which has a different variation ($body \leftrightarrow vessel$, $door \leftrightarrow hatch$, $roof \leftrightarrow mast$ and $window \leftrightarrow sail$), gave rise to images such

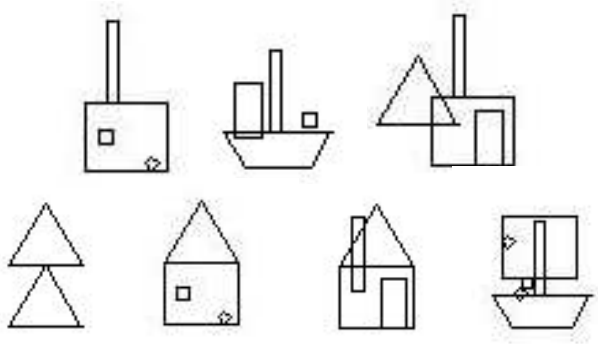

Figure 6.3: Images from the mapping 2.

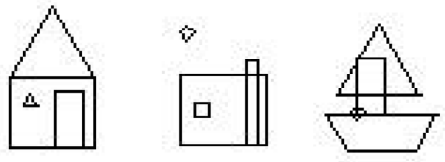

Figure 6.4: Images from the *body* ↔ *vessel*, *door* ↔ *mast* and *window* ↔ *sail* blend.

as in figure 6.5.

The visual quality of the results would vary considerably depending on the application of domain specific knowledge, such as guiding the result to what a house or a boat should look like or which physical/structural rules they should fulfill. Our goal with this experiment was to assess the generativity of the system, regardless of any aesthetical judgment. Let us return to the context given in the beginning of this section, a system with the goal of drawing a house, and imagine the situation in which this system searches for house drawings in the house domain, but cannot find any satisfactory solution. It can then try to diverge gradually from the original domain (where novelty is minimum), and get into a *space of blends*, where novelty increases,

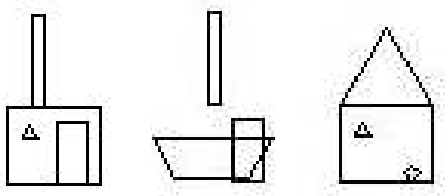

Figure 6.5: Images from the *body* ↔ *vessel*, *door* ↔ *hatch*, *roof* ↔ *mast* and *window* ↔ *sail* blend.

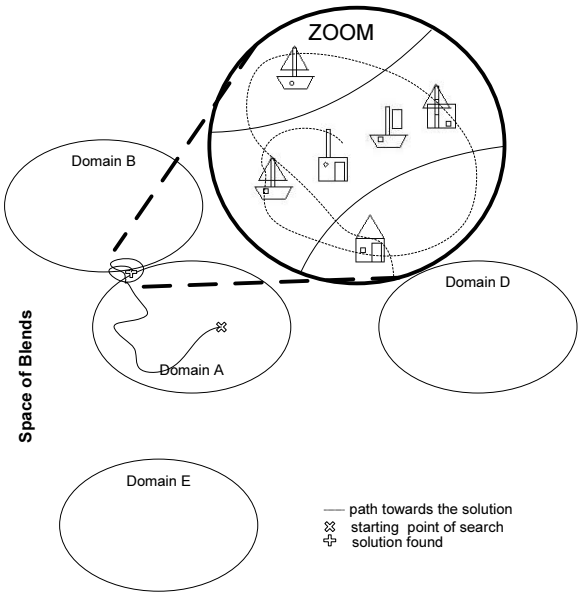

Figure 6.6: A search in a multi-domain environment

a sort of middle space where concepts do not belong to a specific domain, but share knowledge from more than one. In figure 6.6, we present this idea graphically.

Since we cannot have any precise measure of novelty or usefulness for the drawings (that is not obscured by the subjectivity of the image) we cannot do a thorough analysis of these results with regard to the criteria presented in chapter 2.2. However, we can say that Divago produces a large proportion (over 90%) of drawings that are definitely different from typical drawings of houses or boats (they gather different elements of house and boat in the same drawing, sometimes lacking some parts or violating basic drawing principles, like non superimposing objects) in a total of more than 1300 drawings (with repetitions), if counting all the mappings. This basically corroborates that, far from reinventing (i.e. converging), instead it generates novel, yet potentially pointless, results (i.e. diverging).

In general, we can say that, although the two concepts of a house and a boat are close to each other (both are physical structures, used by humans), this can be an example of computational modelling of divergent thought because a large amount of new instances was generated from the blending of two different concepts. According to this perspective, Divago could serve as a meta-level engine for helping another

system with extending its search space. This is particularly feasible in situations where the search space consists of a set of independent knowledge structures, such as in Case-Based Reasoning.

6.2 The Horse-Bird

The Horse-Bird experiment (presented in [Pereira and Cardoso, 2003a][2]) was the first to assess the behavior of the Factory and Constraints module. Actually, it consists of two different kinds of experiments, each with a distinct goal: assessment of the individual effects of each measure on the final results; qualitative evaluation and tuning of the model. After several preliminary GA parameters tuning tests, we decided for 100 individuals as the population size, 5% of asexual reproduction (copy of an individual to the following population), 80% of crossover (combination of pairs of individuals), 14% of mutation and 1% of random generation (to allow random jumps in the search space)[3]. We have three different stopping conditions: appearance of an individual with the maximum value (1); achieving n populations ($n = 500$); being stalled (no improvements in best value) for more than m populations ($m = 20$). We kept these GA configurations throughout the whole experimentation related to the Horse-Bird blending, as described here.

As a result of the problems relating to the interpretation of compounds in the house-boat experiment (e.g. *window|hatch*), we decided to drop this kind of projection, thus reducing the search space, now having from 2^l to $3^{2k} \times 2^{l-2k}$ different blends. Notice that the concept maps of horse and bird (already given in tables 5.1 and 5.2) have (m=) 29 and (n=) 33 different concepts, respectively, which gives an l=m+n=62. The three mappings used (already given in figure 5.4) have sizes k=6,

[2]In the experiments reported in this book, we applied the revised version of the constraints, therefore the results here may slightly differ from the ones given in the paper.

[3]These rather large values for mutation and randomness result from the fact that some mutations have a null effect in the projection scheme (e.g. if an element projection is mutated to nil, it won't have effects if its surrounding elements are already projected to nil). In order to stimulate diversity, these values seemed appropriate. Future experiments are expected to overcome these issues and apply more common values.

k=5 and k=21, thus the search space will have a minimum size of 2^{62} and a maximum size of $3^{42} \times 2^{20}$. Even if discounting the concepts and relations that are repeated in both concept maps (e.g *human*), this corresponds to a very large set of blends.

This experiment also introduces the criteria for defining novelty (the *nov* function, converse of Ritchie's *typ* function) which will be used throughout the rest of this chapter. This function is based on the comparison of the concept map of the blend with those of the inputs. The exact value of distance between an input x and the blend b corresponds to the sum of the relations that belong to b and that are missing in x with those that belong to x and are missing in b. Let us call it $d(b, x)$. This can be seen as an *edit distance* - the set of delete and insert operations needed to transform one into the other. Since this value becomes proportional to the sizes of the concept maps involved, we divide it by the size of the blend concept map (the number of relations), thus getting a normalization that allows us to compare among different experiments and assess the behavior of the system. Following one of the measures of comparison to an archetype ($novelty2(x)$) by [Pease et al., 2001], we define the function *distance* as returning the normalized minimum distance to one of the inputs:

$$distance(b) = \frac{min(d(b, x_1), d(b, x_2))}{size_b}$$

such that x_1 and x_2 are the input concepts for generating b and $size_b$ is the size of the concept map of b. The larger the distance to inputs, the higher is the novelty, therefore, the function *nov* is defined as:

$$nov(b) = \begin{cases} 1 & distance(b) > 1 \\ distance(b) & otherwise \end{cases}$$

Determining usefulness (*use*) presupposes the existence of a purpose. Therefore, we will only apply it when this is explicit in the experiment. In the second part of these experiments, we have the goal of finding a *pegasus*, and the distance to this point in space will give us an estimate to *use*. When analyzing the set of values for *nov* and *use* given in a sequence of runs, we tend to prefer the median since it is not sensitive to outliers, as happens with the mean, and it normally represents a specific

blend that is representative of the mean and that we can inspect. Whenever this assumption becomes unsafe, if there is a large difference between the median and the mean and a large standard deviation, we will also consider other indicators. In any case, the reader will find the values for median, mean, standard deviation and mode in the result files (found in appendix F). Therefore, unless stated otherwise, the statistics presented refer to the median of the results for 30 runs. The inspiring set (following section 2.1) will comprise the two creatures, horse and bird, as well as the pegasus, since the latter was also explicitly defined by us.

6.2.1 Evaluating the Optimality Pressures

This experiment serves to observe the effect of each pressure in the final results, bringing up a way to predict and control the system. For the first part of these experiments, we isolated each optimality pressure, by attributing zero weight to the remaining criteria. Since one of the optimality pressures is not independent (Web) and another (Intensification of V.R.) was not accounted for in our current implementation we have *only* six different criteria to take into account.

The input domains applied were the domains of *horse* and *bird* (in tables 5.1 and 5.2), meaning that the expected results range from the unchanged copy of one (or both) of the concepts to a horse-bird (or bird-horse) which is a combination of selected features from the input domains. The generic domain consists of the general ontology, integrity constraints and a set of frames (already given in table 5.3; see also Appendix F).

We applied the three mappings presented in figure 5.4. For each mapping, we tested the six optimality pressures, each of these comprising 30 runs[4]. The Elaboration module was not used. Each blend was examined by the Constraints module without being subject to any transformation after the projections.

We now present an analysis of the individual effect of each of the measures:

[4]A run is an entire evolutive cycle, from the initial population to the population in which the algorithm stops

- In **Integration**, frames behave as *attractor* points in the search space. Moreover, the frames with a larger coverage tend to be preferred, although when too large (like *aprojection* or *aframe*) they are dropped. The evolution is directed to a compromise of coverage and satisfaction. The complexity of the search space grows with mapping size (the number of cross-space associations found by the mapping algorithm). In fact, when we have a mapping of size 5, six different blends are returned, the best choice being retrieved 43% of the times, while with a mapping size of 21, eight different solutions are found, the best choice being retrieved a mere 6% of the time. This confirms the complexity and dimensions of the search space we discussed in section 5.5. A good compensation for this apparent loss of control is that the returned values are clearly higher (0.68, for the best) than in the small mappings (0.22), suggesting that, with larger mappings, the probability of finding a better solution is higher than in smaller ones. Finally, the novelty was 0.71, meaning that the set of frames used does not lead naturally to any of the inputs, i.e. the system diverges from its input concepts.

- **Pattern Completion** drives the blend to partially complete (i.e. satisfies some conditions but not all) the highest possible number of frames, leading, in each case, to several sets of relations which fit into those frames without satisfying them completely. This means that, isolated, Pattern Completion only leads to disperse, non-integrated results and so it is not very useful. Interestingly, it can be useful when combined with Integration because it gradually brings to the blend the concepts and relations that are needed to complete the frames and so it speeds up the process of finding frames with high Integration value. In respect to the *search landscape*, it seems to be very rich in local maxima. The most constant results came from mapping 2 (of Figure 5.4), with the best results obtained 13% of the time and the second best 20% of the time. An interesting remark is that the local maxima always fall within a very strict range of values (of maximum amplitude 0.11, in mapping 3). The median value for *nov* was 0.79, which confirms our expectancy that Pattern Completion would be close

to Integration, in these terms, since they use the same set of frames.

- From the experiments with **Topology**, we can observe that there is a tendency to bring all the relations from both concept maps to the blend, without being transformed. This means that, at the limit, the blend will comprise the union of the two concept maps from the inputs, thus (if both the inputs have the same size) the novelty will tend to be 0.50 (half the concept map of the blend would have to be deleted to become an exact copy of one of the inputs). This prediction is corroborated by the result (nov= 0.51).

- The influence of **Maximization of Vital Relations** in the results is straightforward, given that its highest value (1) reflects the presence, in the blend, of all the vital relations that exist in the inputs. As the evolution goes on in each run, the value grows until reaching the maximum reasonably early. For each set of the 30 runs, it reached the value 1 a minimum of 93% of the times, and the remaining 7% achieved at least a value of 0.95. As in Topology, the search space of Maximization of Vital Relations is very *simple* since there is a global maximum in the neighborhood of (almost) every point. However, in contrast to Topology, this measure results in very high novelty (0.99), which can be explained by the fact that the number of vital relations in the concept maps is relatively small and that there is no constraint on the arguments of these relations. In other words, it does not matter what the vital relations actually associate with, only that their simple presence in the blend is important in order to get the maximum value in this measure, yielding an apparently random choice of elements projected.

- The results of the **Unpacking** measure show that it drives towards similar results as Topology, with the main difference being that the relations in the blend are clusters of copies of subgraphs from the inputs. I.e., Unpacking only copies those relations that do not imply conflicts (e.g. some concepts that belong to both domains, such as *leg*, can become problematic because its Unpacking is unambiguous). It is therefore a force of inertia. The median value for nov was

0.63, testifying that, whatever was present in the concept map of the blend, it was similar to one of the inputs yet missing some parts that would make it an exact copy, which intuitively agrees with its definition.

- The first part of the test on **Relevance** focussed on making a single relation query. In this case, we asked for "something that flies" ($ability(\_, fly)$). The results were straightforward in any mapping, accomplishing the maximum value (1) in 100% of the runs, although the resulting concept maps did not reveal necessarily any overall constant structure or unity, giving an idea of randomness in the choice of relations other than $ability(\_, fly)$. In other words, the evolution took only two steps: when no individual has a relation $ability(\_, fly)$, therefore with value 0; when a relation $ability(\_, fly)$ is found, yielding a value 1, independently of the rest of the concept map. The second part of the test on Relevance, by adding a frame (`ability_explanation`) to the query, revealed similar conclusions. There was no sufficient knowledge in any of the input domains to satisfy this new frame completely, so the algorithm searched for the maximum satisfaction and reached it 100% of the time in every mapping. So the *landscape* seems to have one single global and no local maxima, reflecting the integration of the two parts of the query. The existence of local maxima would be expected if there were separate frames. Intuitively, the *search landscapes* of Integration and Relevance seem to be similar. As with Integration, the novelty is dependent on the available frames, more specifically on the frames used in the query. With the ones used, the value for *nov* was 1. This is consistent with the observations just made of the apparent randomness of the choice of relations for complementing the concept map.

6.2.2 Finding the pegasus

For our concerns, we define a pegasus as being a "flying horse with wings", so leaving out other features it may have (such as being white). These extra features could also be considered but would need knowledge about any aspects of ancient Greece,

Greek mythology and some ontological associations (e.g. purity is white). Moreover, they would make the generation of the blend considerably more complex, although possibly more interesting. Formally, the pegasus we want to generate has the same concept map as the horse domain augmented with 2 wings and the ability to fly (so, it should also have the relations $ability(horse, fly)$, $motion\_process(horse, fly)$, $pw(wing, horse)$, $quantity(wing, 2)$ and $purpose(wing, fly)$).

For validation purposes, we started by submitting a query with all the relations of the pegasus, so as to check if they could be found in the search space, and the results reveal that only the mapping 3 (see figure 5.4) respects such constraints. This led us to use this mapping exclusively throughout the rest of the experiment. Knowing that the solution exists in the search space, our goal was to find the minimal necessary requirements (the weights, the frames and the query) in order to retrieve it. From a first set of runs, in which the system considers a large set of different frames and no query, we quickly understood that it is not simple (or even feasible) to build the pegasus solely by handling the weights. This happens because the optimality pressures provide control regarding to structural evaluation and general consistency, but only by pure chance can we find the exact weights to match the same relations of the pegasus, a very specific blend that fails to follow all but a few of constraints, but a combination of them. This drives us to the need of queries.

A query may range from specific conditions that we demand the blend to respect (e.g. the set of conditions for flying, enumerated above) to highly abstract frames that reflect our preferences in the blend construction (e.g. the frame `aprojection`: elements from input concept map 1 should all be projected). Intuitively, the best options seem to comprise a combination of the different levels of abstraction.

Since a query is only considered in the Relevance measure, its weight must be large if we intend to give it priority. In fact, using only Relevance is sufficient to bring the concept map of the solution to the blend, when the query is specific enough, as we could test by using a query with `aprojection` and the flying conditions. From a creativity point of view, it is not expected to have very specific queries and we

are more interested in less constrained search directives. In table 6.5, we show the parameters we used, as well as the *nov* and *use* values obtained. *use* is calculated as:

$$use(b) = 1 - \frac{d(b, target)}{size_b}$$

with *target* being the concept map of an optimal blend (in this case, the *Pegasus*). The weights we present correspond to Integrity (I), Pattern Completion (PC), Topology (T), Maximization of Vital Relations (MVR), Unpacking (U) and Relevance (R). The `fly conds.` are the relations that the blend must have in order to be a flying creature, and `aframe`, `aprojection` and `new_ability` are frames as described before (and detailed in appendix F). The values presented correspond to the median in each set of results.

Exp.	Weights						Query	*nov*	*use*	Best blend
#	I	PC	T	MVR	U	R				(*nov*/ *use*)
1	0	0	0	0	0	100	fly conds. + aprojection	0.59	0.53	0.40/0.74
2	0	0	0	0	0	100	fly conds. + aframe	0.86	0.26	0.0/[a]
3	0	0	0	0	0	100	fly conds.+ aprojection + aframe	0.59	0.53	0.40/0.71
4	50	0	0	0	0	50	fly conds.+ aprojection + aframe	0.51	0.62	0.19/0.97
5	33.3	33.3	0	0	0	33.3	fly conds.+ aprojection + aframe	0.78	0.34	0.82/0.32
6	33.3	0	33.3	0	0	33.3	fly conds.+ aprojection + aframe	0.60	0.52	0.49/0.66
7	25	0	25	25	0	25	fly conds.+ aprojection + aframe	0.70	0.28	0.45/0.58
8	20	0	20	20	20	20	fly conds.+ aprojection + aframe	0.62	0.33	0.47/0.51
9	34	0	16	10	4	36	fly conds.+ aprojection + aframe	0.43	0.70	0.44/0.95
10	34	0	16	10	4	36	new_ability+aprojection+aframe	0.26	0.71	0.35/0.73
11	20	0	0	0	0	80	fly conds. +aprojection+aframe	0.16	0.76	0.21/0.90
12	20	0	0	0	0	80	new_ability+aprojection+aframe	0.58	0.47	0.18/0.92

[a]Note: In configuration 2, there is more than one highly scored blend, none with *use* higher than 0.59.

Table 6.5: The 10 different configurations used.

An observation that must be made is that the target is very similar to one of the inputs (the "horse"), its novelty being exactly of 0.26, a very low value that was only acknowledged after the first experiments. Since making it less typical would imply artificial changes in the concept map (actually the Pegasus *is* a horse with wings), we decided to leave it untouched. Furthermore, it is theoretically possible to generate a blend that is close to the pegasus, yet far away from the horse (if it falls in the opposite direction of similarity). As we can see from the experiments, there are useful results that nevertheless fail the threshold of novelty and there is no linear relationship between *nov* and *use*, although when *use* gets high scores, the opposite

happens with *nov*.

The first eight configurations were dedicated to understanding the effect of gradually adding optimality pressures to the fitness function. In the first three, where only Relevance was used, we verified that, although it was *easy* to have all the concepts and relations we expect for a pegasus, often it was complemented by an apparently random selection of other relations. This results from having no weight on Integration, which we added on the configuration 4, yielding the result that was closest to our pegasus: the projection of the entire horse domain, and the selective projection of *wings* and the *fly ability* from the bird domain. There were a few extra bits of knowledge, such as having two claws, feathers or chirping. The majority of the time, the extra knowledge results in blends that are distant to the inputs and to the pegasus, i.e. the pegasus found was more a singularity than the average situation. The straight explanation is that the weight of Integration leads Divago to satisfy frames that compete with the pegasus (e.g. `bframe`, which would project the bird's concept map structure) in many different ways. In configuration 5, the influence of Pattern Completion led the results to minimum incompleteness (e.g. a pegasus with everything except a mane, wings or any other item), which revealed that, by itself, it is not a significant or even positive contribution to the present goal, a reason for dropping its participation in the subsequent configurations.

Adding Topology (conf. 6) essentially brought two different kinds of results. As with configuration 4, it returned the "correct" pegasus with extra features like having *feathers* or a *beak*, each of which was apparently selected at random. These were also given the highest scores in the fitness function. However, in some of the runs (10%), the results contained both creatures (horse and bird) in the same concept map, as if they were connected (e.g. having the same legs or ears). This is a rather unwanted result, and it suggests that the weight of Topology should be relatively small in comparison to others.

The following configuration, the inclusion of Maximization of Vital Relations, confirmed the same conclusions as from Topology, but with more control over the

kind of extra relations transferred to the blend. For example, the blend may have 2 wings (from the relation *quantity*), a beak and feathers (from *pw*), but it is never an oviparous (from *member_of*). On the other hand, we can sense a gradual lack of focus on the overall results (no two runs returned the exact same result) complicating considerably our goal of controlling the system. There is a simple explanation for this: Relevance, Integration, Topology and Maximization of V.R. all have the same weight and some (like Maximization) are more easily satisfied, thus driving the evolution towards their maxima, from wherever the evolution started. This same phenomenon happened in configuration 8, although Unpacking had brought a more stable set of results.

An immediate conclusion we took from these experiments was that each pressure should have a different weight, correspondent to the degree of influence it should have in the result. In our case, we are seeking for a specific object (the pegasus), we know what it is like, what it should not have and some features not covered by the query conditions that we would like it to have. This led us to a series of tests for obtaining a satisfiable set of weights, used in the configurations 9 and 10. Given the huge dimension of the problem of finding these weights, they were obtained from a generate-and-test process, driven by our intuition, so there is no detailed explanation for the exact choice of why these values and not others. Yet, a qualitative analysis can be made and we see a clear strength given to Relevance and Integration. The former serves to "satisfy what we asked" and the latter guarantees overall coherence (centered on the query frames) and consistency (e.g. it prevents the solution from having 2 and 4 legs simultaneously). There is also a more discrete presence of Topology, Maximization and Unpacking, to allow the transfer of extra knowledge. Configuration 9 revealed, possibly, the "richest" pegasus found, in the sense that, although largely failing the target, it contains all of its relations as well as a selection of other relations (having $lungs$, $feathers$, a pair of $claws$). Still, although this result appeared consistently throughout some runs, there is a high variability of results (for configuration 9, the mean of *nov* was 0.64 with standard deviation $\sigma = 0.17$; for conf. 10, the mean for *nov* was 0.57, $\sigma = 0.17$) testifying the difficulties in controlling the

system.

Finally, from what we had learned in the first configurations, we decided to reduce to only two constraints (Relevance - 80% and Integration - 20%), predicting that we would find the best approximation to the target. Indeed, this was confirmed by the results, particularly for configuration 10. Although still not avoiding very bad outliers in two runs which seriously affected some indicators (it yielded $mean_{nov} = 0.24$, $\sigma_{nov} = 0.08$ and $mean_{use} = 0.70$, $\sigma_{use} = 0.30$), the results were very stable (if removing the 6% outliers, we get $mean_{nov} = 0.16$, $\sigma_{nov} = 0.06$ and $mean_{use} = 0.77$, $\sigma_{use} = 0.10$). We will use this specific configuration later on when we know with some confidence the kind of output to get (and we have the necessary frames).

Now, for a more detailed analysis, we will calculate the values for Ritchie's features (or criteria). As we have referred to before, we assume a mapping between the pairs *novelty*/*usefulness* and *typicality*/*value*, such that novelty is the opposite of typicality (typ=1−nov) and usefulness equals value (val=use). The latter may become controversial, yet it may be the best method for applying Ritchie's features in this context and, above all, it is our conviction that, for a formal setting such as the one we are describing, one can only measure the value of something as much as it accomplishes a goal or satisfies a set of conditions. In other words, it must be a solution to a problem, i.e. be useful. According to this philosophy, we obtain the values in table 6.6. We remember that we have assumed the value 0.5 for the parameters α, β and γ

From the first four features, we can say that Divago is producing typicality and value near the half scale (features 1 and 3), but clearly produces more valued than typical items (2 and 4). From criterion 5, we can see that all typical results were valued, which is clearly due to our target falling within the range of typicality. However (from 6 and 7) there are some valued but non-typical outcomes, which is a good indication of creativity. The proportion of valued non-typical outcomes with regard to the typical ones (criterion 8) can be misleading since we are comparing results with different configurations - the last 4 configurations clearly yield more typical, valued

Criterion	Value
1	0.443
2	0.273
3	0.504
4	0.636
5	1.000
6	0.364
7	0.500
8	1.333
9	0.000
10	N/A
11	0.406
12	0.483
13	0.273
14	0.636

Table 6.6: Ritchie's [Ritchie, 2001] features results.

results than the others. Reinventions only occurred in occasional runs, which do not fall into the statistical validity, therefore we cannot say that Divago has consistently produced reinventions, as can be seen by the features 9 and 10. This fact is important in understanding why features 11 to 14 are reproductions of the first four. Indeed, there is such a correlation between these eight features that we suspect they can be reduced to a smaller set.

For determining the fine-tuning of the system according to [Colton et al., 2001] (see Appendix A), we can *at most* determine an estimate, given the participation of so many variables in the definition of each result. From the experiments so far, we can say, for example, that Relevance and Integration are creatively more useful than Pattern Completion, and that `aframe` is less creatively useful than `aprojection` (compare configurations 1 and 2). However, we can also see the high complexity involved. Compare, for example, configurations 11 and 12. The former normally produced better results than the latter, but the latter has a best blend with higher scores. It seems, therefore, that configuration 11 earned more stability at the cost of losing better singularities to the configuration 12. To check the individual influence of the frames in configuration 11, we applied the same weight configuration of 80%

Query	*nov*	*use*	Best Blend *nov/use*
empty list	1.00	0.00	1.00/0.00
fly conds.	1.00	0.00	1.00/0.00
`aprojection`	0.36	0.56	0.19/0.74
`aprojection`, fly conds.	0.35	0.81	0.21/0.95
`aframe`	0.00	0.74	0.00/0.74
`aframe`, `aprojection`	0.15	0.67	0.00/0.81
`aframe`, `aprojection`, fly conds.	0.16	0.76	0.21/0.90

Table 6.7: Checking the fine-tuning of Divago.

Relevance and 20% Integration to all possible combinations of the query (see table 6.7). Notice that the first two queries (void query and fly conds.) completely fail to achieve anything useful. This results from not having applied an organizing frame (such as `aframe` or `aprojection`). In this case, Divago was directed towards satisfying anything (empty list) or just a small set of relations (fly conds.), preventing it from using an organizing frame and therefore making a coherent whole. From this, we can conclude that `aframe` and `aprojection` are important to *organize* the blend (in this case, towards the same organization of the horse concept map).

We can also notice that, when using only `aframe` and `aprojection` (`aframe` alone or combined with `aprojection`), Divago produces either an exact copy of "horse" or a very similar result. However, when put together with the flying conditions, it makes a whole that can lead to (very nearly) the pegasus. This may indicate a high fine-tuning towards the pegasus, however we can also see that the same combination can lead to other results depending on the weights applied (e.g. configuration 5) and other combinations (e.g. configuration 12) can lead to the same results with the same weight configurations. To conclude, if, on the one hand, the frames are a method for controlling/tuning the system, it is also true that their application does not guarantee valuable results and that, in this sense, fine-tuning the system is an extremely difficult task. This results from the complexity of the space and from the specificities of frame combinations (while may be compatible, some others may be competing).

It is clear that the results in this section were driven subjectively by us in the

choice of the concepts and frame design, but the argument we are trying to make is that we can lead Divago to produce novel and useful outputs. Nevertheless, it is a difficult system to control, a good aspect on one side - it is hard to be intentionally biased to specific outputs -, but bad on the other side - it is extremely difficult to test its full potential.

We also developed (in collaboration with Pablo Gervás [Pereira and Gervás, 2003]) an Interpreter for generating textual descriptions of the blends, based on Natural Language Generation techniques. This system made descriptions by comparison with the input concepts of "horse" and "bird". Examples of automatically generated descriptions of blends are:

> (1) A horsebird is a horse. A horsebird has two wings and feathers. It can fly, and it moves by flying.
>
> (2) A horsebird is a horse. A horsebird can fly, it has feathers, a beak, and wings for flying and it moves by flying.
>
> (3) A horsebird is a horse. A horsebird can fly. It chirps, it has wings for flying and it moves by flying.

The example (1) corresponds to a result from configuration 4. Examples (2) and (3) are interpretations from configuration 9.

6.3 Noun-noun combinations

In the experiments with noun-noun combinations (published in [Pereira, 2003]), we show the behavior of Divago with a dataset constructed independently by other researchers ([Costello, 1997]) and make a comparison to C^3 (see section 3.3.1). Each concept (associated to a noun) is represented with a syntax that is equivalent to the one adopted for Divago. Here, we apply for the first time the two-step methodology that will be followed in the subsequent experiments, which starts by "tuning"

the system with preferred outcomes and then allowing it to do free generation, constrained by a given query. Also for the first time, we define a more precise criterion for usefulness (*use*), which will correspond to the score obtained for the Relevance principle. The rationale is that a blend is useful (i.e. valued) if it accomplishes a set of pragmatic conditions that may be specific to a situation (e.g. it should be a clay object that serves to cut food) or a generic demand for an application (e.g. it should be an object with a single color and a single shape), which can be given as a query to the system. Thus, *use* will now correspond to the value of Relevance. This may seem contradictory with the choice for *use* in the previous experiment, but the only difference is that now *use* will participate in the search, which goes in agreement with the model of creativity presented, in which the invention of new concepts should be purpose-driven. The value for *nov* will be given exactly as before.

The dataset used in these experiments comprises 179 concepts (noun conceptual descriptions) borrowed from Fintan Costello's PhD thesis [Costello, 1997] on noun-noun conceptual combination. In this thesis (and in subsequent publications e.g. [Costello and Keane, 2000]), the author describes each concept by a set of attribute-value pairs, as shown below (for "necklace")

Necklace

name:	(necklace)
feature-set:	(solid inanimate static)
color:	(silver gold)
shape:	(small circular)
structure:	
made_of:	metal
parts:	(pendant)
found:	
function:	((wears person3 necklace neck)
	(decorates necklace person3))

The conversion to our concept maps is straightforward: each of the "features" becomes a property relation; the attributes *color*, *shape* and *made_of* become relations with the respective name; each of the *parts* is converted into a *pw* (part whole) relation; each "function" is converted into a set of *actor* and *actee* relations (with third arguments, such as *place* or *instrument*). The *actor* is expected to be the first argument of the function, while *actee* is the second. Therefore, our concept map representation for "necklace" is as follows:

property(necklace, solid)	made_of(necklace, metal)
property(necklace,inanimate)	pw(pendant, necklace)
property(necklace,static)	actor(wears, person3)
color(necklace,silver)	actee(wears, necklace)
color(necklace,gold)	place(wears, neck)
shape(necklace,small)	actor(decorates, necklace)
shape(necklace,circular)	actee(decorates, person3)

In the original dataset, there are interrelationships between nouns. For example, there is also a representation for *pendant*, *person*3 and *neck*, so, along with *necklace*, these nouns can be seen as a small graph representing the knowledge about people and necklaces. Within this small graph, there is normally no repetition of function specifications (e.g. in *neck* or *person*3 representation, there is no *wears* function, although it exists implicitly). For this experiment, we directly and separately converted each noun to a concept map, and there is no communication between our concept maps, which means that many nouns in our knowledge base lose their original implicit data. This was necessary since automatically converting that implicit network into our concept maps would not be a trivial exercise in terms of programming and would clearly fall away from the goals of this project. Another aspect of the dataset

is that some concepts have several different instantiations (e.g. *person*3 is the third representation of the noun *person*). We also converted these directly and separately to our knowledge base, without merging them.

The main goal of these experiments was to observe how Divago behaves with respect to criteria of novelty and usefulness when applied to knowledge from another concept combination system. Another intention was to improve the control over Divago with regard to these measures.

The noun-noun interpretations we consider in the experiments are either *hybrid interpretations* or *property interpretations* (see section 3.3.1). In some tests we made prior to these ones, the Mapper (which is based on structure alignment) was clearly unable to allow other types of interpretations such as relational and known-concept interpretations. This may point in the same direction as Costello and Keane [Keane and Costello, 2001], who argue that conceptual combination cannot be reduced to structure alignment. In our case, the strictness of the structure alignment methodology affects Divago's needed flexibility: 1-to-many mappings should sometimes be considered (rather than 1-to-1); representation of inputs sometimes with variable granularity should be possible (e.g. "fido is a canine, who is a mammal" mapped to "tweety is a bird" should yield $fido \leftrightarrow tweety$ and $mammal \leftrightarrow bird$, rather than $canine \leftrightarrow bird$ or $canine \leftrightarrow tweety$). Still, we cannot argue that these limitations are more than computational and/or representational limitation, so further exploration regarding structure alignment and computation should be taken before claiming it as too rigid.

In order to provide a pragmatic background for the experiments, we invite the reader to consider a situation where one wants to obtain combinations with a specific set of characteristics, we can thus define this set via *scripts* with the same syntax of the nouns described above. A *useful* concept must have specific values for the slots of the script and respect a set of integrity constraints. The slots and values required can thus be grouped together in a query. In all experiments (except in the tuning set), this query consisted of:

property(A,[animate, inanimate]),
property(A,[liquid, solid]),
property(A,[static, mobile]),
made_of(A,_),
shape(A,_), color(A,_),
actor(F,_), actee(F,_)

Square brackets mean disjunction (e.g. the concept A must be *animate* or *inanimate*). The presence of *actor* and *actee* relations means that the concept should have a function.

Beforehand, we could not know exactly which kinds of frames were needed to build "good" combinations, leading to the need of a tuning phase that helped us find a set of appropriate frames. Only after this tuning, are we able to test the system, leaving it to construct its own concepts.

6.3.1 Tuning

The tuning set we used consisted in 30 pairs of randomly selected concepts from the list. For each one, we constructed a solution (called the *tuning target*) correspondent to our own interpretation of the noun-noun combination. This hybrid interpretation considered exclusively the knowledge contained within the selected noun representations and was centered on the head noun, which means that, in any pair A-B of nouns, the interpretation was that "an A-B is a B with such and such A characteristics". In other words, the concept B, the *head*, is always the *focal concept* in our interpretations.

Each experiment consisted of making 30 runs for each pair (each run with the exact same starting conditions), having in the query the set of frames that could be expected to achieve the target. The weight configuration followed was 90% for

Relevance and 10% for Integration, which reflects our intention to test the frames. When the results were mostly missing the target, we either selected other frames or designed new ones and made the 30 runs again. More specifically, this happened when there was an error of more than 2 relations to the target or when this error was due to fundamental relations (i.e. without them, the result would not be novel or valued). In table 6.8, we show a sample with the tuning combinations, target descriptions, resulting difference to the target, novelty score and frames used in the query. It is important to remember that the target interpretations are obtained using only the existing knowledge representation of both nouns, which justifies the appearance of awkward interpretations (e.g. "head hammer_handle", "pen person"). We can also see that the frames were initially tailored to fit the target interpretations and reused later when effective (e.g. the "shape_transfer" was created for "bullet potato", and used often in the succeeding experiments).

The table 6.9 presents the frames that were obtained (or selected from the already existing ones in the generic domain). For the rest of the experiment, this became the set of available frames.

It is clear though that both the target interpretations and the frames were made by us, so introducing a subjectiveness component in these experiments. Since there does not seem to be any simple automatic frame generation mechanism and given that the language itself demands some expertise, the frames had to be constructed with the method described. On the other hand, it would be possible to use other people's interpretations of the randomly generated pairs, requiring a reasonably large set of participants with some expertise to understand the constraints (interpretations are confined to the specific representation). This was done for the next two experiments (established blends and game creatures). Not having done so for this experiment, we tried to follow our intuition and imagination in each case. At worst, the experiments reflect our specific ways of noun combination on the tuning set applied to the free generation set.

The mappings used in all the experiments were automatically generated by our

Combination	**Tuning Target**	**Error**	**Frames**	***nov***
bullet potato	small and cylindrical potato	2	bcore. shape_transfer. slot_set_completion	0.08
cow vehicle_body	black and white vehicle that eats grass	0	bcore, function_transfer, slot_set_completion	0.44
eagle shirt	brown, bird-shaped shirt	1	bcore, structure_transfer, shape_transfer	0.46
engine ball	self-mobile ball	0	bcore, feature_set_contrast	0.29
flower_bloom plant	spherical plant	0	bframe, shape_transfer	0.12
fruit1 paper1	paper with fruit-seeds that humans eat	0	bcore, structure_transfer, function_transfer	0.40
head hammer_handle1	mobile and animate (living) hammer handle	2	bframe, feature_set_contrast	1.00
neck instrument	a small and straight instrument	0	bcore, shape_transfer	0.25
necklace paper	circular paper that people use in the neck for decoration	1	bcore, function_transfer	0.45
patient paper1	paper that has illness	0	bcore, function_transfer	0.14
pen person	thin, long person, that is used (by others) to write on paper	2	bcore, function_transfer	0.67
pencil pendant	thin, long pendant, used to write on paper	0	bcore, shape_transfer, function_transfer	0.46
potato acorn	brown, spherical acorn	1	bframe, function_trasnfer	0.30
potato herring tail	spherical herring tail	2	bcore, shape_transfer	0.12
pottery spoon	spoon made of clay	1	analogy_transfer	0.56
skin stem	thin stem	0	bcore, shape_transfer	0.14
spoon1 frame	brown frame	1	bframe, single_ differentiating_feature	0.00
spoon1_handle lens	straight and long lens	0	bcore, shape_transfer	0.25
thorns hammer1	small and sharp hammer	0	bcore, shape_transfer	0.13
tool boxcar	a boxcar used to make other objects	2	bcore, function_transfer	0.08
torso pencil1	small, animate and mobile (living) pencil	0	bcore, feature_set_contrast	0.36
utensil web	a metal web, used to make food	0	feature_set_contrast, function_transfer	0.40
vegetable person3	static, inanimate person	0	feature_set_contrast	0.37
vegetable spoon	a thing with spoon shape that grows on earth	1	bcore, function_transfer	0.50
vehicle_body vessel1	vessel made of metal	1	bcore, slot_set_completion	0.14
vessel1 food	concave shaped food in which one can put something	0	bcore, function_transfer, shape_transfer	0.37
victim projectionist	projectionist that was damaged by a gun	0	bcore, function_transfer	0.25
wheel sitting_room	circular sitting room	0	bcore, shape_transfer	0.25

Table 6.8: Excerpt of the tuning set (average distance to target (average error)=0.67, standard deviation=0.994)

Frame	**Description**
bframe	The blend has the same relations of head noun (although the arguments may differ)
bcore	The blend has the same relations and arguments (except those related to function) of head noun
analogy_transfer	Transfer all neighbor elements and relations of an element of modifier to the mapping correspondent of head
function_substitution	A function from head is substituted by a function of modifier
single_differentiating_feature	Head and modifier differ only on one feature, which is transferred to head
function_transfer	The head gains a function that was part of the modifier
shape_transfer	The head gains the shape of the modifier
structure_transfer	The head gains the structure of the modifier
slot_set_completion	The slots in head that did not have a value are filled with modifier's corresponding values
feature_set_contrast	The feature-set in the head are replaced by the feature-set of the modifier

Table 6.9: The frames used in the experiments

structure alignment algorithm, with the *seed* dormant bridge connecting the individual identifier symbol of the nouns (for example, in "necklace paper", the seed dormant bridge is *necklace* and *paper*, which then goes to the $made\_of$ relations, establishing a mapping between *metal* and *paper* and so on). It typically established mappings between elements with the same role in both nouns (*color* value with *color* value, $made\_of$ value with $made\_of$ value, etc.)

6.3.2 Free generation

The free generation of noun-noun combinations consisted of selecting randomly a set of 33 pairs of concepts (the free generation set, which is completely distinct from the tuning set) and using the above described query to generate new blended concepts. Every frame shown in table 6.9 was available to the system so that it could find itself the selection of frames that suited the highest scores of the fitness function. The optimality constraint weights were chosen from what we had learned from the previous experiments. In this case, we wanted to give a central role to the frames (thus giving high value to Relevance and to Integration), while also allowing a little control to Topology, Maximization V.R.[2] and Unpacking. The latter received a higher weight to reinforce inheritance of the input concept's main characteristics. The values were: Relevance, 45%; Integration, 30%; Topology, 5%; Maximization of V.R., 5%; Unpacking, 15%. We also added an integrity constraint of having at least two frames being accomplished so as to stimulate knowledge transfer. Apart from these, parameters were equal to those used for tuning.

In figure 6.7, we show examples of the generation of the "fish_tail1 desk" and "fish spider" blends, with the inputs ("fish_tail1", "desk", "fish" and "spider") and the frames that were applied.

In table 6.10, we show the results achieved. For each pair of concepts, we show the best result (in terms of the fitness function) of the 30 runs and describe it textually

[2]The vital relations chosen were *isa*, *pw*, *purpose* and *quantity*

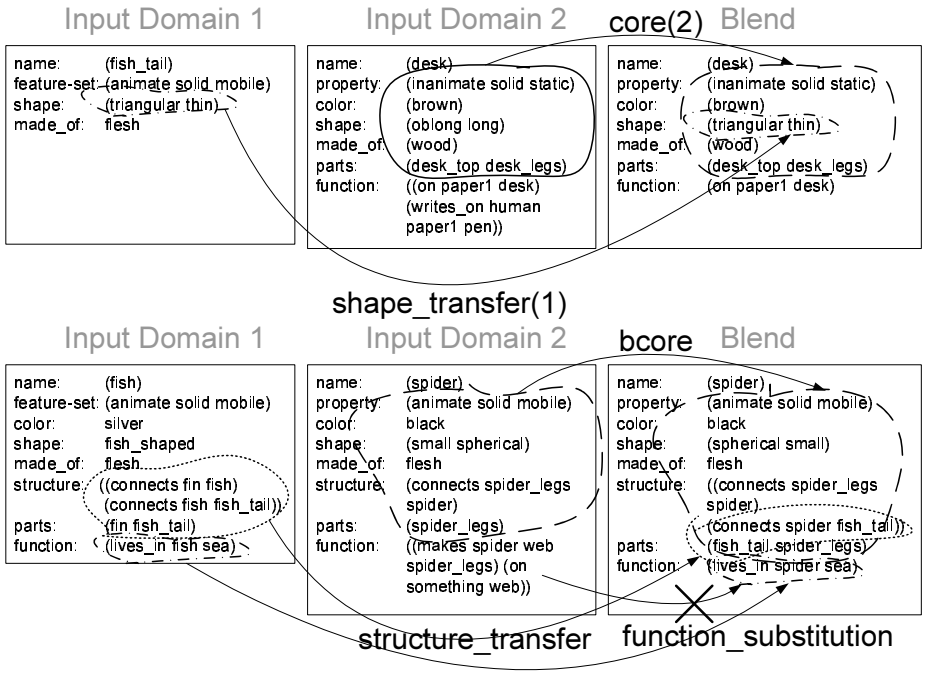

Figure 6.7: Frames used in the construction of "fish_tail1 desk" and "fish_spider"

by enhancing the differences to the head. The *use* score corresponds exactly to the resulting Relevance value. Therefore, a 100% means that every condition of the query was satisfied and no integrity constraints were violated. Other values indicate that either some condition was not satisfied or that integrity constraints were violated (or both).

For example, in Figure 6.7, we can observe that both blends satisfy all requirements of the query (therefore scoring 100%). If, say, there were no values for *made_of* and *color*, then *use* would be 75% since two (in eight) conditions were not satisfied. Another situation could be an integrity constraint violation (e.g. "Something cannot be *black* and made of *flesh* at the same time"), which would lead to a penalty (e.g. supposing *integrity constraint violation penalty* was 20%, "fish spider" usefulness value would be 80%). The frames listed correspond to the frames found in the construction of the best result for each combination.

We notice that every experiment ended satisfying a `bcore` frame. This is not surprising considering the query we used, which comprises a set of relations that coincides almost with the `bcore` frame relations. Still with regard to

Combination	**Interpretation**	***use***	***nov***	**Frames**
barrel spoon	Spoon	1.00	0.09	bcore
bed pips	Oblong pips	0.53	0.25	bcore, shape_transfer
bird1 sea	A bird shaped sea, with wings, and head and made of flesh	1.00	0.67	bcore, shape_transfer structure_transfer, slot_set_completion
bird_head clothes	Curve shaped clothes	0.81	0.12	bcore, bframe, slot_set_completion, shape_transfer
cow_head torso	Conical torso	0.60	0.14	bcore, shape_transfer
desk ornament	Brown, wooden, ornament one can put paper on	1.00	1.00	bcore, function_transfer slot_set_completion
desk1 spoon_bowl	Spoon bowl besides which one puts a chair (and is not used to put food in)	1.00	0.31	bcore, function_substitution
engine apple_tree	Oblong, long and large apple tree	0.43	1.00	bcore, shape_transfer
fish spider	Spider with fish tail that lives in sea, but does not make webs	1.00	0.69	bcore, function_substitution, structure_transfer
fish_tail1 desk	Thin, triangular desk	1.00	1.00	bcore, shape_transfer
flower_bloom hammer	A spherical hammer	1.00	0.13	bcore, shape_transfer
food body_part	A body part that serves to be eaten	0.35	0.33	bcore, function_transfer
herring instrument	A silver, fish-shaped (with fin and tail) instrument that lives on sea and is not used to play music	1.00	0.87	bcore, shape_transfer, function_substitution, structure_transfer
horse_head insect	Insect	0.04	0.00	bcore
insect rodent	A small rodent	1.00	0.12	bcore, shape_transfer
mattress knife	A long knife that is on a frame	0.00	1.00	bcore, shape_transfer, function_substitution
oak horse	A horse that grows on earth, it has a trunk and a crown, but keeps its horse shape	1.00	0.58	bcore, structure_transfer, function_transfer
paper1 chair_seat	White chair seat	0.50	0.17	bcore, slot_set_completion
patient fruit	Human shaped, skin-colored fruit that is ill	1.00	0.60	bcore, shape_transfer, function_substitution
patient plant	Human shaped, skin-colored plant	1.00	0.28	bcore, bframe, shape_transfer, slot_set_completion
person5 paper	Paper that sleeps in bed	1.00	0.14	bcore, function_substitution
person5 stem	Stem that sleeps in bed	1.00	0.50	bcore, function_substitution
potters_wheel desk	A flat and circular desk	1.00	0.38	bcore, shape_transfer
pottery neck	A neck made by a human	1.00	0.37	bcore, function_transfer
rose_bloom desk	Desk	1.00	0.42	bcore
sole bird	A black bird	1.00	1.00	bcore, slot_set_completion
spider_legs carriage	Carriage	0.67	1.00	bcore, bframe
stem vehicle	A straight, green vehicle	1.00	0.18	bcore, shape_transfer, slot_set_completion
train building	A building with the shape and structure of a train, and which serves to transport people	1.00	0.54	bcore, function_transfer, structure_transfer, shape_transfer
utensil pottery	Pottery	0.04	0.50	bcore
victim potters_wheel	Potters wheel	0.05	0.00	bcore
wheel machine	Black and circular machine	1.00	0.22	bcore, shape_transfer, slot_set_completion

Table 6.10: Results (average usefulness=78%; standard deviation=35%; median=100%)

frames, we can also see that the results used essentially 6 different frames (`bcore`, `slot_set_completion`, `shape_transfer`, `structure_transfer`, `function_transfer` and `function_substitution`). A possible explanation may be that the other 4 were either too specific (`single_modifying_feature`) or too generic (`bframe`) to achieve stability in the runs.

Results show that there is no correlation between novelty and usefulness, which seems intuitively plausible. Yet, *use* may contrast with our intuition in some examples (e.g. there is no apparent reason why a "horse_head insect" is less useful than a "rodent insect") and its explanation is simply that, for the context we are dealing with, the new object may lack some fundamental conditions.

Probably because the query is too much centered on the "core" of the object (every aspect except its function), it may lose its function during the blend generation, even when it is vital. For example, in blending "herring" and "instrument", the result *says* it is an instrument, but it lost its musical function, so leading to an empty concept. We also point out to the blends "train building1" and "bird1 sea". Both reveal inconsistencies ("a train building1 is a building that serves to transport people" and "a bird1 sea is a sea with wings...it is made of flesh"). These inconsistencies may be revealed as creative if explored from a metaphoric perspective, a very complex computational challenge although sometimes trivial for humans. Preventing the existence of these extreme examples depends on adding integrity constraints (e.g. "something that serves for transportation cannot be made of bricks") but these will go against the creative potential of the system.

From the observation of the *use* scores, it should be clear that the average of 78% obtained is highly dependent on the specific query and on the specific knowledge contained in the dataset. If the query was less constrained (e.g. having just half of the conditions), the score would certainly be higher, whereas if we added conditions that could not be satisfied within the dataset, *use* would never achieve 100%. What these numbers show is that the model is able to search for the query satisfaction when it is (the knowledge base, the query and the factory) properly configured, thus providing

Criterion	Value
1	0.543
2	0.563
3	0.782
4	0.781
5	0.778
6	0.344
7	0.786
8	0.786
9	0.036
10	16.000
11	0.513
12	0.831
13	0.500
14	0.781

Table 6.11: Ritchie's [Ritchie, 2001] features results.

useful outcomes for the context in question.

Calculating the features from Ritchie, we obtain the results in table 6.11

In comparison with the previous experiments (of horse-bird) shown in table 6.6, we notice an increase of scores for typicality and value (1-4), with or without considering the inspiring set (11-14). Apart from the inherent differences of both experiments, this also reflects a higher control over Divago, due to the methodology followed (tuning+free generation) and to acquired experience in weight choice. This experiment also shows a more realistic setting. For example, not all typical items are valued (criterion 5) and a few reinventions were made (9), although these were a very little proportion of the results (10). Interestingly, Divago produced almost exactly the same proportion of untypical and valued items as in the previous experiments, with regard to the whole set of outputs (6). It even increased, if only considering the untypical items (7).

6.3.3 Comparison to $\mathcal{C}^3$

We had access to a set of analogous experiments that Costello and Keane did with C^3. In these experiments, the authors randomly generated 10 pairs of nouns (e.g. "eagle" and "tulip") and, for each pair, generated interpretations for the two possible combinations (e.g. "eagle tulip" and "tulip eagle"). This gives 20 combinations, for which C^3 provided interpretations (e.g. "An eagle tulip is a tulip that grows on an eagle"). These experiments were intended to model the creativity of concept combination and therefore it makes sense to compare them with Divago. However, we cannot do a comparison that survives subjectivity because both the values referred here (of novelty and usefulness) and by C^3 output (plausibility, informativeness and diagnosticity) are not aligned in the same perspectives. Surely, C^3 interpretations would often fail in the *use* measure suggested by our script, and Divago would not necessarily do well with C^3 constraints, and any of these conclusions would lead nowhere in terms of saying which one is more creative. We can do a different, perhaps more interesting, experiment: check if Divago can arrive to the same results of C^3 (thus proving the possibility of achieving the *same* creativity, whatever it is); and determine which frames would be needed (would they have to be different?). First of all, to level both systems in terms of representation, we had to allow Divago access to the implicit relations with other concepts.

In order to check if Divago could find the same results as C^3, we applied the process described above as *tuning phase* and found that Divago is able to achieve the same results with an average error of 2.4 and median 1. This means that the *normal* error was either 0 or 1 and so the average was strongly affected by two outliers, of errors 8 and 10. These latter cases, in which Divago failed, were interpretations that included knowledge from third nouns, i.e. when there are attributes that do not belong to any of the inputs and come from other elements in the knowledge base. In the rest, it normally achieved the same results of C^3. Another remark is that it tended to include knowledge (e.g. that "an eagle tulip is solid") that C^3 had excluded via the informativeness constraint. Whichever one is more correct in this issue, it was

also clear that, by declaring that the *diagnostic* features of each noun are the features that differentiate the noun in relation to other nouns (an information that is actually available in C^3), Divago could reduce drastically this extra knowledge.

Perhaps the more striking conclusion from this experiment was that Divago could achieve the same results of C^3 (with the error just described) with a very small set of frames. Indeed, only two frames were needed about 85% of the time: `acore` (or `bcore`, depending on whether the focus was the modifier or the head) and `analogy_transfer`. This means that, essentially, C^3 picked one of the nouns (head or modifier), built the combination centered on it - which means it has the same structure and the same "core" attributes-, and also transferred the attributes directly related to the other noun. By *directly related* we mean attributes with distance 1 in its graph representation. This seems to indicate that combinations generated in C^3 were essentially of the *property* type. The other 15% of the results used also the `bframe` (or `aframe`, depending on the focus). The results, representations and C^3 results are listed in appendix F.

To conclude, Divago is able to achieve the same results of C^3 by using a proper set of frames (`aframe`, `bframe`, `acore`, `bcore` and `analogy_transfer`) as goals in the search. This means that, if wanting to configure it as a noun-noun combination interpretation system, only a smaller set of frame combinations should be considered, at least for hybrid and property types, and attention should be paid to other factors, namely to diagnostic features. On the other hand, considering the other experiments in this book, we conclude that Divago offers a much larger set of possibilities, without focusing specifically on the linguistics of combinations. In other words, C^3 models noun combinations[3] and Divago deals with concept combinations, being more open to other problem solving situations. We cannot answer the doubt about the limits of C^3 (could it also achieve the same results of Divago, with a proper configuration?), but it is clear that these are internally very different systems that tackle the same

[3]An aspect to refer is that the combinations, as modelled in C^3, are specific to a set of human languages (English, German, Dutch...). Others, like Portuguese and French, are less ambiguous because of the obligatory use of prepositions.

cognitive problem from different perspectives.

6.4 The creature generation experiment

We now invite the reader to imagine the following context: a game with a knowledge base of objects (creatures, physical objects, scenarios, etc.) coded according to Divago representation. Instead of having a specific object pre-determined for each game situation, let us suppose only a partial specification is given (e.g. for situation x, the game needs a $friendly$ creature that belongs to the *blue team* and should have a *strength* y). Depending on the size of the knowledge base, on the abstractness of these specifications, and on a competent engine for retrieving these objects, such a game could become more unpredictable and surprising, which is one of the current challenges in the area of game development.

We are developing a blending engine for games [Ribeiro et al., 2003] that would fit the context just given, which will partly be a re-implementation of Divago with attention to the specific domain of games and to performance issues, always vital in game development. In order to assess the feasibility of the idea and have a first insight on the problems involved, we made some experiments with generating creatures in Divago.

6.4.1 Tuning

We built an initial battery of 12 creatures, based on the Magic© The Gathering game, which comprises hundreds of different creatures, each one with a strength and defense value pair, a team color, and a *mana* cost (interpreted by us as $food\_consumption$). They could also have functionalities (e.g. protect another creature) and abilities (e.g. fly). Below, we show an example, the $pajem\_angelical$ (all the creatures used are in Appendix F):

```
isa(pajem_angelical, human)
isa(pajem_angelical, bird)
member_of(pajem_angelical, creature)
strength(pajem_angelical, 1)
defense(pajem_angelical, 1)
food_consumption(pajem_angelical, 2)
team_color(pajem_angelical, white)
color(pajem_angelical, human_colored)
made_of(pajem_angelical, flesh)
ability(pajem_angelical, fly)
pw(head, pajem_angelical)
pw(left_arm, pajem_angelical)
pw(right_arm, pajem_angelical)
pw(torso, pajem_angelical)
pw(wing, pajem_angelical)
pw(left_leg, pajem_angelical)
pw(right_leg, pajem_angelical)
actor(strength_enhancement, pajem_angelical)
actee(strength_enhancement, creature)
points(strength_enhancement, 1)
cost(strength_enhancement, 0)
actor(defense_enhancement, pajem_angelical)
actee(defense_enhancement, creature)
points(defense_enhancement, 1)
cost(defense_enhancement, 0)
```

For this stage, we had to obtain blends of creatures (to become the targets). In order to avoid our own bias, we asked another researcher, not aware with Divago's inner processes and having little knowledge of Conceptual Blending, to select randomly pairs of creatures and invent three different combinations for each of them. He chose 14 pairs of creatures, thus making 42 combinations. It was then our task to obtain the set of frames that could help Divago generate the same set of combinations.

Prior to starting the testing and designing of frames, it was necessary to check whether the solution actually existed in the search space, as we did in previous experiments. In other words, given a query with the exact relations of the target, the mapping applied by the designer[4], no integrity constraints (so that, whatever inconsistencies the target may have, it will not be less valued), and a configuration of 90% weight on Relevance (and 10% on Integration), Divago should be able, after a sufficient number of generations (aprox. 30, for the Horse-Bird experiment), to generate the exact same blend. In such a configuration, the search space has only one maximum, containing either the set of relations of the target or the subset that can be achievable by Divago.

[4]In this step, we had to extract this mapping ourselves from the 42 combinations, which was not difficult since every creature has a relatively small set of relations and concepts.

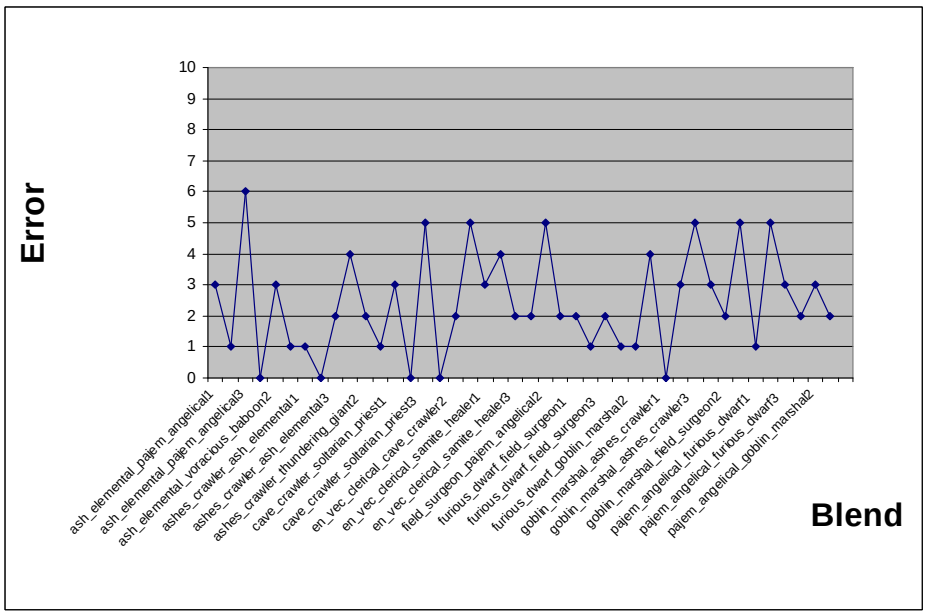

Figure 6.8: Least possible error achievable by Divago for each of the blends (mean values over 30 runs)

In figure 6.8, we can observe that only five of the combinations are completely contained in the search space. The reason for this apparent failure is simple when we inspect the combinations that produced bigger error. Let us analyze one of these combinations, the second combination of *field_surgeon* with *pajem_angelical*, which has an error of 5. The concept map of *pajem_angelical* has been given above and *field_surgeon* is represented as:

```
isa(field_surgeon, clerical)          pw(head, field_surgeon)
member_of(field_surgeon, creature)    pw(left_arm, field_surgeon)
strength(field_surgeon, 1)            pw(right_arm, field_surgeon)
defense(field_surgeon, 1)             pw(torso, field_surgeon)
food_consumption(field_surgeon, 2)    pw(left_leg, field_surgeon)
team_color(field_surgeon, white)      pw(right_leg, field_surgeon)
color(field_surgeon, flesh_colored)   actor(healing, soltarian_priest)
made_of(field_surgeon, flesh)         actee(healing, creature)
size(field_surgeon, medium_size)      points(healing, 1)
```

The designed combination for *field_surgeon*|*pajem_angelical*2 was:

```
member_of(field_surgeon, creature)
strength(field_surgeon, 2)
defense(field_surgeon, 1)
food_consumption(field_surgeon, 1)
team_color(field_surgeon, white)
points(defense_enhancement, 2)
made_of(field_surgeon, flesh)
cost(defense_enhancement, 1)
moves(field_surgeon, jumping)
pw(pajem_angelical_right_arm, field_surgeon)
pw(pajem_angelical_torso, field_surgeon)
pw(pajem_angelical_wing, field_surgeon)
pw(field_surgeon_left_leg, field_surgeon)
actor(defense_enhancement, field_surgeon)
actee(defense_enhancement, creature)
color(field_surgeon, human_colored)
pw(field_surgeon_head, field_surgeon)
pw(pajem_angelical_left_arm, field_surgeon)
```

As we can see, there was a scrambling of all the numbers involved (strength, defense, food_consumption, points and cost). This would be no problem if the substitution was consistent with the projection mechanism of Divago, but this was not the case: sometimes the number 1 becomes projected to 2 (both projection of *strength*), sometimes to 1 (in $defense$), the case becomes even more complicated because 0 projects also to 1 (in *cost*). The problem we are raising is that, by definition, a selective projection can have one and only one projection for each concept in the input concept maps, thus even if, by the mapping function, 1 is mapped to 0, 1 and 2 (which wouldn't be possible anyway with our 1-to-1 structure alignment algorithm), it can only be projected as one of these in the blend. This now seems to us a serious limitation, which we are focussing on in the development of the game engine. Another problem in this blend is that there is one completely new concept, *jumping*, which did not exist in any of the inputs. The reasoning followed by the designer was that, since the new creature has only one leg, then it can only move by jumping. This was important information, since it gave rise to a rule in a knowledge base. After determining the least possible error, we may proceed to determining the frames.

We can see that some of the previous frames (`aframe`, `bframe`, `acore`, `bcore`) were simplified with parameters (`frame(X)` and `core(X)`, respectively), although maintaining the same reasoning. In figure 6.9, we show the best performance achieved and in figure 6.10 we show the difference compared to the best possible result. The weight configurations remained the same (90% Relevance, 10% Integration). In Appendix

creature(X)	The name of the creature is the same as the name of the creature X
frame(X)	The creature maintains the same set of relations of input X
core(X)	The creature contains the core (all attributes) of input X
shape_amputation(L)	The creature has not any of the shape items(*arms*, *legs*, etc.) in list L
shape_transfer(S, X)	The creature inherits from X the shape S
function_transfer(X)	The creature inherits a function from input X
fightAttr(X)	The creature inherits the fight attributes (*strenght* and *defense*) from input X
shape(X)	The creature inherits the overall shape from input X
attr_transfer(L, X)	The creature inherits from input X the set of attributes in list L

Table 6.12: List of frames used in the creature blends

F, we list the generated blends.

Except for a few situations, it was not difficult to find a combination of frames that would give us the best result or a result that was very close to it. Since we tried to avoid tailoring the frames to the specific blends, for some (namely the ones with error bigger than 2, in figure 6.10), it was difficult to find a set of *stable* combinations of frames. Whatever combination of frames given, the search space became much too convoluted and, in order to accomplish some frames, Divago had to drop others, eventually achieving many different local maxima throughout the 30 runs. An example of such incompatible frames are the frames for $shape\_amputation(List)$ and $shape(X)$ (in table 6.12). The former removes a piece of the creature, while the latter tries to make it as a whole.

6.4.2 Free generation

In order to see the capacity of Divago of generating novel creatures, we applied several different configurations and creature combinations. Given the game context, this time we want to apply elaboration and produce a visual output. Due both to copyright obligations and to the availability of a set of three completely defined 3D creatures (a werewolf, a dragon and a horse, see figure 6.11), the content of the knowledge base was changed to include only these latter objects, coded in a similar manner as the creatures from Magic© The Gathering used above.

By changing the knowledge base and keeping the frames and rules obtained, we

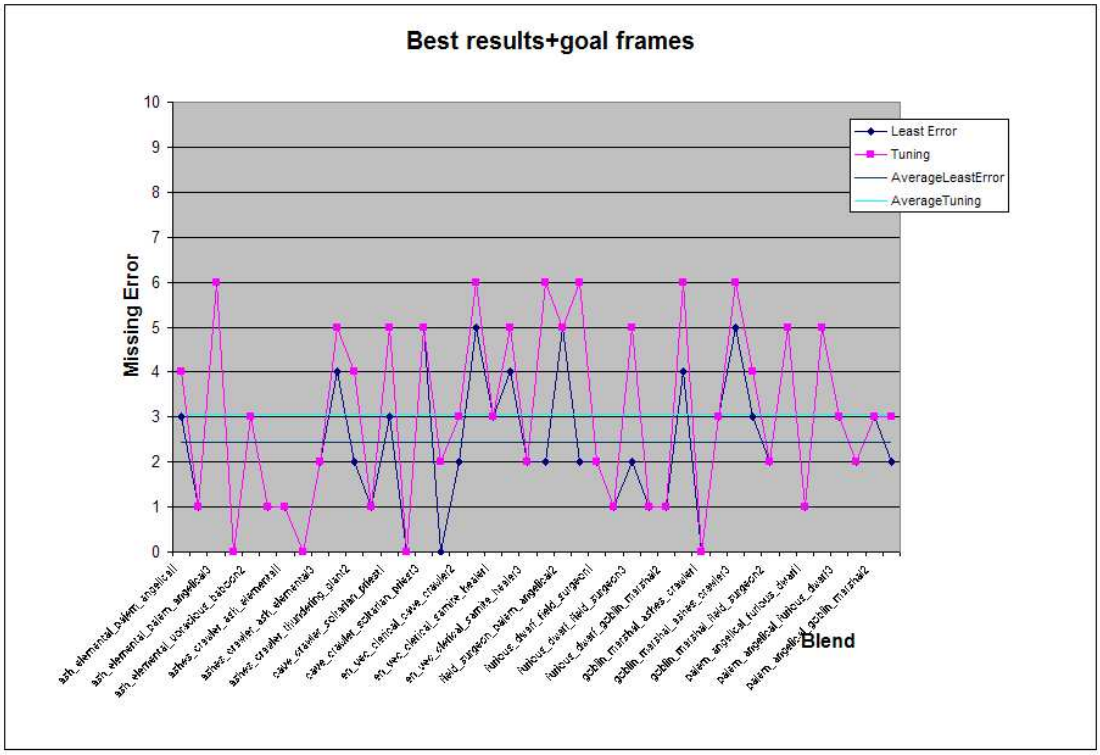

Figure 6.9: Best possible + frame results (mean values)

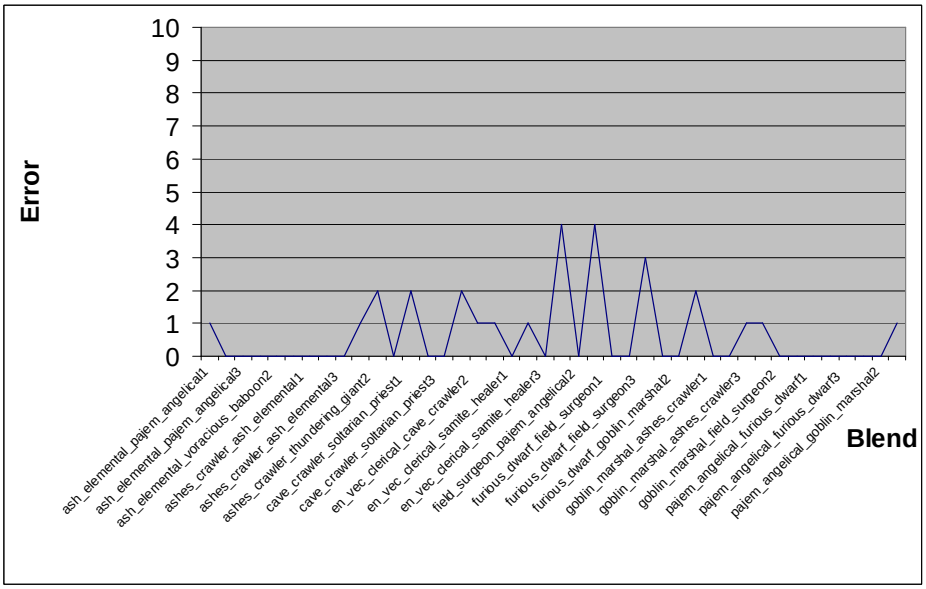

Figure 6.10: Efficency of Divago: difference to best possible values (mean values)

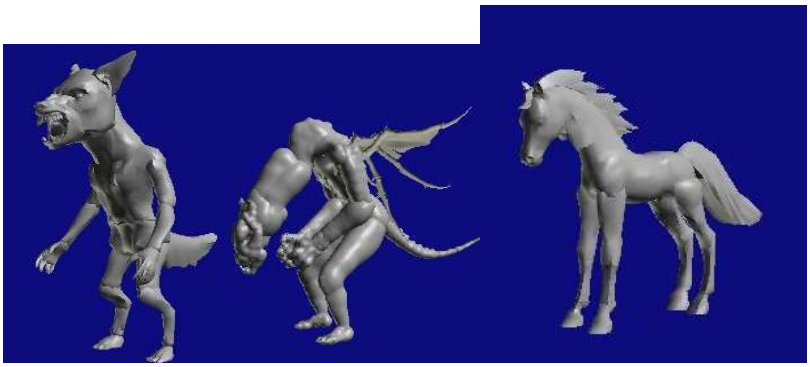

Figure 6.11: The creatures available in the knowledge base.

may verify that Divago has some degree of versatility. In fact, since we maintained the same kinds of relations in the concept maps, only a few rules and integrity constraints had to be created. We added rules for stating that something that has wings should have the ability to fly, something with an odd number of legs should get a wooden leg in the missing connection and it should move by jumping. Another rule calculates the mean when there are two different numerical values for the same attribute (and replaces them with this mean). We also added a rule stating that, when something is dangerous, very strong and very large ($strength > 5$ and $size > 3$), then it should get an *ogre head* and lose the original one. The new integrity constraints state that a creature should not have two different values for the same attribute, they should be symmetric, they should not have two *heads*, two *torsos*, or two members in the same place (e.g. two $left\ arms$)[5].

We only applied a single query throughout this experiment. To determine this query, we analyzed the history of frame combinations used to build the 42 creatures in the previous stage. We concluded that each one had two or three abstract frames such as `creature(X)`, `frame(X)` or `core(X)`. Since `core(X)` can be too specific, we decided to have `creature(X)` and `frame(X)` (the former forces to only have one creature name in the blend, the latter to follow its relational structure). There is also some regularity in the transfer of shape parts from each of the inputs, so we decided to have `shape_transfer(E1, Y)`, `shape_transfer(E2, Y)`, {E1 \= E2}, with X

[5]Of course, this does not imply that these creatures will never be generated. The system is only told that such constructions are to be avoided.

Combination	Weights					*nov*	*use*	Best blend
	I	T	MVR	U	R			(*nov*/ *use*)
horse werewolf	34	16	10	4	36	0.50	1.00	0.56/1.00
horse dragon	34	16	10	4	36	0.31	1.00	0.25/1.00
werewolf dragon	34	16	10	4	36	0.50	1.00	0.62/1.00
horse werewolf	10	0	0	0	90	0.86	1.00	0.86/1.00
horse dragon	10	0	0	0	90	0.69	1.00	0.57/1.00
werewolf dragon	10	0	0	0	90	0.74	1.00	0.65/1.00
horse werewolf	20	0	0	0	80	0.86	1.00	0.86/1.00
horse dragon	20	0	0	0	80	0.87	1.00	0.75/1.00
werewolf dragon	20	0	0	0	80	0.76	1.00	0.65/1.00
horse werewolf	30	5	5	15	45	0.83	1.00	0.71/1.00
horse dragon	30	5	5	15	45	0.37	1.00	0.50/1.00
werewolf dragon	30	5	5	15	45	0.59	1.00	0.65/1.00

Table 6.13: Results of creature combinations

different from Y. Thus, the query is:

$creature(X)$, $frame(X)$, $shape\_transfer(E1, Y)$, $shape\_transfer(E2, Y)$, $\{E1 \setminus = E2,\ X \setminus = Y\}$.

A good blend would therefore consist of the structure of one of the creatures with at least two of the shape parts of the other creature. In table 6.13, we show, for each pair of creatures and a weight configuration, the median results of novelty and usefulness obtained, as well as the scores for the best blend. The weight configurations consist essentially of the ones used in previous experiments.

In these experiments, almost every result entirely satisfied the query, thus giving a value of 100% for each one (the mean was 0.98 with a standard deviation of 0.05 corresponding to a few outliers). With respect to novelty, we can observe the variability of the results with the weight configurations. Indeed, when there is a focusing on Relevance and Integration, the system runs away from typicality, which is understandable when we analyze the used frames. They favor the use of knowledge from both inputs, without significantly favoring one input over the other. When doing so, as also verifiable in the other experiments, novelty tends to increase. When, on the contrary, the frames favor one of the inputs (e.g. the Pegasus is a horse), then the

typicality will tend to increase. When applying optimality principles other than Relevance and Integration, namely Topology and Unpacking, we can notice a decrease of novelty, although also achieving the 100% solution. This results from a heavy weight on Relevance, but also a preference for those blends that, although accomplishing the goal frames, respect the other principles as much as possible (Topology and Unpacking particularly favoring similarity to inputs). From applying Ritchie's features, we obtain the table 6.14, where we can see that the average and ratio of typicality has lowered slightly in comparison to the previous experiments (features 1 and 2). The average and ratio of value is, of course, 100% given that every result entirely satisfied the query (3 and 4). This fact implies also maximum values in other features (5, 7, 12 and 14). Again, this demonstrates the difficulties both in determining the *value* of something as well as in comparing it with other experiments. The ratio of untypical and valuable results, a very important measure for creativity, was raised to 0.667 (remember that in previous experiments, it was rounded to 0.333), expectable given that everything is now maximally valued. The rest of the features confirm the same conclusion: Divago was able to satisfy the value criteria for every combination, it did not reinvent any of the inputs and it was able to produce some proportion of results with low typicality (i.e. high novelty).

The generation of these 3D images was made by an interpreter developed in collaboration with other researchers [Ribeiro et al., 2003]. It receives the concept maps generated by Divago and produces a "wavefront obj" file, which describes the 3D image. The several parts of the creature were coded separately (e.g. $horse\_back\_leg$, $ogre\_head$) and placed together according to the concept map. To give an idea of the creatures generated, we now show some examples (the rest in appendix F). In figures 6.12, 6.13 and 6.14 we show the images of the best blends found in configurations 1, 2, 3 and 4, respectively.

In order to give an insight on the range of generated creatures, we also show the worst results. In these, we can see that they either lack one member (e.g. a wing), they have more than one member in the same point (e.g. a horse|werewolf with four

Criterion	Value
1	0.343
2	0.333
3	1.000
4	1.000
5	1.000
6	0.667
7	1.000
8	2.000
9	0.000
10	N/A
11	0.308
12	1.000
13	0.333
14	1.000

Table 6.14: Ritchie's [Ritchie, 2001] features results.

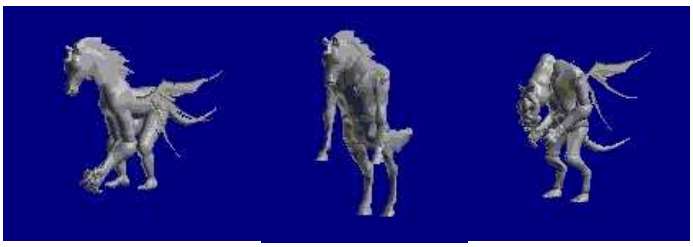

Figure 6.12: The best blends for horse|dragon (*nov*=0.25), horse|werewolf (0.56) and werewolf|dragon (0.62).

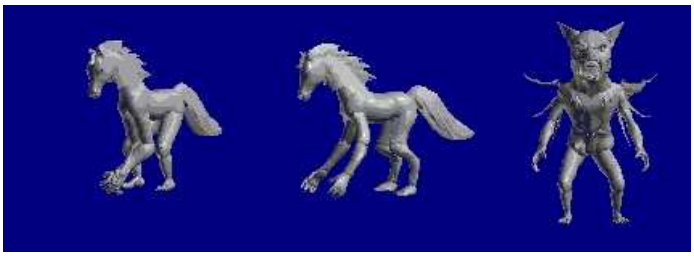

Figure 6.13: The best blends for horse|dragon (0.37), horse|werewolf (0.86) and werewolf|dragon (0.65).

Figure 6.14: The best blends for horse|dragon (0.75), horse|werewolf (0.86) and werewolf|dragon (0.65).

Figure 6.15: The best blends for horse|dragon (0.50), horse|werewolf (0.71) and werewolf|dragon (0.59).

back legs) or a pirate leg. See figure 6.16.

We left the visualization of these creatures to the end of this section to prevent the reader from placing excessive importance on the images. Indeed, there is much more behind each of these creatures, namely characteristics such as the *abilities*, their *strength* and $defense$ values and so on. Therefore their novelty, as reported in the captions, may have been affected by these non-visual characteristics.

The positive conclusion from these experiments was that Divago, as a generative model, can enhance the dynamics of a game environment. Suppose that it is allowed to blend not only creatures, but also scenarios, physical objects, behaviors, and so on.

Figure 6.16: Some worst blends for horse|dragon (nov=0.73), horse|werewolf (0.44 and 0.60) and werewolf|dragon (0.67).

Even more, with appropriate frames, it is theoretically possible to blend creatures and scenarios (with blends that, say, transfer the color or texture of a scenario to a creature, or the function of an object), which would considerably potentiate the possibilities of the game.

Although these experiments were stimulating as a motivation for developing a game, they also revealed some problems that need to be solved:

- The majority of the mappings used in this experiment were not made by the Mapper, they were hand coded, first because the Mapper is easily fooled by the representation simplicity (e.g. it can map *back_leg* to *right_arm* because both are connected to the creature with the same relation, *pw*), and second because some of the mappings are not based on structure alignment. The above situation of the scrambling of numbers is a good example.

- The projections are too restrictive, which prevents them from achieving a lesser error to the targets. Perhaps the game engine should consider multiple projections for the same concept map element.

- Divago is extremely slow in generating each creature. This is acceptable for a Prolog based prototype, but not for an on-line system. For this reason, the game engine must be developed in a computationally faster and lighter setting.

6.5 The established Blending examples

One of the contributions of this work is a computational model of Conceptual Blending and therefore it is fundamental to validate it with a set of examples recognized in literature as being conceptual blends. In section 3.3.2 and in Appendix B, we present what we call the established Blending examples. These are Blending case studies that appear in literature and that we think should be considered when building a computational model of Conceptual Blending. Although more examples could be

included, we restricted ourselves to the ones sufficiently specified (they should they should at least discriminate all the mappings and elements from the inputs) and which considered only two input spaces (there are many examples with multiple input spaces). Moreover, they should be considered "conceptual blends" somewhere in their description, to prevent any subjective evaluation from our side.

In table 6.15, we enumerate the examples, as well as their characterization. We believe that they are representative of a number of situations that have been approached over the last few years in the main CB reference literature. In order to provide Elaboration, a few rules were added to the generic domain, namely the movement laws rule (as presented in section 5.7) and rules stating common sense implication (e.g. "When an x and a y are married, they form a couple"). The Mapper was not used because the mappings for each example were already given in the literature.

Name	**Typology**
The Riddle of the Buddhist monk	Mirror network, Topology preserving
CEO boxing fight	Single-scope network
Gun wound	Nominal compound, Double-scope network
Kant debate	Double-scope, Mirror network,Pattern Completion
Land yacht	Nominal compound, Analogical
Trashcan basketball	Double-scope
Computer desktop	Double-scope,Metaphorical
Computer virus	Double-scope,Category metamorphosis
Same-sex marriage	Double-scope, Category metamorphosis
Sandwich counterfactual	Counterfactual blend, Single-scope
"Mabel is the daughter of Paul"	XYZ blend,Single-scope
Pet fish	Nominal compound

Table 6.15: 12 examples of Conceptual Blending found in literature

From each of these examples, we extracted the input domains, the blend, the generic space and the mapping. Normally, all these were directly available in tables or diagrams. The special cases were "Computer Desktop" and "Computer Virus", which were completed with some common-sense knowledge inserted by us.

For each of the examples in table 6.15, the goal of these experiments was to understand to what extent Divago was able to achieve the "correct" blends (the targets). Unlike the previous experiments, we will not focus on novelty and usefulness as the goal here is not to assess the creativity of the system, but to find how competent

actor(shoot, agent). actee(shoot, human). means(shoot, gun). result(shoot, wound).	error=0.75 $\longrightarrow$	actor(shoot, agent) actee(shoot, target) means(shoot, gun) result(shoot, wound) result(shoot, result)

Figure 6.17: Target blend for the "gun wound" example (left) and a blend (right)

it is in being a model of CB, i.e. in minimizing the error to the target. These targets correspond to the blends described in each example in literature. In table 6.17, we show the target for the "gun wound" blend as well as a (hypothetical) blend with error 0.75. It has 2 relations that do not belong to the target ("actee(shoot, target)" and "result(shoot, result)") and misses one target relation ("actee(shoot, human)"), yielding a sum of 3. Since the size of the target is 4, we have an error of 3/4 (=0.75). Please notice that this error measure (which follows the same reasonings described for calculating distance in the Horse-Bird experiments, in section 6.2) potentiates error values that may be higher than what intuition would say. For example, if a projected relation r erroneously replaces another one, this will count as two (one for *delete*, another for *insert*) instead of one (i.e. there is "only one" wrong relation). This doesn't mean we should divide by two the estimated errors, but we must take this into account when quantitatively analysing the errors found.

As in the Horse-Bird experiments, we divide the experiments into two different stages: isolated constraints and combined constraints. In the former, we will be able to watch the behavior of each isolated optimality constraint of Divago w.r.t. each of the examples. In so doing, it is possible to observe which of these constraints is able to achieve minimum error in the blend generation. The results will also be useful for the last stage of the experiments, in which we will combine constraints according to the results obtained previously. We will not focus on the principles of Web, Maximization and Intensification of Vital Relations. The reasons are now obvious: Web is not considered independent in our implementation, thus it wouldn't make sense to test it in isolation; we are not considering the compression role of Vital Relations, which is after all their reason of existence, thus the results of Maximization/Intensification of

Frame	Description
`aframe`	The blend has the same relations of input space 1 (although the elements may differ)
`aframe`	The blend has the same relations of input space 2 (although the elements may differ)
`aprojection`	The blend has the elements of input space 1
`bprojection`	The blend has the elements of input space 2
`analogy_transfer`	[Part of] the blend results from the transfer of all neighbour elements and relations of mapped elements of input space 2 to their counterparts in input space 1
`role_transfer`	[For a noun-noun compound blend] the head is projected to a role element of the modifier (e.g. for "gun wound", "wound" is projected to "result" in the blend)
`Debate`	The blend has all the relations expected for a debate scenario (see example 3)
`day_compression`	All temporal elements of both inputs become referent to the same day
`head_transfer`	[For a noun-noun compound blend] all relations and elements connected to the head (input space 2) are projected untouched to the blend

Table 6.16: An informal description of the frames used

Vital Relations wouldn't produce conclusions that we could confront with their own corresponding theory.

We also followed the methodology for the noun noun experiments, with a first, *tuning*, step to obtain the frames. The set of frames achieved is listed in table 6.16 and we show their code in appendices B and F. The table 6.17 shows the goals used in each of the examples.

In table 6.17, we can observe some regularity in the choice of frames. Normally, there is at least one generic organizing frame (e.g. `aframe`, `bprojection`), which establishes the general structure of the blend (`aframe` makes the blend maintain the relations of input 1; `bprojection` makes the blend maintain the elements of input 2). Then, there may be other frames that can be transforming (e.g. `analogy_transfer`, `head_transfer`, `day_compression`) or organizing (e.g. `role_transfer`). Finally, pattern identifying frames like `debate` are used in specific situations. A final remark concerns the "sandwich counterfactual" example. As referred to in the creatures experiment, Divago is technically unable to reach the target when a 1-to-many mapping is necessary, as happens with this blend, so we removed it from the rest of the

		Goal
Example	Frames	Relations
Buddhist monk	day_compression	meets(monk1, monk2)
CEO boxing	aframe, bprojection	
Gun wound	aframe, role_transfer	
Kant debate	bprojection, debate	
Land yacht	aprojection, head_transfer	
Trashcan basketball	aframe, analogy_transfer	
Computer desktop	aframe, bprojection, analogy_transfer	
Computer virus	aframe, head_transfer	
Same-sex marriage	aframe	same_sex(person1, person2), married_with(person1, person2)
Sandwich counterfactual		
Mabel is the daughter of Paul	bframe, aprojection	
Pet fish	aframe, bprojection	

Table 6.17: The goals used for the Relevance principle

experiments. Apart from Relevance, no other Optimality principles demand special configuration concerns, therefore their application depends exclusively on their weight being higher than zero.

6.5.1 Experiments with isolated principles

The intention of this part of the experiment is to find the dominant principles of each blend within the first order i.e., it gives us the principle that seems to be immediately prevalent in the blend (within the second order, we would see pairs of principles that seem to be prevalent, and so on), thus giving a first classification for our blends. This latter idea can become even more precise when considering the behavior of all (isolated) principles for each blend and comparing them as a set, so, instead of comparing each value individually, we can compare a sequence of values.

In figure 6.18, we present the overall results of each of the optimality principles. An immediate conclusion can be drawn from this graph: the Relevance principle has consistently smaller error than any other principle; the "CEO Fight" blend beats the record of maximum error in two of the principles. This confirms the results of previous

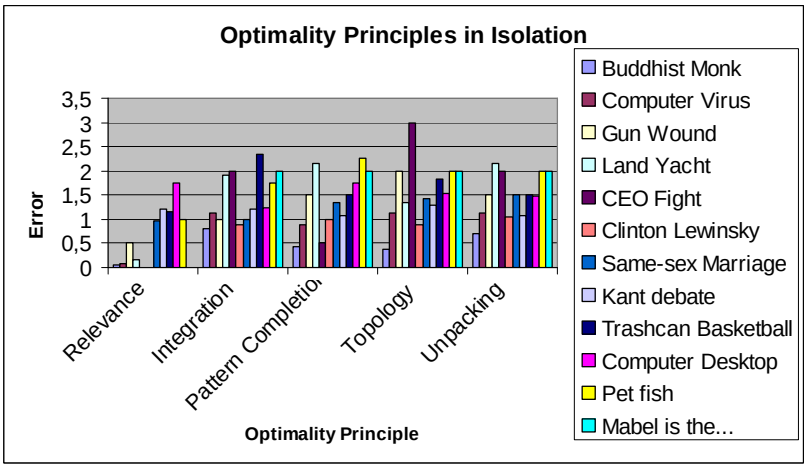

Figure 6.18: The median of the error, for the optimality principles in isolation

experiments and highlights the importance of this principle. On the other hand, it demonstrates that almost none of the blends have a naturally inherent tendency for the other principles in isolation (at least, as we implemented them). Furthermore, since the Relevance principle is configured differently for each blend according to an intuitive, trial-and-error, choice of goals, it seems to indicate that there is no generic, context-independent, principle that can lead to an exact solution. It is also important to say that, since Relevance demands a specific configuration, its value may vary immensely according to the choices of the goals used and so the reader should retain this aspect while interpreting these graphs.

A more practical conclusion from figure 6.18 regards the graphical representation itself. Although using bars seems to be correct given that we have no scale in the x axis, it does not simplify the task of understanding and organizing the several kinds of blends we may be considering. For this reason, we decided to represent it as a line graph (fig. 6.19).

Analyzing the results paying attention to the typology presented in table 6.15 was our first concern for this stage. However, we found no particularly revealing patterns in the results. There may be many different explanations for this, but the most salient one suggests that there may be no correlation between any of the principles and the given typologies. For example, a double-scope blend may be Topology preserving and

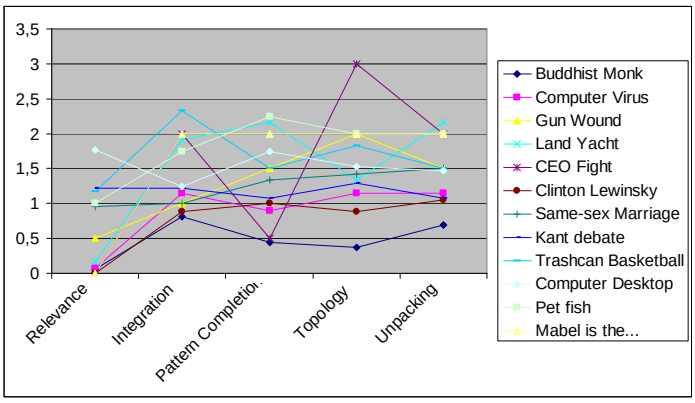

Figure 6.19: Reduced line graph corresponding to the error of isolated optimality principles

another one may demand strong Integration, a mirror network may also preserve (or not) Topology. For this latter case, even if it does preserve Topology, the value of this principle may not have to be very high (i.e. in a mirror network, there is *a priori* topological correspondence between the input spaces, so the preserving effort may be low). Many other arguments that testify to the complexity of blends could be given, eventually ending up in the uncertainty of the typology itself (e.g. the difference between single and double-scope can become extremely subtle).

On the other hand, we found clear patterns in the choices of goal frames for the Relevance principle. Both single-scope blends achieved exactly the target with the pair `xframe` and `yprojection` (being x and y either "a" or "b"). This completely agrees with the idea of single-scope - the elements of one of the inputs are organized according to the frame of the other. Nominal compounds also show a pattern: there is a "projection" of one of the inputs, the one that coincides with the focus of the compound. The exception is "gun wound", which is also double-scope. Double-scope examples normally demand more specific frames (e.g. `debate`, `analogy_transfer`, etc.) or specific relations (e.g. $same\_sex(person1, person2)$) and were less consistent in reaching the exact target. This confirms that "in a two-sided network [i.e. double-scope] (...) it is necessary to use a frame that has been developed specifically for the blend and that has central emergent structure. (...) In two-sided networks, then, we expect to see increasing competition between optimality principles and increasingly

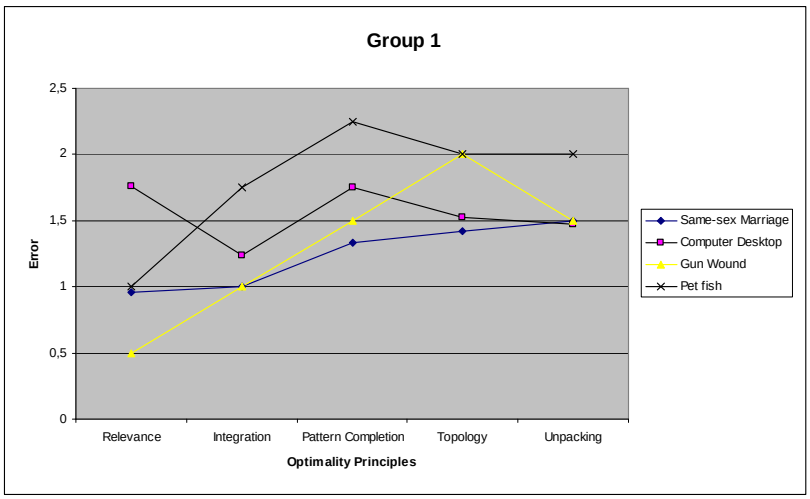

Figure 6.20: Group 1

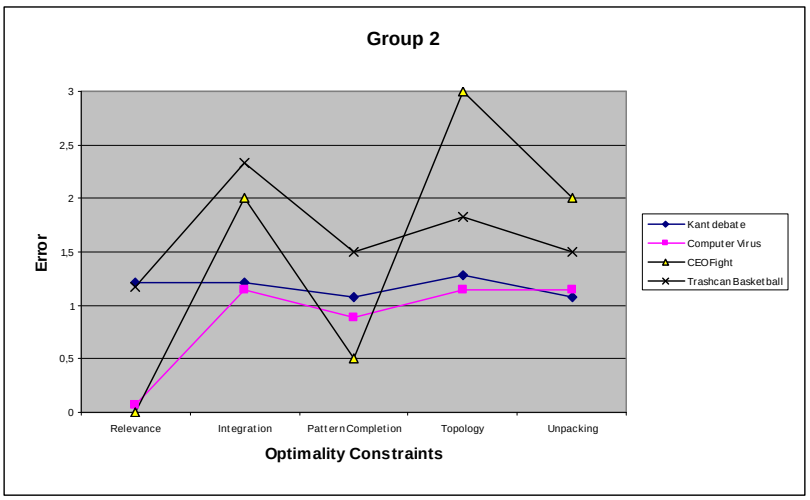

Figure 6.21: Group 2

many opportunities for failure to satisfy them" [Fauconnier and Turner, 1998].

Considering a qualitative evaluation based on similarity in terms of the shape of the graph and of the principles that yield smaller error, we found four different groups of blends: Group 1 ("Same-sex marriage", "Computer Desktop", "Pet fish" and "Gun wound"); Group 2 ("Computer Virus", "Kant Debate", "CEO fight" and "Trashcan Basketball"), Group 3 ("Mabel is the daughter of...") and Group 4 ("Buddhist Monk" and "Land yacht"). These are shown in figures 6.20, 6.21, 6.22 and 6.23.

Except for "Computer Desktop", the blends in Group 1 have Relevance yielding

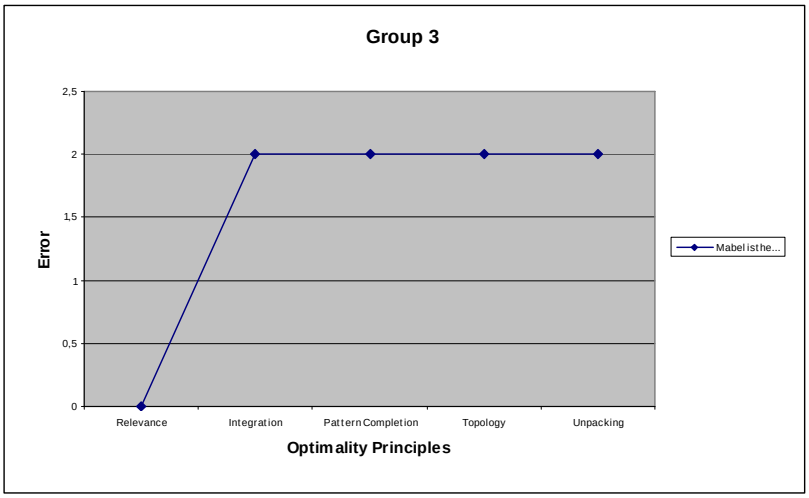

Figure 6.22: Group 3

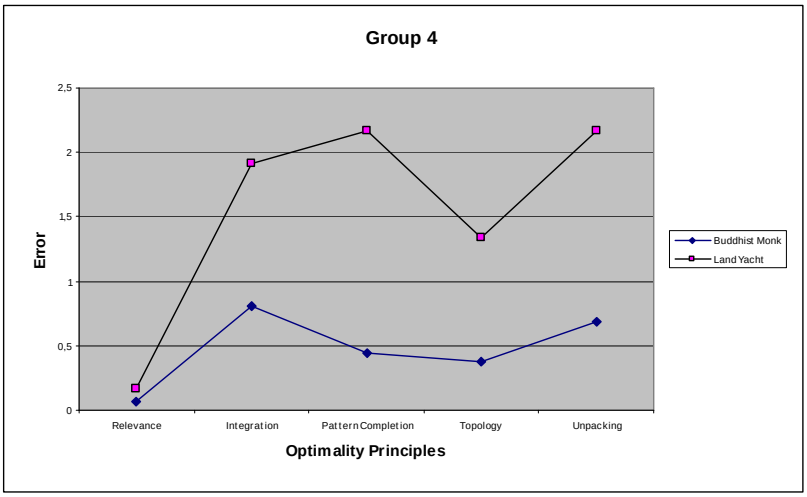

Figure 6.23: Group 4

the smaller error being followed by the Integration and in similar proportion by Pattern Completion. "Computer Desktop" was the only example that we weren't able to find a good set of goal frames for. As a result, isolating Relevance yielded an extremely large error. Thus, the reason for including it in this group lies on the other principles. We can see that, as in the other examples of this group, Integration yields the smaller error, followed by Pattern Completion. Topology and Unpacking show a less constant pattern, although all falling within a small error range.

Figure 6.21 shows Group 2, which consists of blends that have in Pattern Completion the second smaller error and in Integration and Topology the two highest errors. Another interesting remark is the high similarity between the results of "Computer virus" and "Kant debate" (except for Relevance which, as said previously, may vary according to goal configuration). Group 3 has only a single example. As we see in Figure 6.22, there is a pattern of zero error in Relevance followed by stabilization around a specific error value. Every principle (except Relevance) got an error of 2. After analyzing the results carefully, we understand that: a) it is based on a single relation ($daughter\_of(mabel, paul)$); b) being so, the target is only achieved when `bframe` and `aprojection` are achieved simultaneously, which does not happen consistently except in Relevance. In Integration, we observed that the system gives a higher score to the accomplishment of `bframe` alone, instead of its combination with `aprojection`. This agrees with our intuition for Integration presented in section 5.6, in which we argue that, when the blend is totally covered by a single frame, its Integration value should be stronger than when it is totally covered by two different frames.

The examples that have Topology and Relevance as the main principles were gathered in Group 4 (fig. 6.23). In fact, particularly for the "Buddhist Monk", this agrees with the analysis of Fauconnier and Turner [Fauconnier and Turner, 2002, p.45], who stress the role of Topology as being fundamental for this example. The results of "Land yacht" were unexpected and some further analyses revealed that the target is topologically very similar to one of the inputs (the "land") and so the

Combination	Weights (I, PC, T, U, R)
Relevance+Integration (Group 1)	(50, 0, 0, 0, 50)
Relevance+Pattern Completion (Group 2)	(0, 50, 0, 0, 50)
Relevance+Integration+Topology (Group3)	(33.3, 0, 33.3, 0, 33.3)
Relevance+Topology (Group 4)	(0, 0, 50, 0, 50)
Combination 2	(20, 20, 20, 20, 20)
Combination 3	(38, 0, 18, 4, 40)
Combination 4	(21, 0, 21, 5, 53)

Table 6.18: Weights used in the experiments with combination of principles

highest value in Topology may get close to the target, particularly when few elements are projected from the other input (the "yacht"), thus having, in the blend, a copy of the "land" domain. Moreover, since this example had difficulty in achieving small errors (except for Relevance), the salience of Topology is probably magnified.

There seems to be no specific pattern underlying the groupings found. We analyzed issues like concept map size, emergent structure and difference to input domains, but still no patterns were found. In the next section, in which we will apply combinations of weights to each of the groups, it will be possible to check if the same grouping tendencies are maintained. If this happens, then we will have more evidence of the meaningfulness of the groupings made.

6.5.2 Experiments with combination of principles

The second part of these experiments is dedicated to combining the optimality principles in different ways. We made four different combinations: the two principles with least error from the previous stage of these experiments; all principles with equal weight (Combination 2); application of two sets of weights derived from Horse-Bird, normalized to exclude MVR (Combinations 3 and 4, resp.). In Table 6.18, we show the weight configurations used.

We organized the results by maintaining the groupings already discussed and, for each blending example, we also added the best value of error obtained in the previous

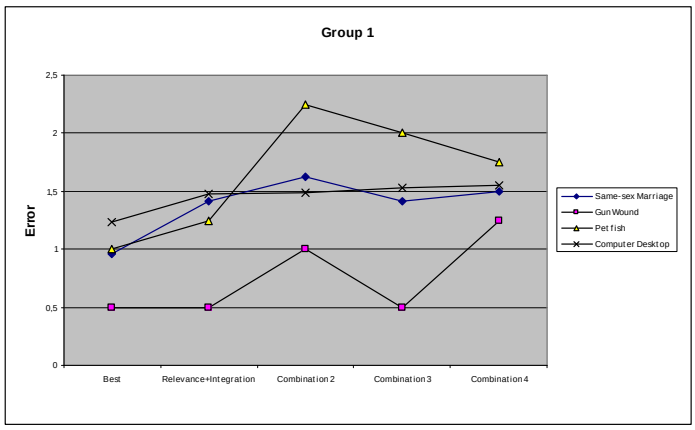

Figure 6.24: Combinations of weights applied to Group 1

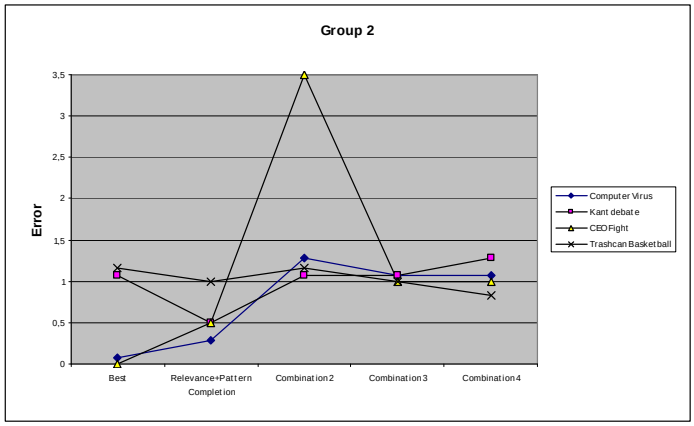

Figure 6.25: Combinations of weights applied to Group 2

section in order to better understand the evolution.

The first observation is that combination 2 tends to obtain the worst results, meaning that adding all principles with equal weights reveals unproductive, supporting the idea that each blend may result from different combinations of these competing pressures. Another immediate observation regards the difficulty in approaching the target, i.e. in comparison to the best so far achieved (with the isolated principles) the error only becomes smaller than this best in the examples of "Kant debate" and "Trashcan basketball" (both when combining Relevance with Pattern Completion), while in "Gun wound" and "Buddhist monk", the first combination reaches the best solution error. For all the others, no combination brought better results. This raises perhaps the most important question for this experiment: is Relevance (or the way

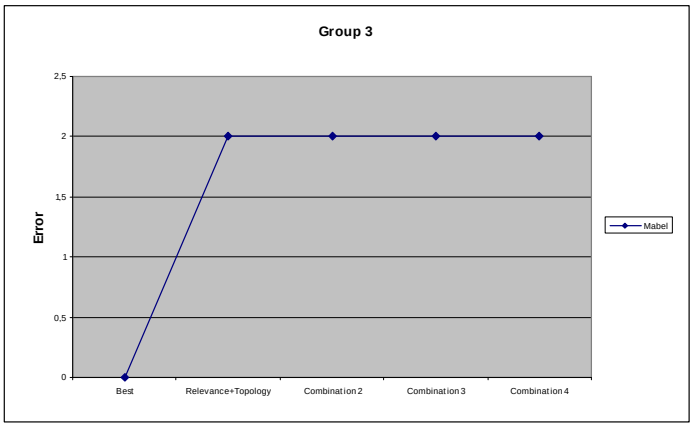

Figure 6.26: Combinations of weights applied to Group 3

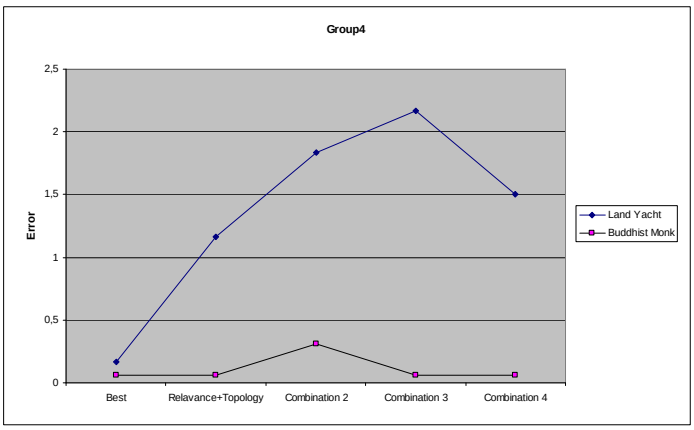

Figure 6.27: Combinations of weights applied to Group 4

we configure it) sufficient to achieve the best blend? From all the experiments done so far, the answer turns out to be "yes". Indeed, with a proper goal choice, it is in principle possible to find the target, independently of its complexity. And this takes us to the perspective of classifying blends by frames rather than by optimality principles.

It is also observable that the four groupings found in the previous section are generally consistent with the new results, although the groupings may seem dubious at points (e.g. we could equally imagine the interchange of some examples from Groups 1 and 2 or from Groups 3 and 4). Therefore, we may now find two distinct major groups (let us call them clusters), the first one comprising groups 1 and 2. In this cluster, Combination 2 normally gets the highest error, followed by Combination 4. Except in Group 2, there is also a tendency for obtaining the same value for Combination 3 and the "two best principles" combination. The other cluster comprises groups 3 and 4, in which Combination 4 seems to acquire much better results than Combinations 2 and 3. This may mean that the blends in question demand less Integration (which can be confirmed in the previous section), since Combination 4 differs from Combination 3 essentially in the weight of Integration. Still in this cluster, we can also see that one example, the "Buddhist monk", is very stable in the obtained errors, getting worse results only when the weight of Relevance is shared with all others (Combination 2).

A final and more practical point regards the computational performance of the system. Given the complexity of some blends and our experiment requirements (30 runs for each blend, for each configuration), our search engine needed a considerable amount of time in some cases. We used an Intel Pentium $IV^{®}$ at a speed of 2.4 GHz, which needed sometimes four to five hours to find a solution (only in the most complex cases like "Computer Desktop" with Combinations 2, 3 or 4). At best, it took 2-5 seconds to find a result. These values can get much lower after an optimization of the system, but it will find it hard to become fast enough to enable comparison with the performance of our own cognitive system, particularly if taking into account the extremely large amount of background knowledge that we are able to cope with.

6.5.3 Some considerations

Given the complexity involving several aspects of Conceptual Blending and the extremely wide range of situations considered, the Blending mechanism of Divago is far from an exhaustive model. Its initial and most fundamental motivation was to be the Bisociation module for Divago. Thus, its scope can be considered very limited in comparison to what we can find in all the CB literature. This fact does not inhibit us from understanding how much our model is capable of simulating and making predictions in the context of established examples. Furthermore, given that, as far as we know, there are no operational models for the study of blends with a stable, commonly used, methodology across researchers, Divago could be a starting point for the validation and discussion of the subset of blends that it can consider. We believe that such a study would bring some interesting outcomes: (possibly new) categorizations of blends; understanding of the underlying frames that are recurrent in some blends; validation of observations previously made (e.g. Topology is important in the "Buddhist Monk"); predictions regarding novel blends (e.g. its underlying frames).

It is a fact that, with the knowledge representation used, Divago was able to find the targets (or very similar solutions) also found in the literature. Achieving such, it is our opinion that it reached a capacity for making Conceptual Blending, although still at a relatively basic level in comparison to our own cognition and to the world of examples discussed in [Fauconnier and Turner, 2002], some considering many input spaces, many consecutive blends, the majority of them not formally described or represented. Clearly, a more dynamic knowledge representation, perhaps not entirely symbolic, would be needed to cope with more elaborate, and more realistic, examples at the level of cognition.

6.6 Discussion

The experiments presented in this chapter raised a set of questions we would now like to discuss. We will start by a set of practical issues and progress towards more

philosophical questions. We intend to focus on the problems and virtues of the implementation and of the model, and lead the reader towards possible evolutions and applications.

From the results obtained (namely regarding *nov* and *use*), it is an undeniable fact that a few predictions can now be made regarding the behavior of Divago. We can now say that, with a problem that can be specifiable via a query, it is able to retrieve a good solution, if this exists in the space of concept combinations and if some prominence is given to Relevance and Integration weights. We can also predict that Divago will diverge from the inputs if this query so implies, or when there are frames available that so imply. Ultimately, if we have no query or frames available, Divago will have the tendency not to reinvent the inputs, but to generate blends that inherit parts from both, although without any specific overall coherence, namely because of its space complexity. This raises the issue that, by themselves, the Optimality Constraints, as we modelled them, did not have success in Divago, except for Relevance and Integration. In this context, and perhaps in general, we believe that the eight constraints of Fauconnier and Turner can be reduced to three: Relevance, for purposefulness; Integration, for internal coherence; and Topology/Unpacking, for external coherence. We must acknowledge, though, that we did not present a thorough account for the Vital Relations and their compression role, which may imply that those three constraints are incomplete regarding an implementation of CB.

With respect to a direct comparison with WASP (the poetry generation referred to in section 2.2.2 [Gervás, 2000b, Gervás, 2002]) regarding the values of Ritchie's features, as we said earlier, such an exercise is merely academic in the sense that, in practice, these are very different from each other. Nevertheless, some new conjectures can be made. First of all, we must provide the values obtained for WASP, as well as a summary of Divago's results (figure 6.19).

From an analysis of these results, the first thing to conclude is that WASP clearly produces a higher average of typicality and lower average of value than Divago (features 1 to 4). This imbalance also affects features 5 to 8, which basically reassures

	Experiment			
Criterion	Horse-Bird	Noun-Noun	Creatures	WASP
1	0.443	0.543	0.343	0.71
2	0.273	0.563	0.333	0.54
3	0.504	0.782	1.000	0.47
4	0.636	0.781	1.000	0.24
5	1.000	0.778	1.000	0.36
6	0.364	0.344	0.667	0.05
7	0.500	0.786	1.000	0.12
8	1.333	0.786	2.000	0.28
9	0.000	0.036	0.000	0.000
10	N/A	16.000	N/A	N/A
11	0.406	0.513	0.308	0.71
12	0.483	0.831	1.000	0.47
13	0.273	0.500	0.333	0.54
14	0.636	0.781	1.000	0.24

Table 6.19: Ritchie's [Ritchie, 2001] features: summary of Divago + WASP results.

that outcomes of Divago were classified as higher valued than WASP's, and features 11 to 14, which get the same conclusions by comparison with the inspiring set. Notice that, by features 9 and 10, neither system tends to reinvent the inspiring set in any way (i.e. these latter features cannot add any new conclusions). We must insist that the actual values should not be taken further in this comparison. At this point, the most one can do (and the actual importance of Ritchie's features in this case) is to conclude that, according to the features used, Divago seems to be more inventive than WASP. To go further, one would have to compare the specific evaluation procedures of each system. This would imply a comparison of a Poetry generation system evaluation methodology (which was based on people's interviews and stylistic analysis) with the ones used in our system. As this is an unsafe comparison to make, we trust that the evaluations just made will be more important for future related works (e.g. of creativity assessment in concept invention systems) than for a competent comparison of Divago and WASP (to read more about this subject, also including a comparison with Dupond, read [Pereira et al., 2005]).

It can also be said that our measures of typicality and value are simplistic and

therefore lead to a high variability in Ritchie's features as well as to some counterintuitive results. As a formal system, Divago needs a set of well defined criteria and the question is whether the effort of building more complex formulae or heuristics would be justified by an added value in results. As we said earlier, there can be no universal measure of *value*, and therefore following the *Occam's razor* principle seems the adequate choice. Moreover, this choice becomes a virtue of Divago in the sense that this system suggests a validation that can be applied to different domains, not being tailor-made for the specific application, as happens in the majority of systems referred to in this book.

The knowledge representation showed itself to be problematic in some experiments. Namely for the noun-noun combinations and for the established blending examples, it is clear that our common sense knowledge of the concepts goes far beyond the representations considered, this being one of the main reasons for such counterintuitive results. Although the frames seemed extremely powerful, they can never compensate for the poor quality of the concept maps. Ideally, the concept maps should be dynamic (such as in the Slipnet of Copycat [Hofstadter and Mitchell, 1988]) and not isolated. Actually, this leads to the first strong self criticism we must make regarding Divago.

While, throughout this book we have been arguing for a multi-domain environment, Divago, as it is implemented, only considers this on a superficial level. We can see this at two levels. At the level of the individual experiments, the pairs of input concepts considered are rarely distant enough to each other such that one can unquestionably consider them from *different domains*. Still at the level of individual experiments, Divago is given externally (or randomly) a pair of concepts, and thus it does not "wander" in the multi-domain environment, but in the space defined by the Blender, which is much more restrictive. At the level of the overall experiments, it did not consider input concepts from two different experiments (e.g. blending a house and a horse, or a werewolf with a paper). After verifying the complexity we faced in the experiments presented, it becomes obvious that our choice for a set of isolated

experiments, some with familiar concepts (horse-bird, house-boat, creatures), some with less familiar concepts (noun-noun combinations, established blending examples), comes from a need to observe the capacity of bisociation of Divago, while avoiding being distracted by other, yet also important, aspects. We trust that, first of all, in order to reason in a genuine multi-domain environment, such a system must be able to deal with simpler situations, with the motivation of gradually being open to a wider scope, as we made with Divago.

We have been recurrently criticizing the structure alignment algorithm used in Mapper, but we must add that this is very much unexplored ground and we have so far found no promising alternatives. The algorithm demonstrated the virtue of being computationally inexpensive and of proposing mappings for Divago in some of the experiments.

Another issue to discuss is the interpretation of the blends. We proposed visual and textual interpretations, yet these describe only a selection of aspects, leaving out some others of potential importance. This is rather a problematic issue. The concept theories and instances were designed to be self explicatory, however in order to avoid ambiguities, large amounts of knowledge are necessary, and each piece of knowledge recursively demands the explanation of its constituents, thus demanding the existence of ground symbols. The semantics of these ground symbols must be context dependent (e.g. the semantics of a *wing* can be a 3D or 2D image, a functionality description, a word, another network of concepts with $feathers$ and *bones*, each one appropriate in a different context). This means that, while knowledge representation for concepts can be domain independent, their interpretation must be domain dependent (or at least context dependent).

Divago's versatility has also been recurrently referred to in that it needs no structural changes for working with any two different pairs of input concepts. For each pair of concepts, it demands their description (via concept theory and instances), a choice (or creation) of frames to use as query and to add to the generic domain, and a choice of the weights for the Constraints module. The description of the concepts

will always demand some effort, while the rest may be picked from those available. Of course, for each application, an interpreter may be necessary, this being the least versatile, although unavoidable, aspect.

We have demonstrated also that Divago is divergent in the sense that it tends not to reproduce the input concepts and agrees with the theories and principles enunciated in the previous chapters regarding Conceptual Blending and our Model of Concept Invention. It was empirically demonstrated that, with an appropriate set of frames (and sufficient Relevance and Integration) the system is able to produce useful and novel results. Indeed, we believe frames have an extraordinary power only superficially explored here. Remember that they may comprise small programs with the expressiveness of the Prolog language.

In spite of the uncertainties in the assessment of issues like creativity or divergence, these experiments show that Divago was able to accomplish two very objective goals: it is able to reach approximately the same results of C^3, with a specific set of frames; it can produce the same blends, or approximate ones, as in the examples listed from the Conceptual Blending literature. The latter is particularly important as it may become a computational methodology for analyzing blends.

As suggested by some of the contexts that we invited the reader to imagine, we assume that the model presented in this book is more useful as a reasoning mechanism (possibly at the meta-level) that can help a computational system to extend its space of possibilities, i.e. *transform* it. Such a system would need to give our model the description of what a valid possibility involves via a language such as used in the queries of Divago), which would then generate bisociations until finding a satisfactory outcome. We argue that this emulates, at least partially, the process of imagination according to Koestler, Guilford and Fauconnier and Turner. However, we are aware that Divago is very incomplete with regard to the implementation of such a model.

As a model of creativity, we have to reassert that it lacks some fundamental parts, namely the interaction with the environment, which is so fundamental, according to Csikszentmihalyi and others. If Divago was a perfect implementation of the model

of Concept Invention discussed here, it would still be somewhat autistic due to a lack of contact with the external world. This is another reason why it should not be considered alone and independent of a specific purpose or environment.

In conclusion, as far as our definition of creative system given in section 2.2.2 goes, Divago clearly falls into that category. For every experiment made, it produced more results that are not replications of previous solutions than copies of its own knowledge; it was able to reach the established goal or just fall short of it in the majority of the situations. It is based on a cognition-centered model - the model of Concept Invention, from chapter 4.2 - and is implemented as a hybrid AI system, since it applies typical Knowledge Based Systems techniques (rules, constraints, knowledge representation) as well as Evolutionary Computation algorithms (the GA of the Factory module). Thus, one can conclude that Divago is also an AI system, an argument for the thesis that Computational Creativity should be part of AI, as much as creativity is part of intelligence.

Chapter 7

Conclusions and Future Directions

Now that we are reaching the end of this book, it is time to draw the main conclusions, both at the level of Creativity modelling and its many associated questions that have been referred to since the beginning and at the level of the practical implementations and models presented here. We cannot finish without elaborating on the main contributions and pointing to possible future directions of research.

Modelling Computational Creativity can be seen, by the most skeptical, as an *a priori* impossible mission. As a formal machine, a computer is deprived of aspects that are often considered fundamental, such as intention or emotion. The same argument is also used against Artificial Intelligence. In either case, one falls into a human-centric view which is monolithic and reductionist at the same time. It is monolithic because it assumes that only a being (or a thing) that has all the characteristics together in a whole can be considered creative or intelligent and it is reductionist because it reduces creativity and intelligence to an all or nothing basis, assuming that only humans fulfill all of these conditions. From the many studies in the areas of Psychology, Philosophy or Cognitive Science, some of which have been described here, we have reached the different conclusion that Creativity is more continuous than discrete and that it is related to many different aspects, some more *computational* than others. We have presented arguments for the construction of Computational Creativity, which does not have to be the same or measured with the same thresholds as the Human Creativity. Computational Creativity must be defined more precisely and we have proposed that

it should be bound by the capacity of generating a reasonable amount of valued and atypical solutions to a problem. One conclusion we took from this work was that this modelling of creativity must not run away from an AI framework. In other words, one should not avoid facing creativity as a problem of search for solutions to problems (although these become normally much less specified) or using mechanisms typically from AI (e.g. genetic algorithms, logic programming, neural networks). This does not mean that creativity can only bring new applications of AI techniques, on the contrary we believe that creativity is a missing part of AI and, in the same way that humans (and nature in general) need creativity to be more versatile, less constrained to usual solutions to usual problems, more open to change and ready to cope with unpredictability, the machine will need to use more creative behavioral skills in order to become more competent.

The model presented here applies many of the well known AI techniques, such as rule-based systems or genetic algorithms, but these are only the means to the broader end of modelling bisociation and divergence, which have not been approached within AI. Traditional AI search has been considered throughout this work, but always with attention to the world beyond the search space, and to methods for how to reach it. It is within this paradox of reaching the unreachable that the study of Creativity can become fundamental within AI. This book brings some contributions motivated by this quest:

- **Model of Concept Invention** based on principles and theories from Psychology, Philosophy and Cognitive Science. This model was the *leitmotif* of this book, representing an ideal system, as opposed to an actual implementation. It proposes a set of modules and their interaction for the invention of concepts via the combination of concepts from distinct areas of the knowledge base. This concept invention is essentially inspired by Koestler's *bisociation* [Koestler, 1964] and Guilford's *divergent thinking* [Guilford, 1967], while still leaving open other forms of concept creation and of concept combination.
- **Computational model of Conceptual Blending**. We present

the first extensive computational approach to Conceptual Blending [Fauconnier and Turner, 1998], which takes into account the several processes (composition, completion, elaboration, selective projection) and principles (optimality principles) described in that framework. This model is directly applicable to the study of blends that are based on one-to-one mappings and two input domains. Once these input domains are represented according to Divago's representation, they can be tested and analyzed, as we showed. A specific suggestion we obtained from this computational model of blending was that the set of optimality principles proposed in the framework are reducible to a smaller set of principles (Integration, Topology and Relevance).

- **Divago**. The two models above ultimately led to the implementation of a system, Divago, which comprises a set of proposals for how to implement the many modules involved. It is a system entirely built in Prolog and completely functional and configurable. It has been thoroughly described in this book and is available for use by other researchers (e.g. for experiments with Conceptual Blending) or for connecting to another system (e.g. to extend this other system's knowledge base). Divago is unique in many aspects, namely its ability to generate results that are valued (according to a purpose known to it) yet untypical[1]. It is a basic argument of this book that this tendency to diverge is fundamental for creative behavior.

- **Multitude of applications**. Divago was tested with a multitude of applications. If not useful for the applications themselves, for they were more hypothetical situations for testing the system than actually directed to specific problems, they can be used for comparison later with similar systems and as a starting point for more specific applications of Divago. More importantly, they allowed an observation of the system within different situations and the analysis of its evolution and behavior. They are also proposals for situations where bisociation can become important and computationally applicable: the creation of

[1]This being, of course, much depending on the situation, configuration and knowledge available.

new concepts for another system (a drawing system in the house-boat; a game system in the creature generation); and the study of conceptual combination (the noun-noun experiments and the established blends).

- **Creativity assessment**. The issue of evaluation is one of the fundamental problems in the study of creativity. Throughout this book, it has been a primary concern and, without promising the holy grail of universal formulae of usefulness or novelty, we propose some ideas for the assessment of creativity in computational systems, and for systems like Divago in particular. The main pillars for defining the criteria used in the creativity assessment of Divago were the works of Ritchie [Ritchie, 2001], Wiggins [Wiggins, 2001, Wiggins, 2003] and Colton et al [Colton et al., 2001]. We characterized the Model of Concept Invention with the more abstract and generic perspective of Wiggins, and analyzed Divago with the more concrete perspective of Ritchie and Colton et al. The latter led to a precise definition of typicality, which consists of the distance to the inputs (what is known), and of value, which is defined by how much the system accomplishes a goal[2]. As these analyses are rare within the field of Creative Systems, we believe that this work contributes to the evolution of the field in general and to the problem of assessment in particular.

There is an extensive set of future directions that this research can pursue, either by us or others and here we list a few that seem to be fundamental:

- **Other processes of invention** other than bisociation. In this book, we have focused almost exclusively on bisociation as a method for concept invention. However, other methods may also apply, such as concept re-representation or interaction with the environment, to name only two candidates. There is no reason to expect that these methods would have to be considerably different or antagonistic to the one presented here, therefore allowing other alternatives

[2]The notion of goal here does not imply a thorough definition, for it can be only partially defined. For example, a thoroughly defined goal can be "draw me a white house, with two windows, a door and a roof", while a less defined can be "draw me a construction where one can live in".

which could result in the extension of the Model of Concept Invention with new modules or further exploration of already existing ones.

- **Evolutions to the Blending model**. The computational model of Blending presented here should also be subject to further developments, namely the redesign of the optimality principles, possibly reduced to a smaller set, the inclusion of the latest changes, namely the vital relations and the compressions. The big leap for this model would be to cope with more than one input space as well as allow a more realistic knowledge base, which would have to be extremely large and organized.

- **Evolutions to Divago**. As this was the main practical part of the work, it was subject to many compromises, namely paths that we did not follow for pragmatic and research direction reasons. These paths are deserving of particular attention in the future:

  - **Real multi-domain environment**. Divago is still not able to work in a real multi-domain environment since the choice of the pairs of concepts to bisociate is either made randomly or externally. In a multi-domain environment, it should, when facing a problem, make an inspection of all its knowledge, which would comprise many different domains and knowledge representations, and would be able to pick for itself the sources for concept invention. Possible algorithms for developing this capability could come from works on analogy retrieval, where, before establishing an analogy between two concepts, the system searches for candidates in the knowledge base (e.g. MAC/FAC, from [Gentner and Forbus, 1995], or ARCS, from [Holyoak and Thagard, 1997]).
  - **Meta-level reasoning**. Being able to do meta-level reasoning would be a giant leap for Divago. It would then reason about its own knowledge and processes, potentially evolving them. A possible inspiration for enabling meta-level reasoning in Divago could be Simon Colton's HR [Colton et al., 1999], which is able to generate theories about its theories,

and theories about its own rules. Analogously, Divago would bisociate its own internal rules, such as the optimality constraints or the blending projections or, a more realistic situation, create new frames by bisociating existing ones. All this seems extremely complex and demanding of a serious research effort.

- **Interaction with the environment**. It was argued in the beginning of this book that environment is important for creativity. For some theories (e.g. the *Systems model* of Csikszentmihalyi [Csikszentmihalyi, 1996]), it is even a necessary condition for the existence of creativity. In this sense, Divago is rather autistic and clearly demands more contact with the environment. Integrating Divago within a multi-agent society environment seems an interesting project to develop, possibly a *hybrid society*, as suggested by [Pazos et al., 2002]).

- **Elaboration**. The Elaboration phase in concept invention is of great importance. It is there that part of the emergent structure of a new concept is constructed. However, the processes by which concepts are elaborated vary a lot depending on the situation. While it is generally agreed that part of the emergent features come from rule-based elaboration, i.e. accomplished by straightforward reasoning about a situation (e.g. a "beach bicycle" must have "large tyres"), other features seem not to have straightforward explanations (e.g. why does "Dracula" hate garlic?). Alternative processes must be sought for the elaboration, for this may have a great effect on the creativity of the results. We suggest that a possible contribution could be the use of other knowledge from the knowledge base (other than the inputs or the generic domain), for example, by searching for similar concepts and bringing new knowledge from them (e.g. when blending "horse" and "bird", the result may become similar to "dragon" and get new knowledge, such as "spitting fire").

In conclusion, the area of Computational Creativity has been growing in the past

few years and is clearly in its early stages. The current need for stronger and consensual definitions is notable, as well as evaluation methodologies, and benchmarks for comparisons. Its relationship with AI and other sciences must be established, if it is to gain its own place and flourish to fulfill its promises. In this sense, our work is only one step in that direction.

Appendix A

The effect of input knowledge

In chapter 6, we make some calculations to estimate the effect of input knowledge in the creativity of Divago. In this appendix, we reproduce the theoretical basis behind those calculations. Simon Colton, Alison Pease and Graeme Ritchie [Colton et al., 2001] propose a set of criteria for evaluating creativity, now with special attention to the effect of input knowledge.

One of the main problems in evaluating computational creativity (and of AI systems in general) relates to the extent to which the system's knowledge is *fine-tuned*, i.e. the system essentially replicates known items to a greater extent than it causes the generation of novel high-valued items [Colton et al., 2001].

Let O_K be the set of output items corresponding to input knowledge K. We define V_K as the set of *high-valued items* in O_K; R_K are the *reinventions* (the items that belong to the inspiring set I); and C_K is the *creative set* (the items in V_K which were not in R_K).

In order to determine the effect of a subset K' of K (the input knowledge), let us first examine the possible effects on $V_{(K-K')}$:

- K' is *creatively irrelevant* if $V_K = V_{(K-K')}$.
- K' is *creatively useful* if $V_{(K-K')} \subset V_K$.

- K' is *creatively destructive* if $V_K \subset V_{(K-K')}$.

For the creatively useful K', Colton et al. define the *dependency set* D'_K, such that $D'_K = V_K - V_{(K-K')}$, which corresponds to the set of items that will not be obtained if we remove K' from input knowledge. We can now say that K' is *fine-tuned* if [Colton et al., 2001]:

$$|D'_K \cap R_K| > 0 \text{ and } |D'_K \cap C_K| = 0$$

This means that the presence of K' in input knowledge only contributes to the replication of high-valued items, without having influence in the production of creative outcomes. For cases where K' still contributes to creativity (i.e. $|D'_K \cap C_K| > 0$), we can obtain a measure of how fine-tuned K' is:

$$ft(K') = \frac{|D'_K \cap R_K|}{|D'_K \cap C_K|}$$

Naturally, when $ft(K')$ is greater than 1, it means that K' is used to rediscover more items than to generate new ones of value. In order to determine whether a program P is fine-tuned when using knowledge K, Colton et al. propose the following two measures (assuming P was constructed using inspiring set I):

- $m_1(P, I, K) = \frac{|K_{ft}|}{|K|}$, where $K_{ft} = \bigcup_{K' \subset K : K' \text{ is fine-tuned}} K'$
- $m_2(P, I, K) = max(ft(K'))$ over $K' \subset K$

If m_1 is greater than 0 or m_2 greater than 1, we can claim that P using K has been fine-tuned to some extent. If m_1 is 1, P using K is completely fine-tuned. If m_2 is greater than 1, then there is at least one such subset of K which is used more to replicate known artifacts than to find new ones [Colton et al., 2001].

Some of the measures presented in this section were applied to the work presented in this book. Maybe due to being quite recent and still demanding refinements of many sorts, these measures have not been applied in practical computational systems,

with the exception of the analysis of Pablo Gervás to his poetry generation system, WASP [Gervás, 2002], who applied Ritchie's criteria, and of Colton's HR fine tuning analysis [Colton et al., 2001].

Appendix B

Established examples of Conceptual Blending

In this appendix, we will show the established blending examples that are used in chapter 5. In chapter 3, we have already described two examples ("Riddle of the Buddhist Monk" and "computer virus"). As with those, we reproduce the diagrams, tables and explanations as close as possible to the original ones.

The "CEO boxing fight" example has two input spaces with different organizing frames [Fauconnier and Turner, 2002, p. 128]. It is a metaphoric scenario that conceptualizes business competition. According to this metaphor, we can say that "one CEO has landed a blow but the other one has recovered", "one of them knocked the other out cold", etc. In other words, it is the structuring of the business domain according to the boxing domain. In figure B.1, we show the corresponding network as proposed by F&T. Since only one input space determines the organizing frame of the blend, this is a single scope blend.

The trashcan basketball example (in figure B.2) refers to the "game" one imagines to play when throwing papers at the wastepaper basket [Coulson, 2000, p. 118]. This involves the integration of the domains of Basketball (imagination) and trash disposal (reality). The emergent structure arises from affordances in the environment. In

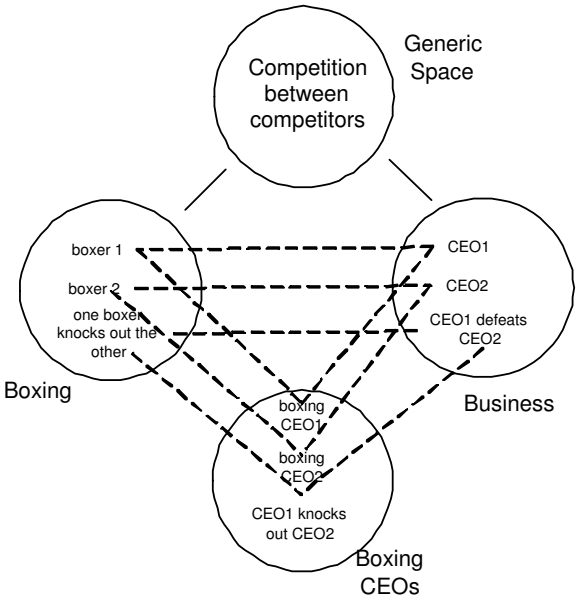

Figure B.1: The blending diagram of CEO fight.

trashcan basketball, some elements are inherited from trash disposal domains ("trash-can", "crumpled-paper") while others from the basketball domain ("shoot(person, paper, trashcan)"). Since the structure of the game comes from both domains (the rules of basketball, the affordances of the room), this is considered a double-scope blend.

The next example is a nominal compound that leads to a permanent category change. More precisely, the notion of "same-sex marriage". One input space is the traditional scenario of marriage, while the other describes an alternative domestic scenario involving two people of the same sex [Fauconnier and Turner, 2002, p. 271]. The cross-space mappings may link typical elements such as partners, common dwellings, commitment, love, sex. Selective projection then pulls to the blend social recognition, wedding ceremonies and mode of taxation from the traditional marriage input, while same sex, absence of biologically common children and culturally defined roles of the partners are projected from the other input (see figure B.3). Thus, this is also a double-scope blend.

Another very classic example is known as the "Debate with Kant" (see figure B.4) . It is about the following monologue (more precisely, an imagined dialogue)

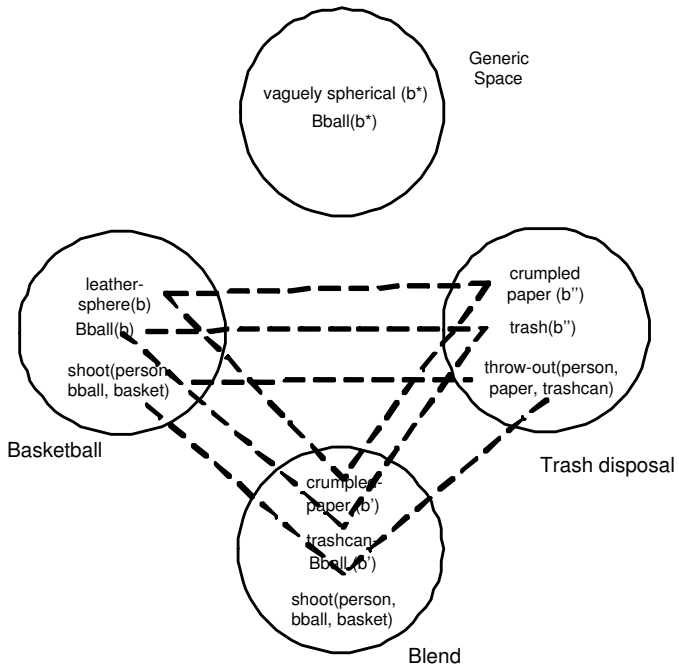

Figure B.2: The blending diagrams of Trashcan Basketball.

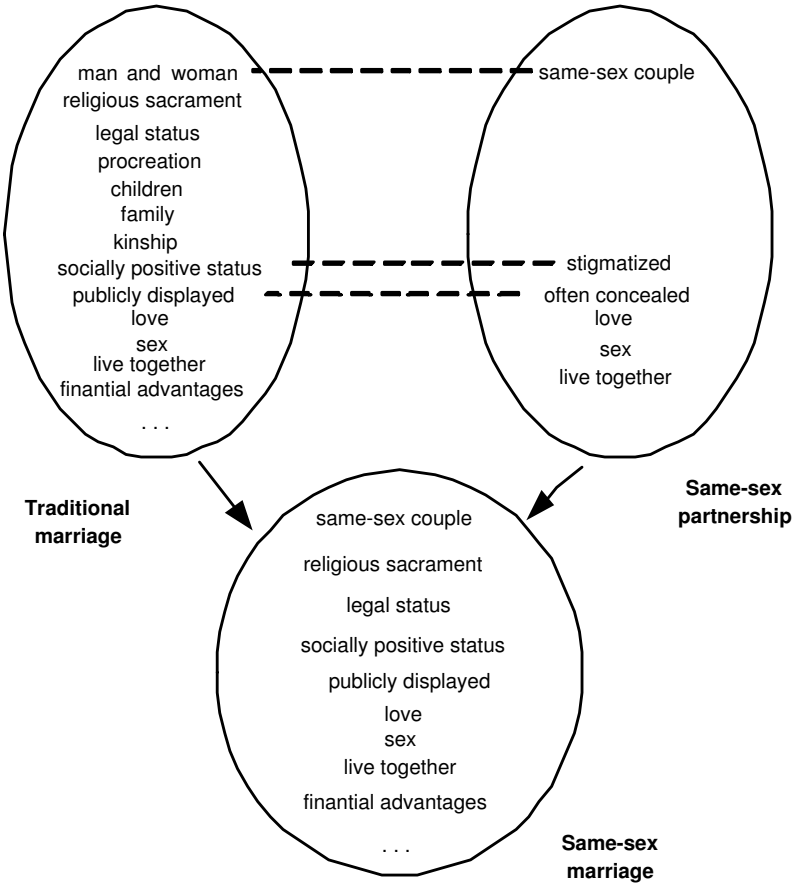

Figure B.3: The blending diagram of Same-sex marriage.

[Fauconnier and Turner, 2002, p. 62]:

> I claim that reason is a self-developing capacity. Kant disagrees with me on this point. He says it's innate, but I answer that that's begging the question, to which he counters, in *Critique of Pure Reason*, that only innate ideas have power. But I say to that, What about neuronal group selection? And he gives no answer.

In one input space, we have the modern philosopher (m) making claims, aware of Kant (k_2) - the eighteenth century philosopher. In a separate input space, we have Kant (k_1) - the living philosopher -, thinking and writing. The blended space has both people and the "debate" frame has been recruited since there is no debate in either input. The debate frame comes to the blend through pattern completion, since so much of its structure is already in place as a result of composition (i.e. many of the elements and relations of the debate frame were already in the blend before it was recruited). Once the blend is established, we can "run the blend", in this case, this is done by instantiating the debate frame with arguments from both input spaces.

In the next examples ("gun wound", "pet fish" and "land yacht"), Seana Coulson approaches a subject that is typical of Conceptual Combination: noun-noun compounds. She thus proposes applying the Conceptual Blending to explain the (conventional) meanings of each of the compounds. A "gun wound" is a "wound" (directly or indirectly) caused by a "gun". The strategy followed to deconstruct this compound is by recruiting the action frames associated with each domain. The generic space contains the generic ViolentAct frame with a cause and a result, while the blend contains the more specific Shoot frame. The input spaces bring the cause (the "gun" domain) and the effect (the "wound") [Coulson, 2000, p. 130] (see table B.1).

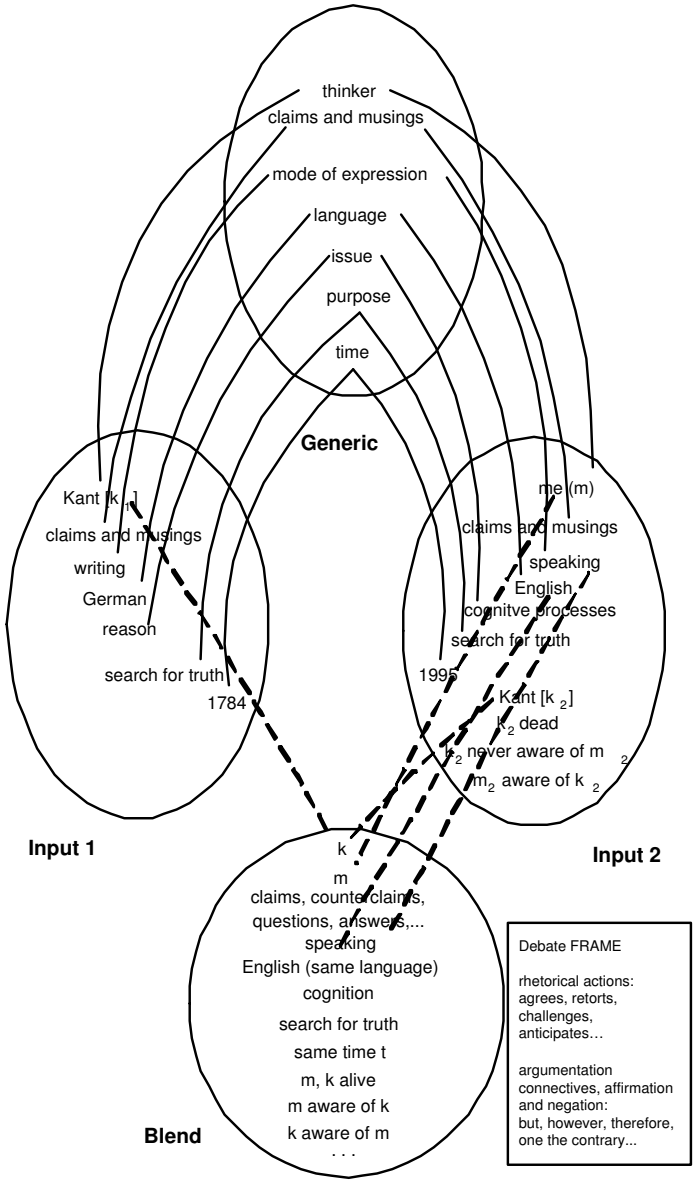

Figure B.4: The blending diagram of Kant Debate.

Input1	**Input2**	**Blend**	**Generic**
Gun	Wound	GunWound	ViolentAct
Elements	**Elements**	**Elements**	**Elements**
Agent	DangerousAct	Agent	ViolentAct
Target	Human	Human	Patient
Gun	Means	Gun	Means
Result	Wound	GunWound	Damage
Relations	**Relations**	**Relations**	**Relations**
Shoot(Agent, Gun, Target)	Cause(DangerousAct, Means, Human)	Shoot(Agent, Gun, Human)	Cause(ViolentAct, Means, Patient)
Result(Result)	Result(Wound)	Result(GunWound)	Result(Damage)

Table B.1: Gun Wound mappings.

Input1	**Input2**	**Blend**
Elements	**Elements**	**Elements**
Pet	Fish	PetFish
Owner	Water	Owner
House		House
		Tank
Relations	**Relations**	**Relations**
Feeds(Owner, Pet)	Lives-in(Fish, Water)	Feeds(Owner, Fish)
Loves(Owner, Pet)		
	Swims(Fish)	Swims(Fish)
		Lives-in(Fish, Tank)

Table B.2: Pet Fish mappings.

"Pet fish" is a blend in which the two inputs ("Pet" and "Fish") are counterparts and map onto the same element in the blended space, i.e. they fuse into the same concept. As in many examples already given, the blend inherits structure from both input spaces. Knowledge having to do with "pet ownership" are inherited from the "Pet" domain, while fish attributes come from the "Fish" domain [Coulson, 2000, p. 143] (see table B.2).

The "Land Yacht" compound demands more subtle reasoning. A land yacht is a very high class luxury car, it inherits the central properties of car and the diagnostic properties of yacht (in relation to other sailboats, a yacht is a luxury boat, extremely expensive and providing high social status to the owner). Therefore, the projection from inputs is more unbalanced than was the case with "Pet Fish", where the central properties of each were projected to the blend. Here, incongruities would arise if so happened (e.g. a car cannot sail). Here goes the cross-space mappings as Coulson presents them [Coulson, 2000, p. 155] (see table B.3).

The example of the "computer desktop" comes from Tim Rohrer, who is interested in the relationship of Metaphor with information technologies [Rohrer, 2000]. The computer desktop interface comes as a metaphorical projection of a physical desktop in an office, with folders, storages, waste basket, documents and the respective actions (moving physical objects from different places, opening folders), to computer data

Input1	**Input2**	**Blend**
Land	Yacht	Land Yacht
Elements	**Elements**	**Elements**
land	water	land
driver	skipper	driver
road	course	highway
car	yacht	luxury car
owner	tycoon	rich owner
Relations	**Relations**	**Relations**
Drives(driver, car, road)	Sails(skipper, yacht, course)	Drives(driver, car, highway)
	Yacht	Luxury Car
	Function: sails	Function: drives
	Sign-of: upper-class	Sign-of: wealth
	Owner: tycoon	Owner: rich person

Table B.3: Land Yacht mappings.

management representation (directories, files) and physical objects (screen, drive). Table B.4 is, therefore, not descriptive of the blending process. It basically shows the direct correspondences between the desktop with the blend, leaving implicit the computer domain elements.

Counterfactuals are also a recursive theme in Blending literature ([Lee and Barnden, 2001]). Counterfactuals are statements about the consequences of things that happen to be false (e.g. "If I were you..."). We present one of Seana Coulson's counterfactual examples: "If I had bread, I could make a sandwich". The inputs are the Actual space (Seana has turkey, cheese and mustard in the fridge), and a Sandwich space, in which there is bread, condiments, meat and cheese [Coulson, 2000, p. 206] (see table B.5).

According to the author, this is a single-scope blend because the organizing structure of the counterfactual comes from the Sandwich space, i.e., it states the individual roles of each element.

The final example we show is also an established theme in blending literature: the "X is the Y of Z" constructions. In this case, it is instantiated as "Mabel is the daughter of Paul"[Coulson, 2000, p. 119]. It is also a single scope blend, since structure comes only from input 2 (see table B.6).

Desktop (input 1 - source domain)		Human-Computer Interface (blend - target domain)
Desktop	→	Screen
Documents	→	Files
Folders	→	Directories
Storage	→	Drive icons
Moving physical objects	→	Dragging icons
Putting physical objects down	→	Dropping icons
Deleting objects	→	Dropping icons in trash (recycle bin)
Focusing on a task	→	'Zooming in', opening window
Putting away a task	→	'Zooming out', closing window

Table B.4: Computer Desktop mappings.

Input1	**Input2**	**Blend**
Actual	**Sandwich**	**Counterfactual**
Seana	Agent	Seana'
Fridge	Fridge"	Fridge'
Turkey	Turkey"	Turkey'
Cheese	Cheese"	Cheese'
Mustard	Mustard"	Mustard'
	Bread"	Bread'

Table B.5: Sandwich counterfactual mappings

Input1	**Input2**	**Blend**
Elements	**Elements**	**Elements**
Mabel	Daughter	Mabel
Paul	Father	Paul
	Relations	**Relations**
	Daughter-of(Daughter, Father)	Daughter-of(Mabel, Paul)

Table B.6: "Mabel is the daughter of Paul" mappings

Appendix C

The Generalized Upper Model Hierarchy

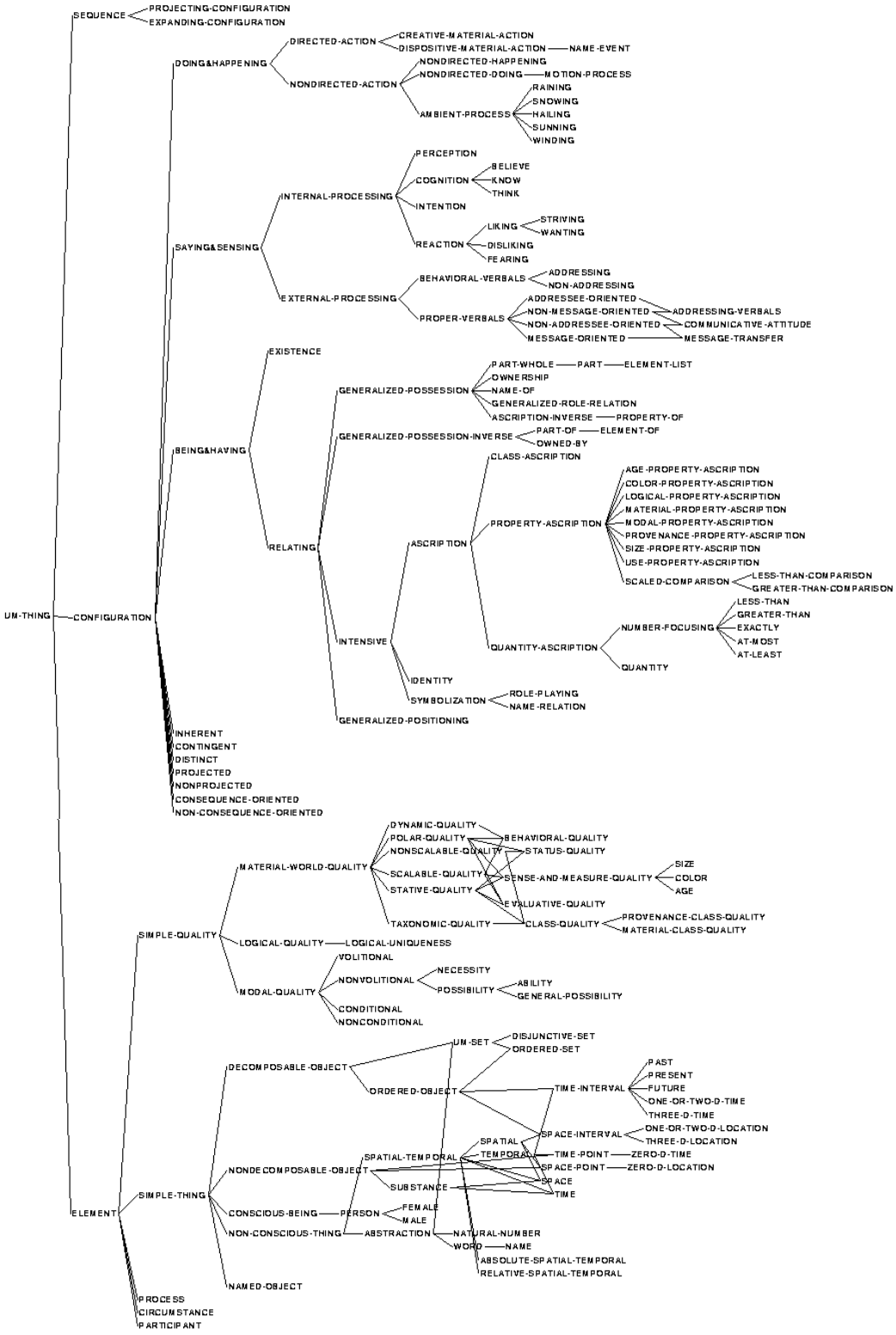

Figure C.1: The Concept Hierarchy

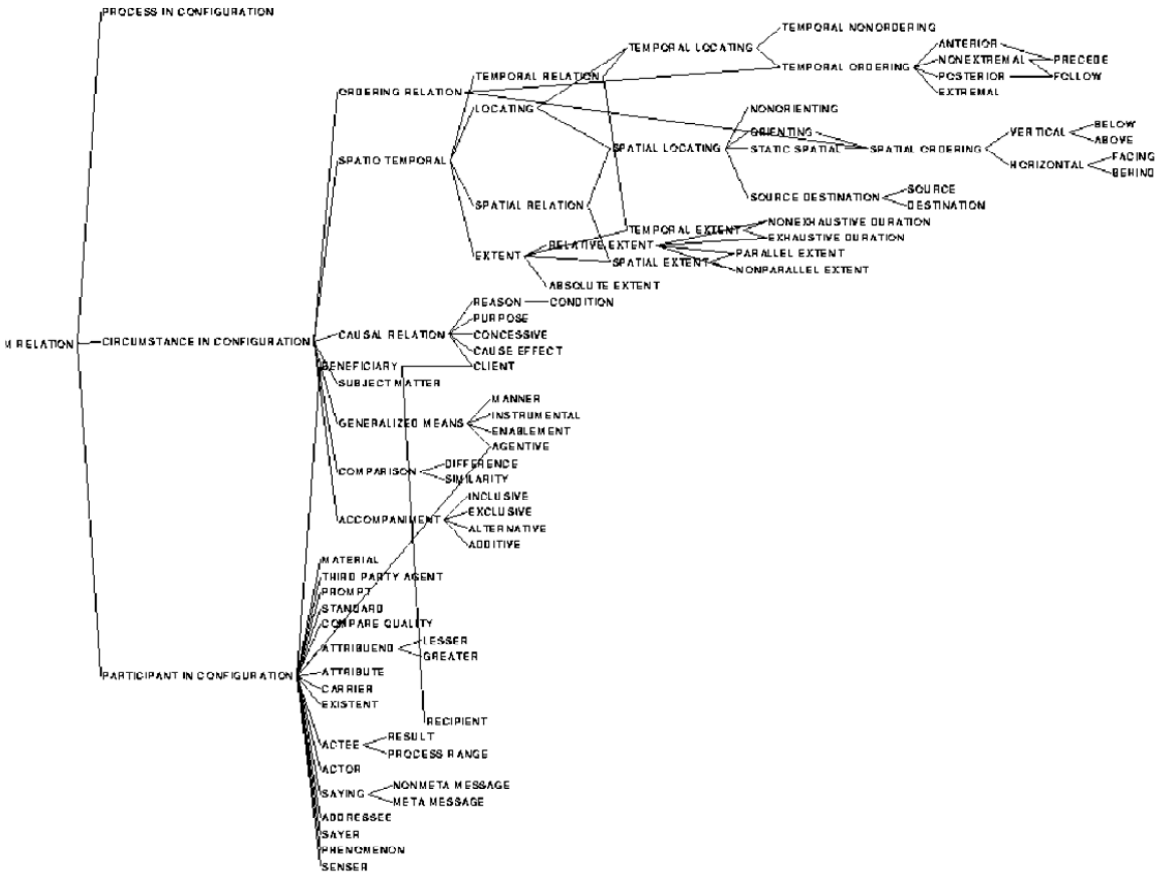

Figure C.2: The Relations Hierarchy

Appendix D

Programming the frames

D.1 Syntax and Overview

The syntax of every frame is:

```
Predicate:  frame(Domain, Name, PosConds, NegConds, PosConc, NegConc)
Domain      identifier of the domain or concept in which
            the frame is included.  (e.g.  generic, eating)
Name        name of the frame (e.g.  aprojection, flying_thing)
PosConds    Positive premisses (e.g.  have(X, wings))
NegConds    Negative premisses (e.g.  weight(X, very_heavy))
PosConc     Positive conclusions (e.g.  ability(X, fly))
NegConc     Negative conclusions (e.g.  habitat(X, water))
```

In first order logic, the frame can be represented as:

PosConc or not (NegConc) ⟵ PosConds and not(NegConds)

For example (prolog like):

ability(X, fly); not (habitat(X, water)) :- have(X, wings), not(weight(X, very_heavy)).

Applied to the blend, this would mean that, if X has wings and it is not very heavy, then it has the ability to fly and its habitat should not be water. If we used the above frame as a query to the blend (measured within the "relevance" optimality

constraint), it would tend to evolve towards a blend that has the concept wings projected (and so the relation has(_,wings) as a consequence) while the very_heavy concept is not projected (it can be projected, but it must be in such a way that weight(_, very_heavy) is not)[1].

Later, in the "elaboration phase", the conclusions are triggered, so the predicate ability(X, fly) is added, while the predicate habitat(X,water) is removed (if it exists).

D.2 Programming of the frame

Four kinds of terms are allowed in any of the frame conditions (or conclusions) part:

1. Regular concept map binary predicates, like "sound(X, neigh)" or "purpose(Y, fly)". There are also special variations in which we can give multiple options for a predicate parameter, specified inside a list. For example, "sound(X, [neigh, bark, chirp])" means that if X neighs, barks or chirps, then the condition is satisfied. A specific binary predicate, isaN, is also considered, being the transitive closure for isa (e.g. "isaN(human, animal)" because "isa(human, primate)" and "isa(primate, mammal)" and "isa(mammal, animal)").

2. Projection specifications, with the syntax projection(Domain, Origin, Destination). And the meaning of this condition is that concept Origin, which belongs to the domain Domain, should be projected into concept Destination, in the blend. For example, if we had the condition "projection(horse, neigh, neigh)" in a frame, it means we are requiring the concept "neigh" to be kept in the blend as it is in the (input space) horse domain. Of course, this doesn't mean the neighbour concepts also keep their original "names" (e.g. it is possible to find "produce(beak, neigh)", instead of "produce(mouth, neigh)".

[1]Notice that, in the condition side we consider "negation by failure", i.e. something is *not* true whenever one cannot prove its truth. For example, if there is not the predicate $weight(fly, very\_heavy)$, the we may assume its falsity. On the other hand, on the conclusion side, negations imply *deletion*, i.e. something will cease to be true if the rule is triggered.

3. Special operators, with the syntax op(Operator), where Operator corresponds to any command the frame interpreter should understand. Currently, we have only one single operator: exists(List). This operator converts the elements of List into the binary predicates and projection format. For example, op(exists([sound/X/neigh, purpose/Y/fly, projection/horse/neigh/neigh])) will just convert the list into "sound(X, neigh), purpose(X, fly), etc" and add it to the list of conditions of the frame.

4. Prolog calls, inside curly brackets ("{}"), just as in DCG grammars syntax. This allows the programming of frames in "regular" prolog. Naturally, some specific predicates have been created, for situations that happen regularly in frame programming:

    - stats(D,X) yields some statistics of the current blending operation (e.g. stats(domain1, X) returns the identifier of domain 1; stats(frame, f) means that the frame f is satisfied in the blend)
    - current_blend(Blend) Blend is the identifier of the blend being created
    - m(R, X, Y) Returns the mapping correspondences according to the vital relation R (e.g. m(analogy, horse, bird) means there is mapping between horse and bird, according to analogy)
    - rel(D, X, R, Y) Direct access to the concept map of domain D (e.g. rel(horse, legs, quantity, 4))
    - projection(B,D,X,Y) Direct access to the projection predicates, where B is the blend in which X, from domain D is projected into Y.
    - other_input_domain(D1,D2) Given D1 or D2, instantiates the other with the "opposite" input domain (e.g. other_input_domain(bird,X) ' X= horse)
    - relationArc(Domain, Action) True if Action has a action/actee configuration in Domain (e.g. relationArc(eating, eating), relationArc(basketball, shoot)"

- descendant(R, GUMConcept) R is descendant of GUMConcept in the GUM hierarchy (e.g. descendant(snowing, ambient_process))

D.3 Examples

We show some example frames, organized according to the level of abstraction. Simple frames only use "regular" binary predicates, while intermediate frames already apply predicates in brackets, still connecting to "lower level" reasonings. The abstract frames deal with the reasoning behind the blend construction (e.g. "project concepts from one domain, while maintaining the structure of the other").

Simple frames

Name:`haunted`

Code: `frame(generic, haunted,`
`[contain(X, Y), cause_effect(Y, fear), attribute(X, [magic, unknown])],`
`[],`
`[property(haunted,X), cause_effect(haunted, interesting)],`
`[]).`

Description: Something is haunted if it contains something that causes fear and is magic or mysterious

Name: `artefact`

Code: `frame(generic, artefact(X),`
`[isaN(X, physical_object), purpose(X, Y), isaN(Y, task)],`
`[],[],[]).`

Description: X is an artefact if it is a physical object whose purpose is a specific task

Name: `habitat_earth`

Code: `frame(generic, habitat_earth(X),`
`[place(X, [land, earth, ground, solid])],`
`[],[],[]).`

Description: The habitat of X is earth if its place is in either land, earth, ground or solid

Name:`habitat_water`

Code:`frame(generic, habitat_water(X),`
`[place(X, [sea, ocean, water, liquid])],`
`[], [], []).`

Description:The habitat of X is water if its place is in either sea, ocean, water or liquid

Intermediate frames

Name: `habitat_water`

Code: `frame(generic, amphibious(X), [stats(frame, habitat_water(X)), stats(frame, habitat_earth(X))],[],[isa(X, amphibious)],[]).`

Description:X is an amphibious if it satisfies frames habitat_water(X) and habitat_earth(X)

Name: `new_ability`

Code: `frame(generic, new_ability(D1),`
`[ability(X,A), purpose(P, A), pw(P,X),`
`{current_blend(Blend), projection(Blend, D1, X, X),`
`other_input_domain(D1,D2), projection(Blend, D2, A, A)},`
`op(exists([projection/D1/X/X, projection/D2/A/A]))],`
`[{rel(D1, X, ability, A)}],`
`[new_ability(X, A)], []).`

Description: A concept projected from domain D1 has a "new ability" in the

blend. I.e. there is a X (projected from D1) that has an ability A that didn't exist in D1 and was projected from D2. In order to be a "well founded" ability, X must have a subpart (P) that enables it to do A.

Name: `quality_transfer`

Code:
```
frame(generic, quality_transfer(D1,Q),
[{current_blend(Blend), rel(Blend, X, Q,A), descendant(Q, simple_quality)},
{projection(Blend, D1, X, X), other_input_domain(D1,D2),
projection(Blend, D2, A, A)},
op(exists([projection/D1/X/X, projection/D2/A/A]))],
[{rel(D1, X, Q, A)}],
[new_quality(X, A)], []).
```

Description: There is a quality transferred from domain D2 onto D1. In the blend, there is a concept X that has a relational quality (i.e. a relation that descends from the simple_quality node in GUM hierarchy) A that didn't exist in the original space.

Name: `living_thing_personificationA`

Code:
```
frame(generic, living_thing_personificationA,
[actor(_,A), {current_blend(Blend), projection(Blend, D, A, A),
rel(D, A, isa, Type)}],
[isaN(Type, living_entity)],
[personification(A, living_thing)],[]).
```

Description: A is able to be an actor of some action, and this becomes a personification of a living thing if A is not a living thing (e.g. a "actor(eating, pencil)" ' we are doing a personification of the pencil, which is not a living thing)

Name: `living_thing_personificationB`

Code:
```
frame(generic, living_thing_personificationB,
[ability(A,_), {current_blend(Blend), projection(Blend, D, A, A),
rel(D, A, isa, Type)}],
```

```
[isaN(Type, living_entity)],[personification(A, living_thing)],[]).
```

Description: The same as above, but A is not expected to be an actor, but instead it should have an ability (e.g. "ability(book, fly)")

Abstract frames

Name: `aprojection`

Code:
```
frame(generic, aprojection(A),
[{stats(domain1, A), current_blend(Blend),
findall(projection/A/X/X, (projection(Blend,A,X,_)), L1)},op(exists(L1))],
[],
[aprojection(A,Blend)],[]).
```

Description: Every concept from domain 1 (A) should be projected (unchanged) to the blend. For example, in "aprojection(horse)", every single concept of "horse" (legs, mouth, snout, mane, neigh, run, cargo, pet, etc.) should be present in the blend.

Name: `analogy_transfer`

Code:
```
frame(generic, analogy_transfer,
[{stats(domain1, A), stats(domain2, B),
findall(projection/A/X/Y, (m(_, X, Y), not(relationArc(A, X))), L1)},
op(exists(L1)),
{findall(projection/B/Y/Y, (m(_, _, Y), not(relationArc(B, Y))), L2)},
op(exists(L2))], [],
[analogy_transfer(B,A)],[]).
```

Description: Every mapped concept X that are part of domain A should be projected to their counterpart Y of domain B (except when X corresponds to an actor/actee action name, e.g. "eating" is projected to "eating" and not to "reading"). And every concept Y from domain B should also be projected to Y in the blend.

Name: `aframe`

Code: `frame(generic, aframe(A),`
`[{stats(domain1, A), current_blend(Blend),`
`findall(R/X/Y, (rel(A, XA, R, YA), projection(Blend, A, XA, X),`
`projection(Blend, A, YA, Y)),L1)}, op(exists(L1)),`
`{findall(projection/A/Action/Action, relationArc(A, Action), L2)},`
`op(exists(L2))], [],`

Description: Every relation R that is present in domain 1 (A), should also be present in the blend, regardless of the projection of the argument concepts (e.g. the ability relation in "ability(bird, fly)" should be present in the blend as in "ability(horse, fly)"). Once again, there is the special case of actor/actee relation descriptions, which should also be projected (e.g. if using aframe in blending "basketball and trash disposal", "shooting" should be projected to the blend, as well as the relations from the basketball domain.

Name: `noun_noun`

Code: `frame(generic, noun_noun(A,B),`
`[{stats(domain1, A), stats(domain2, B), L=[projection/A/A/A,`
`projection/B/B/B]}, op(exists(L)), isa(B, Something), projection(A,C,B),`
`{C\=B}, {(rel(generic, C, isa, Something); rel(A, C, isa, Something))}],`
`[], [], []).`

Description: A proposal of a noun-noun combination. The idea is that A and B have a relational connection such as "property_of", "lives_in", etc.

For the example "house dog", where A=house and B=dog, the frame implies that "house" and "dog" are projected to the blend exactly as they are (they remain the same concepts), but there is also a projection from the "house" domain to the "dog" (e.g. "projection(house, person, dog)"), which yields a dog that inhabits a house ' which is one of the interpretations for "house dog"

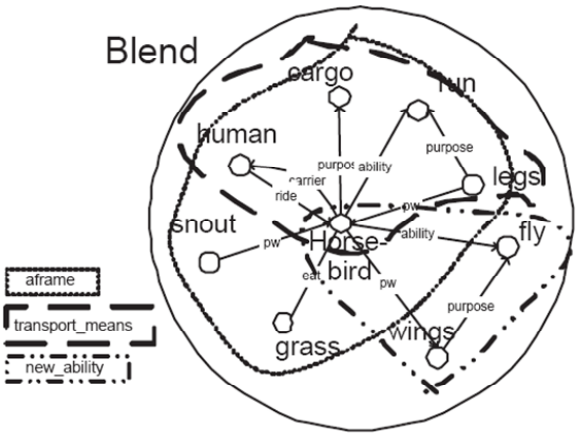

Figure D.1: Three frames in a small blend

In Figure D.1, we give an idea of the application of frames to a blend (to improve readability, both the frames and the concepts are simplified). We say that the blend accomplishes (or satisfies) "aframe", "transport_means" and "new_ability" and that its overall *frame coverage* is 100% (every relation is included in a frame). The coverage of "aframe" is aproximately 72% (the ratio of the blend that is covered by "aframe") and "transport_means" and "new_ability" have coverages of 54% and 27% (respectively).

Appendix E

Instances in the house-boat experiment

In this appendix, we describe the language used in the instances for the house-boat experiment.

An instance follows a hierarchical case representation, each node in the hierarchy written in prolog-like form:

$case(Instance\_Name, Node\_Address, Node\_Name, Command\_List)$.

with $Instance\_Name$ being the identifier of the instance; $Node\_Address$ a unique identifier of the node within the instance with respect to its position; $Node\_Name$ an identifier of the node within the instance (which can be repeated); and $Command\_List$ the list of commands that correspond to the semantics of the specific node. Therefore, this representation is structured top to bottom (the attribute "son" indicates the descendants of a node):

- each level adds a number to the address (e.g., 0 is the root node, 0:0 is the first son of the root node, 0:0:1 is the second son of 0:0)
- each node in the structure corresponds to an area of the drawing, containing one or more shapes.

- some shapes are pre-defined (e.g., parallelogram_boat, oval, rectangle, etc.) in Logo.
- each shape position is relative to a reference point ("in" indicates the commands to apply from the reference point to the starting point of the shape), normally the upper right corner of the smallest rectangle that can include the shape

The Logo keywords for defining shapes are:

- left/X. Rotate 45 degrees left
- right/X. Rotate 45 degrees right
- on/X. Move and write X pixels in the current direction
- off/X. Move without writing X pixels in the current direction

For example, the function that defines the shape $triangle(X)$ is defined by the list $[on/X, right/120, on/X, right/120, on/X, right/120]$. The used shapes are all defined in the file "logoCommands.pl". The representation for sailing boat goes like this:

```
case(b1,0,sailing_boat,[sons=3,size=small, type=simple,son_name=vessel, son_name=
    mast,son_name=sail]).
case(b1,0:0, vessel, [sons=2, in=[left/90,off/14,right/90],son_name=floating_struc-
    ture, son_name=hatch]).
case(b1,0:0:0, floating_structure, [shape=parallelogram_boat, size=small]).
case(b1,0:0:1, hatch, [shape=oval(5,5), size=small, in=[off/25,right/90, off/6,
    left/90]]).
case(b1,0:1, mast, [shape=rectangle(4,30), type=very_thin, in=[off/18]]).
case(b1,0:2, sail, [shape=triangle(30), in=[off/18, right/90, off/7, right/90,
    off/13,right/180]]).
```

And the house is represented as:

```
case(1,0, house, [sons=2, size=small, type=simple, son_name=roof, son_name=body]).
case(1,0:0, roof, [shape=triangle(30)]).
```

```
case(1,0:1, body, [sons=3, in=[left/90,off/25, right/90],son_name=structure,
    son_name=window, son_name=door]).
case(1,0:1:0, structure, [shape=square]).
case(1,0:1:1, window, [shape=square(5), in=[off/20, right/90, off/15, left/90]]).
case(1,0:1:2, door, [shape=rectangle(4, 10), in=[off/3]]).
```

Appendix F

Experiments, Databases and other documents

This appendix corresponds to the CD that comes attached to this document. It has the directory structure shown in the figure F.1. The reader will also find "readme" files with explanations about some of the files and directories.

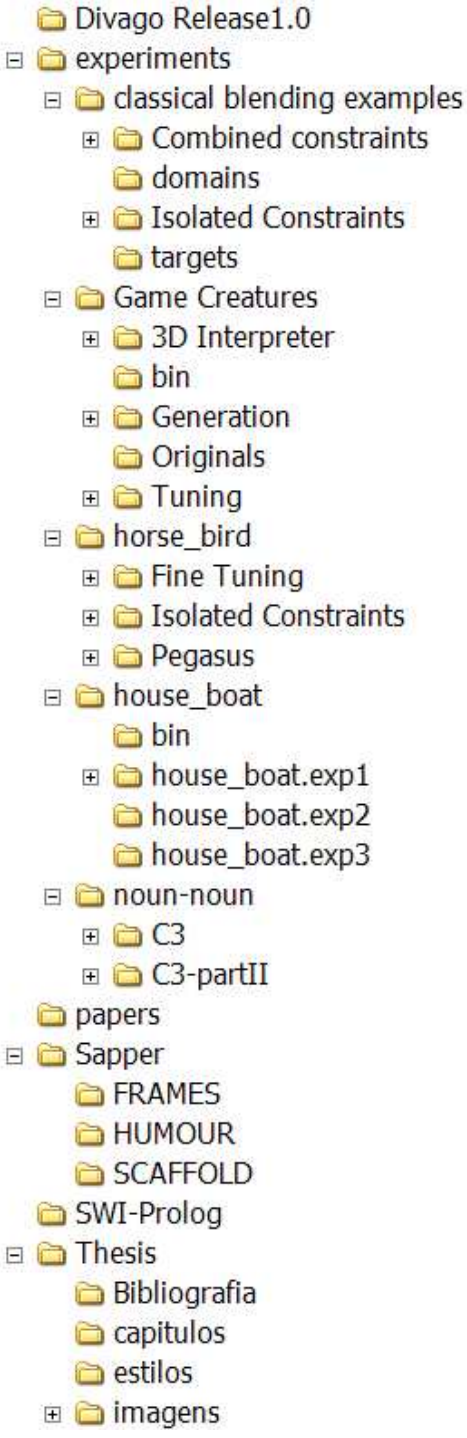

Figure F.1: The directory structure of the CD.

Bibliography

[Abelsson and diSessa, 1981] Abelsson, H. and diSessa, A. (1981). *Turtle Geometry.* MIT Press.

[Albert and Runco, 1999] Albert, R. and Runco, M. (1999). A History of Research on Creativity. In Sternberg, R., editor, *Handbook of Creativity*, pag. 16–35. Cambridge University Press.

[Amabile, 1983] Amabile, T. (1983). *The social psychology of creativity.* New York: Springer.

[Andersen, 1996] Andersen, C. F. (1996). *A Computational Model of Complex Concept Composition.* University of Texas at Austin.

[Barnden, 1998] Barnden, J. A. (1998). An AI system for metaphorical reasoning about mental states in discourse. In Koenig, J.-P., editor, *Discourse and Cognition: Bridging the Gap.* Cambridge University Press.

[Barnden, 1999] Barnden, J. A. (1999). An implemented system for metaphor-based reasoning, with special application to reasoning about agents. In Nehaniv, C., editor, *Computation for Metaphors, Analogy, and Agents, Lecture Notes in Artificial Intelligence*, volume 1562, pag. 143–153. Springer.

[Bateman et al., 1995] Bateman, J., Magnini, B., and Fabris, G. (1995). The Generalized Upper Model knowledge base: Organization and use. In *Towards very large knowledge bases: knowledge building and knowledge sharing.* IOS Press.

[Bentley, 1999] Bentley, P. (1999). *Evolutionary Design by Computers.* Morgan Kaufman.

[Binsted, 1996] Binsted, K. (1996). *Machine humour: An implemented model of puns.* PhD thesis, University of Edimburgh, Scotland.

[Boden, 1990] Boden, M. (1990). *The Creative Mind: Myths and Mechanisms.* London: Weidenfeld and Nicholson.

[Boden, 1999] Boden, M. (1999). Computational models of creativity. In Sternberg, R., editor, *Handbook of Creativity*, pag. 351–373. Cambridge University Press.

[Boring, 1923] Boring, E. G. (1923). Intelligence as the tests test it. *New Republic*, 34:35–37.

[Bringsjord and Ferrucci, 2000] Bringsjord, S. and Ferrucci, D. A. (2000). *Artificial Intelligence and Literary Creativity*. Mahwah, NJ: Erlbaum.

[Bulhak, 2000] Bulhak, A. C. (2000). *On the simulation of postmodernism and mental debility using recursive transition networks*. Dept Computer Science Technical Reports, Dept Computer Science, Monash University, Melbourne Australia.

[Campos and Figueiredo, 2001] Campos, J. and Figueiredo, A. D. (2001). Searching the Unsearchable: Inducing Serendipitous Insights. In Cardoso, A., Bento, C., and Wiggins, G., editors, *Proceedings of the First Workshop on Creative Systems, International Conference of Case-Based Reasoning*. ICCBR-01.

[Cardalda, 1999] Cardalda, J. J. R. (1999). *Metodología Evolutiva para la construcción de modelos cognitivos complejos. Exploración de la 'creatividad artificial' en composición musical*. Universdad de La Coruña.

[Chandler, 2002] Chandler, D. (2002). *Semiotics: The Basics*. Routledge.

[Cohen, 1981] Cohen, H. (1981). On the Modelling of Creative Behaviour. In *Rand paper*. Santa Monica, Calif: Rand Corporation.

[Colton, 2001] Colton, S. (2001). Experiments in Meta-theory formation. In Wiggins, G., editor, *Proceedings of the AISB'01 Symposium on AI and Creativity in Arts and Sciences*. AISB.

[Colton et al., 1999] Colton, S., Bundy, A., and Walsh, T. (1999). HR: Automatic concept formation in pure mathematics. In *Proceedings of the International Joint Conference on Artificial Intelligence*. IJCAI'99.

[Colton et al., 2000] Colton, S., Bundy, A., and Walsh, T. (2000). Agent Based Cooperative Theory Formation in Pure Mathematics. In *Proceedings of the Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science*. SSAISB.

[Colton et al., 2001] Colton, S., Pease, A., and Ritchie, G. (2001). The Effect of Input Knowledge on Creativity. In Cardoso, A., Bento, C., and Wiggins, G., editors, *Proceedings of the First Workshop on Creative Systems, International Conference of Case-Based Reasoning*. ICCBR-01.

[Conklin and Witten, 1995] Conklin, D. and Witten, I. (1995). Multiple viewpoint systems for music prediction. *Journal of New Music Reasearch*, 1(24).

[Cope, 1991] Cope, D. (1991). *Computers and Musical Style*. A-R Editions.

[Costello, 1997] Costello, F. J. (1997). *Noun-noun conceptual combination: the polysemy of compound phrases*. PhD thesis: Trinity College, Dublin.

[Costello and Keane, 2000] Costello, F. J. and Keane, M. T. (2000). Efficient Creativity: Constraint-Guided Conceptual Combination. *Cognitive Science*, 24(2):299–349.

[Coulson, 2000] Coulson, S. (2000). *Semantic Leaps: Frame-shifting and Conceptual Blending in Meaning Construction*. New York and Cambridge: Cambridge University Press.

[Cox, 1926] Cox, C. (1926). *Genetic studies of genius: The early mental traits of three hundred geniuses*, volume 2. Stanford University Press.

[Craft and Cross, 2003] Craft, A. and Cross, I. (2003). A n-gram approach to fugal exposition composition. In *AISB'03 Symposium on AI and Creativity in Arts and Science*. SSAISB.

[Csikszentmihalyi, 1996] Csikszentmihalyi, M. (1996). *Creativity: flow and psychology of discovery and invention*. HarperCollins.

[Eliasmith and Thagard, 2001] Eliasmith, C. and Thagard, P. (2001). Integrating structure and meaning: a distributed model of analogical mapping. *Cognitive Science*, 25:245–286.

[Ernst and Newell, 1969] Ernst, G. and Newell, A. (1969). *GPS: A Case Study in Generality and Problem Solving*. Academic Press, New York.

[Evans, 1968] Evans, T. (1968). *A heuristic program to solve geometric analogy problems*. PhD thesis, M.I.T., Cambridge, Mass.

[Falkenhainer et al., 1989] Falkenhainer, B., Forbus, K. D., and Gentner, D. (1989). The Structure Mapping Engine: Algorithm and Examples. *Artificial Intelligence*, 41:1–63.

[Fauconnier and Turner, 1998] Fauconnier, G. and Turner, M. (1998). Conceptual Integration Networks. *Cognitive Science*, 22(2):133–187.

[Fauconnier and Turner, 2002] Fauconnier, G. and Turner, M. (2002). *The Way We Think*. Basic Books.

[Ferguson et al., 1997] Ferguson, R., Forbus, K., and Gentner, D. (1997). On the proper treatment of noun-noun metaphor: A critique of the Sapper model. In *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, volume 913.

[Figueiredo and Campos, 2001] Figueiredo, A. D. and Campos, J. (2001). The Serendipity Equations. In Cardoso, A., Bento, C., and Wiggins, G., editors, *Proceedings of the First Workshop on Creative Systems, International Conference of Case-Based Reasoning*. ICCBR-01.

[Finke et al., 1992] Finke, R., Ward, T., and Smith, S. (1992). *Creative cognition: Theory, reasearch and applications*. Cambridge:MIT press.

[Forbus and Oblinger, 1990] Forbus, K. D. and Oblinger, D. (1990). Making SME pragmatic and greedy. In *Proceedings of the Twelfth Annual Meeting of the Cognitive Science Society*. Hillsdale, NJ: LEA.

[French, 2002] French, R. M. (2002). The Computational Modeling of Analogy-Making. *Trends in Cognitive Sciences*, 6(5):200–205.

[Gagné and Shoben, 1997] Gagné, C. and Shoben, E. (1997). Influence of thematic relations on the comprehension of modifier-noun combinations. *Journal of Experimental Psychology: Learning, Memory and Cognition*.

[Gentner, 1983] Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2).

[Gentner and Forbus, 1995] Gentner, D. and Forbus, K. (1995). MAC/FAC: a model of similarity-based retrieval. *Cognitive Science*, 19(2):141–205.

[Gervás, 2000a] Gervás, P. (2000a). An Expert System for the Composition of Formal Spanish Poetry. In *Proceedings of the Twentieth SGES International Conferences on Knowledge Based Systems and Applied Artificial Intelligence*. Peterhouse College, Cambridge, England.

[Gervás, 2000b] Gervás, P. (2000b). WASP: Evaluation of different strategies for the automatic generation of spanish verse. In *Proceedings of the AISB'00 Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science*. SSAISB.

[Gervás, 2002] Gervás, P. (2002). Exploring Qualitative Evaluations of the Creativity of Automatic Poets. In *Proceedings to the Second Workshop on Creative Systems*. European Conference of Artificial Intelligence, ECAI'02.

[Goguen, 1999] Goguen, J. (1999). An Introduction to Algebraic Semiotics, with Applications to User Interface Design. In *Lecture Notes in Artificial Intelligence*, volume Computation for Metaphor, Analogy and Agents. Springer.

[Goldberg, 1989] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.

[Gomes et al., 2002] Gomes, P., Pereira, F. C., Paiva, P., Seco, N., Carreiro, P., Ferreira, J. L., and Bento, C. (2002). Case retrieval of software designs using WordNet. In Hermelen, F. V., editor, *Proceedings of the European Conference of Artificial Intelligence (ECAI'02)*. IOSPress.

[Grady et al., 1999] Grady, J., Oakley, T., and Coulson, S. (1999). Blending and Metaphor. In Steen, G. and Gibbs, R., editors, *Metaphor in Cognitive Linguistics*.

[Guilford, 1950] Guilford, J. P. (1950). Creativity. *American Psychologist*, 5:444–454.

[Guilford, 1967] Guilford, J. P. (1967). *The nature of human intelligence*. New York: McGraw-Hill.

[Hampton, 1997] Hampton, J. (1997). Conceptual combination. In Lamberts and Shanks, editors, *Knowledge, Concepts and Categories*. Psychology Press.

[Hampton, 1987] Hampton, J. A. (1987). Inheritance of attributes in natural concept conjunctions. *Memory and Cognition*, 15:55–71.

[Helmholtz, 1896] Helmholtz, H. V. (1896). *Vortäge und Reden*. Brunswick, Germany: Friedrich Vieweg.

[Hofstadter, 1995] Hofstadter, D. (1995). *Fuild Concepts and Creative Analogies*. Basic Books.

[Hofstadter and Mitchell, 1988] Hofstadter, D. and Mitchell, M. (1988). Conceptual slippage and mapping: A report of the Copycat project. In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*. Hillsdale, New Jersey: Erlbaum.

[Holyoak and Thagard, 1989] Holyoak, K. and Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science*, 13:295–355.

[Holyoak and Thagard, 1997] Holyoak, K. and Thagard, P. (1997). The analogical mind. *American Psychologist*, 52(1):35–44.

[Hummel and Holyoak, 1997] Hummel, J. and Holyoak, K. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, 104:427–466.

[Johnson, 1987] Johnson, M. (1987). *The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason*. Chicago: University of Chicago Press.

[Karmiloff-Smith, 1993] Karmiloff-Smith, A. (1993). Constraints on representational change: evidence from children's drawing. *Cognition*, 34.

[Keane and Costello, 2001] Keane, M. T. and Costello, F. J. (2001). Setting limits on analogy: Why conceptual combination is not structural alignment. In Gentner, D., Holyoak, K., and Kokinov, B., editors, *The Analogical Mind: A Cognitive Science Perspective*. Cambridge, MASS: MIT Press.

[Koestler, 1964] Koestler, A. (1964). *The Act of Creation*. New York:Macmillan.

[Kris, 1952] Kris, E. (1952). *Psychoanalytic explorations in art*. New York: International Universities Press.

[Lakatos, 1978] Lakatos, I. (1978). *The Methodology of Scientific Research Programs*. Cambridge.

[Lakoff, 1987] Lakoff, G. (1987). *Women, fire, and dangerous things: What categories reveal about the mind*. Chicago: University of Chicago.

[Lakoff and Johnson, 1980] Lakoff, G. and Johnson, M. (1980). *Metaphors We Live By*. University of Chicago Press, Chicago.

[Lawson et al., 1989] Lawson, A., Abraham, M., and Renner, J. (1989). *A theory of instruction: Using the learning cycle to teach science concepts and thinking skills (Monograph)*. National Association for Research in Science Teaching.

[Lee and Barnden, 2001] Lee, M. and Barnden, J. (2001). Cognitively Plausible Models of Semantic Processing. In *Proceedings of SEMPRO-01*. Edinburgh.

[Leite, 2003] Leite, J. A. (2003). *Evolving Knowledge Bases: Specification and Semantics*. IOS Press.

[Lenat, 1984] Lenat, D. (1984). Why AM and EURISKO appear to work. *Artificial Intelligence Journal*, 23:269–294.

[Levy, 2001] Levy, R. (2001). A computational model of poetic creativity with neural network as measure of adaptive fitness. In Cardoso, A., Bento, C., and Wiggins,

G., editors, *Proceedings of the First Workshop on Creative Systems, International Conference of Case-Based Reasoning*. ICCBR-01.

[Liu and Singh, 2002] Liu, H. and Singh, P. (2002). MAKEBELIEVE: Using Commonsense Knowledge to Generate Stories. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI, 2002)*. AAAI Press.

[Lothe, 2000] Lothe, M. (2000). Knowledge-based composition of minuets by a computer. In *Proceedings of the AISB'00 Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science*. SSAISB.

[Lubart, 1999] Lubart, T. I. (1999). Creativity across cultures. In Sternberg, R., editor, *Handbook of Creativity*, pag. 339–351. Cambridge University Press.

[Macedo and Cardoso, 2001] Macedo, L. and Cardoso, A. (2001). Modelling Forms of Surprise in an Artificial Agent. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*. Mahwah, NJ: Erlbaum.

[Machado and Cardoso, 2002] Machado, F. and Cardoso, A. (2002). All the truth about NevAr. *Applied Intelligence, Special issue on Creative Systems*, 16(2).

[Mandelblit, 1997] Mandelblit, N. (1997). *Grammatical Blending: Creative and Schematic Aspects in Sentence Processing and Translation*. PhD thesis: UC San Diego.

[Manurung et al., 2000] Manurung, H. M., Ritchie, G., and Thompson, H. (2000). Towards a computational model of poetry generation. In *Proceedings of the AISB'00 Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science*. SSAISB.

[Marin et al., 1991] Marin, R., Torre, S. D. L., and Vive, V. (1991). *Manual de la Creatividad*. Barcelona.

[Marshall, 2002] Marshall, J. B. (2002). Metacat: A Program That Judges Creative Analogies in a Microworld. In *Proceedings to the Second Workshop on Creative Systems*. European Conference of Artificial Intelligence, ECAI'02.

[Martin, 1990] Martin, J. H. (1990). *A Computational Model of Metaphor Interpretation*. Academic Press.

[Martindale, 1999] Martindale, C. (1999). Biological Bases of Creativity. In Sternberg, R., editor, *Handbook of Creativity*, pag. 137–153. Cambridge University Press.

[Martindale and Greenough, 1974] Martindale, C. and Greenough, J. (1974). The differential effect of increased arousal on creative and intellectual performance. *Journal of Genetic Psychology*, 124:311–320.

[Medin and Schaffer, 1978] Medin, D. L. and Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review*, 85:207–238.

[Mendes, 2004] Mendes, M. (2004). *Rela cões Lexicais na Gera cão de Língua Natural, MSc Dissertation*. Departamento de Eng. Informática da Universidade de Coimbra.

[Milosavljevic and Dale, 1996] Milosavljevic, M. and Dale, R. (1996). Strategies for comparison in encyclopdia descriptions. In *In Proceedings of the 8th. International Workshop on Natural Language Generation*, pag. 161–170. INLG'96, Herstmonceux, England.

[Mitchell, 1978] Mitchell, T. (1978). *Version Spaces: An Approach to Concept Learning*. PhD thesis, Stanford University.

[Murphy and Medin, 1985] Murphy, G. L. and Medin, D. L. (1985). The role of theories in conceptual coherence. *Psychological Review*, 92.

[Neisser, 1976] Neisser, U. (1976). *Cognition and Reality*. San Francisco: Freeman.

[Oxman, 1997] Oxman, R. (1997). Design by re-representation: a model of visual reasoning in design. In *Design Studies*, volume 18, chapter 4. Elsevier.

[Pazos et al., 2002] Pazos, A., Santos, A., Arcay, B., Dorado, J., Romero, J., and Rodrï¿½uez., J. (2002). Application Framework to Build Computer Evolutionary Systems in Music. *LEONARDO (MIT Press)*, 35(4).

[Pease et al., 2001] Pease, A., Winterstein, D., and Colton, S. (2001). Evaluating Machine Creativity. In Cardoso, A., Bento, C., and Wiggins, G., editors, *Proceedings of the First Workshop on Creative Systems, International Conference of Case-Based Reasoning*. ICCBR-01.

[Peirce, 1958] Peirce, C. S. (1958). *Collected writings from 1931 to 1958*. Cambridge, MA: Harvard University Press.

[Pereira, 1998] Pereira, F. C. (1998). Modelling Divergent Production: a multi domain approach. In *European Conference of Artificial Intelligence, ECAI98*. IOS-Press.

[Pereira, 2000] Pereira, F. C. (2000). *Construiï¿½o Interactiva de Mapas Conceptuais (in Portuguese)*. MSc thesis, Universidade de Coimbra, Portugal.

[Pereira, 2003] Pereira, F. C. (2003). Experiments with Free Concept Generation in Divago. In Cardoso, A., Bento, C., and Gero, J., editors, *Proceedings of the Third Workshop on Creative Systems*. IJCAI-03.

[Pereira, 2005] Pereira, F. C. (2005). *Um Modelo Computacional de Criatividade*. PhD thesis, Universidade de Coimbra.

[Pereira and Cardoso, 1999] Pereira, F. C. and Cardoso, A. (1999). Wondering in a Multi-Domain Environment with Dr. Divago. In *Proceedings of the 8th Cognitive Science and Natural Language Processing Conference (CSNLP-8)*.

[Pereira and Cardoso, 2000] Pereira, F. C. and Cardoso, A. (2000). A Module for Automatic Learning of Concept Maps. In *Proceedings of Diagrams 2000*. Springer.

[Pereira and Cardoso, 2002] Pereira, F. C. and Cardoso, A. (2002). The Boat-House Visual Blending Experience. In *Proceedings to the Second Workshop on Creative Systems*. European Conference of Artificial Intelligence, ECAI'02.

[Pereira and Cardoso, 2003a] Pereira, F. C. and Cardoso, A. (2003a). The horse-bird creature generation experiment. *AISB Journal*, 1(3).

[Pereira and Cardoso, 2003b] Pereira, F. C. and Cardoso, A. (2003b). Optimality Principles for Conceptual Blending: A First Computational Approach. In *AISB'03 Symposium on AI and Creativity in Arts and Science*. SSAISB.

[Pereira and Cardoso, 2003c] Pereira, F. C. and Cardoso, A. (2003c). Optimality Principles for Conceptual Blending: A First Computational Approach. *AISB Journal*, 1(4).

[Pereira and Gervás, 2003] Pereira, F. C. and Gervás, P. (2003). Natural Language Generation from Concept Blends. In *AISB'03 Symposium on AI and Creativity in Arts and Science*. SSAISB.

[Pereira et al., 2005] Pereira, F. C., Mendes, M., Gervás, P., and Cardoso, A. (2005). Experiments With Assessment of Creative Systems: An Application of Ritchie's Criteria. In Gervás, P., Pease, A., and Veale, T., editors, *Proceedings of the IJCAI 2005 Computational Creativity Workshop*. University of Edinburgh.

[Pereira et al., 2000] Pereira, F. C., Oliveira, A., and Cardoso, A. (2000). Extacting Concept Maps with Clouds. In *Lecture Notes in Artificial Intelligence. Proceedings to the Argentine Symposium on Artificial Intelligence (ASAI'00)*. Springer.

[Plate, 1994] Plate, T. A. (1994). *Distributed representations and nested compositional structure*. Ph.D. Thesis, University of Toronto.

[Popper, 1959] Popper, K. (1959). *The Logic of Scientific Discovery*. Hutchinson & Co.

[Ram et al., 1995] Ram, A., Wills, L., Domeshek, E., Nersessian, N., and Kolodner, J. (1995). Understanding the Creative Mind. *Artificial Intelligence Journal*, 79:111–128.

[Reffat, 2002] Reffat, R. M. (2002). Intelligent Agents for Concept Invention of Design Forms. In *Proceedings of the International Workshop on Agents in Design*. WAID'02.

[Ribeiro et al., 2001] Ribeiro, P., Pereira, F. C., Ferrand, M., and Cardoso, A. (2001). Case-based Melody Generation with MuzaCazUza. In Wiggins, G., editor, *Proceedings of the AISB'01 Symposium on AI and Creativity in Arts and Sciences*. AISB.

[Ribeiro et al., 2003] Ribeiro, P., Pereira, F. C., Marques, B., ao, B. L., and Cardoso, A. (2003). A Model for Creativity in Creature Generation. In *Proceedings of the 4^{th} Conference on Games Development (GAME ON'03)*. EuroSIS / University of Wolverhampton.

[Ritchie, 2001] Ritchie, G. D. (2001). Assessing Creativity. In Wiggins, G., editor, *Proceedings of the AISB'01 Symposium on AI and Creativity in Arts and Sciences*. AISB.

[Rohrer, 2000] Rohrer, T. (2000). Even the interface is for sale: Metaphors, visual blends and the hidden ideology of the internet. In Dirven, R., Frank, R., and Ilie, C., editors, *Language and Ideology: Cognitive Descriptive Approaches*. Amsterdam and Philadelphia: John Benjamins.

[Rosch, 1975] Rosch, E. (1975). Cognitive representations os semantic categories. *Journal of Experimental Psychology: General*, 104:192–232.

[Saunders and Gero, 2001] Saunders, R. and Gero, J. (2001). The Digital Clockwork Muse: a Computational Model of Aesthetic Evolution. In Wiggins, G., editor, *Proceedings of the AISB'01 Symposium on AI and Creativity in Arts and Sciences*. AISB.

[Saussure, 1983] Saussure, F. D. (1983). *Course in General Linguistics*. London:Duckworth.

[Sims, 1991] Sims, K. (1991). Art Evolution for Computer Graphics. *ACM Computer Graphics*, 25.

[Sosa and Gero, 2003] Sosa, R. and Gero, J. (2003). Design and Change: A Model of Situated Creativity. In Cardoso, A., Bento, C., and Gero, J., editors, *Proceedings of the Third Workshop on Creative Systems*. IJCAI-03.

[Steel, 1999] Steel, G. (1999). *Cross-domain Concept Formation using HR*. MSc Thesis, University of Edinburgh.

[Sternberg and Lubart, 1996] Sternberg, R. and Lubart, T. (1996). Investing in creativity. *American Psychologist*, 51:677–688.

[Sternberg and Lubart, 1999] Sternberg, R. and Lubart, T. (1999). The Concept of Creativity: Prospects and Paradigms. In Sternberg, R., editor, *Handbook of Creativity*, pag. 3–16. Cambridge University Press.

[Stock and Strapparava, 2003] Stock, O. and Strapparava, C. (2003). Getting Serious about the Development of Computational Humor. In Gottlob, G. and Walsh, T., editors, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers.

[Sweetser and Dancygier, 1999] Sweetser, E. and Dancygier, B. (1999). Semantic overlap and space-blending. In *Proceedings of Sixth International Cognitive Linguistics Conference*.

[Thomas, 1999] Thomas, N. (1999). Are Theories of Imagery Theories of Imagination? An Active Perception Approach to Conscious Mental Content. *Cognitive Science*, 23:207–245.

[Turner, 2006] Turner, M. (2006). Compression and representation. *Language and Literature Journal*, 15(1):17–27.

[van den Bosch, 2001] van den Bosch, A. (2001). *Rationality in Discovery: A Study of Logic, Cognition, Computation and Neuropharmacology*. Institute for Logic, Language and Computation, Amsterdam University: Dissertation Series 2001-02.

[Veale, 1995] Veale, T. (1995). *Metaphor, Memory and Meaning: Symbolic and Connectionist Issues in Metaphor Interpretation*. PhD Thesis, Dublin City University.

[Veale and Keane, 1997] Veale, T. and Keane, M. (1997). The Competence of Sub-Optimal Structure Mapping on Hard Analogies. In *Proceedings of the International Joint Conference on Artificial Intelligence*. IJCAI'97.

[Veale and Keane, 1998] Veale, T. and Keane, M. (1998). Principle Differences in Structure Mapping. In Holyoak, K., Gentner, D., and Kokinov, B., editors, *Proceedings of Analogy '98*. New University of Bulgaria Press: Bulgaria.

[Veale and O'Donoghue, 2000] Veale, T. and O'Donoghue, D. (2000). Computation and Blending. *Cognitive Linguistics*, Special Issue on Conceptual Blending.

[Wallas, 1926] Wallas, G. (1926). *The Art of Thought*. New York: Harcourt-Brace.

[Ward et al., 1999] Ward, T., Smith, S., and Finke, R. (1999). Creative Cognition. In Sternberg, R., editor, *Handbook of Creativity*, pag. 189–213. Cambridge University Press.

[Weisberg, 1999] Weisberg, R. (1999). Creativity and knowledge: a challenge to theories. In Sternberg, R., editor, *Handbook of Creativity*, pag. 226–251. Cambridge University Press.

[Wiggins, 2001] Wiggins, G. A. (2001). Towards a more precise charaterization of creativity in AI. In Cardoso, A., Bento, C., and Wiggins, G., editors, *Proceedings of the First Workshop on Creative Systems, International Conference of Case-Based Reasoning*. ICCBR-01.

[Wiggins, 2003] Wiggins, G. A. (2003). Categorising Creative Systems. In Cardoso, A., Bento, C., and Gero, J., editors, *Proceedings of the Third Workshop on Creative Systems*. IJCAI-03.

[Winston, 1980] Winston, P. H. (1980). Learning and reasoning by analogy. *Commun. ACM*, 23(12):689–703.

[Wisniewski, 1997] Wisniewski, E. J. (1997). Conceptual combination: Possibilities and aesthetics. In Ward, T. B., Smith, S. M., and Vaid, J., editors, *Creative thought.: An investigation of conceptual structures and processes*. Washington DC: American Psychological Association.