

11

Style: A Computational and Conceptual Blending-Based Approach

Joseph A. Goguen¹ and D. Fox Harrell²

¹ Department of Computer Science and Engineering

University of California, San Diego;

² Digital Media Program

School of Literature, Communication, and Culture

Georgia Institute of Technology

Summary. This chapter 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, including interactive multimedia poetry, although the approach generalizes to other media. The central idea is to generate multimedia content and analyze style in terms of blending principles, based on our finding that different principles from those of common sense blending are often needed for some contemporary poetic metaphors.

11.1 Introduction: Motivations and Goals

James Meehan’s 1976 TALE-SPIN [31] 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” (GOFAI), 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?

The word “style” has many different meanings in different contexts. This ambiguity cannot be dispelled by the use of computational technology alone. Statistical

methods and/or formal descriptions of features must be augmented by knowledge of human context and experience to capture a more “true” notion of style than subjective human opinion. Still, we believe that computational techniques can make important contributions to the understanding, analysis, and generation of style.

It is not generally intuitive to both embrace technology and some of its methods at the same time as stressing human values and the limitations (even technical limitations) of the underlying values behind much of current computing. This is our stance however. We propose to analyze and generate works in particular styles by choosing appropriate principles for blending computational structures representing concepts, and more broadly by considering the human conceptual blends that might correspond to the computational structures. On its own this approach is open to some of the same criticisms as other formal top down technical approaches (as in GOFAI). But our claims and theoretical underpinnings are quite different; the formalisms used are not taken to exist explicitly in human cognition and we recognize meaning as embodied, situated, and contextual. Our methods are different as well, for formalization is used as a tool only within the context of human subjectivity, experience, and social interaction.

11.1.1 Style in Interactive and Generative Narrative and Poetry

Our creative experiments have primarily been concerned with new genres of interactive and generative narrative and poetry. Fox Harrell’s GRIOT system provides a framework for implementing expressive computational works in which media elements are composed on-the-fly based upon formal semantic representations and user input [24]. Initial works were text-based, however, the GRIOT framework is meant to generalize for multimedia content. The work “The Griot Sings Haibun” uses a graphical user interface (GUI) interface and the “Generative Visual Renku Project” by Fox Harrell and Kenny Chow uses GRIOT to generate poetic landscapes of iconic images populated by transforming calligraphic human characters [23]. The formal semantic representation framework allows an author to describe the high level structure of the work, e.g. narrative or poetic structure, and more fine grained conceptual content to provide a knowledge base containing information about the narrative or poetic world. User input is used to select particular domains of conceptual content from which new content is composed. For example, a mouse click can select an image that another image should match conceptually as well as visually, or a keyword can select themes from which new descriptive phases are generated to be integrated into a prewritten textual template. The input mechanisms invoked thus far are relatively simple such that an author can construct a robust GUI or gamelike interface atop GRIOT via which the system presents a dynamically structured narrative new events, characters, metaphors, or other creative elements based upon user actions.

These open-ended goals regarding semantically-based content interaction and generation led naturally to questions of style. Primarily, the question arose of how formal methods can yield richly expressive works that adequately represents an author’s goals in light of the variability and generativity that GRIOT allows. We sought to allow authors to maintain coherence at multiple grain sizes, e.g. appropriate metaphors, narrative transitions, conceptual themes, while allowing the system to algorithmically manipulate content in light of user interaction. An algorithm for conceptual blending is a central generative component. The conceptual blending

algorithm, called Alloy, integrates conceptual spaces using mappings between and their shared structure to direct the process to produce “optimal” blended spaces and mappings to them. This optimal blending can lead to structural and conceptual coherence in the generated content. The optimality principles we implemented can be tuned in order to achieve different aesthetic effects depending on an author’s goals. Our approach is amenable to a wide range of perspectives on style, for example, Kevin Burns’s decision theoretic approach to style (also appearing in this book) describes various utilities that an artist or game player may give importance to given a goal at hand. Our research provides a formal approach for an author to implement such aesthetic utilities and maintain them coherently. This search for structural and conceptual coherence, in light of structural and conceptual interactivity and generativity, led us to the formulation of style as a choice of blending principles in this chapter. Though we do not seek to highlight the autonomy of systems created with GRIOT, as in Harold Cohen’s notorious autonomous painting system *AARON* [4] or George Lewis’s *Voyager* [29], our goals overlap with such works that involve tuning algorithmic function to produce a body of works with certain variable attributes and certain invariant notions of “style.” We hope that this chapter provide a useful theoretical lens for analyzing such phenomena across computational media works with similar goals.

11.1.2 Technical Style Versus Human Style

For clarity it is important to express specifically what we have been able to accomplish technically with our approach so far. At the core of our work is the generation and analysis of content using partial composition of sign systems (represented using algebraic semiotics). Secondarily, we combine templates (pre-structured content) according to formalized narrative structures. The strength of our approach is that conceptual blending (the generation component) and structural blending (the media composition component) can be accounted for by the same theoretical underpinnings (algebraic semiotics) and that the work is in sympathy with, and inspired by, recent insights from cognitive science in metaphor theory, embodied cognition, conceptual blending theory, and situated cognition. Finally, the approach is useful for both analyzing media (e.g. user-interfaces) and generating media (e.g. in multimedia artwork or computer games).

Some specific technical contributions of this work (all to be discussed later in the chapter) are:

1. Data structures for sign systems and semiotic morphisms (mapping between sign systems);
2. An algorithm for conceptual blending that is efficient and exhaustive;
3. Four structural optimality principles for conceptual blending;
4. Media morphisms to map conceptual data structures to output in a particular medium (e.g. text or graphical images);
5. Data structures for templates (artist created content resources) that include variables that can be replaced by generated content.
6. An automaton useful for implementing recursive structuralist narrative models (it arranges templates into a particular discourse structure);
7. An interpreter that reads user input and outputs generated content;
8. A scripting format that allows an author to create “improvisational” media.

But can these technical contributions tell us much about style? Have we accounted for style in a structural way that avoids the pitfalls of top-down artificial intelligence, structuralist cultural theory, and cognitivist psychology? We believe so. Though we are inspired technically by some of the approaches above, our approach includes being very forthright about the limitations of computational techniques and to introduce human judgment, subjectivity, and social context at appropriate points in the design, development, and output processes. Thus, for us:

- Style is understood in the interpretation of sign systems via dialogic interaction with media.
- Style is determined by the particular executions of human concepts in media.
- Style is created by developing sign systems and artifacts for both of the above.
- Style exists in the context of interpreting sign systems.
- Style exists in the context of use of artifacts.
- Style is inherent in any knowledge encoded in a sign system by a human.

Implementations based on theories of conceptual blending and semiotic representation are useful for expressing and analyzing style, but are not sufficient. Technically, we systematically examine how humans compose sign systems with particular attention to regularities such as hierarchies, preservation between mappings, information lost or gained, changes of classification of symbols (type casting), and other similar structural features. Computational methods are very good for these purposes. At the same time we pay close heed to the way that humans encode knowledge that is not amenable to computational analysis or manipulation, and we explicitly require human input and judgment in design processes. Our approach is a combination of formal methods, awareness of their limitations, and strategies to partially surmount those limitations.

11.1.3 Overview

As mentioned above, most story generation systems are based on approaches in GO-FAI, which focus on logic and planning. They are in line with a tradition inspired by the Russian formalism of Vladimir Propp [33], 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.

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. We focus on text-based works as they are the most developed applications created using GRIOT and thus they allow us to highlight the issues of style raised most clearly as opposed to challenges raised by working in multiple modalities.

This chapter is especially concerned with how concepts are combined (blended), the principles underlying the different ways that concepts can be blended, and how a computational account of blending can be used to implement and analyze various styles in interactive media. But we believe computer implementations such as the GRIOT system can also contribute to cognitive science as simulations of precise

mathematical theories. Section 11.2 reviews some basic linguistic research used in our approach. Section 11.3 reviews our prior work on semiotic spaces, semiotic morphisms, and structural blending; a rather detailed example is also given. Section 11.4.2 discusses our work on interactive poetry generation, and Sections 11.4 and 11.4.1 discuss the blending algorithm at its core. Section 11.4.3 considers some interesting blends that appear in recent poetry, and finds that very different principles from those proposed in [6] for conventional, common sense blends are needed. This motivates our analysis in Section 11.4.4 of style in terms of the blending principles used to generate works in various media.

11.2 Linguistic Foundations

This section reviews some linguistics research that serves as a foundation for our work on style, including certain topics from cognitive linguistics, sociolinguistics, and narratology. This research is aimed at understanding human cognition, rather than at computer simulation and mathematical precision, as is our own research.

11.2.1 Metaphor and Blending

Gilles Fauconnier and Mark Turner have developed a theory within cognitive linguistics known as **conceptual blending** or **conceptual integration** [6]. Here **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 [5], as in Figure 11.3. George Lakoff, Mark Johnson, and others [27, 26] 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, and perhaps even universal, metaphors, and are also useful in other areas, such as user interface design [10, 11, 12].

Blending two conceptual spaces yields a new space that combines parts of the given spaces, and may also have emergent structure [6]. 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 [6]. 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 [20]. 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, and in a great deal of everyday conversation, as well as humor.

11.2.2 The Cognitive Optimality Principles

Below are five of the 29 optimality principles from the list in Chapter 15 of [6] and discussed in [20]; these cognitive optimality principles are used to determine which among many possible conceptual blends is most appropriate for a given situation. We refer to them here as *cognitive* optimality principles to distinguish them from the *structural*, or *computational*, optimality principles that we have implemented in the Alloy conceptual blending algorithm:

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.; an important inspiration was use in literary criticism, as in [36]. But Section 11.4.3 will show that they do not apply to generating some less conventional language, such as certain metaphors in Pablo Neruda poems. All five of these cognitive 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 one of the structural optimality principles used by Alloy blending algorithm described in Section 11.4. Likewise, the Unpacking Principle holds for the most optimal blended spaces generated by our algorithm.

11.2.3 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. [25, 30], 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. (For example, “You touch that, and you die.”) 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 the 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 languages³. 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 (for extended Backus Naur Form),

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, EBNF 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).

³However, they do not necessarily apply to non-Western langauges and cultures; for example, Balinese narrative does not follow the narrative presupposition [3].

11.3 Algebraic Semiotics

Before briefly discussing algebraic semiotics, it may be helpful to be clear about its philosophical orientation. The reason for taking special case with this is that, in Western culture, mathematical formalisms are often given a status beyond what they deserve. For example, Euclid wrote, “The laws of nature are but the mathematical thoughts of God.” Similarly, “situations” in the situation semantics of Barwise and Perry, which are similar to conceptual spaces (but more sophisticated – perhaps *too* sophisticated) are considered to be actually existing, ideal Platonic entities [2]. Somewhat less grandly, one might consider that conceptual spaces are directly instantiated in the human brain. Our point of view is different: we believe that all such formalisms are constructed by human researchers in the course of particular investigations, having the heuristic purpose of facilitating consideration of certain issues in that investigation. Under this view, all theories are situated social entities, mathematical theories no less than others. Of course, this does not preclude their being accurate representations of reality.

11.3.1 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 [13]. Thus, to use blending as a basis for stylistic analysis, we must generalize conceptual spaces to include structure. Algebraic semiotics captures some major insights of the founders of semiotics, Charles Saunders Peirce [32] and Ferdinand Saussure [34]. Peirce emphasized (among other things) that the relation between a given token and its object is not just a function (as in denotational semantics), but a relation that depends on the situation in which the token is interpreted, while Saussure emphasized (among other things) that signs come in systems. Semiotic systems capture Saussure’s insight, while the blending of semiotic systems captures (and extends) Peirce’s insight.

Algebraic semiotics uses algebraic semantics [18] to describe the structure of complex signs, including multimedia signs (e.g., a music video with subtitles), and to study the blending of such structures. Its basic notion is a (loose algebraic) **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 (loose algebraic) **semiotic system** (also called a **semiotic space** or **sign system**) [10] is a (loose 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. Sorts classify the parts of signs and the values of attributes of signs (e.g., color and size). **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, while 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. Many details omitted here are given in [10, 11, 12].

The grammar for narratives in Section 11.2.3 can be described as a semiotic system. Its top level sort is of course `<Narr>`; the second level sorts are `<Cls>`, `<Eval>`, `<Open>`, and `<Coda>`, while `<Ornt>` and `<Abs>` are third level sorts. It should not be thought that this semiotic system will be blended with conceptual spaces to give a story; this would not be appropriate because narrative structure is at a different level of abstraction from that of narrative content. However, it would be appropriate to blend a narrative structure space with another space that described some other structure, such as the scene/shot/etc. structure of cinema.

Some other examples of semiotic systems are: dates; times; bibliographies (in one or more fixed format); tables of contents (e.g., for books, again in fixed formats); newspapers (e.g., the *New York Times* Arts Section); and a fixed website, such as the CNN homepage (in some particular instance of its gradually evolving format). Note that each of these has a large space of possible instances (i.e., models), but all these instances have the same structure.

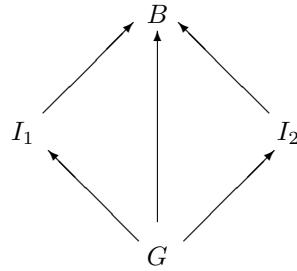
Semiotic systems, like the algebraic theories that they build upon, are *formal* in the precise sense that changing the names used in them does not change their space of models, but only the way that parts of the models are named. Thus, these formal descriptions do not even attempt to capture meaning in any real human sense; however, we do try to choose names that can help the reader's intuition.

Semiotic spaces, like conceptual spaces, are static. Though they describe blends as being dynamic and in some examples carefully describe temporal sequences of blending, 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 so 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 [12], using so called hidden algebra [19]. Also the conceptual blending theory of Fauconnier and Turner 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 in Section 11.3.3 below) and being able to generate more striking and unusual blends by identifying sorts known to be far apart (see the discussion in Section 11.4.3 below). 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.

Using theories for semiotic systems is better for our purposes than concrete model-based approaches (i.e. an approach based in syntax rather an approach such as in [8]) because it is much more natural to treat levels and priorities in the context of theories, and because defining spaces of models with theories makes it convenient to use axioms to possible the allowable models.

11.3.2 Semiotic Morphisms and 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 [10, 11, 12]. Because sign systems are formalized as algebraic theories with additional structure, semiotic morphisms are formalized as theory morphisms that also preserve this additional

**Fig. 11.1.** Blend Diagram

structure. A theory morphism consists 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 may 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 to do so).

Semiotic morphisms are used in structural blending to establish connections between semiotic spaces that indicate which elements should be identified. The simplest form⁴ of blend is shown in Figure 11.1, where I_1 and I_2 are called **input spaces**, and G is called a **base space**⁵. We call I_1, I_2, G together with the morphisms $I_1 \rightarrow G$ and $I_2 \rightarrow G$ an **input diagram**. Given an input diagram, we use the term **blendoid** for a space B together with morphisms $I_1 \rightarrow B$, $I_2 \rightarrow B$, and $G \rightarrow B$, called **injections**, such that the diagram of Figure 11.1 **commutes**, in the sense that both compositions $G \rightarrow I_1 \rightarrow B$ and $G \rightarrow I_2 \rightarrow B$ are “weakly equal” to the morphism $G \rightarrow B$, 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. 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 11.4 uses a set optimality principles based only on the structure

⁴This 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 $G \rightarrow B$, and G plays a different role.

⁵The term “generic space” is used in cognitive linguistics [6], but the term “base space” better describes the role of this space in our approach to interface and active multimedia design.

of blends (hence referred to as *structural* optimality principles), rather than their meaning; these include degrees of commutativity, type casting, and preservation of constants and axioms.

11.3.3 The House/Boat Example

We illustrate blending with the concepts “house” and “boat” shown in Figure 11.2. 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 11.1, 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 “V” 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 11.3 shows a second blend of the same two concepts, which in English is called a “boathouse”⁶. 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 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

⁶Since ‘water’ and ‘land’ are data elements they are handled specially (both are preserved in the blended space and they are not identified – see [10] for details about the handling of datasorts).

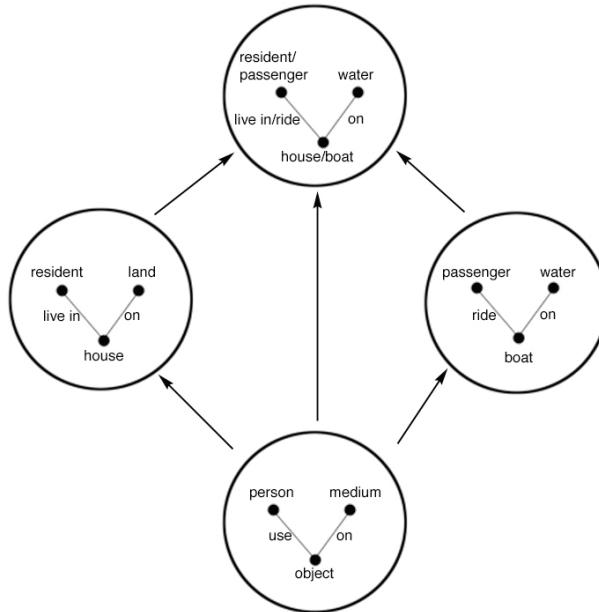


Fig. 11.2. Houseboat Blend Diagram (semiotic spaces and semiotic morphisms between them)

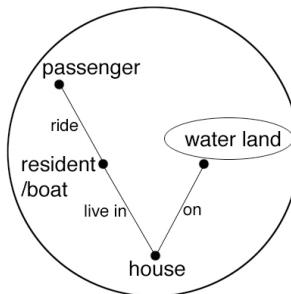


Fig. 11.3. Boathouse Blend Space (in which 'boat' is "type cast" to sort 'Person' and identified with 'resident.')

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 48 blends where all preservation properties hold, and also 736 less perspicuous blendoids.

We can see in this example that the extent to which a blend preserves the structure of an input space corresponds very well with our intuitive sense of how closely the blend resembles that input space. *Quantitative* preservation measures include the number of triangles that commute for the elements of a space, the number for its relations, and the number of type casts required, but *qualitative* measures which are partial orders based on these factors seems even closer to our intuitions [10, 16, 12]. This is a very encouraging result with regard to our plan to use these purely formal optimality criteria in our blending algorithm, though we do not expect it to work as well as it did here in all possible situations.

It may be an interesting exercise for readers to consider blends of “ice” and “house”; this is similar to the “house” and “boat” example, but a bit simpler.

11.4 Blending and the GRIOT System

Our blending algorithm ALLOY 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 of the input spaces that are 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. [17] 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. Since for the houseboat example, the algorithm computes 48 blends where every axiom is preserved, and 736 that fail to preserve some axioms, it follows 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 [6] are powerful, but not computationally effective. Our blending algorithm currently uses degree of commutativity, degree of constant preservation, degree of axiom preservation, and amount of type casting for constants, as *structural* optimality principles. 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). In ALLOY, each optimality principle is 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. Though currently we have implemented poetry only using optimal blends, in future work this may change based upon the style of poetic metaphor we wish to achieve. A

fascinating result is that some metaphors in the Neruda poem in Section 11.4.3 require valuing type casts positively rather than negatively; this seems also to happen in other contemporary art forms.

11.4.1 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 11.2.3. This material does not apply the ALLOY algorithm of Section 11.4, which is for conceptual blending. Also, issues of saliency discussed here have not yet been implemented in GRIOT. 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:

Next, convert the right sides of rules to terms that denote lists of strings (assuming these data structures are in the data algebra); then construct an axiom asserting this term has sort **<Narr>** and saliency⁷ 1,

where $[_:::_,_]$ is a 3-place relation constructor interpreted 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.,

A more sophisticated approach, taken by the system of Section 11.4.2, 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 11.4, 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: 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.

⁷For such rules, our saliency is similar to entrenchment in the sense of [28]; we assume saliency values are in the unit interval $0 \leq v \leq 1$, and that they follow the fuzzy logic of [9].

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; saliences 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 saliences to 1, since Prolog cannot reason with real numbers). A practical system like that described in Section 11.4.2 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 ([35] gives a good introduction) and can be made even closer without much effort, and it is philosophically similar to the cognitive grammar of [28]. Of course, this formalism cannot do everything one might like (see the first paragraph of Section 11.3), but it seems more than adequate for our project of generating interesting new media objects.

11.4.2 Interactive Poetry

This section describes technology for implementing interactive and generative narratives, with an emphasis on its use in creating interactive poetry systems. We have coined the phrases “polymorphic poems” and “polypoems” to describe such works. This research is not intended as part of a project producing a comprehensive model of the human mind or the human process of poetry generation. Instead, our motivation is to improve the algorithms, the theory, and our understanding of blending, as well as to produce interesting texts; but this does not just mean that one cannot draw inferences about human poetry based on the successes and failures of various approaches. The GRIOT implementation has the following components (see Figure 11.4):

1. Theme Domains: Themes are represented as ontologies consisting of sets of axioms expressing properties specific to a particular polypoem theme. Associated with each theme domain is a list of keywords that access that theme domain.
2. ALLOY Conceptual Blending Algorithm: This generates new concepts and metaphors from input spaces selected from the theme domains. ALLOY uses structural optimality principles inspired by those from conceptual

- blending theory [6]. These principles use a measure that quantifies optimality according to: (1) commutativity of mappings from elements in the input spaces to the blended space, (2) type coercions in the input spaces, (3) the number of constants from the input spaces that are preserved in the blended space, and (4) the number of axioms from the input spaces that are preserved in the blended space.
3. Media Morphisms: These map conceptual blends to representations in particular media, such as natural language and graphics. **grammar morphisms** are a special case of media morphisms that provide a mapping from a conceptual space to a representation in natural language text.
 4. Phrase Templates: These are sets of text fragments organized by clause type and featuring **wildcards** that will be replaced by generated content upon each execution of the polypoem.
 5. Narrative (or Event) Structure: This defines how phrase templates can be composed. A polypoem author inputs her or his choice of narrative structure; initial experiments used a version of the Labov narrative structure of personal experience. The templates are selected using a new type of automaton that we invented, called a “probabilistic bounded transition stack machine,” or more simply a “Narrative Structure Engine,” or “Event Structure Engine” in the general case.

Using these components, the output of executing a polypoem is generated according to the rules and themes defined by a polypoem author. Many poetic texts can be output that feature systematic thematic and structural constraints, while at the same time exhibiting great variety in their actual phrases and metaphorical and aesthetic content. This is illustrated in Fox Harrell’s polypoem “The Girl with Skin of Haints and Seraphs” [22], which draws upon a set of theme domains such as skin, angels, demons, Europe, and Africa⁸. After processing a user-input keyword such as “europe,” entered after a “→” prompt, the first line could be:

her tale began when she was infected with white female-it is

or

she began her days looking in the mirror at her own pale-skinned death-figure face

or any of a number of alternate phrases (there are fourteen templates for such opening phrases). As an example of variation within a particular phrase due to wildcard replacement, among many other possibilities the first example above could have also been either:

⁸This work 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 it is unlikely that any one user would read a large number of these).

her tale began when she was infected with tribal-warrior spectre-it is
or

her tale began when she was infected with black demon-it is
depending on how the phrase template was instantiated.

The discussion of template instantiation in the text grammar of Section 11.4.1 gives a theoretical basis for describing wildcard replacement in GRIOT. In the example above, one set of phrase templates contains “(her tale began when she was infected with (* g-singular-noun)-itis)” in the LISP syntax of the implementation, where the inner parenthesis is a wildcard variable that gets replaced with a noun cluster or a noun paired with a modifier. Exactly how the wildcard is replaced is determined by a combination of user input and the content of the wildcard itself. A wildcard consists of two or more parts including a “*” marker that indicates it is a wildcard, and a variable that determines whether it is to be replaced by another phrase (denoted by the prefix “p-” attached to a clause type name) or by content generated using the ALLOY algorithm (denoted by the prefix “g-” attached to a grammatical form name such as “singular-noun”). Optional variables can be used additionally to constrain domains or axioms selected as input to the ALLOY blending algorithm (denoted by “d-” and “a-” prefixes respectively, though in practice we have not had to use axiom determining variables). Optional variables can also be used for structural effects such as forcing repeats of wildcard replacement text from earlier in the poem. User input plays a role in wildcard replacement, since the user entered keywords determine one of the domains to be used in constructing the blends that will be used in template instantiation.

In most of the polypoems implemented so far, phrase templates have been most commonly instantiated by replacing the wildcard with an English language mapping from conceptual blend produced by the algorithm of Section 11.4. After user input and/or optional wildcard variables have been used to select theme domains, conceptual spaces are selected from the chosen domains as follows:

1. Axioms are chosen from the first domain. An axiom is a relation represented as a list such as: The “0”s are initial values of the blend optimality measure, which is updated as execution proceeds; note that users do not see this internal representation.
2. A subdomain is formed from the second domain that consists only of axioms of sorts that match the chosen axioms, axioms are selected randomly from the subdomain.
3. These spaces are used to create an input diagram (a generic space, two input spaces, and morphisms between the spaces).
4. The input diagram is passed to the ALLOY algorithm, which returns a conceptual blend with the two morphisms to it.

After the blend is generated, a grammar morphism (the set of grammar morphisms is implemented as a hash table of Lisp closures) is consulted and used to generate the English text used to instantiate the template.

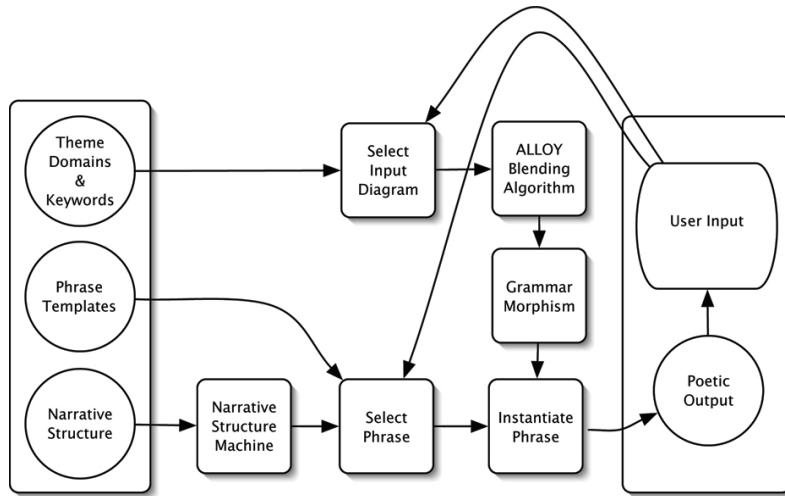


Fig. 11.4. The GRIOT framework as used for polymorphic poetry

The above discussion summarizes material introduced in [21]. A complete sample poem generated by one user's interaction with "The Girl with Skin of Haints and Seraphs" polypoem is given below, where "→" is the prompt, which is followed by the user input:

```

→ evil
every night she wakes covered with hate, awe sweat
→ europe
imperialist and girl thoughts taunted her as a teen
serious times were here
→ africa
drum spiked-tail vapor steamed from her pores when she rode her bicycle
in the rain
→ angel
when twenty-one she was a homely woman
that was nothing lovely
→ skin
tears ran relay races between her girl and european eyes and her ignorance,
longing earlobes and back
she could laugh
→ angel
her dreams were of cupid epidermis
life was a sight gag
→ demon
so she resolved to find bat-wings and pointed-nose passion and be happy

```

title	topic1	eval1	topic2	eval2
clear early morning				
people shaped clouds				
luminous static canopy,				
emptiness, flesh changing, unchanging -				
infinite perfect painful stillness				
(haibun2::	eval2	Domain::other)	
		empty		
self				
other				
haibun 1	haibun 2	haibun 3	haibun 4	haibun 5

Fig. 11.5. Haibun Polypoem Screenshot

The selection of particular phrase templates, i.e., structuring the narrative progression of the polypoem, is accomplished by the Event Structure Engine. The Event Structure Engine has the following format (in a modified BNF notation in which simple phrases (without < >) indicate informal descriptions of atomic elements):

The functionality of the Event Structure Engine can be described using the example of the polypoem “The Griot Sings Haibun,” developed by both authors and based on a poem entitled “November Qualia” by Joseph Goguen [14]. It extends the system by allowing the passing of user input to the system via a graphical or game-like interface (future examples could be navigating a game map or selecting objects in a virtual environment). In “The Griot Sings Haibun” the system was used with a graphical interface (see Figure 11.5) to generate “(neo)haibun,” a combination of prose, haiku, and beat poetry used to narrate personal everyday experiences in a live performance with free jazz musicians [15]. A simple haibun poem can be defined with the following structure: One possible poem output by such an automaton would have the following structure (with clause names standing in for actual clauses): This consists of 3 short poems, preceded by two orientation clauses, where each poem has topic and evaluation clauses; Figure 11.5 shows part of one poem from an actual performance [15].

The most important structure in the description of this format is the <clause>, and the functioning of the Event Structure Engine can be understood by examining these components. A clause consists of a name, pair

of integers, subclause name, exit-to clause name, and a read-flag. A clause is processed as follows:

1. The number of times the clause will be repeated is determined. The first of the pair of numbers indicates the minimum number of repetitions and the second number represents the maximum, hence the automaton is “bounded.” A number, called the “repeat number,” between the minimum and maximum is chosen at with some user-defined probability (the default is equal probability for each integer in the interval), hence the automaton is “probabilistic.”
2. If the read-flag is on (i.e. it is the atom “read”) a prompt is given and user input is accepted.
3. A phrase template of the type indicated by the clause name is selected and output.
4. If the subclause is not the empty list then the clause with the same name as the subclause name is processed.
5. If the exit-to clause is not the empty list then the clause with the same name as the exit-to clause name is processed.
6. If the current clause has not been processed a number of times equal to the repeat number then the clause is processed again.

An interesting philosophical issue is raised by this program: human input might be considered cheating by traditional AI practitioners, since most AI 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 takes precedence over dogmatic Turing test reductionism.

A related point is raised by Espen Aarseth’s analysis [1] of text generation systems, which takes relationships among programmer, system, and reader as a basis for critical analysis. The hypothesis is that 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.”

11.4.3 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 [7]):

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 [6]. 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 11.2.2, 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-level and multi-faceted process, since for example, evidence from poetry suggests that unconventional principles are also necessary at the structural as well as the conceptual levels.

11.4.4 Style as Blending Principle Choice

The GRIOT system, when used for poetry generation, invokes blending at three different levels: (1) large grain structure (e.g., Labov narrative), where structural blending combines clausal units, which are in turn produced by (2) structural blending of phrasal elements, some of which result from (3) 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 nested 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 (1) a set of available domains, (2) items in those domains, (3) optimality principles for choosing among blends, and (4) 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 11.2.3 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. (The situation for comprehension is presumably roughly dual to that for generation, in that here one seeks to understand what principle Neruda may (unconsciously) have used to obtain some typical classes of remote type casts for his blends.)

11.5 Conclusions and 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 poetry or narratives. We were also surprised that the optimality principles proposed by [6] for conventional, common sense blends like “houseboat” often fail for generating poetry; on the contrary, what might be called *disoptimization* principles are needed to generate some metaphors in the Neruda poem in Section 11.4.3. 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. Our theory of style as blending principles will be further developed in future work, and has the potential to be applied to a variety of media and genres, ranging from magazine design, film, music and architec-

ture, to new interactive multimedia narrative forms, e.g., video games with nuances and powerful expressive dimensions. In fact, one major application area for this work is developing narrative computer games. Many computer games are based in narrative, but despite the fact that users act dynamically within the game worlds, the stories are either static, or else feature simplistic branching narrative structure. Providing games with the potential to generate story elements on the fly, constrained stylistically and thematically by the game developer's narrative model, can add value to games, in the forms of greater salience for the users' actions, and greater replayability.

References

1. Espen J. Aarseth. *Cybertext: Perspectives on Ergodic Literature*. Johns Hopkins, 1997.
2. Jon Barwise and John Perry. *Situations and Attitudes*. MIT (Bradford), 1983.
3. Alton Becker. Text-building, epistemology, and aesthetics in Javanese shadow theatre. In Alton Becker and Aran Yengoyan, editors, *The Imagination of Reality: Essays on Southeast Asian Symbolic Systems*, pages 211–243. Ablex, 1979.
4. Harold Cohen. The further exploits of Aaron, painter. *Stanford Humanities Review*, 4(2):141–158, 1995.
5. Gilles Fauconnier. *Mental Spaces: Aspects of Meaning Construction in Natural Language*. Bradford: MIT, 1985.
6. Gilles Fauconnier and Mark Turner. *The Way We Think*. Basic, 2002.
7. Dudley Fitts, editor. *Anthology of Contemporary Latin-American Poetry*. New Directions, 1941.
8. Deidre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170, 1983.
9. Joseph Goguen. The logic of inexact concepts. *Synthese*, 19:325–373, 1969.
10. Joseph Goguen. An introduction to algebraic semiotics, with applications to user interface design. In Christopher Nehaniv, editor, *Computation for Metaphors, Analogy and Agents*, pages 242–291. Springer, 1999. Lecture Notes in Artificial Intelligence, Volume 1562.
11. Joseph Goguen. Social and semiotic analyses for theorem prover user interface design. *Formal Aspects of Computing*, 11:272–301, 1999. Special issue on user interfaces for theorem provers.
12. Joseph Goguen. Semiotic morphisms, representations, and blending for interface design. In *Proceedings, AMAST Workshop on Algebraic Methods in Language Processing*, pages 1–15. AMAST Press, 2003. Conference held in Verona, Italy, 25–27 August, 2003.
13. Joseph Goguen. Musical qualia, context, time, and emotion. *Journal of Consciousness Studies*, 11(3/4):117–147, 2004.
14. Joseph Goguen. November Qualia. *Journal of Consciousness Studies*, 12(11):73, 2005. Four short poems.
15. Joseph Goguen and D. Fox Harrell. The GRIOT sings haibun. Performance at UCSD CalIT2 opening celebration, 28 October 2005. Music by David Borgo, Ryoko Goguen and Bertram Turetzky.

16. Joseph Goguen and D. Fox Harrell. Information visualization and semiotic morphisms. In Grant Malcolm, editor, *Multidisciplinary Studies of Visual Representations and Interpretations*, pages 93–106. Elsevier, 2004. Proceedings of a workshop held in Liverpool, UK.
17. Joseph Goguen and D. Fox Harrell. Foundations for active multimedia narrative: Semiotic spaces and structural blending, to appear. *Interaction Studies: Social Behaviour and Communication in Biological and Artificial Systems*.
18. Joseph Goguen and Grant Malcolm. *Algebraic Semantics of Imperative Programs*. MIT, 1996.
19. Joseph Goguen and Grant Malcolm. A hidden agenda. *Theoretical Computer Science*, 245(1):55–101, August 2000. Also UCSD Dept. Computer Science & Eng. Technical Report CS97-538, May 1997.
20. Joseph Grady, Todd Oakley, and Seanna Coulson. Blending and metaphor. In Raymond Gibbs and Gerard Steen, editors, *Metaphor in Cognitive Linguistics*. Benjamins, 1999.
21. D. Fox Harrell. Shades of computational evocation and meaning: The GRIOT system and improvisational poetry generation. In *Proceedings, Sixth Digital Arts and Culture Conference*, pages 133–143. 2005.
22. D. Fox Harrell. Algebra of Identity: Skin of wind, skin of streams, skin of shadows, skin of vapor. In Arthur Kroker and Marilouise Kroker, editors, *Critical Digital Studies*, pages 158–174. Toronto, 2008.
23. D. Fox Harrell and Kenny K. N. Chow. Generative visual renku: Linked poetry generation with the GRIOT system. In *Proceedings, Electronic Literature Organization Conference*. 2008.
24. Douglas Alan Harrell Jr. (a.k.a. D. Fox Harrell). *Theory and Technology for Computational Narrative: An Approach to Generative and Interactive Narrative with Bases in Algebraic Semiotics and Cognitive Linguistics*. PhD thesis, University of California, San Diego, 2007. Department of Computer Science and Engineering.
25. William Labov. The transformation of experience in narrative syntax. In *Language in the Inner City*, pages 354–396. University of Pennsylvania, 1972.
26. George Lakoff. *Women, Fire and Other Dangerous Things: What categories reveal about the mind*. Chicago, 1987.
27. George Lakoff and Mark Johnson. *Metaphors We Live By*. Chicago, 1980.
28. Ronald Langacker. *Foundations of Cognitive Grammar*. Stanford, 1999.
29. George E. Lewis. Too many notes: Computers, complexity and culture in voyager. *Leonardo Music Journal*, 10:33–39, 2000.
30. Charlotte Linde. *Life Stories: the Creation of Coherence*. Oxford, 1993.
31. James Meehan. TALE-SPIN. In Roger Shank and Christopher Riesbeck, editors, *Inside Computer Understanding: Five Programs Plus Miniatures*, pages 197–226. Erlbaum, 1981.
32. Charles Saunders Peirce. *Collected Papers*. Harvard, 1965. In 6 volumes; see especially Volume 2: Elements of Logic.
33. Vladimir Propp. *Morphology of the Folktale*. University of Texas Press, 1928. Reprinted 1968.
34. Ferdinand de Saussure. *Course in General Linguistics*. Duckworth, 1976. Translated by Roy Harris.
35. Stuart Shieber. *An Introduction to Unification-based Approaches to Grammar*. CSLI, Stanford, 1986.
36. Mark Turner. *The Literary Mind*. Oxford, 1996.