AISB Journal **1**(3), © SSAISB, 2003
http://www.aisb.org.uk

The horse-bird creature generation experiment

Francisco C. Pereira and Amílcar Cardoso

Centro de Informática e Sistemas da Universidade de Coimbra (CISUC)
Dep. Engenharia Informática, Pólo II, Pinhal de Marrocos, 3030 Coimbra
camara@dei.uc.pt ; amilcar@dei.uc.pt

Abstract

This paper presents the process and results of experiments regarding the generation of blends of a concept of "horse" with a concept of "bird". The blending process is based on the framework of Conceptual Blending (Fauconnier and Turner, 1998) and its development is achieving some stability. We present an overview of our system, Divago, namely of its newest developments around the optimality constraints.
The results demonstrate the behavior of the system with regard to each of the optimality constraints and also give an insight on its ability for the generation of new concepts from the combination of pre-existing ones, although highlighting problems and further developments that must be taken.

1 Introduction

One big challenge to AI, more specifically to Computational Creativity, is that of the generation of new concepts. The first issue to approach is the very definition of *concept* and its representation, interesting issues on their own right. Assuming a concept representation and semantics, we are then faced with the problem of the *process*. What kind of processes can yield new and valid concepts?
In this paper, we apply a model of creative process that follows a framework, named Conceptual Blending (CB) (Fauconnier and Turner, 1998), and present some results of recent experiments. Although lacking in formalization and scientific proof in some aspects, this framework suggests principles and processes that explain many creative cognitive phenomena, such as metaphor, analogy and conceptual combination. CB is, at the least, a very elegant model of creativity, a motivation that led us to attempt to a computational basis. In the system we are developing, *Divago*, those principles and processes are applied iteratively until a *stable* solution is found. This solution should be a *blend*, a new concept (or web of concepts) that shares structure and knowledge from the inputs, yet having an emerging structure of its own (e.g. a "pegasus", as a blend of "horse" and "bird").
We start this paper by a short review of similar systems, namely from a related area named Conceptual Combination, after which we give an overview of the CB framework. Divago is presented afterwards and, finally, we present and analyse the experiments we made with the "horse" and "bird" domains, which constitutes the main contribution of this paper. The reader will also find a final discussion, in which we make a reflection around the results, the presented model and its creative aspects.

2 State of the Art

The first computational work on Conceptual Combination we find in literature is that of Carl Andersen (Andersen, 1996), which presents a system for "joining of information from two existing concepts to form a third, more complex concept". He gives a set of very interesting ideas, but the paper is lacking argumentation and validation, thus oversimplifying conceptual combination. An example of combination of "house" and "boat" is given, but the definition of these two initial concepts, from our point of view, biases the results because of their overt simplicity. Another issue is the lack of *background knowledge*, i.e., each concept is considered in isolation (a fact the author himself acknowledges), so there are no ontological explanations or means of relating the concepts in question other than from their structure, leaving to an external entity the task of establishing a mapping between them.

Fintan Costello and Mark Keane (Costello and Keane, 2000) bring us a computational model, C^3, for the interpretation of noun-noun compounds (e.g. "Cactus fish", "pet shark"), proposing one or more solutions for each concept pairing and validating them against empirical tests on people. C^3 searches for concept explanations that use differentiating properties from each of the nouns (the *diagnosticity* constraint), that are consistent with background knowledge (the *plausibility* constraint) and that avoid redundancy or vagueness (the *informativeness* constraint). In so doing, their system provides different sorts of noun-noun combinations, thus resulting in the polysemy we also find in humans. Noun-noun compounds are clearly one example of the conceptual combination and creativity we do regularly.

On the side of Conceptual Blending, Tony Veale and Diarmuid O'Donogue (Veale and O'Donogue, 2000) describe, from a computational perspective, a proposal inspired by Veale's Metaphor interpretation framework, *Sapper*. As we argue in (Pereira and Cardoso, 2001), this proposal lacks some fundamental points of CB, namely the emergence of a new domain, the blend, independently of the initial inputs. Furthermore, it takes into account only *metaphoric blends*.

Our system, Divago, initially proposed in (Pereira, 1998), formalized in (Pereira and Cardoso, 2001; **?**; Pereira and Cardoso, 2003)), makes use of a computational version of Conceptual Blending as a process for transformation of the search space. This motivation was discussed in (Pereira and Cardoso, 2002b), and the first experiments with blending a "house" and a "boat" are shown in (Pereira and Cardoso, 2002a) and demonstrate a change to the search space (e.g. containing houses with circular windows). Divago has a knowledge base composed of domains, instances and rules and blends them following eight optimality principles (described below). It makes use of a *generic domain* to find mappings between concepts, generic frames and rules and integrity constraints. It is expected to do concept combination as in (Andersen, 1996) and make noun-noun compound interpretations as in (Costello and Keane, 2000). The experiments shown in this paper focus on the former, while the ones presented in (Pereira, 2003) concerned to the latter.

3 Conceptual Blending

Conceptual Blending was initially proposed by (Fauconnier and Turner, 1998) as part of a major framework concerning cognition and language. Its role was to explain the integration of knowledge coming from distinct sources into a single, independent and coherent unit, the Blend. 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 find examples of blends in many sorts of situations. A blend can be an effective way to get attention and curiosity towards advertising a product (e.g. Sony's AIBO robot uses all sorts of Sony products behaving as if it were a real human) or spreading a message (e.g. the Marlboro cowboy with impotence problems). People have been making blends at least from the times of Greek mythology (e.g. *pegasus*) till today (e.g. *pokemons*) and is present throughout our daily communication (e.g. "John digested the book", "Sue sneezed the napkin off the table"). Many more examples and situations could be listed and studied in detail, demonstrating the ubiquity of CB.

In the *canonical* model of Conceptual Blending, we have four different *spaces*: two input spaces, one generic space and the blend. Each space corresponds to what Fauconnier and Turner call a "mental space", a cognitive structure that corresponds to a concept, a set of concepts, a frame, a reasoning or *lower level* entities like the perception of movement or the feeling of physical pain. Mental spaces may have internal connections (inner-space relations) between their constituent elements and connections to other mental spaces (outer-space relations). The input spaces correspond to two mental spaces (e.g. horse and bird) that will be integrated in the blend (e.g. a horse with wings). The generic space contains knowledge that is not specific to any of the inputs but may relate to both (e.g. biology taxonomies) or is common sense (e.g. Greek mythology).

An essential step in the process of blending is the establishment of a (partial) mapping between elements of the input spaces. This mapping may be achieved through different processes (e.g. identity, structure alignment, slot-filling, analogy) and doesn't have to be 1-to-1. 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.

From the projections, some new relations emerge that relate elements either as a direct result from the projection or from "running the blend", which consists of performing cognitive work within the blend, according to its own emergent logic. There is a set of *governing principles*, the *Optimality Pressures*, that should drive the process of generating a "good blend" (Fauconnier and Turner, 1998):

- Integration - The blend must constitute a tightly integrated scene that can be manipulated as a unit. More generally, every space in the blend structure should have integration.

- Pattern Completion - Other things being equal, complete elements in the blend by using existing integrated patterns as additional inputs. Other things being equal, use a completing frame that has relations that can be the compressed versions of the important outer-space vital relations between the inputs.

- Topology - For any input space and any element in that space projected into the blend, it is optimal for the relations of the element in the blend to match the relations of its counterpart.

- Maximization of Vital Relations - Other things being equal, maximize the vital relations in the network. In particular, maximize the vital relations in the blended space and reflect them in outer-space vital relations.[1]

[1]Fauconnier and Turner 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.
- Unpacking - The blend alone must enable the understander to unpack the blend to reconstruct the inputs, the cross-space mapping, the generic space, and the network of connections between all these spaces
- 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.
- 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. Conversely, an outer-space relation between the inputs that is important for the purpose of the network should have a corresponding compression in the blend.

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 decrease. As far as we know, there is no work yet towards an objective study of the optimality pressures, measuring examples of blends or specifying these principles in detail. This, we believe, inhibits considerably the appreciation and application of Conceptual Blending in scientific research, making a particular motivation for this work being that of testing and specifying a formal proposal of these optimality pressures.

4 Overview of the Model

The architecture of Divago has four central modules: The Knowledge Base, the Mapper, the Factory and the Constraints module. The Knowledge Base comprises data structures (the concept maps, the frames, the integrity constraints and the instances) that are organised in *spaces*, according to their scope and generality. The Mapper establishes mappings between the *input* spaces (e.g. some concepts from the input space "horse" get mapped onto concepts from the input space "bird"), and provides these associations to the Factory, which deals with the process of blend generation, controlled through the evaluation made by the Constraints module. This module implements the optimality constraints which, when together applied to a given blend, return a value between 0 and 1. A fifth module, Elaboration, will be added to the final version of the system and will transform the blend according to knowledge from the generic space (by applying rules and triggering frame conclusions).

4.1 The Knowledge Base

As in any other AI system, knowledge representation is the first fundamental issue to decide. Here, we are particularly concerned about the representation of a "concept", for it is the goal of Divago to generate new "concepts". Assuming a symbolic approach (as opposed to sub-symbolic ones, such as neural networks or genetic algorithms), we decided for a semantic network based representation, in which a concept does not stand alone as an isolated symbol, its definition and explanation being dependent on the relationships it has with the surrounding concepts. More specifically, we take our concept networks as being *Concept Maps*. A Concept Map is a graph in which nodes represent *concepts* and arcs represent *relations*. This is far from a novel perspective. It goes in consonance with Murphy and Medin's mini-theories (Murphy and Medin, 1985) or CYC (Lenat, 1995) and WordNet (Miller, 1995) representations. Even more important, any of the previously mentioned works (Costello and Keane, 2000; Andersen, 1996; Veale, 1997) suggest this view of concepts.

From a semiotics perspective, this representation of concepts seems Saussurian because “everything depends on relations”(de Saussure, 1983). We try to escape from this extreme position through the possibility of association of effective semantics to each concept (e.g. the concept “window” may be realized as a set of instructions for “drawing a square”) and by the association of the concepts to practical examples, the *instances* (as in (Pereira and Cardoso, 2002a)). Now, from a Peircian point of view, imagining the *meaning triangle* (Ogden and Richards, 1923), we have the individual *symbol* (e.g. the word “window”) as standing for a *concept* (e.g. the concept network around “window”) and corresponding to an *object* (e.g. a drawing of a window).

The choice of symbols for concepts and relations in our concept maps is arbitrary, yet we are following a set of normalization principles. The first one is that relations must belong to the Generalized Upper Model hierarchy (GUM) (J. Bateman and Fabris, 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 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 also used as relations (e.g. “color(mane, dark)”, “ability(horse, run)”). Other principles we follow in the construction of the concept maps is that concepts in our knowledge base may only be represented as nouns, adjectives, preferably in the singular form, or numerals (in the particular case of numbers). 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 concept”. In Divago, there are also other elements (such as instances), but for the scope of this paper, the reader needs only to understand the notion of *concept map*.

In Table 2, we show examples of concept maps of “horse” and “bird”. These maps are necessarily arbitrary in the sense that each person would draw their own maps, a result of the different conceptualization and points of view one can take individually. Yet, we assume these as being the conceptualization of the domains of “horse” and “bird” and so, when we interpret a new concept as being a “bird with a moustache”, we refer to that specific “bird” concept map with an attached subgraph that represents a “moustache”.

The semantics of each individual relation is arbitrarily defined by the user. By default, each relation is simply a symbolic connection between two concepts (e.g. in “purpose(leg, stand)”, “purpose” is a connection between “leg” and “stand”), being its interpretation dependent on a context (e.g. in (Pereira and Cardoso, 2002a), the “shape” relation and its parameters were converted into a set of drawing primitives). Structurally, the user is allowed to add *integrity constraints* to the relations (e.g. a “pw”, or part-whole, relation cannot be circular, i.e. “pw(X,X)” is not possible) as well as to attach it to the GUM hierarchy (e.g. “eat”, not in the original GUM, descends from “dispositive material action”). There is a particular relation that has a special role, “isa”, which attaches a concept to a taxonomic hierarchy (the general ontology) that is included in Divago Knowledge Base.

Two other important knowledge structures to refer here are the *frames* and the *integrity constraints*. The frames have the role of describing specific composite concepts, situations

isa(horse,equinae)	pw(leg, horse)	purpose(horse, food)
isa(equinae,mammal)	purpose(leg, stand)	sound(horse, neigh)
existence(horse, farm)	pw(paw, leg)	purpose(mouth, eat)
existence(horse, wilderness)	purpose(horse, traction)	purpose(ear, hear)
pw(snout, 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(paw, 4)	quantity(leg, 4)	purpose(horse, cargo)
pw(eye, snout)	quantity(eye, 2)	taxonomicq(horse, ruminant)
pw(ear, snout)	quantity(ear, 2)	ride(human, horse)
pw(mouth,snout)	purpose(eye, see)	motion_process(horse,walk)
isa(farm, human_setting)		

Table 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 2: The concept map of *bird*

or idiosyncracies. For example, we could specify that we are in face of a "new ability" if some concept X has, in the blend, the ability A, which was not present in X's input space, d1. We can even say that this "new ability" should have a minimal explanation, i.e., there must be a subpart P of X whose purpose is to provide ability A. Furthermore, we can also require that X and A be projected from different inputs ($d1$ and $d2$, resp.) to the $blend$.

$$\begin{array}{lll} \multicolumn{3}{l}{frame(new\_ability(d1)):} \\ \quad new\_ability(X,A) & \longleftarrow & ability(X,A) \wedge not\, rel(d1, ability(X,A)) \wedge \\ & & purpose(P,A) \wedge pw(P,X) \wedge \\ & & projection(blend, d1, X, X) \wedge \\ & & projection(blend, d2, A, A) \end{array}$$

Frames can represent very abstract reasonings (e.g. the blend should have the same structure as the input space 1 - the "aframe") or very specific (e.g. the "transport_means" frame). The generic space we use in the experiments has the frames of Table 3.

In Figure 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

Frame name	**Conditions**
aframe	The blend contains identical structure from input 1
aprojection	The blend contains the same concepts of input 1
bframe	The blend contains identical structure from input 2
bprojection	The blend contains the same concepts of input 2
pw_based_explanation	The blend contains a concept that has associated a set of part-whole relations (i.e. it is explained by a set of these relations)
transport_means	The blend contains a concept that has associated the set of features of a generic transport means
purposeful_subpart	The blend contains a a concept that has a subpart that has associated a set of relations that justify its existence (e.g. purpose, cause-effect)
new_ability	A concept has an ability relation not existent in any of the inputs
new_creature	A concept is a living thing that did not exist (or wasn't such) in any of the inputs
new_feature	A concept has a feature relation not existent in any of the inputs

Table 3: Frames of the generic space

"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).

The integrity constraints serve to specify logical impossibilities. 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:

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

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

The violation of an integrity constraint does not imply the elimination of a blend; it only brings a (configurable) penalty to its value. Thus it must have strong arguments to

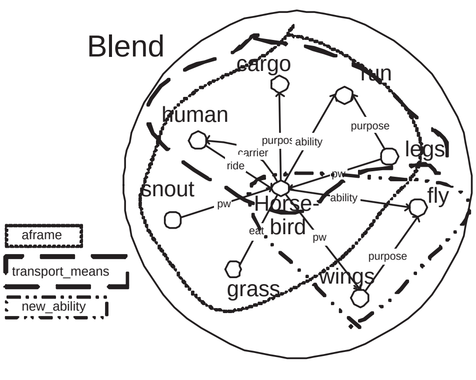

Figure 1: Three frames in a small blend

violate an integrity constraint and still be a "good blend". For space restrictions, we don't show the generic domain concept map, yet the reader should only know it has a very long list of "isa" relationships, establishing an ontological basis for the concepts (e.g. *isa(red, color), isa(human, primate), isa(physical_object, object)*, etc.).

4.2 Mapper

The Mapper currently takes an optional role in the architecture. Its purpose is to generate mappings between the concept maps of the input domains automatically. It uses an algorithm of structure matching inspired in Tony Veale's Sapper framework (Veale and Keane, 1993). Basically, it uses a spreading activation algorithm to look for the largest isomorphic pair of subgraphs from the input domains. In this context, two graphs are considered isomorphic if they have the same relational (arcs) structure, independently of the concepts (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 concepts[2]. However, since it only allows alignment when it finds equal relations in both graphs, the number of possible solutions can be drastically reduced, yet still demanding Mapper to make the search in such huge space. Furthermore, the algorithm starts with a randomly selected pair of concepts, so the "perfect choice" (or even the same choice) is not guaranteed every time we run it.

This module generated three different mappings for input spaces of "horse" and "bird", as shown in Figure 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 very unintuitive associations (e.g. "4" associated with "2"; "snout" with "bird"). The existence of "wrong" associations, however, doesn't necessarily affect the results because, when their use implies low-valued outcomes, the system will ignore them due to the "selective projection" algorithm we will describe in the next section.

4.3 Factory

Each concept x, from the input domains, has a "projection" in the blend that can be either x, nil (meaning it is not projected to the blend), y (the mapping counterpart of x) or a composition of x and y (represented by $x|y$). The latter two only possible when there is a mapping counterpart for x. To a string of "projections" of all concepts of the input domains (one "projection" for each concept), we call a "selective projection". The name "selective projection" comes from the fact that, in blends, some aspects of the input spaces are not present (i.e., not projected), some change and some remain the same.

The role of the Factory module is to make a search for blends that best fit the optimality constraints in the space of "selective projections". This space has a very high complexity. Taking a close look on this issue, we notice that, for an *input* domain 1 with m concepts and an *input* domain 2 with n concepts (with $m \geq n$), we may have the maximum of $m!$ different mappings (if we use the isomorphic mappings, as in the Mapper), with the largest mapping having a size $k = min(m, n)$. This projection selection is made independently on each concept, which means we have $l = m + n$ different concepts for each blend, each one with its own projection. So, in the "least complexity scenario", the size of the

[2]Assuming n as the number of concepts of the largest (in number of concepts) of the two concept maps, we will have a search space of $n!$ possible mappings. So, with an exponential k^n, as n approaches infinity, $\frac{k^n}{n!}$ will be 0, meaning that the search space will expand more than exponentially as the number of concepts grows

ear	$\leftrightarrow$	wing
snout	$\leftrightarrow$	bird
eye	$\leftrightarrow$	lung
mouth	$\leftrightarrow$	feathers
2	$\leftrightarrow$	2
hear	$\leftrightarrow$	fly

1

mouth	$\leftrightarrow$	beak
snout	$\leftrightarrow$	bird
eye	$\leftrightarrow$	lung
ear	$\leftrightarrow$	feathers
eat	$\leftrightarrow$	eat

2

vegetable_food	$\leftrightarrow$	vegetable
food	$\leftrightarrow$	food
horse	$\leftrightarrow$	bird
equidean	$\leftrightarrow$	aves
animal	$\leftrightarrow$	animal
human_setting	$\leftrightarrow$	house
wilderness	$\leftrightarrow$	wilderness
ruminant	$\leftrightarrow$	oviparous
run	$\leftrightarrow$	fly
cargo	$\leftrightarrow$	pet
neigh	$\leftrightarrow$	chirp
snout	$\leftrightarrow$	lung
mane	$\leftrightarrow$	feathers
tail	$\leftrightarrow$	beak
leg	$\leftrightarrow$	eye
paw	$\leftrightarrow$	wing
4	$\leftrightarrow$	2
eye	$\leftrightarrow$	leg
ear	$\leftrightarrow$	claw
hear	$\leftrightarrow$	catch
grass	$\leftrightarrow$	grass

3

Figure 2: The three mappings

mapping is 0, meaning that we have only two choices for each of the l concepts (either it gets projected to the blend or it is not projected), thus we have 2^l "selective projections". If the size of the mapping is k (the maximum possible), we have four choices for each of $2k$ concepts (k concepts in each of the domains) because each concept x mapped to y can be projected either to x, y, $x|y$ or nil. Apart from these $2k$ concepts, the rest $(l - 2k)$ has only two possibilities. This leads us to the conclusion that we have a range of 2^l to $4^{2k} \times 2^{l-2k}$ different "selective projections" to choose, which is a very large search space. For example, for $m = n = 20$ (a "small" size pair of networks), we have at least 2^{40} different solutions.

Given these space dimensions and expecting that the optimality pressures (a set of competing constraints, described in next section) would complexify the search landscape, we decided to implement a parallel search algorithm, which wouldn't depend on following a specific sequence of application of those pressures. The solution we found was a genetic algorithm (GA): a framework inspired by evolutionary theory in which we have a sequence of *populations* of *individuals*, each *individual* with a *fitness* value that represents its *survival* and *reproduction* possibilities. This well-known framework has had much success in problems with a search space with the characteristics we described. The detailed formal and technical explanation of GA's is far out of the scope of this paper, so we direct the interested reader to (Goldberg, 1989). On the other side, those not interested in the technical details of our GA implementation may skip to the next section and retain the general idea that this is a parallel search that does not guarantee the "best blend" but is able to search a vast area of the space and return, with correct parameters, relevant solutions.

In our GA, the *individuals* we are *evolving* are blends, each one determined by a "selective projection". The *individual* is then an ordered sequence of projections (the *genes*), each one with an allowed value given by the projection function (from the range x, y, $x|y$ and $\emptyset$). The evaluation of a blend is made by the application of the optimality pressures, which then participate in a weighted sum, yielding the fitness value. We have populations of individuals (currently 100) that are then stochastically selected according to this value. After the selection of the individuals, the step of generation of the following population is made by using 4 operations: direct reproduction (the individual is copied to the next population); crossover (two individuals exchange part of their list of projections); mutation (random changes in the projections); random individual. The system stops when a predefined number of iterations of this process has been done, when it stabilized around a maximum for more than a predefined number of iterations or when an individual was found that has a satisfactory (predefined) value.
Through this process, 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.

4.4 Constraints

The Constraints module implements the optimality pressures. The general role of this module is to make a preprocessing of each blend (checking frame satisfaction and completion, integrity constraint violation, vital relation projection, etc.) and then obtain a value for each of the eight measures. These values then participate on a weighted sum, which yields the *value* of the blend (normalized to fall into the [0,1] interval) that is returned to the Factory. The weight attributed to each optimality pressure is defined by the user. The optimality pressures are formalized and described in (Pereira and Cardoso, 2003), and so, an entire paper is needed to specify in detail our implementation of these. Therefore we give an informal explanation below. Beforehand, we would like to say that we make no claims in respect to the cognitive realization of each measure. These eight suggestions of quantification concern totally to the representation and scope of this model which moves towards a computational account of conceptual blending. This doesn't mean that this proposal should not be verified or tested with regard to cognition and the blending phenomena in general, it states that we didn't make our measures based on cognitive experiments, but only tried to follow the philosophy behind the description that F&T give in (Fauconnier and Turner, 2002) projected to our formal model.

4.4.1 Integration

Frames have an integration role. The reasoning behind a frame relies in the idea that concepts within it should be tightly integrated according to a situation, structure, cause-effect or any other relation that ties a set of concepts onto one, more abstract or broad, composite concept. For example, the frame of "transport_means" corresponds to a set of concepts and relations that, when connected together, fit the abstract notion of "transport means". Frames may get much more abstract and, as in the example of "aprojection" or "new_feature", represent construction directives for the blend. The integration role of these frames is not perceived as clearly as in "transport_means" but, without them, the blend will lack global consistency (as in "aprojection") or novelty (as in "new_feature"). We see frames as *information moulds* and building a blend for a given situation should depend much on the choice of these structures. Thus, the Integration value of a blend is

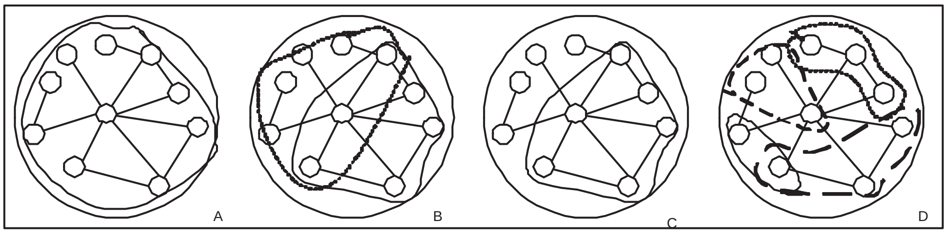

Figure 3: The role of frame coverage in Integration value

calculated with regard to a set of frames.
Assuming the set F of frames that are satisfied in a blend, we define the *frame coverage* of a domain to be the set of relations from its concept map that belong to the set of conditions of the frames in F. The larger the frame coverage of the blend, the higher its Integration value is. Yet, a blend that is covered by many frames should be less integrated than a frame 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 domain. In the diagrams of Figure 3, we give an informal idea of this reasoning. Blend A has the maximum Integration value (100%) because a single frame is able to cover the complete concept map. B will have a lower value since it needs two frames to cover the same area. Blends C and D will have lower values than the other two and C will have higher value than D because frame coverage of D is too much dispersed.

The Integration measure we propose also takes integrity constraints into account so that, when a frame violates such a constraint, it is subject to penalty. Integration belongs, along with Relevance, Topology and Unpacking, to the fundamental bricks of the blending process of Divago. It is intended to lead the choice of the blend to be something *recognizable* as a whole, fitting patterns that help to determine and understand what a *new* concept is.

4.4.2 Topology

The Topology optimality pressure brings *inertia* to the blending process. It is the constraint that drives against change in the concepts because, in order to maintain the same topological configuration as in the inputs, the blend should maintain exactly the same neighborhood relationships between every concept, ending up being a projected copy of the inputs. This pressure is normally one that is disrespected without big loss in the value of the blend. This is due to the *imagination* context that normally involves blends, i.e., novel associations and changes in previous ones are not necessarily penalized.
We calculate our Topology measure by simply obtaining the ratio of relations ($r(x, y)$) in the blend that also exist in the inputs. This means, for example, in blending "horse" and "bird", that obtaining an exact "horse" or an exact "bird" would give the highest Topology value (every relation would also exist in the corresponding input domain), while adding the relation "ability(horse, fly)" to the "horse" would lower that value (there is no "ability(horse, fly)" in any domain).

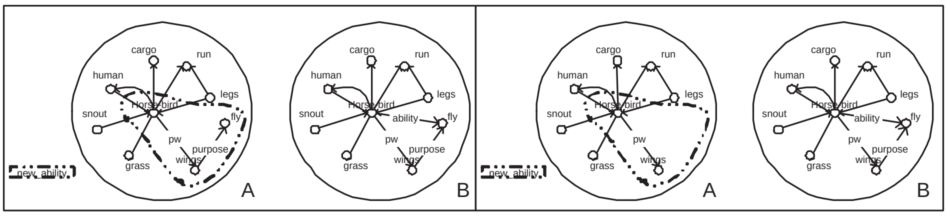

Figure 4: Pattern Completion examples

4.4.3 Pattern Completion

The Pattern Completion pressure brings the influence of patterns present either in the *inputs* or in the *generic* space. Sometimes, when reasoning about a concept (or a set of concepts), it may make sense to *complete* it with new knowledge. For example, if we have a "horsebird" defined as having "2 wings made of feathers", we may *complete* it with the "flying ability" by matching the concept of "horsebird" with the pattern of "flying creatures": "Flying creatures have 2 wings. They are made of feathers and serve to fly".

At present, in the context of this work, a pattern is described by a frame, i.e. we don't distinguish these two concepts, and therefore pattern completion is basically frame completion. Here, as in the definition of this principle, the completing knowledge becomes available from "outside", not as a result of projection. This means that the act of completing a frame consists of asserting the truth of the ungrounded premises from frames of the generic domain, a process that happens only after a sufficient number of premises is true. We call this the *evidence threshold*. In Figure 4, we show two examples of the evidence thresholds of a frame ("new_ability") with regard to two different blends. In the first one, the frame has an evidence of 67% approximately (the frame has three relations, and two are true in the blend) and so its completion is made by adding the relation "ability(horse-bird, fly)". In the second one, the evidence threshold is approximately 33% (and so it is completed by adding two new relations).

As in the integration pressure, 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 domain, it is difficult to find such a linear rationale (e.g. would two patterns each with individual completion x be valued higher than three having each slightly less than x?). As a result, the value of Pattern Completion of a blend corresponds to the evidence threshold of the union of the frames (which will be the frame with all conditions that appear in all the frames).

4.4.4 Maximization of Vital Relations

For the maximization of vital relations, we estimate the impact of the inner-space and outer-space vital relations to the blend. Fauconnier and Turner list a set of 15 vital relations (in footnote of page 3 of this document) that exist between elements within a mental space (inner-space relations) and between different mental spaces (outer-spaces). These are *special* relations in the sense that they can be *compressed* in blend. Since in our current model we don't explicitly approach compression, vital relations will be a set of (customizable) relations whose projection should be valued.

The measure of Maximization of Vital Relations is calculated as a ratio of the actual num-

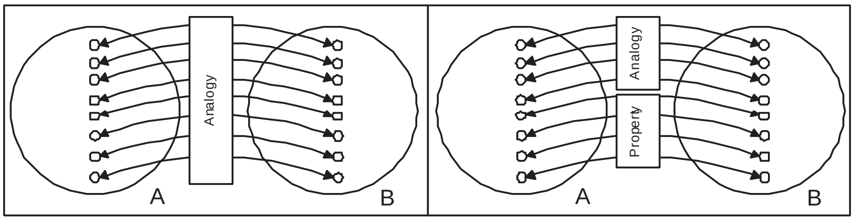

Figure 5: Intensification VR examples

ber of vital relations in the blend w.r.t. the maximum possible number of vital relations (that would appear in the blend if every concepts were projected).

4.4.5 Intensification of Vital Relations

The difference between Intensification and Maximization of Vital Relations is not clear in the definition so we propose a perspective that may disagree with what was originally meant by Fauconnier and Turner. In our case, Intensification of Vital Relations is a measure concerning (exclusively) outer-space relations, more specifically the relations established by the mapping algorithm. The rationale is that, for each vital relation, there is a mapping algorithm that connects elements from the two input spaces with the respective vital relation and the use of these connections will yield an estimate of how "intense" this vital relation is in the blend. In our case, the (only) vital relation established is "analogy" (e.g. there is an "analogy" relation between "run" and "fly" in mapping 3, Figure 2) and its intensity is measured by the *systematicity principle* (if x is associated to y, its neighbors should also be associated). For different vital relations (e.g. "disanalogy"), different intensity measures could be applied. We have only implemented "analogy" so far, so our proposal for this measure is far less solid than the others.

The calculation of the value for Intensification pressure takes the point of view that a blend that applies mappings generated by only one vital relation (suppose it has an intensity value x) should have higher measure than a blend that apply n vital relations (suppose each with intensity value x/n). We want to favor "concentration", therefore there is a penalty for the proliferation of different vital relations. In Figure 5, we show an example of two choices for mappings. The mapping on the left will get higher Intensification value because it is concentrated around a single vital relation ("analogy").

In the experiments, our mapping was based on a single vital relation therefore this measure could not yet be tested.

4.4.6 Unpacking

Unpacking is the ability to reconstruct the whole process starting from the blend. From our point of view, such achievement underlies the ability to reconstruct the input spaces. 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 so 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"

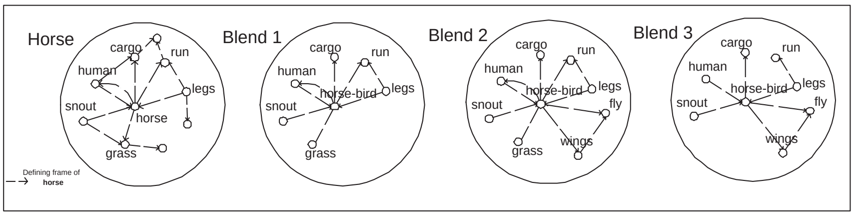

Figure 6: Unpacking examples

concepts. Once again we use the idea of *frames*, more specifically the *defining frame* of a concept, which comprises its immediately surrounding relations. In the blend, if we can identify clearly the defining frames of the original concepts, then its Unpacking value is high. In Figure 6, we present the defining frame for “horse”, in the “Horse” domain. In Blend 1, the concept “horse-bird” (the projection of “horse”) will have the highest Unpacking value because it fits exactly its defining frame. In Blend 2, the value is lower because there are two new relations (with “fly” and “wings”), meaning it is not the exact same concept. Blend 3 will get the lowest Unpacking value of all three because it also lacks some relations (e.g. with “run” and “grass”).

The calculation of the Unpacking value of a blend corresponds to an average of the individual Unpacking values of all the concepts in the concept map. Each of these individual Unpacking values takes into account the completion ratio of the defining frame and the extra relations added.

4.4.7 Web

The Web principle concerns to being able to “run” the blend without cutting the connections to the inputs. It is our opinion that 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. It is not to say that Web is the same as Topology or Unpacking, what we are arguing is that, on one side, Topology provides a pressure to maintain the most fundamental connection to the input: the same set of relations; on the other side, Unpacking evaluates the easiness of finding the connections to the inputs. The weighted sum of these two values yield, we propose, an estimation of the strength of the web of connections to the inputs.
Not being an independent variable, we don’t apply the Web constraint in the tests we show here.

4.4.8 Relevance

The notion of “relevance” or “good reason” for a blend is tied to the pragmatics of the situation, or, in other words, the context and goal of the blending generation. Once again, frames take a fundamental role, they are “context specifiers” (i.e., the set of constraints within a frame describe the context within which the frame is fulfilled). Thus, Divago allows the specification of a *query* that may contain a set of conditions (e.g. if we want to find a concept that “flies”, we could add the condition $ability(x, fly)$) and a set of frames to be accomplished (e.g. the blend should accomplish the “transport_means” frame). This query will then correspond to a set of frames (the set of conditions is also considered a

frame itself) to which we call *goal frames*.
Having a set of goal frames, which could be selected from any of the existent domains or specified externally, a blend gets the maximum Relevance value if it is able to satisfy all of them. In this measure we must also take into account partial completion of the goal frames. A blend that "almost" satisfies a goal frame should be valued in relation to a frame that doesn't (assuming both are equal in the other features). Regarding this, we consider a factor for the partial completion of the goal frames following the same procedure as in Pattern Completion.
Intuitively, this measure takes two parts: the satisfied goal frames and the unsatisfied goal frames. The value of the latter depend on completion (e.g. if Completion=50%, these count as "half" satisfied goal frames).
The Relevance principle allows a user to specify his notion of *usefulness*. In fact, the usefulness of a concept is always a pragmatic matter. Something can be extremely and obviously useful in a context and the opposite in another. For example, in a graphical environment, useful concepts should have a color, a shape, a position and so on, while in a text-based role-playing game, they are expected to have the game's attributes (e.g. mood, strength, role, etc.). Thus, a blend that has a 100% Relevance value satisfies all the conditions of the query, meaning it contains all the knowledge needed for a given context, i.e., it is useful with regard to that context. This is not to say that usefulness is easily specified, as it is not straightforward to find all constraints and requirements that need to be respected in a given context.

5 Experiments

We made two different 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), 5% of mutation and 1% of random generation (to allow random jumps in the search space). 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 two experiments.

5.1 Evaluating Optimality Pressures

This test serves to observe the real 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.) only applies one mapping algorithm (based on analogy), we did not test them, so we had six different criteria to take into account.
The input domains we applied were the domains of *horse* and *bird* (in Tables 1 and 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 a simple general ontology (essentially an "isa" tree with concepts from high-level like "information_entity" or "physical_object" to low-level like "digit" or "dog"), a set of frames and integrity constraints (see Table

3). We applied the mappings presented in Figure 2. For each mapping, we tested the six optimality pressures, each of these comprising 30 runs[3].
We present now a detailed analysis of the individual effect of each of the measures:

- 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 away. The evolution is directed to a compromise of coverage and satisfiability. 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, it returns six different blends, being the best choice retrieved 43% of the times, while with a mapping size of 21, it finds eight different solutions, being the best choice retrieved only 6% of the times. This confirms the complexity and dimensions of the search space we discussed in section 4.3. 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.

- **Pattern Completion** drives the blend to partially complete (i.e., satisfy some of conditions but not all) the highest possible number of frames, leading, in each case, to several sets of relations that fit into those frames without satisfying them. 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 brings gradually to the blend the concepts and relations that are needed to complete the frames and so speeding up the process of finding frames with high Integration value. In which respects to the *search landscape*, it seems to be very rich in local maxima. The most constant results came from mapping 2 (of Figure 2), with the best results obtained in 13% of the times and the second best in 20%. An interesting remark is that the resulting local maxima always fall within a very strict range of values (of maximum amplitude 0.11, in mapping 3).

- In all the experiments with **Topology**, the final results were valued 100%, meaning that this constraint is easily fully accomplished, independently of the mapping. An interesting fact is that there is a multitude of solutions in the *search landscape* of Topology, showed by the amount of different final results in each mapping. Intuitively, and observing the short duration of each run, this means that, wherever the search starts, there is always a Topology optimal point in the neighborhood. From observation of the relations contained in the final results, we see that this constraint brings a tendency for *disintegration*, i.e, small isolated graphs appear in the blend. Each isolated graph is either a copy of a (normally unmapped) subgraph of one input source or consists of complete structure matching (there are concepts from both domains, but only the relations that exist in both are present)

- 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 30 runs, it reached the value 1 a minimum of 93% of the times, and the remaining 7% achieved

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

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.

- The results of the **Unpacking** measure show that it has a deleterious side effect on the size of the blend, it drives it to very small sets (between 0 and 5) of relations. The interpretation here is straightforward: the ratio of *unpackable* concepts is highly penalized in bigger sets because of the projected relations that come as side effect of the projection of (*unpackable* or not) concepts. These relations *confuse* the unpacking algorithm so that it leads the evolution to gradually select the smaller results. The maxima points also correspond to the value 1, but it seems, from the experiments, that there is a very limited set of such individuals, achieved in the majority (at least 93% for each mapping) of the experiments.

- 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 times 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. If there were separate frames, it is expectable the existence of local maxima. Intuitively, the *search landscapes* of Integration and Relevance seem to be similar.

It is important to stress that, in the current version of Divago, no inference is done within the blends. Each blend is examined by the Constraints module without being subject to any transformation after the projections. In other words, there is yet no "running the blend", an aspect that will be focussed in the next developments of this work.

5.2 Qualitative evaluation

In this stage of the experiments, we tried to understand the behavior of the system by generating and observing different blends, each one with a specific goal. The first goal was to generate a *well known* blend of a horse and a bird: the *pegasus*. Then, we allowed more variations of this creature, by changing the mapping or the weights of the optimality pressures. Finally, we tried to generate different creatures that, from our point of view, reveal interest.

5.2.1 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 concerning to the several 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, even if 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 (the relations "ability(horse, fly), motion_process(horse, fly), pw(wing, horse) and quantity(wing, 2)").
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 obviously the results reveal that only the mapping 3 (see Figure 2) respects such constraints. This led us to use exclusively this mapping throughout this section.
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 big set of different frames and no query, we quickly understood that it is not simple (or even possible) to build the pegasus solely by handling the weights. This happens because there is no controlling device that allows a user or an evaluation function to drive the evolution to a particular place. The optimality pressures provide control regarding to structural evaluation and general consistency and may yield interesting results, but only by chance a pegasus, which 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 space 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 find the solution if 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 (in these cases, the search wouldn't be needed, in the first place) and we are more interested in less constrained search directives. In the Table 6, we show the parameters we used. 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 the blend must have in order to be a flying creature, and aframe, aprojection and new_ability are frames as described before.

Exp.	Weights						Query
#	I	PC	T	MVR	U	R	
1	0	0	0	0	0	1	fly conds. + aprojection
2	0	0	0	0	0	1	fly conds. + aframe
3	0	0	0	0	0	1	fly conds.+ aprojection + aframe
4	1	0	0	0	0	1	fly conds.+ aprojection + aframe
5	1	1	0	0	0	1	fly conds.+ aprojection + aframe
6	1	0	1	0	0	1	fly conds.+ aprojection + aframe
7	1	0	1	1	0	1	fly conds.+ aprojection + aframe
8	1	0	1	1	1	1	fly conds.+ aprojection + aframe
9	8.5	0	4	2.5	1	9	fly conds.+ aprojection + aframe
10	8.5	0	4	2.5	1	9	new_ability+aframe+aprojection

Table 4: Parameters used in each experiment.

The first eight experiments 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 experiment 4, yielding the most strict pegasus, the projection of the entire horse domain, and the selective projection of wings and the fly ability from the bird domain, in more than 90% of the runs. In experiment 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 following experiments. Moreover, it suggests a revision of the implementation of this measure.

Adding Topology (exp. 6) brought essentially two different kinds of results. In 60% of the runs, it returned the "correct" pegasus with extra features like having feathers or a beak (which was not constrained in the query), either of each apparently selected at random. These were also given the higher scores in the experiment. In other 37% of the runs, the results were either "simple" horses or a compromise between a bird and a horse (e.g. two legs, a beak, two wings, ruminant, a mane, paws, etc.). A possible interpretation is that, on one side, the frames *aprojection* and *aframe* already imply strong topological maintenance, and Topology itself brings knowledge that, although not considered in the frames, strengthens this value. Yet, this does not avoid the existence of local maxima that represent stable results, in terms of the weights considered. The following experiment, the inclusion of Maximization of Vital Relations, confirmed the same conclusions, 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 *taxonomicq*). 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 around their maxima.

The eighth experiment brought a more stable set of results. Adding Unpacking to the other pressures reassures the prominence of the "basic" pegasus, but, as happened with the majority of sixth experiment results, augmented with features projected from the bird domain. This time, some of these new features came isolated to the blend, i.e., not connected to the rest of the blend (e.g. there are 2 claws that serve to catch, but they don't make part of anything).

An immediate conclusion we took from these first 8 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 to 12. 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 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 discreet presence of Topology, Maximization and Unpacking, to allow the transfer of extra knowledge.

The experiment 9 revealed, possibly, the "best" pegasus we could expect. As we can see

quantity(wing, 2)	conditional(wing, fly)	motion_process(horse, fly)
ability(horse, fly)	purpose(wing, fly)	pw(wing, horse)
isa(horse,equinae)	pw(leg, horse)	purpose(horse, food)
isa(equinae,mammal)	purpose(leg, stand)	sound(horse, neigh)
existence(horse, farm)	pw(paw, leg)	purpose(mouth, eat)
existence(horse, wilderness)	purpose(horse, traction)	purpose(ear, hear)
pw(snout, 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(paw, 4)	quantity(leg, 4)	purpose(horse, cargo)
pw(eye, snout)	quantity(eye, 2)	taxonomicq(horse, ruminant)
pw(ear, snout)	quantity(ear, 2)	ride(human, horse)
pw(mouth,snout)	purpose(eye, see)	motion_process(horse,walk)

Table 5: Example 1 (from experiment 9)

purpose(claw, catch)	pw(claw, leg)	purpose(lung, breathe)
pw(lung, horse)	conditional(wing, fly)	motion_process(horse, fly)
ability(horse, fly)	purpose(wing, fly)	pw(wing, horse)
isa(horse,equinae)	pw(leg, horse)	purpose(horse, food)
isa(equinae,mammal)	purpose(leg, stand)	sound(horse, neigh)
existence(horse, farm)	pw(paw, leg)	purpose(mouth, eat)
existence(horse, wilderness)	purpose(horse, traction)	purpose(ear, hear)
pw(snout, 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(paw, 4)	quantity(leg, 4)	purpose(horse, cargo)
pw(eye, snout)	quantity(eye, 2)	taxonomicq(horse, ruminant)
pw(ear, snout)	quantity(ear, 2)	ride(human, horse)
pw(mouth,snout)	purpose(eye, see)	motion_process(horse,walk)

Table 6: Example 2 (from experiment 9)

in the two results presented in tables 7 and 8, it has all the horse features, the specified "flying" requirements and some added knowledge that we consider valid, like having 2 wings, lungs or claws. It is clear that these results were subjectively driven by us in the choice of the concepts and frame design, but the argument we try to bring is that it produces a new concept that, not only respects the query, but also brings new knowledge that was selectively projected.

In the final experiment (10), we decided to give a more vague specification, asking only for a *new_ability* in the blend, as well as the generic frames *aprojection* and *aframe*. As a result, we found the exact pegasus in 23% of the times. This gives the first evidence that the system can be used for generating concepts without a very constraining and specific query and led us to the following experiments, in which we tried to assess its generative possibilities.

5.2.2 Other creatures

In order to explore the potential of the system, we made additional tests, without imposing specific goals beforehand. We didn't make significant variations on the weights of the previous tests. For two, we removed some weights from the configuration and reduced Integration in the latest ones. In table 9, we show all the configurations (the omitted weights are 0, as in the other experiments). We made variations on the query and checked the results, trying not to bias for particular outcomes. Therefore, these tests aim to give an informal insight on the generative potential of the system.

We found several "creatures" that we'd like to describe. To the first (experiment 11), we call "dumborse", a flying horse that uses its ears as wings (like *Dumbo*, the flying elephant). This "creature" is possible to find in mapping 1 (*ears* are mapped onto *wings*). It is exactly a horse, but it has wings instead of ears, which serve to fly and to hear. With *Dumbo* in mind, we tried to go further to a horse with ears that serve to fly and hear (instead of wings in place of ears), and this was achieved by allowing only weights on Integration and Relevance (experiment 12). A simple explanation is that, while it satisfies entirely Relevance and, almost totally, Integration, it has less topology and less Unpacking (ears don't ever relate to fly in the bird domain).

Another creature to report is the "flying snout" (which appeared in 23% of the runs of configuration 13, see table 9), a snout that has all the features of the bird. This is a "weak" blend in the sense that an isolated concept (the "horse snout") gets projected to the "bird" structure without any surrounding support such as its shape or its purpose. The third creature is the transport bird, which has all the features of the bird, but also carries humans, it serves for cargo and traction. It appeared occasionally during the previous experiments, but was triggered now by the frame "transport_means" in the query (in configuration 20), meaning indeed we had it in mind. Yet, its appearance throughout the tests (only when dealing with mapping 3, though) led us to include it in this section. The fourth creature is an oviparous horse, with two legs (instead of four), two wings and claws. It appeared in less than 10% of the results in the configuration 20, but it was the one that got the highest score.

In configurations 14, 15, 21 and 22, the results were essentially copies of the "bird" concept map, whereas 19 and 21 yielded highly unstable partial projections of both the "horse" and the "bird" concept maps simultaneously to the blend, since each of the 30 runs returned a different concept map. In the latter, we find it difficult to interpret anything. A possible explanation for these unsuccessful configurations is that the frames used are too much abstract, leaving no concrete goal to the system.

Exp.	Weights					Query	Mapping
#	I	T	MVR	U	R		
11	8.5	4	2.5	1	9	new_ability+aframe+aprojection	1
12	8.5	0	0	0	9	new_ability+aframe+aprojection	1
13	8.5	0	0	0	9	new_ability+ bprojection + bframe	1
14	8.5	4	2.5	1	9	new_ability + bprojection + bframe	1
15	8.5	4	2.5	1	9	bprojection + bframe	1
16	8.5	4	2.5	1	9	new_ability + bprojection + bframe	3
17	8.5	4	2.5	1	9	bprojection+ aframe	1
18	8.5	4	2.5	1	9	bprojection+ aframe	3
19	8.5	4	2.5	1	9	aprojection+ bframe	3
20	4	4	2.5	1	10	transport_means+bframe+bprojection	3
21	4	4	2.5	1	10	transport_means+bframe+bprojection	1
22	4	4	2.5	1	10	transport_means+bframe+bprojection	5

Table 7: Parameters for configurations 11 to 22

These ad-hoc experiments reveal that the system can produce novel concepts, yet it also demonstrates clearly that we face a very large search space, demanding a serious reflection about the tuning of the system.
It is the capacity to create novel and valid (with regard to the queries) creatures that testifies the potential of this model towards computational creativity. On one hand, it surely allows the creation of new concepts, a vital feature of a creative process, but on the other hand, the ultimate control always needs to be parameterized by a user (or another system?). There seems to be a paradox here: one must orient the system towards novelty and usefulness, but if doing so exhaustively, the emergent *creativity* is set *a priori* by the parameters. Yet, this apparent paradox seems to be present in discussions around creativity regarding issues like intentionality or evaluation. In fact, the boundaries between what is and what is not a creative product are very controversial and fragile. In our case, this boundary may lie within the level of abstractness given in the specification, which should comprise the mandatory conditions (e.g. specific frames) and more abstract preferences (e.g. abstract frames, like *aframe*).

6 Discussion

As we expected, the experiments raised several fundamental issues, some of which demanding a short reflection. Does the system agree totally with the Conceptual Blending framework? Does this system implement any kind of Computational Creativity? What can we expect from this model?
Since Fauconnier and Turner do not present a formal perspective on Conceptual Blending, it is not straightforward to validate our work in this respect. Starting from the representation of a mental space, we decided for a static, generic notion, the *domain*. We believe the representation we use (or an extension of it) could lead to mental spaces in general, but we are not confident to claim so much yet. This reduction of the knowledge basis of Conceptual Blending - the mental space - brings, *a priori*, limitations to our model. If successful, it should be able to produce the specific types of blends that result from blending static knowledge, such as domains, as opposed to dynamic knowledge, such as we have in discourse. In the latter case, we would need to extend our language to consider modalities, tense, mood, perspectives, or any other subjective, pragmatic or circumstantial components of discourse. Assuming that concepts, like “horse” and “bird”, can be validly defined by domains (those of “horses” and “birds”, from a common sense perspective), our model is expected to generate new concepts, like “horse-bird”, described in the same language. Above all, this “horse-bird” must be understandable from *a)* the chain of explanatory connections that appear in the new domain; *b)* the reference to the input domains, in the end-points of the explanatory connections. This agrees with the notions of *emergent structure* and *web of connections* that are present in the fundamentals of Conceptual Blending.
In (Pereira and Cardoso, 2002b) and (Pereira and Cardoso, 2002a), we discussed the potential of this framework from the point of view of Computational Creativity, namely in transforming the search space by changing the meta-level description of a domain. We showed that, having a level of instances (in the example of that paper, visual objects of a “house” and a “boat”), a theory for explaining the concepts involved in them, and assuming these instances as the search space for the problem (in the example, “drawing a house” or a “boat”), it is possible to obtain new ideas via blending the theories. After generating a blend (with the first version of this model, formally described in (Pereira and Cardoso, 2001)), the system reinterpreted each of the instances according to the new

relations (e.g. a house with a round window). There were no criteria for assessing the value of the blends or even selective projection. The idea was to generate the whole new search space at the instance level starting from different mappings. Currently, we focus on domain theories and on the evaluation of the blends via optimality pressures, leading to further conclusions about the creative aspects of this model. Creativity has, without controversy, two important aspects: novelty and usefulness. The work described in (Pereira and Cardoso, 2002b) and (Pereira and Cardoso, 2002a) is centered on novelty, leaving the task of choosing the "useful" results a responsibility of the search procedure. The combination of the two could then be novel and useful. A step further, the model we present now brings two components that may be valuable for usefulness: the frames and the optimality pressures. Frames provide low-level specifications or directives that should be valued in the blend, whereas the 8 optimality pressures work as high-level directives that allow the system to evaluate each blend according to several aspects. Thus, without having to exhaustively specify the query, it is possible to generate a novel concept that conforms a set of constraints. From the assumption that the ability to create concepts is a factor of creativity, we argue that ours is a computational model of creativity.

7 Acknowledgements

The authors would like to thank the reviewers for the patience and effort they spent for giving their very constructive commentaries for this paper.

References

Andersen, C. F. (1996). *A Computational Model of Complex Concept Composition*. University of Texas at Austin.

Costello, F. J. and Keane, M. T. (2000). Efficient creativity: Constraint-guided conceptual combination. *Cognitive Science*, 24(2):299–349.

de Saussure, F. (1983). *Course in General Linguistics*. London:Duckworth.

Fauconnier, G. and Turner, M. (1998). Conceptual integration networks. *Cognitive Science*, 22(2):133–187.

Fauconnier, G. and Turner, M. (2002). *The Way We Think*. Basic Books.

Freeman, M. (1999). The role of blending in an empirical study of literary analysis. In *Proceedings of the 6th International Cognitive Linguistics Conference*.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.

Hutchins, E. (2002). Material anchors for conceptual blends. *forthcoming*.

J. Bateman, B. M. and Fabris, G. (1995). The generalized upper model knowledge base: Organization and use. *Towards very large knowledge bases: knowledge building and knowledge sharing*.

Lakoff, G. and Nunez, R. (2000). *Where Mathematics Comes From: How the Embodied Mind brings Mathematics into Being*. New York:Basic Books.

Lenat, D. B. (1995). Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11).

Miller, G. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, 38(11).

Murphy, G. L. and Medin, D. L. (1985). The role of theories in conceptual coherence. *Psychological review*, (92).

Ogden, C. K. and Richards, I. A. (1923). *The Meaning of Meaning*. London: Routledge and Kegan Paul, Ltd.

Pereira, F. C. (1998). Modelling divergent production: a multi domain approach. In *European Conference of Artificial Intelligence, ECAI98*. IOSPress.

Pereira, F. C. (2003). Experiments with free concept generation in divago. In A. Cardoso, C. B. and Gero, J., editors, *Proceedings of the Third workshop on Creative Systems*. IJCAI. Forthcoming.

Pereira, F. C. and Cardoso, A. (2001). Knowledge integration with conceptual blending. In *Proceedings of AICS 2001*.

Pereira, F. C. and Cardoso, A. (2002a). The boat-house visual blending experience. In *Proceedings of the Symposium for Creativity in Arts and Science of AISB 2002*. AISB.

Pereira, F. C. and Cardoso, A. (2002b). Conceptual blending and the quest for the holy creative process. In *Proceedings of the Symposium for Creativity in Arts and Science of AISB 2002*. AISB.

Pereira, F. C. and Cardoso, A. (2003). Optimality principles for conceptual blending: A first computational approach. In *AISB'03 Symposium on AI and Creativity in Arts and Science*. SSAISB.

Veale, T. (1997). Creativity as pastiche: A computational treatment of metaphoric blends, with special reference to cinematic "borrowing". In *Proceedings of the Mind II: Computational Models of Creative Cognition*. Dublin, DCU.

Veale, T. and Keane, M. (1993). A connectionist model of semantic memory for metaphor interpretation. In *Neural Architectures and Distributed AI Workshop*. Centre for Neural Engineering, U.S.C. California.

Veale, T. and O'Donogue, D. (2000). Computation and blending. *Cognitive Linguistics*, Special Issue on Conceptual Blending.