In *Proceedings of the 2nd Workshop on Creative Systems: Approaches to Creativity in AI and Cognitive Science*, ECAI 2002, Lyon, France, 2002.

The Boat-House Visual Blending Experiment

Francisco Câmara Pereira and Amílcar Cardoso

camara@dei.uc.pt, amilcar@dei.uc.pt

Centro de Informática e Sistemas da Universidade de Coimbra (CISUC)

Departamento de Engenharia Informática, Polo II – Pinhal de Marrocos

3030 Portugal

Abstract

The main subject of this paper is to describe an experiment of blending two domains using our previous work, Blender (Pereira and Cardoso, 2001), a computational attempt to model Fauconnier and Turner's (98) Conceptual Blending. In this experiment, we visualize the resulting instances using a logo-like language, allowing us to produce visual blends. We discuss the results and suggest further improvements.

Introduction

In (Pereira and Cardoso, 2001), we proposed and formalized a framework that has the ultimate goal of modeling Conceptual Blending (CB) theory (Fauconnier and Turner, 98). We then implemented it and present now the results of our experiment on blending the domains of *houses* and *boats*. Being the first results of such a recent model, they contribute more to reflections and discussions on this approach than for demonstrating its value as a finished work.

Blender is part of a wider project, Dr. Divago (Pereira and Cardoso, 99), in the area of Computational Creativity research.

In this experiment, we generated exhaustively all the resulting instances, enabling us to see the search space that may become available for systems that recur to domain knowledge, such as the ones of Case Based Reasoning or Expert Systems.

In the next section, we will talk about some interesting works that relate to Conceptual Blending and Computation. Then, in section 3, we will give a brief overview of Conceptual Blending theory. Section 4 is dedicated to describe our model and in section 5 we show the results. Finally, the last section is dedicated to a short discussion on the results and issues that emerged in this research.

Conceptual Blending and Computation

Conceptual Blending was initially proposed by (Fauconnier and Turner, 98), and its value has been increasingly acknowledged as a wider range of researchers is becoming interested in studying it. The works of (Mandelblit,97), (Sweetser and Dancygier, 99), (Coulson, 97) and (Veale and O'Donogue, 2000) are examples of how this theory is an important contribution to Linguistics, Creative Cognition, Analogy and Metaphor.

To explain it in some detail, we must introduce the concept of Mental Space. According to (Fauconnier and Turner, 98), Mental Spaces are partial structures that proliferate when we think and talk, allowing a fine-grained partitioning of our discourse and knowledge structures. As we talk or think, our reasoning focus flows from space to space, transporting and mapping concepts according to points of view, presuppositions, beliefs, changes of mood or tense, analogical counterfactuals and so on, each giving birth to a different mental space.

Blending is generally described as involving two input mental spaces that, according to a given structure mapping, will generate a third one, called Blend. This new domain will maintain partial structure from the input domains and add emergent structure of its own.

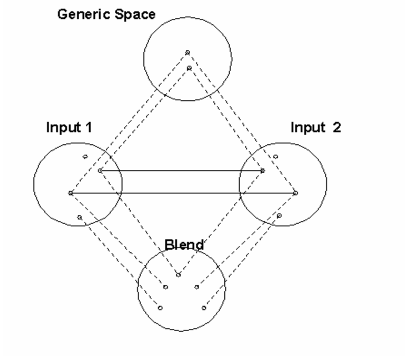

Figure 1 – Conceptual Blending Theory

As can be seen in figure 1, a generic space is also considered. This can be seen as having a unification role, such that concepts mapped onto each other are considered as belonging to the same, generic, concept.

Some examples of blends are: the title of this paper, a blend of the "the quest for the holy grail" and "research towards creative processes"; evolutionary computation, a blend of "natural evolution theories" and "problem solving in computation"; swatch, blend of "swiss" and "watch"; Mussorgsky's "pictures of an exhibition" and many others. In these examples, like in many others, we can see the presence of Metaphor. For a discussion on Metaphor and Blending, read (Grady et al, 99).

The Blend has emergent structure not provided by the inputs. This happens in three (unrelated) ways (Fauconnier, 97):

1. Composition - Taken together, the projections from the inputs make new relations become available that did not exist in the separate inputs

2. Completion - 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.

3. Elaboration - The structure in the blend can then be elaborated. This is "running the blend". It consists of cognitive work performed within the blend, according to its own emergent logic.

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 behaviour, expectation or common sense. In other words, there are several issues clearly hard to model. Yet, the intersection of AI and CB may bring, if not the computational model of the theory, at least processes or algorithms that may extend the power of AI in problem solving. In our opinion, this is especially true for the field of Computational Creativity.

The Blender

In (Pereira and Cardoso, 2001), we find a detailed formalization of a simple blending architecture: the Blender. This project is part of a major framework, Dr. Divago[1] (Pereira and Cardoso, ??), a model of creativity inspired on the idea of Divergent Thinking (Guilford, 67). In Dr. Divago, we propose the use of a multi-domain knowledge base as a problem solving resource. When searching for a solution, Dr. Divago may *diverge* to a domain different from that in focus, through the use of a mapping function that works as a cross-domain bridge, or to a domain blend, i.e., a domain that results from blending the domain in focus with a different domain, a process discussed in this paper.

Outside of Dr. Divago framework, we conceive the Blender as a way of generating new concepts, an idea we believe very valuable for creativity research. This paper presents essentially this facet of the Blender. Another, opposite, application of this work would be in the interpretation of concepts if we reverse the direction of process, i.e., instead of generating a new domain out of two input domains, we could interpret a concept in the light of sets of domains. Still considering Creativity, this probably more complex task would be valuable for the problem of evaluation since it could search for some possible meaning or interpretation for concepts. Imagine a program that generates new artifacts, the Blender could *explain* some of them in the light of the input domains it perceived. We didn't study yet the applicability of this idea, but we hope it prompts some divergent thinking in the reader's mind.

The research on Blender started by a first formalization of the Conceptual Blending theory (Pereira, 2001). We are aware we left out several important issues, and so we cannot say it is a complete and definitive formalization of the theory. We prefer to see it as a systematization of some fundamental issues like the mapping function, the blending projection and the definition of domain. With these first developments, we were able to make

[1] In Portuguese, "Divago" means "I wonder"

some tests in domains and discuss practical issues around the task of implementing CB.

The first issue to be raised was the problem of interpretation and semantics of a blend.

The semantics definition we assumed to the blend presupposes its self-containment. Meaning of concepts would be defined by the relations and surrounding concepts, ultimately linked to the generic space. Thus, concepts like "house|sailing boat"[2] are defined exclusively according to its connections to other concepts: "house|sailing boat" is a "physical structure|boat" and it has a "roof|mast", among other things. While this may make sense (in fact, the blend of the two concepts can result from the blend of all their related sub-concepts) and be non-contradictory with our semantics definition (in fact, concepts always mean something, within the blend), it becomes very indeterminate and arbitrary to interpret specially when the chain of concepts doesn't achieve unambiguous points ("physical structure" is unambiguous, as opposed to "physical structure|boat"). This is particularly troublesome when one wants to realize the content in some practical way, like drawing the concept such as we will see in this paper. In the final section, we do a critical evaluation of this problem and point out several issues and solutions.

The need for interpreting and *visualizing* the inside of a blend lead us to the experience we are showing in this paper in which we projected the generated space of instances, subsumed in blend. This way, we could test the generative potential of the current model.

In this work, a domain comprises two different kinds of knowledge: domain theory and domain instances. Domain theory consists of a declarative description of the concepts by means of rules and a simple concept graph with binary relations between concepts (to which we call *concept maps*). Domain instances are examples of artifacts of the domain in question.

A mapping function is also fundamental for the blend to be built and we are now using a structure-matching algorithm, inspired in Tony Veale's Sapper (Veale, ??). We intend to work further on this issue, exploring other mapping algorithms.

[2] The signal "|" means co-reference, i.e., x|y means that and x and y both refer to the same concept (say the concept xy). The operator is commutative,so x|y ⇔ y|x

As referred before, this system is formally described in (Pereira and Cardoso, 2001) and is currently implemented under XSB-prolog (??).

Blender: the house-boat experiment

To have an insight on the potential concept space generated by Blender, we decided to take the example of Goguen (99), the "house" and "boat" blend. The goal was to blend these two domains and interpret the newly generated instances according to an unambiguous process. In this case, we decided to define them according to a simple language (logo (ref)), which enabled us to draw simple objects (a house and a boat) and see the generated space without heavy computational work.

For domain knowledge, we built simple concept maps for each domain (using Clouds (Pereira and Cardoso, 2000)), having only superficial knowledge, as shown in the excerpts in figures 2 and 3.

The house concept map

```
isa(house,physical_structure).
isa(human,mammals).
isa(mammals, animals).
isa(animals, living_entity).
isa(physical_structure, physical_entity).
isa(night, time_object).
isa(time_object, information_entity).
isa(skyscrapper, physical_structure).
isa(door, physical_object).
isa(window, physical_object).
isa(roof, physical_object).
isa(observation, task).
isa(entrance,task).
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).
purpose(roof, protection).
purpose(body, container).
have_many(skyscrapper, house).
have_many(house, room).
property(skyscrapper, very_big).
```

Figure 2 – The house domain concept map

```
The boat 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).
```

Figure 3 – The boat domain concept map

A short interpretation of these concept maps tells us facts like "a sailing 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 body ".

For domain instances, we coded drawings of a house and a boat in simple logo-like language (ref), having commands like on/5 ("draw line for 5 pixels"), off/5 ("move 5 pixels without drawing") or left/45 ("turn left 45 degrees"). The logo-representation of the "boat" and the "house" are shown in figures 4a and b. They follow a hierarchical case representation, in prolog-like form:

case(case name, node address, node name, commands).

As we will see, this representation is structured top to bottom (the attribute "son" indicates the descendants of a node), with each level adding 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); some shapes are pre-defined (e.g., parallelogram_boat, oval, rectangle, etc.); 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. Since this representation is just a simple choice for representing drawing domain instances, as many others could be, we don't think it is worthwhile to detail further in this paper.

```
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_stru
cture,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]]).
```

Figure 4a – instance representation of boat

```
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]]).
```

Figure 4b – instance representation of house

The interpretation of these instances gives the drawings of figure 5.

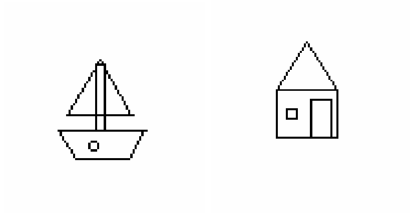

Figure 5 – the boat and the house, as specified in figures 4a and b

We made several runs, with random mapping starting points[3] and a general comment on the

[3] Different seeding pairs (x,y), as described in (Pereira and Cardoso, 2001)

results is that, as we expected, a big amount of different instances appeared, with some emerging patterns.

Among the 4 generated mappings, the one below in figure 6 was the most common to appear.

```
entrance<-M->movement
task<-M->task
protection<-M->support
roof<-M->mast
door<-M->sail
house<-M->sailing_boat
physical_structure<-M->boat
window<-M->hatch
body<-M->vessel
water_proof<-M->slow
container<-M->container
observation<-M->observation
```

Figure 6 – A house-boat mapping

While some concept mappings come naturally (like "window-hatch" or "body-vessel"), others, less intuitively acceptable, appear as consequence of the exhaustiveness of the mapping function. For example, "water_proof-slow" comes because both can be "properties" of something (e.g. "physical_structure can be water_proof", and "boat can be slow"). Below, we can see the concept map of the new domain.

```
isa(entrance|movement, task).
purpose(door|sail, entrance|movement).
blended_with(task, task).
blended_with(protection, support).
isa(protection|support, task).
purpose(roof|mast, protection|support).
isa(protection|support, task).
blended_with(roof, mast).
isa(roof|mast, physical_object).
have(house|sailing_boat, roof|mast).
purpose(roof|mast, protection|support).
blended_with(physical_object, physical_object).
blended_with(door, sail).
isa(door|sail, physical_object).
have(house|sailing_boat, door|sail).
isa(door|sail, physical_object).
shape(door|sail, triangle).
purpose(door|sail, entrance|movement).
blended_with(house, sailing_boat).
isa(house|sailing_boat, physical_structure|boat).
have(house|sailing_boat, window|hatch).
have(house|sailing_boat, body|vessel).
have_many(house|sailing_boat, room).
property(house|sailing_boat, water_proof|slow).
live_in(human, house|sailing_boat).
have_many(skyscrapper, house|sailing_boat).
have(house|sailing_boat, door|sail).
have(house|sailing_boat, roof|mast).
place(house|sailing_boat, sea).
use(human, house|sailing_boat).
sail(human, house|sailing_boat).
blended_with(physical_structure, boat).
isa(physical_structure|boat, physical_structure).
isa(house|sailing_boat, physical_structure|boat).
blended_with(window, hatch).
isa(window|hatch, physical_object).
purpose(window|hatch, observation).
isa(window|hatch, physical_object).
shape(window|hatch, circle).
property(window|hatch, tiny).
have(house|sailing_boat, window|hatch).
blended_with(body, vessel).
isa(body|vessel, physical_object).
purpose(body|vessel, container).
isa(body|vessel, physical_object).
have(body|vessel, floating_structure).
have(house|sailing_boat, body|vessel).
```

Figure 7 – The blend concept map

According to the self-contained semantics discussed in the previous section, we can read 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.

Apart from definitions of shape that emerge in the blend (like, "shape(door|sail, triangle)"), we don't know exactly what is the visual blend of objects (e.g., what is the visual shape of "door|sail"?), so their visual interpretation will be either corresponding to the original one in each of the domains or blank ("door|sail" can be represented in the new objects by a door, a sail or nothing). This was an inevitable choice because we needed to stop in some level of abstraction (otherwise, we had to blend logo representations themselves). As a result of this, a large set of visual proposals for the house|boat blend appear.

The mapping presented in figure 6 generates a about 80 different images, such as those shown below:

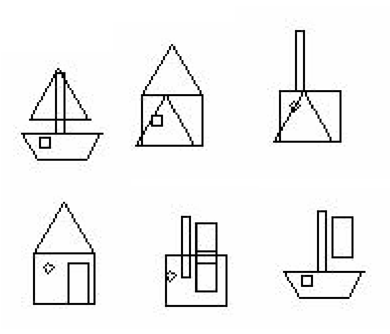

Figure 8 – Images that result from mapping of figure 6.

Analysing these house|boats, we can see that it makes sense to have a window|hatch with window shape (first image) or with a hatch shape (third and fourth images). A door|sail with the shape of a triangle also agrees with the definition (second and third images), as didn't happen with the last three images (all have rectangular door|sails). It is also of relevance to say that there are also strange instances that appear as consequence of not having specific domain-knowledge for generating a drawing or just because of unfortunate combinations (second, third and fifth images). For example, if we had a rule for making a triangle "fit" in a square, some images would result better (second and third).

Now for a quick overview of the other mappings (which generated approximately the same amount of different images as the one above), figure 8 shows two examples of the blend that has the concepts "roof|mast" and "window|hatch".

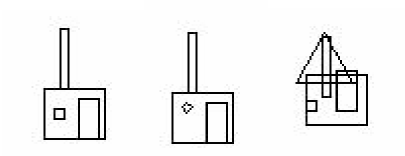

Figure 8 – Two examples of the "roof|mast" and "window|hatch" blend

And now a blend that has the concepts "body|sail", "door|hatch" and "window|mast".

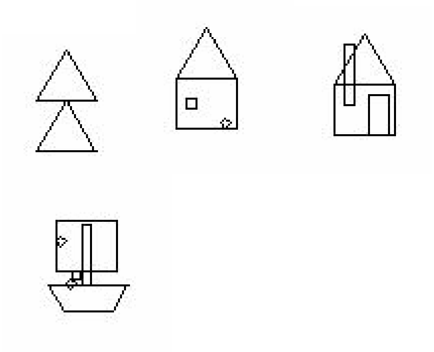

Figure 9 – Images from the "body|sail", "door|hatch" and "window|mast" blend.

Finally, a blend with "roof|mast", "door|hatch" and "window|sail".

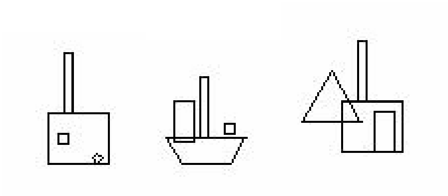

Figure 10 – images from the "roof|mast", "door|hatch" and "window|sail".

Results and Discussion

First of all, we are aware that the visual quality of the results can vary lot depending on the application of effort on domain specific knowledge, such as guiding the result to what a house or a boat should seem like or which physical/structural rules should they fulfill. Our goal with this experiment was to test the capacity of generating new and different instances coherent with the blend, regardless of any aesthetical judgment. In general, we can say this was achieved and that, although these two domains are close to each other (both are physical structures, used by humans), this is a reasonable example of computational modeling of divergent thought. A big amount of new instances was generated not attached to a single domain and consistent with the defined blend. We remind that the goal of Dr. Divago is to model divergent thought in problem solving by proposing solutions outside the domain in focus. E.g., in the problem of designing a new house, it would be able to propose a concept that came from the house|boat blend, presenting its drawing and defining the meaning of its components according to the generated new theory (although, as discussed before, this is not an unambiguous issue).

We found our concept maps too simplistic, leading to mappings in which too few interconnected concepts supported correspondences, i.e., concepts could be blended together just because they share a common, general feature (e.g., *door* and *sail* only share the fact that they are *physical objects*). If more specific knowledge were brought, stronger connections would surpass these candidate mappings. As a result of that, we found some nonsensical drawings (such as the last ones in 8 and 9), estimated at about 40% of the overall generated results. We believe more detailed domain theory would enhance this performance.

Another question we can point out is the lack of concern on the Optimality Constraints (??), which are central to the Conceptual Blending theory. Up to now, there is a free projection of concepts, allowing us to see the limits of our model but showing us that it is incomplete in which regards to CB theory. In (Pereira and Cardoso, 2001) there is a short reference to the Optimality Constraints that the Blender fulfills, but we expect a more explicit and precise approach will be followed in near future.

As a result of the free projection, each mapped concept is co-referent with another one, which not always is the more correct (e.g., while it may seem correct to project the mapping of *window* and *hatch*, it is not so to project *physical structure* and *boat* or *water proof* and *slow*). Yet, this correctness bias has the risk of being too arbitrary because, when forcing our view of *house|boats*, we may be cutting the blend in important points (why should we say *water proof|slow* is less correct that *window|hatch*?).

As the reader could see, even in concepts that now we consider correct, like *window|hatch*, there is still ambiguity in its visual realization (is it a circle or a square?). In this experiment, we had to make the compromise of allowing both, in a disjunctive approach (either it is round or square), because we didn't define the blend at the logo-language level.

Conceptual Blending theory brings the notion of Organizing Frame, "a frame that specifies the nature of the relevant activity, events, and participants" (Fauconnier and Turner, 98). Organizing frames can be the missing glue for the creation of meaningful blends. While our blends are valuable to our goal of modeling divergent thinking, they are fragile in the sense that there is not always a *good reason* (Fauconnier and Turner, 98), for each concept. Next development of Blender will have organizing frames as a structuring role in the construction of new domains.

The introduction of organizing frames would help solving ambiguities and clarify the semantics of a blend since it presupposes a situation or a specific concept map that aims to a goal. An organizing frame, we are sure, would strengthen the blend once concepts that *don't fit* either don't get mapped or aren't projected. For example, in the house-boat blend, concepts like "entrance|movement" wouldn't appear once we used a "house" frame (this frame would specify the expected characteristics in a house: it must clearly contain an entrance, whatever its shape).

References

(Coulson and Oakley, 2000) Seana Coulson and Todd Oakley, Blending Basics. *Cognitive Linguistics 11*. 2000

(Coulson, 97) Seana Coulson, *Semantic Leaps: The Roleo of Frame-Shifting and Conceptual Blending in Mapping Construction*, UC San Diego, 1997

(Fauconnier and Turner, 98) Gilles Fauconnier and Mark Turner, Conceptual Integration Networks, *Cognitive Science*, 1998

(Fauconnier, 97) Gilles Fauconnier, *Mappings in Thought and Language*, Cambridge University Press, 1997

(Goguen, 99) Joseph Goguen, An Introduction to Algebraic Semiotics, with Applications to User Interface Design, *Computation for Metaphor, Analogy and Agents in Springer Lecture Notes in Artificial Intelligence*, Springer, 1999

(Grady et al, 99) J. Grady, T. Oakley and S. Coulson. Blending and Metaphor. In *Metaphor in Cognitive Linguistics.* G. Steen and R. Gibbs (Eds.). John Benjamins, 1999

(Guilford, 67) J. P. Guilford *The Nature of Human Intelligence*. New York: McGraw-Hill. 1967.

(Pereira and Cardoso, 2000) F. C. Pereira and A. Cardoso. A Module for Automatic Learning of Concept Maps. In *Proceedings of Diagrams 2000*. Springer. 2000

(Pereira and Cardoso, 2001) F. C. Pereira and A. Cardoso. Knowledge Integration with Conceptual Blending. In *Procs. of 12th Irish Conference on Artificial Intelligence & Cognitive Science*, AICS 2001. 2001.

(Veale and O'Donogue, 2000) Tony Veale and Diarmuid O'Donogue, Computation and Blending, in *Cognitive Linguistics* (special Issue on Conceptual Blending). 2000

(Veale, 97) Tony Veale, Creativity as Pastiche: A computational treatment of metaphoric blends, with special reference to cinematic "borrowing", in *Procs. of Mind II: Computational Models of Creative Cognition,* 1997