Style as a Choice of Blending Principles

Joseph A. Goguen and **D. Fox Harrell**
Department of Computer Science & Engineering
University of California, San Diego

Abstract

This paper proposes a new approach to style, arising from our work on computational media using *structural blending*, which enriches the conceptual blending of cognitive linguistics with structure building operations in order to encompass syntax and narrative as well as metaphor. We have implemented both conceptual and structural blending, and conducted initial experiments with poetry generation, although the approach generalizes to other media. The central idea is to analyze style in terms of blending principles, based on our finding that different principles from those of common sense blending are needed for some creative metaphors.

1 Introduction & Background

James Meehan's 1976 TALE-SPIN (Meehan 1981) was perhaps the first computer story generation system. It explored the creative potential of viewing narrative generation as a planning problem, in which agents select appropriate actions, solve problems in the simulated world, and output logs of their actions using syntactic templates. Here is a sample:

> Henry Squirrel was thirsty. He walked over to the river bank where his good friend Bill Bird was sitting. Henry slipped and fell in the river. Gravity drowned.

The logic behind this non-sequitur is impeccable: Gravity is pulling Henry into the river, and it has no friends, arms, or legs that can save it from the river; therefore Gravity drowns. But we know Gravity is not something that can drown; there is a startling type check error here. Subsequent systems were better, but still mainly followed "good old fashioned AI," which assumes human cognition is computation over logic-based data structures, and which largely ignores (or even denies) the embodied and socially situated nature of being human. Such systems lack elegance and style. But how can we do better? And what is style anyway?

While building our poetry generation system (Section 3.3) and the blending algorithms at its core (Sections 3.1 and 3.2), and in considering blends that appear in recent poetry (Section 3.4), we found that very different principles from those proposed in (Fauconnier & Turner 2002) for conventional, common sense blends are needed. This led us to analyze style in terms of the blending principles used to generate various kinds of work, as discussed in Section 3.5. Future projects discussed in Section 4 include interactive poems, and computer games that generate new plot as they are played. Some theoretical background needed for our approach is briefly reviewed in the following subsections, and an approach to cognitive grammar is sketched in Section 3.2.

1.1 Metaphor & Blending

Gilles Fauconnier and Mark Turner have developed a theory within cognitive linguistics known as **conceptual blending** or **conceptual integration** (Fauconnier & Turner 2002). In this theory, **conceptual spaces** are relatively small, transient collections of concepts, selected from larger domains for some purpose at hand, such as understanding a particular sentence; this basic notion builds upon Fauconnier's earlier notion of mental spaces, which are formally sets of "elements" and relation instances among them (Fauconnier 1985). George Lakoff, Mark Johnson, and others (Lakoff & Johnson 1980; Lakoff 1987) have studied metaphor as a mapping from one conceptual space to another, and shown that metaphors come in families, called **image schemas**, having a common theme. One such family is MORE IS UP, as in "His salary is higher than mine," or "That stock rose quite suddenly." The source UP is grounded in our experience of gravity, and the schema itself is grounded in everyday experiences, such as that having more beer in a glass, or more peanuts in a pile, means the level goes up. Many image schemas, including this one, are grounded in the human body, and are called **basic image schemas**; these generally yield the most persuasive metaphors, and are also useful in other areas, such as user interface design (Goguen 1999).

Blending two conceptual spaces yields a new space that combines parts of the given spaces, and may also have emergent structure (Fauconnier & Turner 2002). Simple examples in natural language are words like "houseboat" and "roadkill," and phrases like "artificial life" and "computer virus." Blending is considered a basic human cognitive operation, invisible and effortless, but pervasive and fundamental, for example in grammar and reasoning. It also gives a new way to understand metaphor. For example, in "the sun is a king," blending the conceptual spaces for "sun" and "king" gives a new **blend space** together with **conceptual mappings** to it from the "king" and "sun" spaces. Although there is no direct mapping between the two original spaces, there are "cross space" identifications, certainly including the identification of the "sun" and "king" elements, so that they are the same element in the blend space. **Metaphoric blends** are *asymmetric*, in that the **target** of the metaphor is understood using only the most salient concepts from the other **source** space (Grady, Oakley, & Coulson 1999). For example, aspects of "king" may be *blocked* from mapping to the blend space – usually the sun does not wear a crown or charge taxes. Additional information needed to construct a coherent blend may be recruited from other spaces, as well as from **frames**, which encode highly conventionalized information. **Conceptual integration networks** are networks of conceptual spaces and conceptual mappings,

used in blending the component spaces for situations that are more complex than a single metaphor. These are of course the norm in literary texts.

1.2 The Classical Optimality Principles

Below are five optimality principles from the most recent version of this list (Fauconnier & Turner 2002); they are used to determine which among many possible blends is most appropriate for a given situation:

1. *Integration:* The scenario in the blend space should be a well-integrated scene.
2. *Web:* Tight Connections between the blend and the inputs should be maintained, so that an event in one of the input spaces, for instance, is construed as implying a corresponding event in the blend.
3. *Unpacking:* It should be easy to reconstruct the inputs and the network of connections, given the blend.
4. *Topology:* Elements in the blend should participate in the same kinds of relation as their counterparts in the inputs.
5. *Good Reason:* If an element appears in the blend, it should have meaning.

These principles apply to common sense blends, such as are typically found in ordinary language, in advertisements, etc.; Section 3.4 will show that they do not apply to generating some unconventional language, such as certain metaphors in Pablo Neruda poems. All five of these optimality principles require human judgement, and so cannot be implemented in any obvious way. However, when the relations involved are identities, the Topology Principle does not involve meaning, and so can be implemented; indeed, it is part of the blending algorithm described in Section 3.1.

1.3 Narrative

In many games and art works, narrative provides a deeper and more satisfying sense of involvement. Temporal and causal succession are essential for narrative, but values also play a key role, by connecting events in the story to the social worlds and personal experiences of users. These two aspects of narrative provide the sense that a work is "going somewhere" and "means something," respectively. Sociolinguists, e.g. (Labov 1972; Linde 1993), have done extensive empirical study of narratives of personal experience, which are told orally to a group of peers under natural conditions. The following briefly summarizes this research:

1. There is an optional **orientation section**, giving information about the setting (time, place, characters, etc.) for what will follow.
2. The main body is a sequence of **narrative clauses** describing the events of the story; by a default convention called the **narrative presupposition**, these are taken to occur in the same order that they appear in the story.
3. Narrative clauses are interwoven with **evaluative material** relating narrative events to values.
4. An optional **closing section** summarizes the story, or perhaps gives a moral.

The interpretation of narrative also employs the **causal presupposition**, which says that, other things being equal, given clauses in the order A, B we may assume that A causes B. An additional principle is **accountability**, that the person telling such a story must establish to the audience the relevance of the actions reported. This is accomplished by *evaluative material*, which relates narrative events to social values shared by the narrator and audience; it provides a warrant for inferring the values involved.

The above assertions are thoroughly grounded in empirical research on contemporary American small groups, but appear to apply more broadly to contemporary Western languages[1]. Although developed for oral narratives of personal experience, the theory also yields insight into many other media and genres, such as novels and human computer dialogues, because their structure has a basis in oral narratives of personal experience.

It may be surprising that values are an integral part of the internal structure of stories, rather than being confined to a "moral" at the end; in fact, values pervade narrative, as justifications for the narrator's choice of what to tell, or a character's choice of what to do, as well as via modifiers such as "very" or "slightly." The default narrative presupposition can be overridden by explicit markers of other temporal relations, such as flashbacks and flashforwards, so that even narratives that involve multiple times, multiple places, or multiple narrators, are still composed of subsequences that conform to the above structure.

The purely structural aspects of this theory can be formalized as a grammar, the instances of which correspond to the legal structures for narratives. The following uses so called EBNF notation,

```
<Narr> ::= <Open> (<Cls> <Eval>*)+ [<Coda>]
<Open> ::= ((<Abs> + <Ornt>) <Eval>*)*
```

where `[...]` indicates zero or one instance of whatever is enclosed, `*` indicates zero or more instances, infix `+` indicates exclusive or, superscript `+` indicates one or more instances, and juxtaposition of subexpressions indicates concatenation. Here `<Narr>` is for narratives, `<Cls>` for narrative clauses, which potentially include evaluation, `<Eval>` for stand-alone evaluative clauses, `<Open>` for the opening section, which may include an orientation and/or abstract, and `<Coda>` for the closing section.

Of course, BNF is far from adequate for describing many other aspects of narrative, e.g., coherence of plot, development of character, and dialogue. The above grammar also fails to address the variety of ways in which evaluation can occur. Some alternatives to explicit evaluative clauses include repetition of words or phrases (which serves to emphasize them), noticeably unusual lexical choices (which may serve to emphasize, de-emphasize, or otherwise spin something), and noticeably unusual syntactic choices (which also may serve to emphasize or de-emphasize).

2 Algebraic Semiotics & Structural Blending

It may help to first clarify our philosophical orientation, since mathematical formalisms are often given a status be-

[1] However, they do not necessarily apply to non-Western langauges and cultures; for example, Balinese narrative does not follow the narrative presupposition (Becker 1979).

yond what they deserve. For example, Euclid wrote, "The laws of nature are but the mathematical thoughts of God." However, our viewpoint is that formalisms are constructed by researchers in the course of particular investigations, having the heuristic purpose of facilitating consideration of certain issues in that investigation; theories are situated social entities, mathematical theories no less than others.

Section 2.1 below describes the semiotic space generalization of conceptual spaces and its origin in algebraic semiotics, Section 2.2 describes semiotic morphisms and structural blending, and Section 2.3 gives a detailed illustration of these ideas for the special case of conceptual spaces; the general case of structural blending is illustrated in Section 3.2 with text generation for Labov narrative structure.

2.1 Semiotic Spaces

Whereas conceptual spaces are good for concepts, they are inadequate for structure, e.g., how a particular meter combines with a certain rhyme scheme in a fixed poetic form; music raises similar issues, which again require an ability to handle structure. Thus, to use blending as a basis for stylistic analysis, we must generalize conceptual spaces to include structure; we do this by enriching conceptual spaces with structure building operations, called **constructors**, and with axioms to describe how these behave; we also allow hierarchical type systems. Thus conceptual blending differs from **structural blending** or **structural integration**, in allowing spaces that have non-trivial constructors.

Structural blending comes from **algebraic semiotics** (Goguen 1999), which uses algebraic semantics to describe the structure of complex signs (e.g., a music video with subtitles) and the blends of such structures. Algebraic semantics has its origin in the mathematics of abstract data types (Goguen & Malcolm 1996); its basic notion is a **theory**, which consists of type and operation declarations, possibly with subtype declarations and axioms. It is usual to use the word **sort** instead of "type" in this area.

A **semiotic system** (or **semiotic theory** or **sign system**) is an algebraic theory, plus a **level ordering** on sorts (having a maximum element called the **top sort**) and a **priority ordering** on the constituents at each level (Goguen 1999). Ordinary sorts classify the parts of signs, while **data sorts** classify the values of attributes of signs (e.g., color and size), and a **data algebra** contains values and operations for data sorts. **Signs** of a certain sort are represented by terms of that sort, including but not limited to constants. Among the operations are **constructors**, which build new signs from given sign parts as inputs. **Levels** express the whole-part hierarchy of complex signs, whereas **priorities** express the relative importance of constructors and their arguments; social issues play a key role in determining these orderings. Conceptual spaces are the special case with only constants and relations, only one sort, and only axioms asserting that certain relations hold on certain instances. See (Goguen 1999; 2003) for more detail.

Here is a simple semiotic theory for books: `Book` is the top sort, `Chapter` the secondary sort, `Head` and `Content` tertiary sorts, and `Title` and `PageNo` fourth level sorts. One constructor build chapters from their head and content, and another builds heads from a title and page number. Among the constituents of `Head`, `Title` has priority over `PageNo`, and among those for `Chapter`, `Head` has priority over `Content`. The grammar for narratives can also be considered a semiotic system. Its top sort is of course `<Narr>`; its second level sorts are `<Cls>`, `<Eval>`, `<Open>`, and `<Coda>`, while `<Ornt>` and `<Abs>` are third level sorts. Among the second level sorts, `<Cls>` has highest priority and `<Eval>` next highest.

Semiotic spaces, like conceptual spaces, are static. Fauconnier and Turner do not attempt to capture the behavior of dynamic entities, with changeable state, in their theory. However (given the necessary mathematics), it is not very difficult to extend semiotic spaces to include dynamic structures; in fact, such an extension is needed for applications to user interface design, and is carried out in detail, with examples, in (Goguen 2003), using so called hidden algebra. The conceptual blending theory of Fauconnier and Turner also does not assign types to elements of conceptual spaces; this makes sense, due to the very flexible way that blends treat types, but it also represents a significant loss of information, which in fact can be exploited in some interesting ways, such as being able to characterize some metaphors as "personifications" (see the discussion below) and being able to generate more striking and unusual blends by identifying sorts known to be far apart. Another difference from classical conceptual blending is that we do not first construct a minimal image in the blend space, and then "project" it back to the target space, but instead, we build the entire result in the blend space.

2.2 Semiotic Morphisms & Structural Blending

Mappings between semiotic systems are uniform representations for signs in a source space by signs in a target space, and are called **semiotic morphisms**; user interface design is an important application area for such mappings (Goguen 1999). Because sign systems are formalized as algebraic theories with additional structure, semiotic morphisms are formalized as theory morphisms that also preserve this additional structure. A theory morphism consists of a set of mappings from one theory to another that preserves the basic constituents, which are sort declarations, operation declarations, and axioms; semiotic morphisms in addition preserve levels and priorities. However, these mappings must be *partial*, because some sorts, constructors, etc. are not preserved in the intended applications. For example, the semiotic morphism from the conceptual space for "king" into the blend space for the metaphor "The sun is a king" discussed above does not preserve the throne, court jester, queen, and castle (unless some additional text forces it).

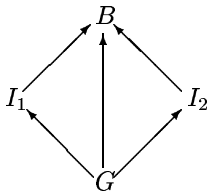

Figure 1: Blend Diagram

Semiotic morphisms are used in structural blending to establish connections between semiotic spaces to indicate which elements should be identified. The simplest form[2] of blend is shown in Figure 1, where I_1 and I_2 are called **input spaces**, and G is called a **base space**. We call I_1, I_2, G together with the morphisms $I_1 \to G$ and $I_2 \to G$ an **input diagram**. Given an input diagram, we use the term **blendoid** for a space B together with morphisms $I_1 \to B$, $I_2 \to B$, and $G \to B$, called **injections**, such that the diagram of Figure 1 **commutes**, in the sense that both compositions $G \to I_1 \to B$ and $G \to I_2 \to B$ are "weakly equal" to the morphism $G \to B$, in the sense that each element in G gets mapped to the same element in B under them, provided that both morphisms are defined on it. In general, all four spaces may be semiotic spaces; the special case where they are all conceptual spaces gives conceptual blends. We call the composition of the two morphisms on the left of Figure 1 its **left morphism**, the composition of the two morphism on the right its **right morphism**, to the middle upward morphism its **center morphism**, to the triangle on the left its **left triangle**, and the triangle on the right its **right triangle**. Since there are often very many blendoids, some way is needed to distinguish those that are desirable. This is what optimality principles are for, and a **blend** is then defined to be a blendoid that satisfies some given optimality principles to a significant degree. The blending algorithm of Section 3.1 uses optimality principles based only on the structure of blends, rather than their meaning; these include degrees of commutativity and of type casting.

2.3 An Example

We illustrate conceptual blending with the concepts "house" and "boat" shown in Figure 2. Each circle encloses a conceptual space, represented as a graph, the nodes of which represent entities, and the edges of which represent assertions that a certain relation, the name of which labels the edge, holds between the entities on its nodes. As in Figure 1, the bottom space is the generic or base space, the top is the blend space, and the other two are the input spaces, in this case for "house" and "boat." The arrows between circles indicate semiotic morphisms. In this simple example, all four spaces have graphs with the same "vee" shape, and the five morphisms simply preserve that shape, i.e., each maps the bottom node of the "vee" in its source space to the bottom node in its target. To avoid clutter, types are not shown, but in this case, it happens that the types correspond to the entity names in the generic space.

For this blend, the two triangles commute for all three sorts in the base space; similarly, the two base constants `object` and `person` are preserved. Thus we have commutativity for this blend, so that corresponding elements of the input spaces are identified in the blend; e.g., `house` and `boat` are identified in "houseboat", and the merged element is named `house/boat`. Similarly, the two relations in the base space map to the same relation in the blend via the three paths, so that the relations `live-in` and `ride` are identified. Finally, for each pair of elements in the base space for which a relation holds, the corresponding elements in the blend space satisfy the corresponding relation, which means that all three paths preserve the axiom in the same way.

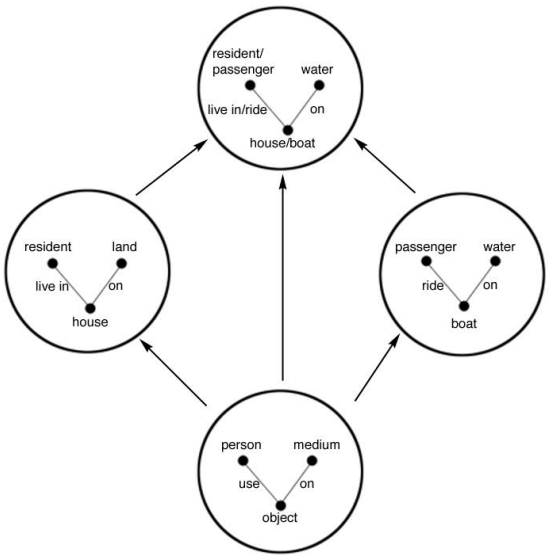

Figure 2: Houseboat Blend Diagram

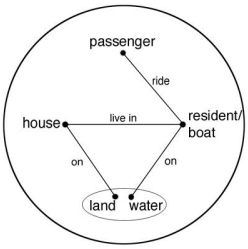

Figure 3: Boathouse Blend Space

Figure 3 shows a second blend of the same two concepts, which in English is called a "boathouse." In it, the boat ends up in the house. Notice that mapping `resident` to `boat` does not type check unless `boat` is "cast" to be of type `person`; otherwise, the boat could not live in the boathouse. This is the kind of metaphor called **personification** in literary theory, in which an object is considered a person. For this blend, neither triangle commutes, because the base element `object` is mapped to `boat` in the blend by the right morphism, and to `house` by the left morphism, but is not mapped to `boat/house` in the blend. Similarly, the central morphism cannot preserve the base element `person`, and the same goes for the base `use` operation. On the other hand, the base relation `on` goes to the same place under all three maps. A third blend is similar to (in fact, symmetrical with) the above "boathouse" blend; in it, a `house/passenger` ends up riding in the boat. (There are real examples of this, e.g., where a boat is used to transport

[2]This diagram is "upside down" from that of Fauconnier and Turner, in that our arrows go up, with the generic G on the bottom, and the blend B on top; this is consistent with the basic image schema MORE IS UP, as well as with conventions for such diagrams in mathematics. Also, Fauconnier and Turner do not include the morphism $G \to B$, and G plays a different role.

prefabricated houses across a bay for a housing development on a nearby island.)

A fourth blend is less familiar than the first three, but has very good preservation and commutativity properties, and hence is very pure, even though its physical existence is doubtful. This is an amphibious RV (recreational vehicle) that you can live in, and can ride on land and water. A fifth blend has an even less familiar meaning: a livable boat for transporting livable boats; perhaps only an algorithm could have discovered this counter-intuitive blend. Finally, a sixth blend gives a boat used on land for a house; it omits axioms that a house/boat be on water and a passenger ride a house/boat. (There are also many less perspicuous blendoids.)

The extent to which a semiotic morphism preserves source space features helps to determine its quality (Goguen 1999; Goguen & Harrell 2004b; Goguen 2003). It is therefore encouraging that our intuitive sense of the relative purity of the blends discussed above, and the degree to which they seem boat-like and house-like, corresponds to the degree to which the appropriate morphisms preserve commutativity and axioms in the input spaces. This supports our use of preservation measures for the quality of blends.

3 Blending & Style

Most text generation systems have followed a tradition inspired by the Russian formalism of Vladimir Propp (Propp 1928), implementing a discourse level syntax with a fixed set of textual templates plus rules for combining and instantiating those templates, although there are certainly differences in the theoretical foundations they propose for templates and rules, in the generalizability and soundness of those foundations, and in the success of the experiences they generate. In contrast, cognitive linguistics does not focus on syntax, but on mental spaces, prototypes, blending, metaphor, etc.; cognitive linguistics is cognitive in this sense.

The subsections below show how such a cognitive view of language can be implemented and applied to various kinds of text; in particular, we report some initial experiments on poetry. A significant finding is that the optimality principles proposed in (Fauconnier & Turner 2002) do not work for generating some poetic metaphors. As a result, we suggest a much broader view of blending principles in Section 3.5, under which different works may be controlled by different principles; for example, the choice of domains for themes, imagery, local knowledge, etc. is considered a blending principle, because these domains contribute to both the conceptual and structural blends that constitute the work. We then explore the idea that style may be determined by such principles. Before this, Section 3.1 describes our conceptual blending algorithm; Section 3.2 describes structural blending for syntax; Section 3.3 reports on an experiment in poetry generation using those algorithm, and Section 3.4 gives examples where principles quite different from those of (Fauconnier & Turner 2002) are needed.

3.1 Conceptual Blending Algorithm

Our blending algorithm is programmed in LISP, and given an input diagram, it can compute one good blend, or else compute all blendoids over the diagram. It does a depth first traversal over two binary trees, which describe possible ways to identify relations and to identify constants. Different elements from the same input space are never identified. Data sorts and data constants are never identified. Non-data sorts are identified only if required by being mapped to from a common sort in the base space. Elements in the input spaces not mapped to from the base space are included in the blend space. When constants of different sorts are identified, both choices for the sort of the blended constant are considered acceptable. (Goguen & Harrell 2004a) gives more detail on the implementation.

Even for simple inputs, the number of blendoids is so large that it is difficult for humans to find them all. In the houseboat example, the algorithm computes 48 primary blendoids (in which every possible axiom is preserved), and 736 including those that fail to preserve some axioms. This implies that efficient techniques for computing high quality blends are necessary for the algorithm to be useful for content generation and analysis. There are three distinct ways that one can go about this; all are needed. The first is just to optimize a given procedure, e.g., by using more efficient data representations. The second is to improve the procedure to reduce the search space, so that low quality blendoids are neither generated nor examined (as opposed to finding and then ranking all blendoids). The third is to use more discriminating measures of quality, which we hereafter also call *optimality principles*.

The optimality principles of (Fauconnier & Turner 2002) are powerful, but not computationally effective. Our blending algorithm currently uses degree of commutativity as its only optimality principle, but we will add other computationally feasible optimality principles, including degree of axiom preservation, and amount of type casting for constants. A *type cast* means that a constant in the blendoid has been given an unnatural type; without type casting, blended items must have compatible types (i.e., the same type, or else one a subtype of the other). Future work will make optimality principles a user-set parameter, with each optimality principle measured on a numerical scale and given a weight (possibly negative), to yield a single weighted sum. Thresholds can be set for component measure and for the sum, to avoid processing low quality blendoids. A fascinating result is that some metaphors in the Neruda poem in Section 3.4 require valuing type casts positively.

3.2 Syntax as Blending

This subsection develops an approach to text generation inspired by cognitive grammar and based on structural blending; it is illustrated by the Labov narrative syntax of Section 1.3. This material does not apply to the algorithm of Section 3.1, which is for conceptual blending. The approach assumes a context free grammar, so we first convert the two EBNF Labov rules to this form; this yields many rules, one of which (depending on how it is done) is:

`<Narr>` $\rightarrow$ `<Open> <Cls> <Eval> <Coda>`

Next, convert the right sides of rules to terms that denote lists of strings (assuming these data structures are in the data algebra) where infix period denotes append, e.g.,

<Open>.<Cls>.<Eval>.<Coda>; then construct an axiom asserting this term has sort <Narr> and saliency[3] 1,

[<Open> <Cls> <Eval> <Coda> :: <Narr>, 1]

where [_::_,_]is a 3-place relation constructor interpretaed as above. Terms in such axioms are called **templates**. The set of all such axioms is the Labov space, call it L.

To get an actual narrative, we need a domain space D for phrases to instantiate the bottom level non-terminals in L. These are asserted as axioms, just as above, e.g.,

[Once upon a time, :: <Ont>, 1]

A more sophisticated approach, taken by the system of Section 3.3, uses more cognitively oriented domains with axioms for relationships, which are then converted to syntactic templates for instantiation. Note that templates may contain variables that call for a conceptual blend produced by the algorithm of Section 3.1, drawing on conceptual domains different from those used for syntax.

Next, the generic space G contains: a constant of sort NT for each non-terminal in the grammar; variable symbols of sort Var; the above relation constructor [_::_,_]; and another relation constructor [_:_] that is explained below.

The last ingredient is a set of deduction rules to enable instantiation, also given as axioms, one of which is

[X : s'] & [t(X) :: s,v] & [t' :: s',v']
⇒ [t(t') :: s,vv']

where [X : s']indicates that variable X has sort s', t(t') indicates substitution of t' for X in t, and where vv' indicates multiplication of real numbers v and v'. Intuitively, the axiom says that if X has sort s', and if t has sort s and contains X, and if t' has sort s', then the substitution of t' into t for each instance of X has sort s (and saliency vv'). The generic space (and hence all input spaces) should also contain versions of this rule for templates t(X,Y) with two variables, for templates with three variables, etc., up to the maximum arity in any domain (alternatively, an inference space could be defined and imported into every space). The data algebra should include the operation for substituting lists into lists.

Finally, we blend the input spaces L and D over G, with the evident morphisms, and consider the deductive closure of the blend space B, which contains all axioms that can be deduced from the given ones. Those axioms with terms of sort <Narr> containing no variables are the narratives. When several templates are available, a random choice is made; saliencies can be used to compute probabilities, and the saliency of a template can be reduced after it is used, to help avoid repetitions. All this is easily coded in Prolog, to both produce and parse narratives (but declarative coding will require setting all saliencies to 1, since Prolog cannot reason with real numbers). A practical system like that described in Section 3.3 can just take the above as a semantic specification and implement it using standard tricks of the trade. A different formalization also seems possible, in which rules are constructors and processing is done at the basic level, instead of though axiomatization at the meta-level.

More complex blending than instantiation can use constraints as axioms, e.g., for tense and number agreement, or to handle anaphoric coreference of a noun phrase and pronoun. This seems a new approach, considering syntax as emergent from real-time processing and integrated with conceptual processing. It is technically similar to unification grammar ((Shieber 1986) gives a good introduction) and can be made even closer without much effort, and it is philosophically similar to the cognitive grammar of (Langacker 1999). Of course, this formalism cannot do everything one might like (see the first pragraph of Section 2), but it seems more than adequate for our project of generating interesting new media objects.

3.3 Active Poetry

This section describes a poetry generation system. It is not intended as part of a project producing a comprehensive model of the human mind. Instead, our motivation is to improve the algorithms, the theory, and our understanding of blending, as well as to produce interesting texts. Fox Harrell created an instance of this system called "The Girl with Skin of Haints and Seraphs" (Harrell 2004). This LISP program draws on a set of theme domains such as skin, angels, demons, Europe, and Africa, given as sets of axioms. It constructs input spaces by extracting axioms from two different domains, and then infers relations, sorts, and constants from these axioms. A base space is generated by instantiating shared structure between the spaces. Morphisms from the base space to the input spaces are generated, and the input spaces, base space, and morphisms are passed to the blending algorithm. Blending knowledge domains with theme domains requires selecting appropriate conceptual spaces from them[4]. Knowledge domains provide background.

In generating a metaphor, one input space is chosen as target, and attributes from the other input space are blocked, which can greatly reduce the search space. The generated blends are then placed in poetic phrase templates, which are then placed in larger grain templates for Labov narrative structure. Only conceptual blends with maximal commutativity are output. A sample poem generated by the system is given below (with parentheses removed from the LISP output and corresponding punctuation added):

her tale began when she was infected with smugnessloveitis.
she began her days looking in the mirror at her own
itchy entitled face.
her failure was ignoring her tormented angel nature.
life was an astounding miracle.
nordic-beauty death-figure vapor steamed from her pores
when she rode her bicycle.
that was nothing lovely.
when 21 she was a homely woman.
she decided to persevere;
in the rain, she fears only epidermis imperialists.
she believes that evil pride devours and alternates with
pride of hope.

[3]For such rules, our saliency is similar to entrenchment in the sense of (Langacker 1999); we assume saliency values are in the unit interval $0 \leq v \leq 1$, so they follow the fuzzy logic of (Goguen 1969).

[4]Selecting by priority of sorts and relations is a promising idea for future implementation.

it was no laughing matter.
she snuggles in angel skin sheets and sleeps.
inside she was resolved to never find
a smug or paranoid love.

This poem is a commentary on racial politics and the limitations of simplistic binary views of social identity. The dynamic nature of social identity is also reflected in the way the program produces different poems with different novel metaphors each time it is run (though reading a large number of these could be tiresome).

The text grammar of Section 3.2 gives a basis for rationally reconstructing and enhancing the poetry system (the current implementation was not conceived this way when built, and is less general). The Labov space of Section 3.2 gives top level structure for poems. One poetic domain contains the template `(her tale began when *r*)`, in the LISP syntax of the implementation, where `*r*` is a variable that gets instantiated with a noun phrase containing a past tense transitive verb, such as `(was infected with *s*)` where `*s*` is a variable that gets instantiated with a conceptual blend produced by the algorithm of Section 3.1. The axiomatic form of the first template is

```
[her tale began when X. :: <Ont>, .9]
```

where we assign saliency .9 (although the current implementation does not have saliencies). Arguments of other templates are instantiated with elements from domains for persons (e.g., a protagonist), places, objects, etc.; it is a major task of the artist to choose such material appropriately.

An interesting philosophical issue is raised by this program: human input might be considered cheating by traditional AI practitioners, since most text generation projects are oriented towards total automation and Turing test competence. But our quite different goal is to use the blending algorithm in a human designed system that generates poetry containing novel metaphors in real-time; just as with computer games, it is desirable and necessary for humans to provide rich content. For such projects, artistic freedom must take precedence over dogmatic Turing test reductionism.

A related point is raised by Espen Aarseth's analysis (Aarseth 1997) of text generation systems, which takes relationships among programmer, system, and reader as a basis for critical analysis. This is useful because readers' authorial models affect their interpretations of works, causing the approaches of traditional literary criticism to fail when computers are putative authors. Our view is that an instantiation of the poetry generation system with domains should be viewed as a work of art, produced by the designer, programmer and instantiator of the system, and judged by the corpus of poems produced by that instance; we consider it entirely wrong to view an individual poem as produced by "the computer."

3.4 Unconventional Blends

The poem "Walking around" by Pablo Neruda has the form of a narrative. Its first stanza serves as an orientation, introducing the protagonist, the place, and the time (the latter two in a condensed poetic form); the location is perhaps a small city in Chile. Each subsequent stanza explores aspects of some area within that city, using metaphors that are often quite striking. The general theme of the poem is weariness induced by consumerism. Here are its first two stanzas (out of ten, from (Fitts 1941)):

It so happens that I am tired of being a man.
It so happens, going into tailorshops and movies,
I am withered, impervious, like a swan of felt
navigating a water of beginning and ashes.

The smell of barbershops makes me weep aloud.
All I want is a rest from stones or wool,
all I want is to see no establishments or gardens,
no merchandise or goggles or elevators.

Neruda's metaphors often blend concepts in unconventional ways that require optimality principles quite different from those of (Fauconnier & Turner 2002). For example, the phrase "water of beginning and ashes" violates the first three principles (and thus requires much effort to satisfy the fourth and fifth) of Section 1.2, by combining things of enormously different type, so that casting to very remote types is required. It follows that to generate such metaphors, type casts would have to be valued positively rather than negatively. A less drastic example in the same text is "swan of felt." Similar reversals of optimality principles are needed to generate some images in Rilke's *Duino elegies*, e.g., "cheap winter hats of fate" in the fifth elegy. Neruda's imagery, objects, and cultural contexts can be implemented using domains, e.g., a `Town-location` is a place such as a tailorshop, movie theater, or barbershop, which has `town-objects`, such as goggles, elevators, wool, and stones, where attributes of wool might be heavy and impervious. For us, blending is a multi-grain process, and evidence from poetry suggests that unconventional principles are also necessary at the structural as well as conceptual levels.

3.5 Style as Blending Principle Choice

Our poetry generation system uses blending at three different levels: large grain structure (e.g., Labov narrative), where structural blending combines clausal units, which are in turn produced by structural blending of phrasal elements, some of which result from conceptual blending. Different choices of constructors at the top two levels can produce very different styles, such as a randomized "postmodern" ordering, or a deeply embedded narrative structure (as in *A Thousand and One Nights*), or a sonnet; constructors at these levels could also be used to control transitions among such styles (these would correspond to conditional rules). Other stylistic parameters at the second level include syntactic complexity, and tense and mood of verbs; different domains for themes, places, etc. can also be selected at different times. In addition to blended metaphors, the phrasal level includes noun clusters, verb phrases, etc., again potentially taken from different domains at different times. At each level, different optimality principles can be used for making choices, and these too can be different at different times (note that randomization is an optimality principle in our broad sense of that phrase).

This gives rise to 12 parameters for controlling style: each of the three levels has a set of available domains, items in those domains, optimality principles for choosing among

blends, and controls for changing domains. Since the content of domains may include not just constructors and relation instances, but also axioms for templates and for semantic relationships, if we count these as parameters, then we get 18 parameters. Of course, we could cut this cake more finely or more coarsely to get different numbers, and we may later find other parameters that are also important for style. Every parameter can be considered a principle that controls blending, but by far the most interesting and least explored are non-classical optimality principles. The narrative, causal, and accountability principles of Section 1.3 are also interesting to consider. It is clear that all principles must be carefully tuned to achieve a reasonable approximation to an existing style, but it is also clear that the results are unlikely to be close to the genius of a great poet like Neruda.

4 Conclusions & Future Work

A surprising result of our experiments is that a combination of conceptual and structural blending can produce interesting poetry, which some critics have even considered superior to prior computer generated works. Another is that both large grain structure and syntax can be handled by blending in ways that are close to, but extend, what has been done in prior text generation programs; this use of blending also gives rise to a somewhat novel view of grammar as emergent from processes of blending, rather than fixed in advance. A third result is that it is easy to extend this approach to interaction, to media other than text, and to forms other than narrative. We were also surprised that the optimality principles proposed by (Fauconnier & Turner 2002) for conventional, common sense blends like "houseboat" often fail for generating poetry; on the contrary, what might be called *dis*optimization principles are needed to generate some metaphors in the Neruda poem in Section 3.4. This led us to consider a range of different principles, and to analyze style in terms of the principles used for blending texts, where "text" is understood in a broad sense to include cinema, video games, and even living. The resulting view of style differs radically from views based on estimating parameters in statistical models of media objects.

Future work will build a version of the Neruda poem with output depending on user navigation through a computer map of a small Chilean town, and build computer games with story lines depending on interaction history. Our theory of style as blending principles will be further developed and appled to a variety of media and genres, such as video games, film music, architecture, and magazine design.

References

Aarseth, E. J. 1997. *Cybertext: Perspectives on Ergodic Literature*. Johns Hopkins.

Becker, A. 1979. Text-building, epistemology, and aesthetics in Javanese shadow theatre. In Becker, A., and Yengoyan, A., eds., *The Imagination of Reality: Essays on Southeast Asian Symbolic Systems*. Ablex. 211–243.

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

Fauconnier, G. 1985. *Mental Spaces: Aspects of Meaning Construction in Natural Language*. Bradford: MIT.

Fitts, D., ed. 1941. *Anthology of Contemporary Latin-American Poetry*. New Directions.

Goguen, J., and Harrell, F. 2004a. Foundations for active multimedia narrative: Semiotic spaces and structural blending. To appear in *Interaction Studies: Social Behaviour and Communication in Biological and Artificial Systems*.

Goguen, J., and Harrell, F. 2004b. Information visualization and semiotic morphisms. In Malcolm, G., ed., *Multidisciplinary Studies of Visual Representations and Interpretations*. Elsevier. 93–106. Proceedings of a workshop held in Liverpool, UK.

Goguen, J., and Malcolm, G. 1996. *Algebraic Semantics of Imperative Programs*. MIT.

Goguen, J. 1969. The logic of inexact concepts. *Synthese* 19:325–373.

Goguen, J. 1999. An introduction to algebraic semiotics, with applications to user interface design. In Nehaniv, C., ed., *Computation for Metaphors, Analogy and Agents*. Springer. 242–291. Lecture Notes in Artificial Intelligence, Volume 1562.

Goguen, J. 2003. Semiotic morphisms, representations, and blending for interface design. In *Proceedings, AMAST Workshop on Algebraic Methods in Language Processing*. AMAST Press. 1–15. Conference held in Verona, Italy, 25–27 August, 2003.

Grady, J.; Oakley, T.; and Coulson, S. 1999. Blending and metaphor. In Gibbs, R., and Steen, G., eds., *Metaphor in Cognitive Linguistics*. Benjamins.

Harrell, D. F. 2004. Algebra of Identity: Skin of wind, skin of streams, skin of shadows, skin of vapor. Performed at *Powering Up/Powering Down*, a Festival of Radical Media Arts, organized by Teknika Radica, 30 January.

Labov, W. 1972. The transformation of experience in narrative syntax. In *Language in the Inner City*. University of Pennsylvania. 354–396.

Lakoff, G., and Johnson, M. 1980. *Metaphors We Live By*. Chicago.

Lakoff, G. 1987. *Women, Fire and Other Dangerous Things: What categories reveal about the mind*. Chicago.

Langacker, R. 1999. *Foundations of Cognitive Grammar*. Stanford.

Linde, C. 1993. *Life Stories: the Creation of Coherence*. Oxford.

Meehan, J. 1981. TALE-SPIN. In Shank, R., and Riesbeck, C., eds., *Inside Computer Understanding: Five Programs Plus Miniatures*. Erlbaum. 197–226.

Propp, V. 1928. *Morphology of the Folktale*. University of Texas Press. Reprinted 1968.

Shieber, S. 1986. *An Introduction to Unification-based Approaches to Grammar*. CSLI, Stanford.