

# UC San Diego

## UC San Diego Electronic Theses and Dissertations

### Title

Theory and technology for computational narrative : an approach to generative and interactive narrative with bases in algebraic semiotics and cognitive linguistics

### Permalink

<https://escholarship.org/uc/item/1tv6c68g>

### Author

Harrell, Douglas Alan

### Publication Date

2007

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Theory and Technology for Computational Narrative: An Approach to Generative and  
Interactive Narrative with Bases in Algebraic Semiotics and Cognitive Linguistics

A dissertation submitted in partial satisfaction of the  
requirements for the degree Doctor of Philosophy

in

Computer Science and Cognitive Science

by

Douglas Alan Harrell, Jr.

Committee in charge:

Professor Geoffrey Voelker, Chair  
Professor Gilles Fauconnier, Co-chair  
Professor Lev Manovich  
Professor Simon Penny  
Professor Victor Vianu

2007

Copyright

Douglas Alan Harrell, Jr.

(a.k.a. D. Fox Harrell)

All rights reserved.

The dissertation of Douglas Alan Harrell, Jr. is  
approved, and it is acceptable in quality and form for  
publication on microfilm:

---

---

---

---

Co-chair

---

Chair

University of California, San Diego

2007



## DEDICATION

*Dedicated with love and honor to the memories of*

*Mrs. Janet L. Williams  
and  
Dr. Joseph A. Goguen*

*Special thanks for support to*

*D.L.H, D.A.H, Sr., A.N.H.,  
and  
S.*

## TABLE OF CONTENTS

Signature Page .....	iii
Dedication.....	iv
Table of Contents.....	v
List of Figures.....	viii
Acknowledgements.....	ix
Curriculum Vitae.....	xiii
Abstract.....	xiv
Chapter 1 Introduction .....	1
1.1 Goals and Perspectives .....	1
1.2 Contributions.....	8
1.3 Dissertation Outline.....	11
Chapter 2 Theoretical Framework .....	16
2.1 Computational Narrative .....	16
2.1.1 Narrative Intelligence .....	25
2.1.2 Sociolinguistic Narrative .....	28
2.1.3 Non-Digital Works .....	31
2.1.4 Text Based Interactive Fiction .....	33
2.1.5 Story Generation.....	36
2.1.6 Interactive Drama and Virtual Reality.....	41
2.1.7 Games and Dynamic Systems .....	48
2.2 Cognitive Linguistics.....	52
2.2.1 Overview of the Cognitive Linguistics Enterprise .....	52

2.2.2 Remarks on Cognitive Linguistics Methodology.....	57
2.2.3 Key Points in Metaphor Theory .....	60
2.2.4 Key Points in Conceptual Blending Theory .....	63
2.2.5 Critique of Conceptual Blending Theory and Responses .....	74
2.2.6 Computational Approaches to Analogy, Metaphor, and Conceptual Blending.....	79
2.2.6.1 Computational Approaches to Analogy .....	81
2.2.6.2 Computational Approaches to Conceptual Blending.....	87
2.2.6.3 A Computational Metaphor Theory Based Approach to Narrative.....	93
2.2.6.4 Discussion of Computational Approaches to Cognition.....	96
2.3 Algebraic Semiotics .....	98
2.3.1 Basic Semiotics Concepts .....	99
2.3.2 Algebraic Semantics .....	104
2.3.3 Algebraic Semiotics Definitions and Key Concepts .....	106
2.3.3.1 Semiotic Spaces .....	108
2.3.3.2 Semiotic Morphisms .....	113
2.3.4 Algebraic Semiotic Account of Blending .....	115
2.3.4.1 An Example of Blending Using Algebraic Semiotics .....	118
Chapter 3 Alloy.....	126
3.1 Knowledge Representation for Alloy .....	129
3.2 The Alloy Blending Algorithm .....	132
3.3 Summary Discussion of Alloy .....	136
Chapter 4 GRIOT.....	139

4.1 Formalization of GRIOT's Functionality: Narrative Construction as Structural Blending .....	140
4.2 The GRIOT Architecture .....	143
4.2.1 The Event Structure Machine.....	149
4.3 Levels of Use .....	152
4.4 Examples of Interactive Polymorphic Poetry .....	154
4.4.1 The Girl with Skin of Haints and Seraphs .....	155
4.4.2 The Griot Sings Haibun .....	161
4.4.3 Walking Blues Changes Undersea .....	164
4.4.3.1 Narrative Imagining .....	164
4.4.3.2 <i>Walking Blues Changes Undersea</i> Functionality and Structure.....	165
4.4.4 Remarks on Aesthetic Analysis of Computational Poetry.....	169
Chapter 5 Conclusion.....	174
5.1 Recapitulation of Results.....	175
5.2 Toward Dimensions of Aesthetic Style.....	178
5.3 Directions for Future Work.....	181
Appendices .....	185
Appendix A: Alloy Output for the Houseboat Blend.....	185
Appendix B: The Girl with Skin of Haints and Seraphs .....	198
Appendix C: The Griot Sings Haibun .....	202
Appendix D: Walking Blues Changes Undersea.....	210
References .....	213

## LIST OF FIGURES

Figure 2.1: The “Sun is a King” metaphor .....	62
Figure 2.2: Blending in an 18 <sup>th</sup> century political cartoon.....	64
Figure 2.3: A diagram of a conceptual integration network (Turner, 2006) .....	67
Figure 2.4: The Conceptual Spaces in the Buddhist Monk Riddle.....	68
Figure 2.5: Cross-Space Mappings in the Buddhist Monk Riddle .....	69
Figure 2.6: Selective Projection in the Buddhist Monk Riddle.....	70
Figure 2.7: The sign for the Tirthankara Suparshvananth.....	100
Figure 2.8: The sign for the Nazi party .....	101
Figure 2.9: A sign from the “Youth Against Racism in Europe” website .....	101
Figure 2.10: Peirce’s Semiotic Triad .....	102
Figure 2.11: Three signs warning of the danger of a child being on the road.....	103
Figure 2.12: A Blend Diagram .....	116
Figure 2.13: The Houseboat Blend .....	122
Figure 2.14: The Boathouse Blend .....	122
Figure 3.1: The House/Boat Input Graph.....	132
Figure 4.1: The GRIOT architecture.....	148
Figure 4.2: A screenshot from <i>The Griot Sings Haibun</i> .....	162
Figure 4.3: A screenshot from <i>The Griot Sings Haibun</i> .....	163
Figure 4.4: GUI design screenshot for <i>Walking Blues Changes Undersea</i> .....	168
Figure 4.5: A GUI design screenshot for <i>Walking Blues Changes Undersea</i> .....	169
Figure A.1: Seven screenshots from <i>The Griot Sings Haibun</i> .....	209
Figure A.2: Two GUI design screenshots for <i>Walking Blues Changes Undersea</i> .....	212

## ACKNOWLEDGMENTS

The San Diego morning fog, night ocean, manzanita trees, torrey pine trees, anise scented chaparral...what made you so? Thank you for being.

Joseph Goguen, gardener of algebraic flowers, I hope that the bulbs you planted here blossom in the ways you always imagined they could. You exist in all of these pages, I honor your memory with deep gratitude. I shall cherish your many gifts always.

Geoff Voelker, thank you for your kindness, chairmanship of my committee, insight into the research process, and guidance. Gilles Fauconnier, thank you for your visionary approach to cognitive science, for research values, methods, and results, and disciplined passion for imagination. Lev Manovich, thank you for your deep insights into the nature of the computational medium, inspirational ability to thrive as both an artist and theorist, mentorship, and generosity with your time. Simon Penny, thank you for your activism, values-based engagement with technology and art, and for the invigorating experience of wind and sea spray on your sailboat. Victor Vianu, thank you for your warmth, interest in both technical and artistic practices, and your deeply engaging and informative course on logic, it was sincerely a favorite of mine.

Rafael Núñez, George Lewis, Charles Elkan, David Borgo, Bertram Turetzky, Seana Coulson, thank you all for being faculty members willing to share your time, interest, friendship, and insight.

Guilian Wang, Kai Lin, Jon Phillips, Dana Dahlstrom, Doug Turnbull, Monica Marcus, Abbie Gray, and the many other members passing through the Meaning and Computation Lab, thank you for your friendship and collegiality. Ernesto Morales-Perea, Erez Perelman, Koji Nakano, thank you for your friendship and camaraderie as fellow UCSD graduate students. Rod Burstall, Werner Kuhn, Kokichi Futatsugi, Karl Meinke, and other friends of the Meaning and Computation Lab, thank you for your support of Joseph Goguen and his Meaning and Computation Lab, your visits inspired me greatly.

Ken Knoespel, Janet Murray, and the faculty, staff, and students I met in the School of Literature, Communication, and Culture at Georgia Tech, thank you for your generosity, beautiful program, the opportunity to share ideas with you, and the burst of inspiration you gave me as I brought this dissertation to a close.

Michael Mateas, Noah Wardrip-Fruin, Nick Montfort, Andrew Stern, Mary Flanagan, Scott Rettberg, Ian Bogost, Pat Harrigan, Eric Zimmerman, Frank Lantz, the organizers and attendees of DAC 2005, the organizers and attendees of the 1999 AAAI Symposium on Narrative Intelligence, thank you for the influence you have had on my path, and for providing a community of shared interests and practices.

Ryoko Amade Goguen, thank you for your deep, gorgeous, and engaging music, your caring for Joseph, and your kinship.

Capoeira N’Golo Preto Velho, Capoeira Angola Center of Los Angeles, thank you for providing me with examples of humility, grace, and strength, a rich community of shared values, and for allowing me to train, within the small roda, practices that are even more valuable in this large one. Mestre João Grande and his students, thank you for providing me the foundation to receive this.

Kevin Ellington Mingus, thank you for brotherhood.

Diane Harrell, Douglas Harrell Sr., Alana Harrell, Anika Harrell, Elan Harrell, Jason Ryan Harrell Diaz (welcome to the world), Shantinath Veeragoudar, Sarojani Veeragoudar, the Harrell, Williams, and Veeragoudar families, thank you all with love.

Sneha Veeragoudar Harrell, thank you, with love and devotion.



The text of **Subsection 2.1.2**, **Section 2.3**, and **Section 3.2**, in part, contains material that is a reprint of, or has been submitted for publication in: Goguen, J., & Harrell, D. F. (2007); Foundations for active multimedia narrative: Semiotic spaces and structural blending (in preparation); *Interaction Studies*. The dissertation author was a co-author of this paper.

The text of **Chapter 4**, in part, contains material that is a reprint of, or has been submitted for publication in: Harrell, D. F. (2005); *Shades of Computational Evocation and Meaning: The GRIOT System and Improvisational Poetry Generation*; Paper presented at the 6th Digital Arts and Culture Conference, Copenhagen, Denmark; and Harrell, D. F. (2006). *Walking Blues Changes Undersea: Imaginative Narrative in Interactive Poetry Generation with the GRIOT System*; Paper presented at the AAAI 2006 Workshop in Computational Aesthetics: Artificial Intelligence Approaches to Happiness and Beauty, Boston, MA. The dissertation author was the sole author of these papers.

The text of **Chapter 4**, **Section 5.1**, and **Section 5.2**, in part, contains material that is a reprint of, or has been submitted for publication in: Goguen, J., & Harrell, D. F. (2007); Style as a Choice of Blending Principles (submitted); In S. Argamon, K. Burns & S. Dubnov (Eds.), *The Structure of Style: Algorithmic Approaches to Understanding Manner and Meaning*; Berlin: Springer. The dissertation author was a co-author of this paper.

## CURRICULUM VITAE

**Education**     **University of California, San Diego;**  
Department of Computer Science and Engineering, La Jolla, CA  
**Doctor of Philosophy** Computer Science and Cognitive Science, 2007

- National Science Foundation Graduate Research Fellowship
- Ford Foundation Dissertation Fellowship
- Eugene Cota-Robles Fellowship
- AGEP - MASEM Fellowship

**New York University;** Tisch School of the Arts, New York City, NY  
**Master of Professional Studies** Interactive Telecommunications, 2000

- Jacob K. Javits Fellowship

**Carnegie Mellon University;**  
School of Art, Department of Philosophy, School of Computer Science,  
Pittsburgh, PA

- Carnegie Mellon University Merit Scholarship

**Bachelor of Fine Arts** Art,  
• concentration Electronic and Time Based Media,  
*with University and College Honors*, May 1998

**Bachelor of Science** Logic and Computation,  
• concentration in Artificial Intelligence,  
*with University and College Honors*, May 1998

**Minor** in Computer Science

**University of California, Berkeley,** Berkeley, California,  
August 1995 - August 1996

*Phi Beta Kappa honor society member*

### Academic Employment

**Georgia Institute of Technology;** Assistant Professor  
Digital Media, School of Literature, Communication, and Culture  
Tenure Track, August 2007 (Visiting, Spring 2007)

## ABSTRACT OF THE DISSERTATION

Theory and Technology for Computational Narrative: An Approach to Generative and Interactive Narrative with Bases in Algebraic Semiotics and Cognitive Linguistics

by

Douglas Alan Harrell, Jr.

Doctor of Philosophy in Computer Science and Cognitive Science

University of California, San Diego, 2007

Professor Geoffrey M. Voelker, Chair

Professor Gilles Fauconnier, Co-Chair

This dissertation presents theoretical and technical support for, and implementations of, narrative computational media works with the following characteristics: generative content, semantics-based interaction, reconfigurable narrative structure, and strong cognitive and socio-cultural grounding.

A system that can dynamically compose media elements (such as procedural computer graphics, digital video, or text) to result in new media elements can be said to generate content. The GRIOT system, a result of this dissertation, provides an example of this. It has been used to implement computational poetry that generates new narrative poems with varying particular concepts, but fixed themes, upon each execution. This generativity is enabled by the Alloy system, which implements an algorithm that models key aspects of Gilles Fauconnier and Mark Turner's theory of

conceptual blending. Alloy is the first implementation of Joseph Goguen's algebraic semiotics approach to blending. (Fauconnier & Turner, 2002; Goguen, 1998) This research also contributes to the theory of algebraic semiotics by developing a blending-based notion of style.

Semantics-based interaction means here that (1) media elements are structured according to the meaning of their content, and (2) user interaction can affect content of a computational narrative in a way that produces new meanings that are constrained by the system's author. "Meaning" in this case indicates that the author has provided formal descriptions of domains and concepts pertinent to the media elements and subjective authorial intent.

Meaning can also be reconfigured at the level of narrative discourse. The formal structure of a computational narrative can be dynamically restructured, either according to user interaction, or upon execution of the system as in the case of narrative generation.

Strong cognitive and socio-cultural grounding here implies that meaning is considered to be contextual, dynamic, and embodied. The formalizations used derive from, and respect, cognitive linguistics theories with such notions of meaning. Furthermore, the notion of narrative here is not biased toward one particular cultural model. Using semantically based media elements as a foundation, a cultural producer can implement a range of culturally specific or experimental narrative structures.

# **Chapter 1 Introduction**

## **1.1 Goals and Perspectives**

Imagination, computation, and expression blend in the goal of inventing new narrative forms for new media. New narrative forms may become lost to history as mere curiosities, or they may revolutionize and invigorate our possibilities for communicating our thoughts, feelings, dreams, social constructions, and even our senses of selves to one another. New media technologies, likewise, may fade to grey obsolescence, or they may become the basis for new conventions sending its artists and theorists scrambling to discover its language – a language that allows one to manipulate the characteristics of the medium to convey a range of expressive work ranging from functional data to evocative fictions. The hope here is to offer theory and technology as a step on the path toward the revolutionary and convention-establishing fates, while leaving behind the disappointments of the lost and obsolete.

The basis for this hope is a novel and integrated perspective on creative imagination, computer science, and cultural expression.

While this dissertation offers specific advances in computational narrative, perhaps its central contributions are theoretical and methodological. Unified by the overarching goal of expression and a central concern for narrative, this research tightly integrates approaches from computer science, cognitive science, and artistic production such that each of these areas mutually informs the others:

1) **Imaginative Cognition | Computer Science**

Computer science allows for formalization, precise modeling, implementation, and experimentation with recent cognitive science theories that understand cognition as distributed, embodied, and situated.

2) **Computer Science | Artistic Production**

Artistic production drives computational solutions and enables critical technical practices by introducing subjectivity and human-centricity.

3) **Artistic production | Imaginative Cognition**

Cognitive science investigation of imagination provides scientific accounts of how narrative imagining, metaphor, conceptual integration, and related phenomena are fundamental to human thought, and of how imaginative fiction maps on to real human conditions.

The goals and approaches outlined above are important because I believe that computing can contribute to understanding and expressing nuances of the human social condition, a domain traditionally investigated by the humanities. Narrative is central to human communication. However, many forms of computer-based interactive narrative have failed so far to capture the popular consciousness. Computer gaming has produced many examples of popular narrative experiences, and both computer gaming artifacts and theory of computer gaming deeply inform the work here. However, many computer games are not primarily narrative, and even those that are often do not feature technical support for allowing user interaction to dynamically generate narrative meaning (though players undoubtedly generate their own narrative interpretations of gameplay). I believe that an approach that takes seriously culturally based forms of narrative imagination and its expression, at the same time as considering the (cognitive) scientific underpinnings of that imagination and its expression, and looking critically at the ways that computational media can allow for manipulation of those statements at the *semantic* level, holds great promise for the development of effective computational narratives.

This dissertation does not define a singular eventual form of computational narrative such as virtual reality, hypertext, or narrative computer games. Instead, the starting point is meaning and narrative cognition. I propose a model to allow an author to represent her or his subjective meanings, and I theorize how these meanings can interact with knowledge bases of media assets, data structures for representing a wide range of discourse structures, and an engine to orchestrate interactive events. This basis allows the author of an interactive narrative to utilize a combination of

semantic data structures, algorithmic blending of those structures, a reconfigurable model of narrative structure, and a subjective (but less easily manipulable by computational means because semantics are not explicitly represented) knowledge base. My approach starts by considering what meaningful content an author has to express, and how structure and interaction can enable expression of that content.

This approach encourages an author to utilize whatever computational means necessary in order to convey her or his particular narrative. At the same time, it also allows an author to create constructive spaces in which user interaction or content production are the core expressive aspects of the work. This point of view contrasts to those that start from imagining one futuristic form of computational narrative as a “holy grail,” for example, allowing users to manipulate objects and interact with characters within a narrative virtual world. The theoretical bases proposed here are sufficiently general as to be used to articulate models of immersive virtual worlds, computationally generative poetry, narrative computer games, and other related forms.

The view of narrative and poetry taken in this dissertation can be captured by integrating the following quotations:

Poetics deals with problems of verbal structure, just as the analysis of painting is concerned with pictorial structure. Since linguistics is the global science of verbal structure, poetics may be regarded as an integral part of linguistics. (Jakobson, 1960)

... models of language and linguistic organization proposed should reflect what is known about the human mind, rather than purely aesthetic dictates such as the use of particular kinds of formalisms or economy of representation ... (Evans, Bergen, & Zinken, 2006)

Poetry, narrative, and their analyses are seen as within the domain of inquiry of linguistics (if not the structuralist model of Jakobson’s time). Linguistic phenomena



are seen as observable manifestations of human cognitive processes. (Fauconnier, 2000) Since I am interested in developing new media forms and genres that may or may not possess all of the characteristics of narrative or poetry from strictly literary traditions, my view is necessarily broad. I begin with cognitive perspectives on narrative imagining, and its component structures and processes, as basic semantic “building blocks.” Poetry is taken to be the wide domain of verbal art, which is not given status greater or less than other linguistic phenomena. Indeed, a major insight of the cognitive linguistics enterprises is that the same cognitive processes involved in everyday common sense reasoning also underlie literary creativity.

This does not mean that I ignore the cultural specificity of narrative and poetic forms or the entire history and insight to be gained by literary theoretic or other approaches to narrative. Rather, I invoke cognitive definitions in order to admit a wide range of specific influences and insights. The following is intended to roughly distinguish this approach from related approaches in the area of narratology (the study of narrative). This discussion is meant to provide an orientation for readers unfamiliar with this field.

There is no consensus on definitions of terms such as “story” or “narrative.”

N. Katherine Hayles summarizes the situations well in (Hayles, 2005) as follows:

The binary established by the Russian formalists of *fabula* and *sjuzhet* followed the distinction, dating back to what Gerard Genette calls the “pre-history” of narratology, of story and plot. Mieke Bal defines *fabula* as the “material or content that is worked into a story,” while the story itself is “defined as a series of events.” This definition is more or less echoed by Genette, Seymour Chatman, Shlomith Rimmon-Kenan, and others. The *sjuzhet*, on the other hand, is the order of appearance of the events in the work itself, or [as] Chatman, quoting Boris Tomashevsky, puts it, “how the reader becomes aware

of what happened.”” Different theorists transpose these older terms into binaries with slightly different inflections, including story and discourse (Chatman), fabula and story (Bal), and story and narrative (Genette), who sees both these terms deriving from a third term, narrating. As these examples show, there is no consistent terminology...

Within cognitive science, Mark Turner has described stories as “dynamic interactions of events, actors, and objects,” presenting a quite minimal model of narrative aimed at capturing the skeletal pattern underlying narrative imagining. (Turner, 1996) This does seem to parallel the minimal literary theoretic definition provided by Manfred Jahn in (Jahn, 2005), in which stories are sequence of events involving characters (‘events’ including “both natural and nonnatural happenings”). Jahn also presents a minimal definition of narrative as “story presented via media,” as in Genette’s terminology in the Hayles quotation above. As a broad frame, this dissertation shall accept Turner’s view of narrative, with the realization, however, that it is quite a general description narrative when analyzing the rich conventions, innovations, and meanings of any specific cultural form. Under such view of narrative, the examples of computational poetry discussed here are also considered to be narrative, though the underlying technology is not limited to producing narrative discourse.

It is informative to distinguish how this cognitive linguistics perspective on narrative affects the framework for analysis. Jahn describes two major approaches to narrative analysis as “discourse narratology,” which “analyzes the stylistic choices that determine the form or realization of a narrative text,” and “story narratology,” which

“focuses on the action units that ‘emplot’ and arrange a stream of events into a trajectory of themes, motives and plot lines.” (Jahn, 2005) David Herman describes another approach called “cognitive narratology,” in which “both narrative theory and linguistics should instead be construed as resources for cognitive science. Or rather, narratology, like linguistics, can be recharacterized as a subdomain of cognitive-scientific research. From this perspective, both language generally and narrative specifically can be viewed as tool-systems for building mental models of the world.” (Herman, 2000) From the perspective of cognitive linguistics, cognitive narratology would focus on elaborating, for example, projection of action-stories onto event-stories, how these arise stories arise from image schema, conceptual metaphor and blending, and related cognitive phenomena described later in **Section 2.2**.

“Computational narratology,” in the sense employed occasionally by the author and Joseph Goguen, is deeply influenced by the cognitive linguistics approach to narrative, and provides techniques from computer science to provide a language to describe cognitive insights and to implement narrative effects of the type analyzed in discourse narratology.

This approach reflects a paradigm shift in cognitive science away from elegant formal views of language that are not necessarily cognitively grounded (or even plausible), toward grounding in empirical results regarding cognitive phenomena underlying language. This computational approach to narrative also represents a paradigm shift in computer science. The computational narrative system presented in this dissertation does not attempt to implement a model of creativity (human or computer). Instead, it aims to provide a language for representing the human author’s

expressive intentions along with subjective and constructive possibilities for user interaction and content generation. That is, the goal is to enable a human author to construct computational narratives featuring semantic underpinnings informed by cognitive science, and all of the potential for interpretively meaningful interaction and generation that such an underpinning provides. Human narrative imagining, conceptual blending, and related processes allow for user and author to negotiate meanings. The underlying computational structures an algorithms merely exploit regularities of such human cognitive processes, but do not simulate them.

## **1.2 Contributions**

The research described here consists of theoretical and technical support for, and implementations of, narrative computational media works with the following characteristics: generative content, semantics based interaction, reconfigurable narrative structure, and strong cognitive and socio-cultural grounding.

Generative content means that the system should be able to compose new content from media elements (in the forms of text, computer graphics, audio, or other media) on the fly. The GRIOT system, one of the results of this effort, provides an example of such generation. (Harrell, 2005a, 2005b, 2007) It has been used to implement computational poetry that generates a new narrative poem with fixed themes but varying metaphors upon each execution. This computational poetry is implemented on the basis of several knowledge bases entered by the author of a

computational poem. The author constructs a knowledge base of domains (sets of typed binary relations), a narrative structure implemented as a new type of automaton called a “probabilistic bounded transition stack machine” or “Event Structure Machine,” a knowledge base of textual templates organized according to narrative type and featuring variables that can be replaced with generated content. (Goguen & Harrell, 2007b; Harrell, 2006) A key aspect is that the author defines the semantic rules that determine how content is generated.

Generativity is enabled by the Alloy system. Central to Alloy is an algorithm that implements key aspects of Gilles Fauconnier and Mark Turner’s theory of conceptual blending; it is also the first implementation of Joseph Goguen’s algebraic semiotics approach to blending (in turn, the first formal mathematical approach to blending). (Fauconnier & Turner, 2002; Goguen, 1998) Alloy takes in a data structure called an “input diagram” consisting of data structures representing Fauconnier and Turner’s conceptual spaces and mappings between them. Alloy returns an output diagram consisting of a “blended” conceptual space that integrates elements from conceptual spaces in the input diagram, and mappings to the blended conceptual space. Alloy, however, is not considered to model human cognitive processing. It is seen as a system to allow a researcher to formally describe conceptual spaces empirically determined or hypothesized by cognitive science researchers and to explore structural combinations of the conceptual spaces. As such, it is not intended as an “artificial intelligence” project, but rather as a system that allows researchers to invoke the utility of computational experimentation when appropriate, while respecting the limitations of computational models for describing human conceptual

processes. The formalization used in Alloy and GRIOT is based in Joseph Goguen's algebraic semiotics, which uses algebraic semantics and specification to describe sign systems.

Semantics-based interaction means that (1) media elements are structured according to the meaning of its content, and (2) user interaction can affect content of a computational narrative in a way that produces new meanings that are constrained by the system's author. "Meaning" in this case means that the author has provided formal descriptions of domains and concepts (as described above) pertinent to the media elements and authorial intent.

Reconfigurable narrative structure means that the formal structure of a computational narrative can be dynamically restructured, either according to user interaction, or upon execution of the system as in the case of narrative generation. The "Event Structure Machine" is the component that affords reconfigurable narrative structure. (Goguen & Harrell, 2007b; Harrell, 2006)

Strong cognitive and socio-cultural grounding here implies that despite the use of formal descriptions of semantic concepts, meaning is considered to be dynamically constructed, distributed among our selves and our artifacts, situated in social contexts, and embodied in our physical experiences. (Fauconnier & Turner, 2002; Hutchins, 2000; Lave & Wenger, 1991; Varela, Thomson, & Rosch, 1991). The formalizations used also are inspired by, and respect, cognitive linguistics theories with such notions of meaning, and in practice a system author must be sensitive to these issues to effectively utilize the technical framework provided. Furthermore, the notion of narrative here is emphatically not biased toward one particular cultural model. The

architecture is layered so that, atop a technical layer that provides an authoring platform, a cultural producer can implement her or his desired model of narrative (whether Labov's empirically based narrative structure of personal experience from sociolinguistics, or Gerald Vizenor's narrative theory of Native American trickster tales). (Harrell, 2005a; Labov, 1972; Vizenor, 1989)

Toward all of these ends, the work here also contributes to the theories of algebraic semiotics and cognitive semantics by developing a blending-based notion of style and some mathematics in order to extend the algebraic semiotics model to allow for generativity and the specific applications described above.

### **1.3 Dissertation Outline**

A challenge, and boon, of an interdisciplinary dissertation is the fact that such work invites readers from differing disciplinary backgrounds to consider it. While a boon, of course, is that the work may be influential and interesting to people of varying intellectual traditions, one challenge is that necessary background for one audience may prove to be either rudimentary, advanced, or auxiliary for another. Worse, each tradition carries with it a set of values, methods, and goals which may be challenging to integrate, and at worst may be incommensurable.

The challenge is handled in this dissertation by defining an overarching set of goals and presenting individual results on the path toward those goals in the language of the discipline to which the results are most relevant (e.g., the description of the Alloy algorithm is presented within the tradition of computer science and

engineering). At times, a hybrid discourse is constructed because a natural alignment between the goals of multiple disciplines is formed (e.g., the work in computational poetry speaks to an integrated framework of creative writing, literary theory, cognitive science, and computer science). At other times, it is necessary to clearly articulate the relationship between disciplines and that clear articulation becomes a new theoretical result (e.g., clearly describing the role of computational modeling for scientific experimentation with conceptual blending theory). At other times again, it is important to offer an introductory account of an entire field of study because practitioners from one field may have had no previous exposure to the field (e.g., some practitioners from engineering may have had little exposure to semiotics theories). This dissertation attempts to balance all of these concerns into a coherent and compelling text. A gentle request is made to the reader to put aside disciplinary biases and submit to the possibility of integrated interdisciplinary goals. The read will be more pleasurable, and in that light I believe the contributions here can be most sensitively received and evaluated.

Bearing in mind the interdisciplinary relationships described above, **Chapter 2** of this dissertation describes the theoretical foundations of the dissertation's contributions. **Section 2.1** and its subsections describe various approaches to narrative ranging from sociolinguistics to experimental literature, and including various forms of narrative in computational media. By presenting this overview of approaches and antecedent computational narrative forms, **Section 2.1** comprises a review of literature that influences the approach here and a vision of how they are integrated into a coherent research project.



**Section 2.2** presents ideas from the enterprise of cognitive linguistics within the field of cognitive science that are central to the research problem here. This account is aimed primarily at readers outside of the field of cognitive linguistics and helps to situate the theories developed within that enterprise in relation to earlier theories. The account of cognitive linguistics is also aimed at relating the aims of the enterprise to issues in linguistics that have proven influential in computer science and artificial intelligence. To better situate the empirical foundations and theoretical import of the specific cognitive linguistics theories, **Subsection 2.2.1** describes a high level view of the philosophical commitments and historical context of the cognitive linguistics enterprise. **Subsection 2.2.2** describes methods used in cognitive linguistics. **Subsection 2.2.3** presents an overview of the influential theory of metaphor. **Subsection 2.2.4** presents an overview of conceptual blending theory, one of the central theoretical influences on the research here. Conceptual blending theory is a young and developing theory and, as such, is not without its controversies. **Subsection 2.2.5** presents some of the central controversies and criticisms of the theory and offers some remarks on them. Since one of the results of this dissertation is an algorithm that implements aspects of conceptual blending, **Subsection 2.2.6** presents a review of other work that applies computational techniques to conceptual blending and related areas in cognitive science.

**Section 2.3** presents Joseph Goguen's theory of algebraic semiotics. (Goguen, 1998) Together we pioneered the use of his theory toward the purpose of computational narrative. Indeed this theory forms the bridge between cognitive science and computer science used here. It also forms the bridge between computer

science and socio-cultural and artistic meaning, so it is one of the most crucial aspects of the theory and technology underlying all of the results of this dissertation.

Algebraic semiotics is used to formalize conceptual blending theory, but just as important is the philosophy underlying algebraic semiotics, i.e., formalization is used only as a descriptive (and implementable) tool. Language and meaning themselves are *not* viewed as formal by nature, and this work respects the insights of the cognitive linguistics enterprise in which human meaning is considered to be embodied, distributed, and situated. Formalization alone does not offer any greater cognitive plausibility to research results. This balance between formal methods that are necessary in computer science and deep understanding of the socially, culturally, bodily, and cognitively constructed nature of meaning is a reason why the algebraic semiotics approach is desirable for computational narrative applications. This section is not merely a literature review, but includes new results by Joseph Goguen and the author.

**Chapter 3** presents the Alloy system, including its knowledge (data) representation structures in **Section 3.1**, and its algorithm for conceptual blending in **Section 3.2**. **Section 3.3** presents an account of the limitations of the system and how, in part, these limitations reflect the natural limitations of computational methods as tools for cognitive linguistics research. In this regard, the algorithm itself, and the theoretical discussion of its relationship to human meaning construction, are both presented as important results.

**Chapter 4** presents the GRIOT system for implementing computational narratives. **Section 4.1** formally describes the problem of narrative construction

within the framework of algebraic semiotics as the blending of structures. **Section 4.2** presents a detailed view of the GRIOT architecture. **Section 4.3** describes various levels of use including multiple levels authorship, readership, performance (and uses that exist in-between all three) enabled by GRIOT. **Section 4.4** presents several examples of computational poetry as case studies implemented with GRIOT. Each of these case studies raises unique theoretical issues and introduces particular implementation advances. **Section 4.4** concludes with remarks toward evaluating the “success” of the case studies, understanding that engineering and scientific models of success may not apply to aesthetic works.

**Chapter 5** concludes the dissertation with a recapitulation of the results and contributions, remarks on how the approach here broadly can lead to new accounts of “style” in aesthetic and cultural expression, and reflection on the possibilities and future directions left to explore along the trajectory begun here.

## **Chapter 2 Theoretical Framework**

This chapter presents the theoretical framework used in this dissertation.

Because of the multi/interdisciplinary nature of the endeavor, this section is in part a literature review, but in part a novel framework that integrates a wide-ranging set of academic concerns centering the issue of computational narrative. The focus here is on the computer science and cognitive science dimensions of the work, but at times it is impossible to separate these from issues in digital media art, literary theory, semiotics, computer game engineering and studies, and related fields.

### **2.1 Computational Narrative**

Narratives have different characteristics depending upon the medium through which they are delivered. This is as true for computational media as for literature or cinema. Unearthing the characteristics of computational media is a unique problem because the computer is a general machine. It can emulate previous technological

media ranging from spoken word to digital video. It is instructive then to ask what it means to move a media form, in particular a narrative media form, onto the computer.

Computational media have their own specific characteristics such as abstract data structures, dynamic execution, polymorphic representation, user feedback channels, distributed networks, and massive storage. I believe that powerfully evocative new narrative forms can be constructed to take advantage of these unique characteristics of computational narrative. There have been many approaches to making narrative structure and meaning amenable to computational manipulation. This section introduces some influential attempts to theorize and create computational narratives, and some theoretical and cultural precursors to the computational narrative approach proposed in this dissertation.

In (Murray, 1997), Janet Murray describes a set of characteristics of computational media especially relevant for interactive narrative. Toward this end she defines the four essential properties of digital environments as: procedural and participatory (which give rise to interactivity), and spatial and encyclopedic (which give rise to immersion). Her set of properties has been influential for research in computational narrative, though other authors have varied in their emphases. Lev Manovich, interested in new media forms both narrative and otherwise, proposes a different set of characteristics of computational media in general: numerical representation, modularity, automation, variability, transcoding. There is overlap between the two, Murray's procedural property contains elements of each of Manovich's categories. In this regard, Manovich's characteristics can be said to refine the notion of the procedural nature of the medium. However, numerical representation

and modularity also contribute to the encyclopedic nature of digital environments. Murray's encyclopedic property is enabled by both increases in the capacity of data-storage hardware and by the economy of data representations and the efficiency of accessing data. In contrast, Manovich's set seems primarily concerned with software issues. The properties and characteristics offered by both authors offer insights into general affordances offered uniquely by computational media for producing cultural artifacts, but operate at different grain sizes and vary in their degrees of specificity and adherence to categories used within computer science. Here, it is important to combine the more granular analysis of Manovich with the more restricted (in its focus on digital environments) application of analysis of Murray. I offer the set of characteristics below in dialogue with those of Murray and Manovich with the goals of (1) focusing on those characteristics I feel are most salient for the issue of narrative, and (2) describing the characteristics in further dialogue with those recognized in computer science.

The critical characteristics of the computational medium that are especially relevant for computational narrative are:

- (1) Abstract Data Structure
- (2) Dynamic Execution (Algorithmic Processing)
- (3) Polymorphic Representation
- (4) User Feedback Channels
- (5) Distributed Networks
- (6) Massive Storage

Abstract data structuring is often invoked as non-linearity and data-tagging in discussion of interactive narrative forms. Though it is standard to explicitly compose structures and enforce data typing in computer science, strict enforcement of data structure and type becomes novel when imported into the realm of narrative. Narrative information stored in a printed book is not modular or dynamically manipulable; it is not possible to add a new node and have it trickle down to its appropriate location in a tree when using film. The granularity of recombination of narratives and the meaningful definition of narrative elements are implemented by means of abstract data structuring.

Dynamic execution provides the means for activating narrative information. It can allow new information to be deduced from a set of information and is the means to enact the restructuring of data structures. In a computational narrative, rules can be defined for search, manipulation, and retrieval of narrative information. Furthermore, non-narrative execution such as allowing user manipulation of synthetic computer graphics is possible.

Polymorphic representation is an especially unique quality of computational narrative. One way of accounting for this notion is the “semiotic morphism,” a mapping between two representational forms. This is invoked as the ability to separate structure from representation so that one structure can be visualized in a multiplicity of ways. An example of this is the ability of information to be displayed differently in a text only on a Lynx browser as compared to a full graphical representation on a more robust browser such as Firefox. Using XML to describe data, while not enforcing a particular representation, is another example. Hence, a

single story could be defined in such a way that it can be presented in text only, 3-D graphics, or digital video depending upon the situation.

User feedback channels present possibilities including collaboration, competition, and conversation between the reader/user/player and the narrative. This feedback can be used to affect the dynamic execution of the narrative and to navigate the narrative data structures. The form of this feedback can vary over a wide range of input devices and conventions.

Distributed networks enable the story to be stored and executed in a wide range of localities. This can increase processing power, but also can allow for remote users to access the narrative and for interaction among remote users.

Massive storage is simply the ability to retain extremely large volumes of data in compact form. This feature naturally interacts with the other features. Abstract data structuring is necessary to make large data volume manageable. Polymorphic representation along with massive storage and dynamic execution make 3-D graphical representation possible.

The characteristics discussed above are meant to give rise to computational narrative possibilities enabled by computational media in a broad, but comprehensive manner. While those characteristics seem very fundamental within contemporary computer science, they radically change the nature of narrative representation. By themselves however, these characteristics are inert. They combine to create specific possibilities for computational narratives that often distinguish them from narrative in other media (although, as is discussed in (Murray, 1997), there are examples in earlier media that can be thought of as “harbringers” or precursors that exhibit one or more of



these properties). Some interesting narrative possibilities within the computational medium are:

- (1) Emergent narrative
- (2) Device mediated narrative interaction
- (3) Spatialized narrative information
- (4) Narrative abstract data structures
- (5) Narrative and non-narrative interleaving
- (6) Synthesis of previous narrative media

Emergent narratives often arise from an author developing a system of rules that interact to allow stories to develop in unpredictable ways. Simulation games such as *The Sims* use this model for narrative. (Maxis, 2000) Although simulation games are not primarily narrative, they do create causal chains of events that a user interprets as the story of a city, character, family, or whatever the subject matter may be. These systems tend to be open ended, but constructed in such a way that users can interpret coherent meaning and causality from their experience of the system.

Massive multi-player online games allow for highly structured, yet socially open-ended, narrative driven, interaction between users at remote locations. The structure of the interaction is imposed by the user interface design and the perceived social affordances it represents. This social mediation by devices is not limited to sedentary console interaction. This mediated interaction can occur using cellular phones<sup>1</sup>, fax machines, global positioning devices, etc. A good example is the game

---

<sup>1</sup> The author along with Ho Ming (Sheldon) Chow also created an interactive narrative that played over cellular phone network called "Storyline."

*Majestic* that immersed users in a world of conspiracy theory and contacted players via telephone, email, fax, and other media to give them the sense that the story played out in the real world. (Anim-X, 2001) Social interaction can occur at multiple levels for users, within the context of a fictitious experience as in a game, within the context of real life interaction, or as a blend of the two (which commonly occurs in chat rooms).

Spatialized narrative information structure echoes the insights of Janet Murray. Here it is not introduced as a property characterizing digital environments, but rather as a possibility for computational narrative arising from abstract data structuring and massive information storage. Visualizing space in the computational realm often also involves dynamic execution and polymorphic representation to a lesser degree. Spatial visualization has become necessary for the dual reasons of making information management more tractable and for modeling the spatial nature of the real world. The definition of spatialized structure introduced here has a dual meaning. The first is that information can be used to model 3-D interactive worlds. The second is that information can be imagined as a space through which a user must navigate. It is noteworthy that this metaphor has become pervasive in contemporary thought about data.

Narrative abstract data structures are important because they potentially capture the computational enforcement of discrete choices humans make in life and the way those choices organize and reconfigure information. For example, considered interactively, branching paths present the user with options within a given interface; considered structurally branching paths represent the use of traditional computer

science data structures such as trees and graphs. Hypertext fiction is the dream of creating a new form by adding explicit hyperlinks within a literary text. Though hypertext fiction is generally read only by a small community of aficionados, in a larger sense the internet represents the triumph of hypertext narrative. It is such a ubiquitous form on the world wide web that it becomes easy to overlook the new styles of reading people have adopted when perusing webpages. Hypertext fiction has been absorbed into adventure games, online journalism, and many other computational narrative forms.

Because of the general-purpose nature of the digital computer, it is commonplace to interleave narrative interaction or presentation with other forms of interaction or presentation. Educational websites that accompany mathematics or science textbooks often include some pedagogical narrative interspersed with applets that allow users to experiment with visualizations or simulations of the concepts. Computer role-playing games often interleave random encounters that allow players to increase statistics of their characters with extremely cinematic linear narratives. Adventure games likewise often intersperse logical puzzles within branching narratives. This blending of narrative and non-narrative interaction often becomes conventionalized and the transition between the different forms of interaction can create a seamless (or frustratingly disjointed) experience for users.

For the same reason that the computer can interleave narrative and non-narrative interaction, it can emulate many previous media. Users typically and fluidly alternate between using a personal computer for conveying video, audio, games, text, and a variety of other applications adapted from other media. The fact that the

general-purpose computer can emulate previous media within the context of its own unique characteristics described above is a unique characteristic of the computational medium in its own right. Conventions can be adopted from media as ancient as oral narrative or as recent as 3-D platformer games. The interaction of these conventions and the ability to draw upon any of them is a new phenomenon in narrative media and computational media as a whole.

Within this view of the computational medium, the remainder of **Section 2.1** describes approaches to narrative that contribute to theory and technology of computational narrative presented in this dissertation. **Subsection 2.1.1** describes the research area of narrative intelligence, in which the interrelationship between both human and artificial intelligences and narrative is explored from a variety of disciplines. **Subsection 2.1.2** presents an approach to narrative from socio-linguistics that was used to inspire some of the first models of narrative structure developed using the Alloy and GRIOT systems. **Subsection 2.1.3** presents examples of non-digital narratives with experimental characteristics that act as inspiration points for aesthetic and formal possibilities for computational narrative. **Subsection 2.1.4** presents previous work in story and poetry generation. **Subsection 2.1.5** presents work in text-based interactive fiction along a brief description of seminal work in human-computer dialog. **Subsection 2.1.6** presents work in interactive drama, which uses advanced computing techniques with the aim of developing character driven narratives in virtual worlds that offers users a high degree of agency. **Subsection 2.1.6** presents several relevant issues from computer gaming.

### 2.1.1 Narrative Intelligence

Narrative intelligence<sup>2</sup> began as a term to describe a reading group that explored the contrasting ways in which texts are read in computer science and in literary theory. (Davis & Travers, 1999) It has developed into a unifying theme that brings together researches and artists in a range of disciplines to approach overlapping problems. Narrative intelligence considers the intersection of the development and interpretation of computational narratives, the development and interpretation of traditional media and conceptual narratives, and artificial intelligence research involving narrative. Narrative intelligence offers a useful account of different ways in which the relationship between narrative and computing can be viewed. Originally, the theme of narrative intelligence drew upon the following streams of influence: cognitive science and artificial intelligence, literary theory, art, drama, media studies, narrative psychology and sociology, user interface theory, software engineering, and social computing. (Mateas & Sengers, 1999)

Some observations made by narrative intelligence researchers Marc Davis and Michael Travers break down the important differences between the relationship with texts held by literary theorists and computer scientists. In some cases they did not account for literary theorists' analysis and creation of computational texts so the account of their observations below has been slightly supplemented. Some of these differences, in a form they admit as somewhat caricatured, are (Davis & Travers, 1999):

---

<sup>2</sup> The term "narrative intelligence" arises from the Narrative Intelligence Reading Group at the MIT Media Lab, which was started by Marc Davis and Michael Travers.

- (1) Reading natural language texts: The literary theorist closely analyzes style, implicit meaning, and structure while the computer scientist values quick extraction of core concepts and utility
- (2) Reading programming language texts: The literary theorist typically does not address programming language texts (though recent theorists such as Espen Aarseth and Ian Bogost have begun to do so) while computer scientists are able to develop detailed understanding of text functionality so that it can be executed or appropriated for new text production
- (3) Writing natural language texts: The literary theorist considers this to be the primary form of text production. Typically this production is done to analyze other texts. Stylistic innovation is rewarded. Computer scientists often produce these texts secondarily to programming language texts. They often are analyses and documentation of programming languages texts. Stylistic innovation is not highly rewarded, while clarity of exposition is valued.
- (4) Discussing and presenting work: The literary theorist uses speech as a rhetorical art form where innovation, cleverness, and complexity are rewarded. Computer scientists often value lack of ornament and simplicity of exposition. Demonstration of implemented work is highly valued.

The discussion above does not deal directly with technical issues within narrative intelligence, but it highlights some of the challenges within interdisciplinary

research. The schism between values between involved communities can strain communication of ideas and hinder researches drawing on multiple methodologies.

Some the main research and development agendas within narrative intelligence are (Mateas & Sengers, 1999):

- (1) Narrative Based Design of User Interfaces
- (2) Design of Narrative Agents
- (3) Support for Human Storytelling
- (4) Story Database Systems
- (5) Story-understanding Systems
- (6) Storytelling Systems
- (7) Interactive Fiction and Drama
- (8) Narrative Based Cultural Analysis

Though such research agendas are widely divergent they also are often mutually enriching. In this sense, narrative intelligence does not actually define a field of study proper, but rather a “family of resemblance concept, a cover term for a rich set of ideas.” (Mateas & Sengers, 1999) This does not dilute the value for creating a forum for collaboration, shared research knowledge, and establishment of some common terminology for interdisciplinary research. These are perhaps the most valuable contributions of the concept of narrative intelligence as a whole.

### 2.1.2 Sociolinguistic Narrative

Sociolinguists have done extensive empirical study of narratives of personal experience, which are told orally to a group of peers under natural conditions. (Labov, 1972; Linde, 1993) These are important here because these narrative forms do not arise from critical analysis of written texts, but represent narrative “in the wild” as encountered in every day interpersonal interaction. Such models can prove to be useful for developing computational narrative experiences that parallel aspects of human communication. The model below has been favored in the work of Joseph Goguen in his work on social issues of user-interface design, where it may have first been presented in the context of computer science research. (Goguen, 2001) Goguen and the author have presented this account as applied to computational narrative in joint work in (Goguen & Harrell, 2007b) and elsewhere.

The result of this work by William Labov, as refined by Charlotte Linde, can be summarized as follows (Goguen, 2001; Labov, 1972; Linde, 1993):

- (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.

This interpretation of narrative also employs **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 (however, they do not necessarily apply to non-Western languages and cultures; for example, Balinese narrative does not follow the narrative presupposition (Becker, 1979)). 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.

In (Goguen & Harrell, 2007a), we assert that 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,

$$\begin{aligned} \langle \text{Narr} \rangle &::= \langle \text{Open} \rangle (\langle \text{Cls} \rangle \langle \text{Eval} \rangle^*)^+ [\langle \text{Coda} \rangle] \\ \langle \text{Open} \rangle &::= ((\langle \text{Abs} \rangle + \langle \text{Ornt} \rangle) \langle \text{Eval} \rangle^*)^* \end{aligned}$$

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  $\langle \text{Narr} \rangle$  is for narratives,  $\langle \text{Cls} \rangle$  for narrative clauses, which potentially include evaluation,  $\langle \text{Eval} \rangle$  for stand-alone evaluative clauses,  $\langle \text{Open} \rangle$  for the opening section, which may include an orientation and/or abstract, and  $\langle \text{Coda} \rangle$  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).

### 2.1.3 Non-Digital Works

This brief subsection, rather than presenting a general account of harbringers of computational narrative as found in previous media (for such an account see chapter 2 of (Murray, 1997)), is meant to present several important examples relevant to the goals of the GRIOT system and works created with it discussed later in **Chapters 4 and 5**. An early relevant work is Raymond Queneau's 1961 "Cent Mille Millions de Poèmes" ("One Hundred Thousand Billion Poems"), originally published as a set of ten sonnets with interchangeable lines, but later made available in computer implementations. (Queneau, 1961) This work is relevant because of its exploration of the idea of writing as a combinatorial exploration of possibilities, which exemplifies the experimental literary group Oulipo's often whimsical use of mathematical ideas. This view is well explicated by another Oulipo member, Italo Calvino, in his essay/lecture "Cybernetics and Ghosts." (Calvino, 1982b) Calvino claims that writing is a combinatoric game and cites work such as Vladimir Propp's morphology of the folktale to support his thesis. (Propp, 1968) Calvino states "the operations of narrative, like those of mathematics, cannot differ all that much from one people to another, but what can be constructed on the basis of these elementary processes can present unlimited combinations, permutations, and transformations." In Calvino's novels, such as *If on a winter's night a traveler*, there is also a strong sense of narrative coherence and a concern for a careful balance between experimental form and meaningful expression. (Calvino, 1982a) The computational approach to narrative in this dissertation is more influenced by this concern for coherent subjective

expression in Calvino's work than merely the idea of utilizing mathematical techniques to arrive at poetry with variable structure for its own sake.

A classic earlier example of this concern for ways in which subjective meaning can emerge from experimental literary structure is Ryunosuke Akutagawa's 1922 short story "In a Grove" (and Akira Kurosawa's famous film adaptation *Rashomon*). (Akutagawa, 1999; Kurosawa, 1950) "In a Grove" is the tale of a brutal rape and murder told and retold from a variety of perspectives: from the vantage point of the victims, the perpetrator, and a by-stander. Meaning is constructed through the concrete knowledge that the event did take place and the shifting, conflicting reports of the event given by the characters. The conflicts between the different points of view are used to create an emergent statement about the human condition and the absence of truth as exemplified in the following dialogue from Kurosawa's film *Rashomon* (Kurosawa, 1950):

Priest: If men don't trust one another, then the earth becomes a hell.

Commoner: Right. The world's a kind of hell.

Priest: No! I don't want to believe that!

Commoner: No one will hear you, no matter how loud you shout. Just think. Which one of these stories do you believe?

Woodcutter: None makes any sense.

Commoner: Don't worry about it. It isn't as if men were reasonable.

In the hands of a careful poetic system implementer GRIOT provides a means to make use of "meaningful difference" between output to allow a global meaning to emerge from repeated execution of the system.

### 2.1.4 Text Based Interactive Fiction

Early attempts at interactive fiction focused solely on textual exposition. Some of these tackled the problem of believable character design, some addressed issues of interactivity and immersion within a fictional world (early adventure games), and some focused on branching structure within narrative. Joseph Weizenbaum's ELIZA is an oft-cited early example of an artificial intelligence "bot" designed to communicate with a human in a convincing manner. (Weizenbaum, 1966) ELIZA is limited to imitation of a Rogerian psychotherapist, greatly reducing the type of utterances that the program is expected to write. The interesting point about ELIZA here is that it only formally manipulates text; it attempts no understanding of user input whatsoever. The following example, taken from ("MIT course webpage (ELIZA assignment),"), is an excerpt of a conversation between a young woman and ELIZA (ELIZA's responses are in all caps) :

```
>Men are all alike.
IN WHAT WAY
>They're always bugging us about something or
other.
CAN YOU THINK OF A SPECIFIC EXAMPLE
>Well, my boyfriend made me come here.
YOUR BOYFRIEND MADE YOU COME HERE
>He says I'm depressed much of the time.
I AM SORRY TO HEAR THAT YOU ARE DEPRESSED
>It's true that I am unhappy.
DO YOU THINK COMING HERE WILL HELP YOU NOT BE
UNHAPPY
>I need some help, that much seems certain.
WHAT WOULD IT MEAN TO YOU IF YOU GOT SOME HELP
>Perhaps I could learn to get along with my
mother.
TELL ME MORE ABOUT YOUR FAMILY
```

Of course since all ELIZA can do is respond based upon the patterns of the user's inputs it is possible to thwart ELIZA and end up with a nonsensical dialogue ("MIT course webpage (ELIZA assignment),"):

```

HOW DO YOU DO? PLEASE STATE YOUR PROBLEM
>I'm wondering whether computers can think.
WHY DO YOU MENTION COMPUTERS
>Since you are one I thought I would ask you —
can you think?
DID YOU THINK THEY MIGHT NOT BE ONE YOU THOUGHT
YOU WOULD ASK I — CAN I THINK
>Let's talk just about you — not me. Can you
think?
YOU LIKE TO THINK I — NOT YOU — DON'T YOU
>Pardon?
WHAT DOES THAT SUGGEST TO YOU?
>What does is suggest to you?
TELL ME ABOUT WHAT DOES IT SUGGEST TO I
>That's nonsense — which of us do you mean by
I?

```

Still, in the general case, the pattern matching artificial intelligence computer program ELIZA constructs entertainingly effective conversations using formal manipulation of text.

Early adventure games allowed users to interact with the narrative system via a set of commands that do not differ too greatly from natural language. The user had to explore an imaginary space and solve logical puzzles in order to win the games. The following is an excerpt from the game *Zork* as presented in (Murray, 1997):

```

Welcome to Zork.

West of House.

Your are in an open field west of a big white
house with a boarded front door.  There is a
small mailbox here.

>go north

```

North of House.

You are facing the north side of a white house.  
There is no door here, and all the windows are  
barred.

>east

Behind house.

You are behind the white house. In one corner  
of the house there is a small window which is  
slightly ajar.

>open the window

With great effort, you open the window far  
enough to allow entry

>go in

A difference between such systems and strictly pattern matching systems such as ELIZA is that they kept track of state and categories of objects. They did not have to have replies for every single user action; they could check the action and objects involved in the action to see if they were of the proper sort. If so, then the action could be executed, if not, then a clever response was often returned. (Murray, 1997)

The witty writing of such games contributed to their success as much (if not more) than the programming techniques used to implement them. Basic dialogue in contemporary adventure games does not use techniques far more advanced than in these early games. Perhaps one difference is that interaction in current games often affects the global narrative state rather than only local state. This is the case in games such as *Ultima IV* discussed above where user choices affected the narrative development to result in (minimally) variably repeatable narratives. (Origin, 1985)

Hypertext represents the dream of creating a new form of narrative by combining literature with hyperlinking to provide immediate connections between different segments of the work. It allows stories to have explicit branching structures. As discussed above hypertext has come into ubiquitous use on the world wide web, though it has not proven as influential within literature. Branching and linking is such a fundamental part of computer science implementation that this shall play a role within a wide range of computational narratives. Truly flexible models, however, should not use branching as the sole means of interaction and structuring. Taking full advantage of the nature of the computational medium should often involve the use of dynamic algorithms that create emergent structure.

### **2.1.5 Story Generation**

Automated construction of compelling stories is another problem that computer scientists have tried to tackle. Influential systems include Meehan's TALE-SPIN and Bringsjord and Ferrucci's BRUTUS. (J. Meehan, 1981; J. R. Meehan, 1976) For contrast, the seminal TALE-SPIN will be depicted in comparison to the more recent BRUTUS. An excerpt from a typical story told by TALE-SPIN follows (Bringsjord & Ferrucci, 2000):

Once upon a time John Bear lived in a cave.  
John knew that John was in his cave. There was  
a beehive in a maple tree. Tom Bee knew that  
the beehive was in a maple tree. Tom was in  
his beehive. Tom knew that Tom was in his  
beehive...



A typical BRUTUS story (about betrayal) excerpt proceeds as follows (Bringsjord & Ferrucci, 2000):

Dave Striver loved the university – its ivy-covered clocktower, its ancient and sturdy brick, and its sun-splashed verdant greens and eager youth. The university, contrary to popular opinion, is far from free of the stark unforgiving trials of the business world: academia has its own tests...  
 ...At the defense, Dave eloquently summarized chapter 3 of his dissertation. There were two questions, one from Professor Rodman and one from Dr. Teer; Dave answered both, apparently to everyone's satisfaction. There were no further objections.  
 Professor Rodman signed. He slid the tome to Teer,; she too signed, and then slid it in front of [Dave's adviser] Hart. Hart didn't move...

TALE-SPIN essentially locates characters in a miniature world models with individual problems and then runs a simulator in which the characters try to solve the problem.

BRUTUS uses formalized themes and then uses first-order logic to construct narratives in accordance with these themes. Incidentally, the power of the natural language generator in each system is a separate matter than the generation of narrative structure. While TALE-SPIN seems to the reader to use a more obvious planning algorithm and simply output the results, BRUTUS uses rich textual descriptions. For example a university is an object implemented with both *positive iconic features*: {clocktowers, brick, ivy, youth, architecture, books, knowledge, scholar, sports} and *negative iconic features*: {tests, competition, 'intellectual snobbery'}. These features provide BRUTUS with a richer set to draw from in its natural language generation while TALE-SPIN only outputs results of its simulation algorithm.

These systems raise questions about the nature of authorship. The authors of BRUTUS have provided it with a rich set of primitives and structures to work with. In some sense the authors nearly explicitly seem to determine the set of possible stories that BRUTUS can tell. BRUTUS's stories are (arguably) interesting, but are they surprising or emergent? Perhaps with a large enough data-set and less highly specific data BRUTUS could construct a story that surprises its creators. Though the goal of BRUTUS is not to surprise its creators, within their AI and "literary Turing text competence" perspective, it seems that a story generation system should be able to come up with unanticipated output if it is to be considered robust. If all output could be anticipated from the structure of the program, then there is a good chance that a database narrative approach naively combining story elements would suffice. BRUTUS succeeds in its natural generation model and its narrative structure model, but to be useful more broadly to develop interactive narrative forms, an architecture should be more robust and flexible with a coherent model of narrative as opposed to only employing what Bringsjord and Ferrucci call "engineering tricks."

The knowledge engineering approach employed by these story generation systems is in stark contrast to the techniques used by the Oz project (described in **Subsection 2.1.6** below), which relied on behavioral emergence<sup>3</sup>. For example, Bringsjord and Ferrucci's believe that they can find logical representations for concepts that make stories interesting, and can use an encoding of these concepts to generate interesting stories. As a notable omission, they do not comment upon the contextual information they have encoded into their system by their choice of the

---

<sup>3</sup> The classic AI issue of top down vs. bottom up methodologies.

narrative terms they provide it to tell a story. BRUTUS tells stories that invoke the theme of betrayal; all of the sample stories were set within a university setting and (as seen above) revolve around a graduate student's being encouraged and supported by a Professor, only to have either that professor refuse to approve the thesis in the final committee meeting, or to have the student rebel against the professor and what the university represents at that same final meeting. The system's designers did not readily address the issue of social contextual cues they were placing within their story, and seem to imply that betrayal could be individually extracted as the only thematic element within the story, when clearly it is not. They seem not to have considered the relationship of the story to the audience's view of the author, and have scarcely addressed how the knowledge that the author of a story has been presented as being a machine can influence audience interpretation of a story. They seem to reinforce Italo Calvino's opinion that a computational system can play the combinatoric game of words and sentences, but cannot produce a revolutionary form without acknowledging the "ghosts" of social values. (Calvino, 1982b) Interestingly, the authors of BRUTUS have either subconsciously imbued their system with some of their social values, or they have failed in their task of creating an autonomous story-telling system, as BRUTUS does not generate a representation of social context explicitly at all. These story generation techniques represent some engineering tools that may be useful for the development of variably repeating narrative, but in the end do not seem to rely on the type of foundational theory for computational narrative design that allows many of the representational problems discussed in **Chapters 4 and 5** to be addressed properly.

In contrast to those computer science based *story* generation systems, William

Chamberlain and Thomas Etter's dialogue based program Ractor, and Ractor's (Chamberlain's) 1984 book, *The Policeman's Beard is Half Constructed*, used syntactic text manipulation to support conversation with users having text input and *poetic* output. (Racter, 1984) This was not intended as scientific research, but rather as entertainment, with humorous clever output. Chamberlain described such output as being computer authored, exploiting the novelty of being "written by a computer program." Many text generation projects have been oriented towards total automation and Turing test competence. The goal with the GRIOT system, presented later in this dissertation, was quite different: it is designed to provide a technical framework for humans to provide rich content and poetic systems created with GRIOT are meant as cultural products themselves (as opposed to merely seeking consideration of the instances of output of such poetic systems). Charles Hartman's 1996 work in automated poetry generation was presented as literary experimentation, but Hartman realized that it is better not to ask "whether a poet or a computer writes the poem, but what kinds of collaboration might be interesting." (Hartman, 1996) Hartman's work emphasizes how a computer can introduce "randomness, arbitrariness, and contingency" into poetry composition. This is another significant difference from the GRIOT system, which uses structured principles of meaning and narrative to guide poetry generation as opposed to random template selection.

A final influential work in richer media and developed as a joint enterprise of documentary filmmaking, computational media art, and artificial intelligence is *Terminal Time* developed by Steffi Domike, Michael Mateas, and Paul Vanouse at Carnegie Mellon University. (Domike, Mateas, & Vanouse, 2003) In *Terminal Time*

an audience is polled in real-time with questions about the state of society today and on the fly a “Ken Burns style” documentary of the last two millennia of history is created to attempt to mirror and often exaggerate their biases and desires.” Its basis in what Michael Mateas terms “expressive AI” allows *Terminal Time* to utilize formal representations subjective meaning. Like *Rashomon*, *Terminal Time* invokes “meaningful difference” as a central strategy for artistic expression and is an important example of computational art because computational (artificial intelligence) concerns are intertwined with expressive content: recombining content structure and shifting rhetorical differences along with the technology used to produce them are central to the work. GRIOT has similar concerns as *Terminal Time* in this respect, however a novel contribution of the GRIOT system to this type of work is its theoretical underpinnings and contributions to research in the frameworks of algebraic semiotics and cognitive linguistics.

### **2.1.6 Interactive Drama and Virtual Reality**

The Oz project, which was headed by Joseph Bates at Carnegie Mellon University, is a strong example of interactive drama research. (Bates, 1992) Bates began with the observation that most previous virtual reality research focused on interface issues and how to present a simulated world in a convincing fashion. The Oz project instead focused on computational techniques for and theories for cognitive/emotional agents, narrative guidance, cinematic presentation style, and dramatic agency. If the interface and display technology issues are considered to be

the surface structures in interactive storytelling then Bates considers organization and content of narratives to be the “deep structure.”<sup>4</sup>

The work of the Oz project had three main trajectories as stated above. These are:

- (1) Design of cognitive/emotional agents
- (2) Techniques for presentation style (cinematically inspired)
- (3) Integration of dramatic constraints

In the design of cognitive/emotional agents the Oz researchers tried to create agents that took advantage of the observation that “people see subtlety, understanding, and emotion in an agent so long as the agent does not actively destroy the illusion.” (Bates, 1992) Toward this end, initially they developed an architecture called Tok (succeeded by Hap and extended by Michael Mateas and Andrew Stern to ABL) that modeled internal goals, reactivity, emotion, natural language ability, and social knowledge of agents in a simulated world. (Mateas, 1999) Though extensions of the research would focus on drama management and multi-agent coordination, earlier they focused on believable character and behavioral animation, a technique inspired by behavior based AI research. Michael Mateas characterizes their approach as focusing on the development of broad/shallow, embodied, distributed state, reactive agents. (Mateas, 1999) This type of modeling is accomplished using transition functions, emotional states, and associated behaviors for each agent. Agents try to satisfy local goals. For example, if agent-1 is hungry and agent-2 thwarts agent-1 from eating, the fact that

---

<sup>4</sup> The use of the phrase “deep structure” here is not related to Noam Chomsky’s use of the phrase in formal linguistics.

one agent-1 has failed in accomplishing a goal due to another agent could result in the state “angry.” If the agent had been thwarted by the environment, rather than another agent, then perhaps its emotional state would instead have become “frustrated.”

Agents exhibit behavior in alignment with their emotional states via body language, facial expression, and movement. The goal of such modeling is to imbue characters with coherent and believable personalities. For Oz type interactive drama emotional believability is the most important trait an artificial agent can possess.

For presentation style, the Oz researchers drew upon film theory more than computer science theory. They believed in importing expertise from other disciplines rather than attempting to reinvent it from scratch. Toward this end, their account of presentation includes discussion of cinematic techniques such as: lap dissolve, pan shots, flashbacks, symbolic camera movement and positioning, visual rhythm, zoom freeze, flash frames, and others. The central observation here is that computational narratives can also import and account for, and most importantly, *implement* conventions from previous media.

Oz researchers introduce dramatic constraints by often visualizing interactive drama as an adversarial game. The simulated world is considered to be an environment within which the dramatic engine tries to present the user with a coherent story while the user tries to act freely to explore the world. The dramatic engine then searches for appropriate events to maintain the dramatic structure. Peter Weyhrauch claimed (rather reductionistically) that an author’s aesthetic sensibility could be encoded by an evaluation function. (Weyhrauch, 1997) He further claimed that an adversarial search algorithm could take advantage of this to guide the user’s

experience of interactive drama systems. Toward this end he developed the Moe Architecture for dramatic guidance. Moe has various “moves” that it can make on a given time step such as introducing new characters, giving characters strong emotion, or eliminating characters. These moves are selected against an evaluation function with arbitrary criteria such as “emotional intensity.” Because Moe does not model the player as an optimal opponent (because the player’s goal is not to win but to have an evocative dramatic experience) it uses a variation of a min-max algorithm that Weyrauch calls “avg-max.” In anticipating the user’s next move it uses the average expected value of the next nodes rather than the worst possible user move (from Moe’s perspective).

Using such evaluation functions should be seen as techniques for narrative guidance, rather than as encodings of aesthetic values. The adversarial search approach also seems inadequate to model a user’s engagement with an interactive narrative system. Still, search algorithms have a place in design of computational narratives where appropriate media events or objects need to be selected in order to maintain a coherent narrative structure. The difference in the cognitive linguistics based approach of this dissertation is that narrative concepts are explicitly modeled (perhaps in conjunction with state functions) according to theories of how humans construct and represent narrative

The most recent extension of this work is the *Façade* system created by Michael Mateas and Andrew Stern. (Mateas & Stern, 2004) The system is a first-person interactive one-act play in which a user takes the role of a dinner guest of an unstable, bourgeois couple. The user can interact with the characters via a natural



language interface, and can also interact with object in the world through both iconic (such as kissing a character by clicking the character when the mouse cursor reveals that option) and “physical” interaction). *Façade* consolidates and advances many of the concerns of the Oz project, but also features significant extensions. Its various components include: an integrated drama architecture, a reactive planning language for believable agents, a (“broad and shallow”) natural language processing system, a drama manager, and a modular system of story design. (Mateas, 2002) These components combine to exemplify an approach that Mateas calls “expressive AI,” which is ambivalent about the division between top-down logic and knowledge based approaches and bottom up, or connectionist approaches to AI, opportunistically taking from either with the goal of generating expressive content. Despite this ambivalence, Mateas also claims the approach as a “critical technical practice ” in the sense defined by Phil Agre, in which there is active reflection on the philosophical underpinnings of the work and carries the possibility of inventing new technical directions through the integration of culturally situated concerns. (Agre, 1997; Mateas, 2002)

Related work can be found in procedural and behavioral animation systems *Improv*, developed by Ken Perlin and Athomas Goldberg at New York University, and *Alive*, developed Bruce Blumberg others at the MIT Media Lab. (Blumberg & Galyean, 1995; Perlin & Goldberg, 1996) These systems also define algorithms that give the illusion of intention to animated characters. The authors of these systems worked backwards from encoding representations of intentionality (at the level of behavior in Blumberg’s case, at the level of movement in Perlin and Goldberg’s case) and tried to generate the effects of these intentions in characters modeled in computer

graphics. This approach need not only apply to visual images, but can be applied to narratives as well. Blumberg uses models of animal behavior taken from ethology to determine agent action. Perlin and Goldberg both used blends of behaviorally annotated animation cycles to construct new animation cycles. For example, a character in an angry state performing a walk cycle will perform an “angry walk.” The goal of the *Improv* system is not to be a comprehensive interactive drama tool, but rather to be an authoring tool for animators to create animations at a higher level of abstraction. Using the *Improv* system, animators can specify procedural movements, combine them, and then refine the blended movements by hand and keyframing. There is no general theory for how animation procedures should be annotated or blended, though perhaps algebraic semiotics could provide insight into this problem.

Ken Perlin is a strong believer in the power of the “texture of experience.”<sup>5</sup> This means that the details and meaningful structure of experiences do not need to be modeled explicitly, but can be evoked by appropriate functions. In this model, a narrative does not need to be preconceived, but needs only to convey a general feeling of “narrativity” to a user. Such an approach could be valuable for an online environment such as massively multi-player online role-playing games (MMORPGs) in which users explore and interact with other users, virtual objects, and occasionally artificial intelligence characters. The task of coordinating one large-scale narrative among multiple users is a great challenge, and if an emergent feeling of coherent story could be made to evolve algorithmically it would be a strong model for computational narrative.

---

<sup>5</sup> This was communicated via personal conversation.

The problem with this approach is that it does not account for the guidance of any specific type of meaning, if such is desired. The author in this case is deciding that the message she or he is conveying to the user is primarily regarding the types of interaction that can occur and the space that can be explored. The specific content that can be conveyed is more akin to the type that can be conveyed by a physical space, or by a telecommunications device, which is quite different than the types of meaning conveyed by film or literature for example. To convey specific meaning, the model that the audience has of the narrative world in relationship to the real world must be addressed. The authorial model must seem to wish to convey some sort of complex narrative content; otherwise the audience will not expect it. The author in this case gives up control over specific narrative content and leaves it to the algorithm. The specific content must then be encoded into the algorithmic representation, or overdetermination of linguistic signs (the natural human inclination to construction of meanings from even quite minimal signals) will cause the audience to generate some type of unexpected meaning that may not be desirable. This is not a negative in itself, but if such an experience is what the author is providing the audience, she or he should be aware of it and should not expect that a novelistic type of experience can “emerge.”

It is important to keep in mind that the audience could easily interpret the authorial intention to be “providing an exciting or fun or interesting experience” and overlook other narrative intentions because it is conventional to read computer games in this manner, and computer gaming is not popularly associated with transformative power in “real life.” Even if the author desires to provide an experience that references primarily the activity of play and winning a game, her or his creation can be

informed by examining the narrative relationship of the audience to the representation, in order to design the experience and the feeling of play, or in order to control conventional and potentially undermining meanings to interfere with the experience of play.

### **2.1.7 Games and Dynamic Systems**

There is a subset of computer games that comprise a quite popular cultural form of computational narrative. There are striking differences, however, between computer games and more general computational narrative models. One major aspect of these differences can be explored through the following definition of a game by Elliott Avedon and Brian Sutton-Smith: “[a game is] an exercise of voluntary control systems, in which there is a contest between powers, confined by rules, in order to produce a disequilibrium outcome.” (Avedon & Sutton-Smith, 1979) For the purposes of this paper this definition of a game should be contrasted with the following definition of play. Johan Huizinga defines play as (Huizinga, 1950):

...free activity standing quite consciously outside “ordinary” life as being “not serious,” but at the same time absorbing the player intensely and utterly. It is an activity connected with no material interest, and no profit can be gained by it. It proceeds within its own proper boundaries of time and space according to its own fixed rules and in an orderly manner. It promotes the formation of social groupings which tend to surround themselves with secrecy and to stress their difference from the common world by disguise or other means.

The definitions given above are not meant here to account for every instance of interaction that could be considered a game or play. Katie Salen and Eric Zimmerman present a summary of other definitions in (Salen & Zimmerman, 2004). Indeed, the

definitions have generated a fair amount of controversy. For example, many activities recognized as games include elements of cooperation or fantastic role-play as opposed to conflict and competition. While one could argue that labeling such activities as “play” versus “games” is merely a matter of distinguishing categories, another view is that as computer games become increasingly popular and powerful (economically and socially) it is crucial to consider who defines what is considered to be a game. Thus, the definitions above are merely meant to give a suggestion for the types of activities that are may considered games as distinguished from other activities since boundary objects will exist for any category of game that is established by a definition (see (Bowker & Star, 1999) for a discussion of “boundary objects” in the sociological theory of classification).

Salen and Zimmerman identify three levels at which games can be examined, these are: (1) formal, (2) social, and (3) cultural. The formal level accounts for the system of rules that the game employs, the social level accounts for social interaction occurring during the game that is not restricted by the formal rules, and the cultural level consists of the role that the game plays in culture at large. They also consider a hallmark of good game design to be the implementation of a strict system of “desire elicitation and fulfillment.” This means that within a game there is a series of constraints designed to give the user a feeling of freedom. This is an observation that is absent from accounts of interactive narrative where interaction is pitted against free will of the user. A good design will constrain the user to make choices that give the illusion of free exploration. Miyamoto Shigeru’s games are good examples of this principle. Characters such as Mario and Link in his *Super Mario* and *Legend of Zelda*

series of games often just offer the player tiny hints of what is possible for a character to accomplish. (Nintendo, 1987, 1996) As another example, in the game *Super Metroid* there may be a thin sliver of a platform at the top of the screen that appear only to frame the space. (Nintendo, 1993) An inquisitive player may actually try to jump onto that platform (testing the limits of the system and free range of motion within the screen). Such a player is rewarded to find that she can walk atop of that platform that formerly seemed inaccessible. In this matter the player feels that she or he has escaped the boundaries of what *should* be possible in the game. The player's desires have been teased and fulfilled within a strict system of constraints. These constraints need not always be strict however. Other game design strategies have found ways to stretch the boundaries of these constraints.

Computer games and multimedia works based upon dynamic systems use a style of programming called event-oriented-programming. A good example of this is *The Eye of Wodon*, developed by Peter Bøgh Anderson. (Anderson, 1997) It is a multimedia exhibit about Vikings that was developed for a museum in Denmark. Anderson asserts that the computer is an elastic medium in that the nature of what it presents is determined by physical actions performed by users. The narrative trajectory in a system of this type can be said to arise from a set of elastic constraints that grow more rigid as a user attempts to avoid them. If the user resists enough then these constraints can become rigid. In the absence of an overall specified narrative structure these types of constraints offer some narrative guidance.

*The Eye of Wodon* functions by means of a phase-space and event triggers.

Phase space can be described by the following equation:

$$v(t+1) = v(t) + \text{delta } a + \text{delta } u$$

where  $v$  is the vector in phase space,  $t$  is the time,  $a$  is the increment provided the author, and  $u$  is the increment provided by the user, noting that both these increments are also computed at time  $t$ , and that their values depend on the current state of the system. The phase space represents the state of the world at a given time step and the transition function to the next time step based on the actions a user takes. Events are preplanned narrative sequences that occur when certain states are entered. The phase-space can be structured to guide the users actions by introducing features such as “basins of attraction” where some states are more weighted more heavily than others as time passes. A basin of attraction constrains the user’s choice of action and draws the user into the center of the basin and thus a particular state (usually to trigger a particular event). After the user reaches the bottom of the basin she or he can be reset at a state specified by the system. The Will Wright designed computer game *The Sims* uses a similar model. (Maxis, 2000)

In both *The Sims* and *The Eye of Wodon*, the narratives created are largely emergent. This type of emergence is more specific than the general phenomenon of emergent narrative structure described earlier in this section, in the context of dynamic systems emergence means that the user’s experience is shaped primarily by execution of state transition functions in response to user interaction. Emergent narratives can result in unexpected consequences to user decisions made in the course of the experience. They can also have some inconsistency of the quality and coherence of the narrative. To some degree, event triggers help to mollify this issue. More

promising for computational narrative design are systems that merge flexible representation of narrative structure with dynamic logics for user interaction.

## **2.2 Cognitive Linguistics**

### **2.2.1 Overview of the Cognitive Linguistics Enterprise**

The theories of conceptual blending and metaphor both are concerned with issues of cognitive semantics within the field of cognitive science and are central components of the approach to computational narrative presented in this dissertation. These theories are important constituents of the more general enterprise of cognitive linguistics. This section describes the fundamental and unifying features of the enterprise, with a focus on issues related specifically to addressing conceptual blending using a methods involving computational techniques. The cognitive linguistics enterprise is dedicated to investigating “the relationship between human language, the mind and socio-physical experience.” (Evans et al., 2006) In very general terms, the cognitive linguistics enterprise centralizes the fact that language is in service of “constructing and communicating meaning” and it provides a window to our cognitive processes. (Fauconnier, 2000) It is characterized by a rejection of earlier formal approaches to linguistics (exemplified by Noam Chomsky’s Generative Grammar) and by a collection of shared philosophical commitments.

Two key philosophical commitments of cognitive linguistics have been described as “The Generalization Commitment” and “The Cognitive Commitment.”



(Evans et al., 2006; Lakoff, 1990) In turn, I shall provide definitions and discussion of the underpinnings of each commitment in the discussion below. The Generalization Commitment refers to a commitment to understanding and describing general principles of human processes of meaning making using language as a means of gaining insight to those processes. This commitment has lead cognitive linguistics researchers to focus upon commonalities among language phenomena and the application of successful methods of investigation to language phenomena at as general a level as is possible. This commitment to generalization impacts research even at the neurobiological level. Cognitive linguists interested in the neural basis for cognition are oriented toward examining the “reuse of existing structures for new purposes, both on evolutionary and developmental timescales.” (Evans et al., 2006) This is not accidental, it reflects our knowledge that many biological structures exhibit this characteristic.

In support of the generalization commitment, empirical research in cognitive linguistics suggests that language activity is only the observable result of processes in which humans draw upon “a vast array of cognitive resources” involving “innumerable models and frames, set up multiple connections, coordinate large arrays of information, engage in creative mappings, transfers, and elaborations.” (Fauconnier, 2000) Gilles Fauconnier has referred to these process of meaning construction as “Backstage Cognition,” and asserts that “backstage cognition includes specific phenomena including:

...viewpoints and reference points, figure-ground/profile-bases/landmark-trajectory organization, metaphorical, analogical, and other mappings, idealized models, framing, construal, mental spaces,

counterpart connections, roles, prototypes, metonymy, polysemy, conceptual blending, fictive motion, [and] force dynamics. (Fauconnier, 2000)

The assertion that many aspects of backstage cognition are based upon shared cognitive structures or operate on the basis of general principles is referred to as “operational uniformity.” (Fauconnier, 2000) On this basis, the Generalization Commitment can be seen as a philosophical and methodological response to empirical evidence suggesting operational uniformity for such a wide range of cognitive phenomena.

The Generalization Commitment contrasts strongly with the commitment of older linguistics models of investigating language as a set of distinct areas such as syntax, semantics, and discourse. The older modular approach to the study of language is a characteristic of formalist linguistics, and carries with it an assumption that each distinct module of language production and use is likely enabled by distinct neural modules for language facility within the brain. The empirical results cited by cognitive linguists that focus on the operational uniformity of a great deal of backstage cognition refute this.

The goal of accounting for the diverse, astonishingly complex, and often unnoticed, processes of backstage cognition underlying meaning construction also reflects the “Cognitive Commitment.” It is a commitment to finding principles underlying meaning construction in accord with known and accepted results about the mind/brain complex from a range of disciplines. (Evans et al., 2006) This poses a challenge for cognitive linguistics researchers, since new results must “present convergent evidence for the cognitive reality of any proffered model of explanation –

whether or not this research is conducted by the cognitive linguist.” (Evans et al., 2006) Specifically, models of meaning construction and language should reflect results from the component disciplines of cognitive science such as psychology, neuroscience, philosophy, and artificial intelligence. Along these lines, cognitive linguistics researchers often emphasize relatively recent findings in cognitive science, which assert that (1) “the same neural and cognitive mechanisms that allow us to perceive and move around also create our conceptual systems and modes of reason” (Lakoff & Johnson, 1999), (2) cognitive processes may be distributed across members of a social group, external artifacts, and even time (Hutchins, 1996, 2000), and (3) cognition is always situated in particular social and cultural contexts (Lave & Wenger, 1991). In short, such theories consider cognition to be embodied, distributed, and situated. Such results of cognitive linguistics research have profound implications for the study of meaning and language phenomena, not the least of which is its critical challenge to previous theories. This is because the commitment to building models only upon accepted empirical results about the mind/brain complex, as opposed to “purely aesthetic dictates such as the use of particular kinds of formalisms of economy of representation,” (Evans et al., 2006) leads to a rejection of logic based theories in analytic philosophy, formal models of “good old fashioned artificial intelligence” (hereafter ‘GOFAI’), and extremely abstracted and idealized models in connectionist artificial intelligence. Indeed, the cognitive linguistics enterprise has been posed as a radical challenge to central tenets of Western philosophy including Cartesian dualism, Kantian radical autonomy, Fregean analytic philosophy, or Chomskian formal linguistics. (Lakoff & Johnson, 1999)

Of particular interest here are challenges to analytic philosophy and Chomsky's formal approach to linguistics. These challenges are especially relevant since this dissertation proposes computational contributions that are compatible with the commitments of cognitive linguistics. These contributions then must be understood as distinct from other incompatible approaches involving formalization such as classical GOFAI approaches in computer science, which typically have either an explicit or implicit basis in analytic philosophy where meaning exists in the world at large and the role of language (in particular logic as the language of mathematics) is to provide symbolic correspondences to external objects. Chomsky's generative grammar has been a valuable and important contribution for research into programming languages, however it is in conflict with empirical results from cognitive linguistics. George Lakoff broadly states the central conflicts with the Chomskian picture by asserting it is not the case that for humans:

...language is pure syntax, pure form insulated from and independent of all meaning, context, perception, emotion, memory, attention, action, and the dynamic nature of communication. Moreover, human language is not a totally genetic innovation. Rather, central aspects of language arise from sensory, motor, and other neural systems that are present in "lower" animals. (Lakoff & Johnson, 1999)

These challenges are important because, since I agree with the tenets and commitments of cognitive linguistics, they force the question of how computation can be used in order to describe human meaning making within a world view that rejects a view of language and meaning as being formalizable. Brief discussion of this issue follows below at the conclusion of **Subsection 2.2.2** and the task will be taken up in **Section 2.3**.

### 2.2.2 Remarks on Cognitive Linguistics Methodology

The cognitive linguistic enterprise represents a paradigm shift. We have seen that influential earlier approaches to linguistics focus upon language phenomena at various levels which, while obviously integrated in human communication, have typically been described using methods and terminologies that are largely incommensurable. Cognitive linguistics researchers consider linguistic distributions (language phenomena across various levels of specificity) as only examples of observable manifestations of processes of backstage cognition with striking operational uniformity. (Fauconnier, 2000) From the perspective of cognitive linguists, this constitutes a paradigm shift because of three revolutionary transformations within cognitive science that occurred after empirical evidence caused a crisis for former traditions. (1) The goals of the enterprise have necessarily changed. Language is now seen as one means of revealing general aspects of human cognition, as opposed to a self-contained area of study (Fauconnier, 2000). (2) The world view of the enterprise has dramatically changed. Different levels of language phenomena are now seen with a “vertical” orientation grounded in operationally uniform cognitive processes, as opposed to “horizontally” oriented modules each with separate underlying neural modules (Evans et al., 2006). (3) Earlier methods have been rejected as having precluded the development of the cognitive linguistics approach. Data such as grammaticality or acceptability judgments of native speakers cannot reveal the operational uniformities across phenomena and media discovered by

cognitive linguists and furthermore “form and meaning in language can no longer be seen as separate from everything else outside of language” (Fauconnier, 2000).

The evidence underlying this paradigm shift has converged from a variety of results within the component disciplines of cognitive science, Fauconnier cites a series of these convergences in (Fauconnier, 2000):

Backstage cognition operates in many ways uniformly at all levels. Figure-ground and viewpoint organization pervades the sentence (Langacker, 1987/1991; Talmy, 1978), the Tense system (Cutrer, 1994), Narrative structure (Saunders & Redeker, 1996), in signed and spoken languages, and of course many aspects of non-linguistic cognition. Metaphor builds up meaning all the way from the most basic levels to the most sophisticated and creative ones (Grady, Oakley, & Coulson, 1999; Lakoff & Turner, 1989). And the same goes from metonymic pragmatic functions (Nunberg, 1978) and mental space connections (Fauconnier & Sweetser, 1996; Liddell, 1996; Van Hoek, 1997), which are governed by the same general Access principle. Frames, schemas and prototypes account for word level and sentence level syntactic/semantic properties in cognitive and construction grammar (Fillmore, 1985; Goldberg, 1994; Lakoff, 1987; Langacker, 1987/1991), and of course they guide thought and action more generally (Bateson, 1972; Goffman, 1974). Conceptual blending and analogy play a key role in syntax and morphology (Mandelbilt, 1997), in word and sentence level semantics, and at higher levels of reasoning and rhetoric (Coulson, 1997; Robert, 1998; Turner, 1996). Similarly, we find force dynamics and fictive motion (Talmy, 1985, 1998) operating at all levels (single word, entire systems, like the modals, and general framing).

In addition to producing and interpreting such examples of convergent data in light of the commitments of cognitive linguistics, cognitive linguists discover and explicate examples which seem anomalous within existing frameworks, and develop more general frameworks in order to account for the new evidence. For example, conceptual blending theory was invented by Gilles Fauconnier and Mark Turner first as an attempt to accommodate anomalies that could not be resolved using existing

models within metaphor theory. (Fauconnier, 2000) The original results in metaphor theory arose from examination of regularities (both syntactic and conceptual) of mappings and motor-sensory based metaphor within of large corpi of linguistic data, however the results leading to the generalization of the models in metaphor theory to those of conceptual blending theory were realized in a different manner. Seana Coulson and Todd Oakley have described the methodology of blending theory as “mining for golden events,” meaning that, holding as a desideratum “the treatment of disparate cases in a principled way,” blending theory proceeds by detailed analyses of individual cases in their particularities in order to discover and describe the systematicity of new generalizations. (Coulson & Oakley, 2000) At the same time, conceptual blending theory also appeals to linguistic corpi, for example finding widespread examples of syntactic regularity, in order to support observations found in anomalous cases. It is important to note, however, that syntactic analysis is insufficient a primary method since a central tenet within conceptual blending theory is the notion of “economy of language” and the fact that syntax alone is insufficient to account for the immense array of mental activations and elaborations that are triggered by even minimal linguistic information. (Fauconnier, 2000) Furthermore, since operational uniformity is present across a variety of meaning making phenomena, the specific cases investigated by conceptual blending theorists may include images, computer interfaces, utterances, and a range of phenomena that are not necessarily linguistic. While not all research under the banner of cognitive linguistics proceeds in this manner, a general description of some of the methods of conceptual blending theory is important here because it captures aspects of the Generalization and

Cognitive Commitments and because conceptual blending theory is an influence upon the computational work presented in this dissertation.

There have, of course, been criticisms of methodologies and results within the field of cognitive linguistics. At a general level one concern is the fact that, despite increasing evidence from a variety of disciplines, much of the data is linguistic and insight to cognitive processes at the level of the mind/brain complex is largely inferential. (Evans et al., 2006) Conceptual blending theory faces its own challenges because of its attempt to powerfully account for a wide range of phenomena and the prevalent analyses anomalous cases (though many commonplace cases have been analyzed as well) as a means for generalization. Some of these challenges and responses to them will be mentioned later in the **Subsection 2.2.5**.

### **2.2.3 Key Points in Metaphor Theory**

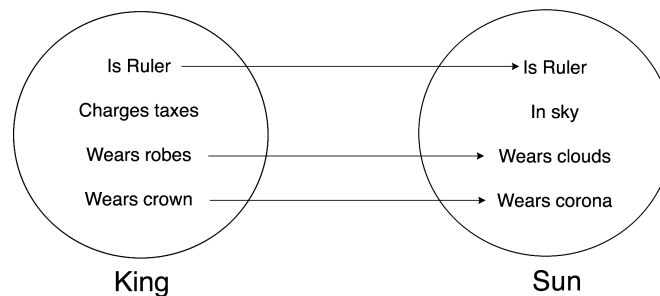
Metaphor theory, initiated by the work of George Lakoff and Mark Johnson (Lakoff & Johnson, 1980), postulates metaphor as a fundamental mechanism of human conceptual thought. Key characteristics of the theory include its focus on the cognitive, as opposed to purely linguistic, nature of metaphor (Grady et al., 1999), the embodied basis of metaphor (Johnson & Labov, 2002; Lakoff & Johnson, 1999), the systematicity of mappings between mental spaces in metaphor (Lakoff, 1987; Lakoff & Johnson, 1980), and the ubiquity of metaphor in human conceptual processes at multiple grains sizes (the most basic thoughts to elaborate literary works) (Lakoff & Johnson, 1980).



George Lakoff, Mark Johnson, and others shown that metaphors come in families, called “image schemata,” having a common basis in pre-linguistic motor-sensory experience. (Lakoff, 1987; Lakoff & Johnson, 1980) Image schemata are “skeletal patterns” that recur in our motor-sensory experiences such as **Motion Along a Path**, or **Container** as expressed respectively by entrenched basic metaphors such as **Life is a Journey**, or **Emotional States are Containers**. Grounded in our embodied experiences of image schemas, and building upon Gilles Fauconnier’s theory of mental spaces (first-order, mostly binary, relational structures) (Fauconnier, 1985), metaphor is modeled as selective projection from one mental space called a “source space” to another called a “target space.” “Links” between the analogous elements structure the projection. Lakoff and Johnson have shown that there are many basic, entrenched metaphors that people use to express everyday concepts. These basic metaphors in turn can be projected onto other concepts in particular contexts, for example **More is Up** can project onto the concept of income “His salary is higher than mine” or the economy “The national deficit rises quite quickly.” The source **Up** is grounded in our experience of gravity, and the schema itself is grounded in everyday experiences, such as that when there is more milk in a glass, or more peanuts in a pile, the level goes up. Some image schemas, including this one, are grounded in the human body, and are called “basic image schemas;” they tend to yield the most persuasive metaphors, and can be useful at a variety of levels, for example, in user interface design. (Goguen, 1998) Similarly, the source **Container** is grounded in our experience of objects or environments containing other objects, and **Emotional States are Containers** can be projected onto concepts as depression, e.g., “She was in a deep

depression,” or love, e.g., “He is in love.”

We have seen that metaphor is not only a higher level “literary” mechanism (though it is undoubtedly that is well). In everyday thought and complex cultural works (artistic, mathematical, scientific or otherwise), metaphor is typically used in order to express something that is not well understood (the target space) in terms of something more well understood (the source space), and can be found in phenomena as diverse as simple propositional statements, simile, analogy, metonymy, or metaphysical conceit. For example, in the sentence “The sun is a king,” we wish to understand the target space of “sun,” which is a remote entity, via projection of elements from the source space of “king,” which is a well known human being (see **Figure 2.1**).



**Figure 2.1: The “Sun is a King” metaphor**

One of the ways that metaphors serve to explicate target spaces is by allowing for systematic analogical insights via projection. For example, a common metaphor is: “Theories are constructed objects,” (Lakoff, 1994) which features the following projections.

- Major premises are foundations
- Major claims and arguments are its structure

- Facts are material constituents
- Arguments are mortar between facts and claims
- Logical strength is design or architecture
- A theorist is an architect
- Believability is strength
- Persistence is successful standing
- Failure is collapse

Piecewise then, the components of theories are seen as having characteristics of physical objects that are well know, and inferences that hold true regarding constructed objects can, by projection, be seen as applying to theories as well.

#### **2.2.4 Key Points in Conceptual Blending Theory**

Gilles Fauconnier and Mark Turner propose conceptual blending as a fundamental cognitive process underlying human creativity. Their theory describes the means by which concepts are integrated and elaborated, guided by systematic principles, both unconsciously in everyday thought and in more complex abstract thought, such as in literary arts or rhetoric. It is proposed to underlie meaning construction in realms as diverse as grammar and reasoning, invisible and effortless, but pervasive and fundamental. (Fauconnier & Turner, 2002) Key characteristics of conceptual blending theory include a model that builds upon mental space theory and generalizes the unidirectional two-space mappings in metaphor theory, an account of processes of conceptual integration underlying multiple grains of meaning construction across media, and an account of the systematic structural principles underlying blending including the emergence of new structure and content beyond the elements of the concepts being integrated, and an account of principles guiding the

creation of optimal blends. Each of these characteristics of conceptual blending will be described in the discussion below.

Blending is common not only in natural language, for example, in basic metaphors such as “she was deeply in love,” compound words like “houseboat” and “starfish,” phrases like “artificial life” and “computer virus,” counterfactuals such as “if I were the king I would mandate a daily siesta,” but also in other media such as images. **Figure 2.2** is a political cartoon by Benjamin Franklin exhorting the American colonies toward solidarity in the “French and Indian War” by blending the idea of divided colonies with the image of a severed snake. (Franklin, 1754)



**Figure 2.2: Blending in an 18<sup>th</sup> century political cartoon**

In this blend, the superstitious belief that if a severed snake is rejoined before sunset it will come back to life (or else die) is blended with the concept that the American colonies needed unity in order to defend themselves. This blend has expressive and explanatory power for humans in part because the concept of the abstract union of the colonies is compressed to the more human scale image of a snake. Such **compression** of abstract relationships toward human scale human scale is a common feature of conceptual blends.

The basic units of integration within the theory are **conceptual spaces**.

Conceptual spaces, building (like the source and target spaces in metaphor theory) upon Fauconnier's theory of mental spaces (Fauconnier, 1985), are sets of "elements" and relations, "relatively small, transient collections of concepts, selected from larger domains for some purpose at hand, such as understanding a particular sentence." (Fauconnier & Turner, 2002) Although concepts are often viewed as packets of meaning, really they arise from connections across multiple meanings that are dynamic and distributed. The structure of these dynamic connections is described using **conceptual integration networks**, which generalize the two-space projections of metaphor theory to multi-space (prototypically four-space) networks.

Fauconnier and Turner assert that the process of blending is structured by a set of "constitutive" principles, and constrained by "governing" principles that exert pressure to produce optimal blends. These two sets of principles will be described respectively in the following discussion. The constitutive principles represent the structural systematicity of blending and are captured by the structure of multi space conceptual integration networks and the process of blending described below.

The basic elements of a conceptual integration network are (Fauconnier & Turner, 2002):

- (1) **Input Spaces** (the conceptual spaces to be combined)
- (2) **Cross-Space Mappings** (links indicating counterpart connections between analogous elements in different input spaces)
- (3) The **Generic Space** (a conceptual space mapped to both of the input spaces that describes shared structure between the input spaces)
- (4) The **Blended Space** (the space in which elements from the input spaces are integrated)

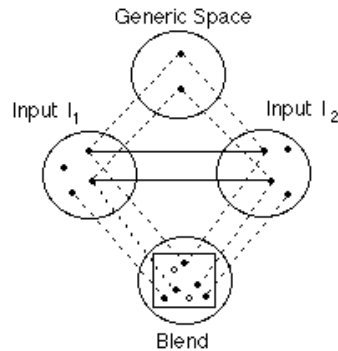
Instead of a source and target space with elements projected between them, in blending theory input spaces are **selectively projected** to a new third category of space called the “blend space.” When two input spaces are blended there is a fourth category of space called the “generic space” that is also involved. This generic space indicates the counterpart structure, more general relationships and elements found in both of the input spaces, shared between the two input spaces. Mark Turner uses an example taken from a 1993 issue of the sailing magazine “Latitude 38” to explain the generic space (Turner, 1996):

As we went to press, Rich Wilson and Bill Biewenga were barely maintaining a 4.5 day lead of the ghost of the clipper Northern Light, whose record run from San Francisco to Boston they’re trying to beat. In 1853, the clipper made the passage in 76 days, 8 hours.

In this blend two boats from two different times are thought of as making the journey at the same time in a race. The generic space contains counterpart structure such as the journey, the boat, and the duration of time of the journey. Furthermore, this blend is structured by a frame that allows for the recruitment of additional structure. For example, the concept of race exists in neither input space. It is an emergent concept from the blend the results from the interaction of the input spaces with frame<sup>6</sup> representing the precompiled notion of what a race is.

---

<sup>6</sup> Conceptual spaces are distinct from, but related to, other grains of conceptual units, for example a frame is a densely interconnected system of concepts such as “Family” with components such as father, mother, son, daughter, etc., or “Chair” with components such as legs, seat, back, etc.. A domain is a larger collection of more loosely connected concepts, e.g., law or education. (Evans et al., 2006)



**Figure 2.3: A diagram of a conceptual integration network (Turner, 2006)**

The structure of a specific conceptual space network is well illustrated with the following example of a quite creative blend by Fauconnier and Turner called the *Buddhist Monk* riddle, which reads as follows:

A Buddhist Monk begins at dawn one day walking up a mountain, reaches the top at sunset, meditates at the top for several days until one dawn when he begins to walk back to the foot of the mountain, which he reaches at sunset. Make no assumptions about his starting or stopping or about his pace during the trips. Riddle: Is there a place on the path that the monk occupies at the same hour of the day on the two separate journeys? (Fauconnier & Turner, 2002)

Following Fauconnier and Turner, I shall exhort the reader unfamiliar with this riddle to close this dissertation for a moment and try to solve the riddle without any hints in order to increase its effect.

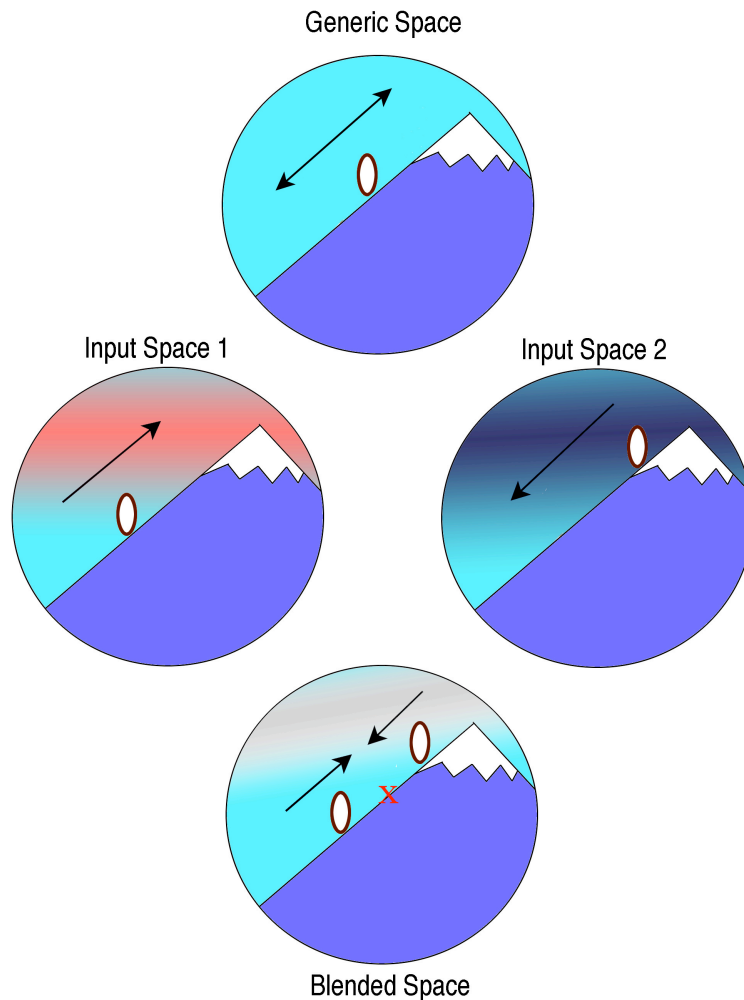
\*\*\*

Returning to the matter at hand, their solution reads:

Rather than envisioning the Buddhist Monk strolling up one day and down several days later, imagine that he is taking both walks on the same day. There must be a place where he meets himself, and that place is the one we are looking for. We don't know exactly where this

place is, but we do know that, whatever its location, the monk must be there at the same time of day on his two separate journeys. For many people this is a compelling solution to the riddle.

The conceptual spaces in the conceptual space network prompted by the solution to this riddle contain the following elements (depicted in **Figure 2.4**):



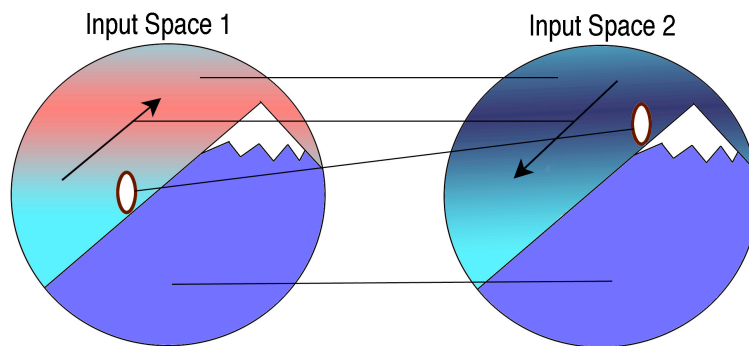
**Figure 2.4: The Conceptual Spaces in the Buddhist Monk Riddle**

- (1) The generic space illustrates the shared structure between the input spaces and thus contains a mountain, an undirected path (depicted in the figure by the line with arrowheads on both ends), a time of day (depicted in the figure by the sky), and a moving Buddhist monk.



- (2) Input space 1 contains the elements of the generic space except that the path is unidirectional (uphill), and the monk is in a specific location at a specific time of day.
- (3) Input space 2 contains the elements of the generic space except that the path is unidirectional (downhill), and the monk is in a (possibly different) specific location at a specific time of day.
- (4) The blended space contains the elements of both input spaces except there is a (possibly different) specific time of day, only a single path for both monks, and most notably there is the additional element of a meeting point between the two monks.

The cross-space mappings between the input spaces illustrates the counterpart elements and relations in the input spaces. The cross-space mappings are indicated by horizontal lines in **Figure 2.5**.



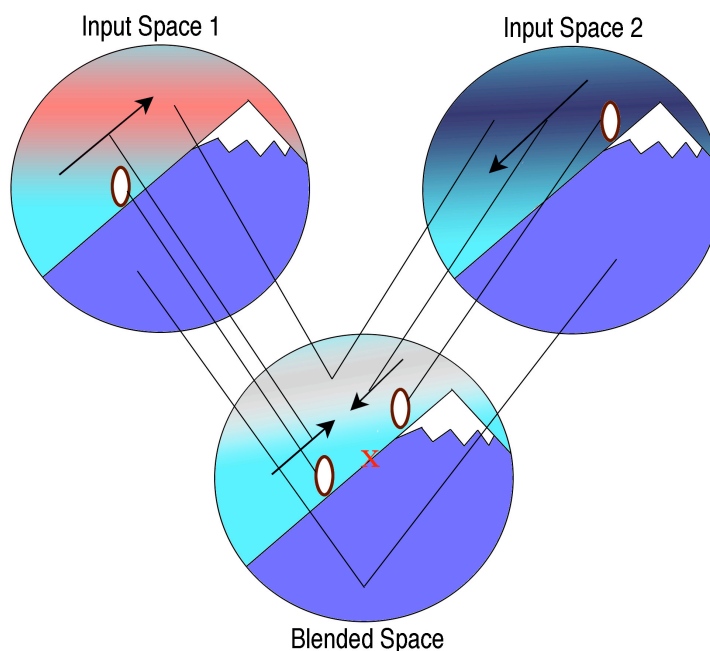
**Figure 2.5: Cross-Space Mappings in the Buddhist Monk Riddle**

The elements of the input spaces are not projected to the blended space uniformly.

The selective projection in the Buddhist Monk Riddle is illustrated in **Figure 2.5**.

Note that while both directional paths and monks are projected from the input spaces, there is only a single time of day (depicted in the figure by the white gradient in the

sky as opposed to both gradients from the input spaces). Selective projection here means that sometimes the elements of a given input space are projected, while other times they are not. Likewise, sometimes counterpart elements are fused (identified), sometimes they are not, and sometimes they are projected asymmetrically (only one of the counterparts is projected to the input space).



**Figure 2.6: Selective Projection in the Buddhist Monk Riddle**

In light of this discussion of selective projection, it is worthwhile to note how metaphor is handled in a conceptual blending framework. As mentioned above in **Subsection 2.2.2**, conceptual blending theory, while originating from an attempt to account for anomalous data in, or limitations of, metaphor theory (Fauconnier, 2006), now generalizes both the structure and range of cases that can be systematically explained through selective projecting. In conceptual blending theory, metaphoric blends are accounted for as asymmetric in that all of the elements from one input space (in metaphor theory the “target space”) are integrated in the blend space with

only some of the elements from the other input space (in metaphor theory the “source space”). (Grady et al., 1999) For example, in the “the sun is a king” metaphor introduced in **Subsection 2.2.3**, aspects of “king” are “blocked” from mapping from the “king” input space to the blend space – usually the sun does not wear a crown or charge taxes. In this manner it is possible to describe blends as being more or less metaphorical depending on the asymmetry of the selective projection. In (Fauconnier & Turner, 2002) and elsewhere Fauconnier and Turner have identified many families of blends that exhibit structural regularities, of which asymmetric four space metaphorical blends are only one example.

Although the discussion of the conceptual integration network above has described the partial composition of input spaces underlying the blending process, the Buddhist Monk Riddle example shows that content that does not appear in either input space can emerge from the blending process. This emergence of structure in the blended spaces arises from three separate stages of the blending process. These stages are (Fauconnier & Turner, 2002; Grady et al., 1999):

- (1) Composition (elements are composed in the blended space to provide relations that are not in individual input spaces)
- (2) Completion (pattern completion is performed based on other knowledge and additional structure is integrated)
- (3) Elaboration (running the blend dynamically with emergent structure)

These stages of emergence in the blending process can also be illustrated using the Buddhist Monk Riddle. Composition is used to introduce relationships to the blended space that were not in either of the input spaces. As an example of composition, Fauconnier and Turner assert that “composition yields two travelers making two

journeys at the same time on the same path, even though each input only has one traveler making one journey.” (Fauconnier & Turner, 2002) Completion allows for recruitment from frames in order to enrich our interpretations of patterns recognized in the blend. As an example of completion, the authors assert that “the composition of two monks on the path is completed so automatically by the scenario of two people journeying toward each other that it take some thinking to see that the “journeying toward each other” scenario is much richer than the “two monks” composition.” Elaboration allows us to simulate scenarios and imagine using the content of the blended space. As an example of elaboration the authors write that “We run the Buddhist monk blend to get the “encounter” in the blend that provides the solution to the riddle. We are able to run the blend because we know the dynamics of the scenario of two people making opposite journeys along a path, which was brought in by patter completion.” Fauconnier and Turner go on to make the point that a blend can be elaborated in an unlimited number of possible ways and that this ability underlies many forms of human creativity.

The structures and processes described above represent the basic model of conceptual blending. This model is further refined by the introduction of **vital relations** that describe specific types of cross-space mappings such as shared identity or causality. The idea of **compression** as a common characteristic of blends further refines the model. As an example of compression, recall the systematic reduction of scale mentioned in the discussion of the “Join or Die” cartoon of **Figure 2.2**. In that example, links that were formerly cross-space now exist within the blended space as integrated elements from the input spaces. In order to elaborate the blended space and

for emergent structure to arise, information needed to understand a blend may be recruited from other spaces, as well as from frames, which encode highly conventionalized information. This model of blending provides a general template for how concepts, including narrative concepts, can be dynamically combined and elaborated. Within this model it is useful to remember that blended spaces can then be used as input spaces in other blends and that many ideas arise as networks of blends.

Fauconnier and Turner also describe blending as being constrained by a set of “governing” principles. These principles, called **optimality principles**, optimize emergent structure in the blends all “other things being equal.” (Fauconnier & Turner, 2002) Six of the optimality principles that Fauconnier and Turner describe as governing the blending process are (Fauconnier & Turner, 2002):

- (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.
- (6) Metonymic Tightening: When metonymically related elements are projected into the blended space, there is pressure to compress the “distance” between them.

Because we have seen that blending is, in many ways, an unlimited resource for meaning construction it is necessary for the theory to describe what principles are used to constrain this process. (Grady et al., 1999) Optimality principles, as a means to describe pressures producing the most satisfying blends for common sense thought in

the absence of mitigating context, bolster the specificity of the descriptive power of conceptual blending theory.

When it comes to the computational aims of this dissertation, all of these optimality principles require human judgment, and cannot be implemented in any obvious way. However the Topology Principle, in the special case where the relations involved are identities, does not involve meaning, and so can be implemented; indeed, it is part of the Alloy system's conceptual blending algorithm discussed in **Section 3.2.** and introduced in **Subsection 2.2.7** below.

### **2.2.5 Critique of Conceptual Blending Theory and Responses**

Conceptual blending theory is a developing framework and it has been subject to several criticisms. Several main criticisms have been articulated by the psychologist Raymond Gibbs, who sees blending theory as an attractive hypothetical model that, as of yet, lacks some of the criteria of good scientific theories as deemed necessary in experimental psychology. (Gibbs Jr., 2000) In line with his stated desire to make good “psychology” out of blending theory, his criticisms address possible issues with the methodology and framework that apply more broadly to conceptual blending theory as an empirical scientific hypothesis.

The issues raised by Gibbs can be summarized as the following suggestions that conceptual blending theory should:

- (1) acknowledge the importance of falsification,
- (2) contrast its predictions with predictions made under alternative

hypotheses,

- (3) more clearly distinguish linguistic products (interpreted meanings) from linguistic processes (backstage cognitive operations accompanying comprehension),
- (4) and appropriately characterize the role and nature of mental representations in conceptual blending.

Regarding (1), Gibbs suggests that analyses within conceptual blending theory tend to be post hoc descriptive explanations as opposed to predictive falsifiable hypotheses, even intimating that the explanations are upon the theorists intuitions and motivated by a pre-theorized model. Regarding (2), Gibbs claims (following naturally from his first point) that if predictions are made and positive results are claimed in conceptual blending theory, then the results should be supported by evidence that favors the conceptual blending model as opposed to other models. Regarding (3), Gibbs questions the accuracy of inferences made about cognitive processes based on analysis of linguistic products such as the interpretation of an English sentence. He contends that the process by which humans produce linguistic understanding is not constituted by a single activity, but rather “occur along a variety of temporal dimensions, starting in the first milliseconds of unconscious processing and extending up to long-term reflective analysis.” He suggests that conceptual blending theory should clarify the psychological status of blending operations. Suggestion (4) is Gibbs’s most involved suggestion. He recommends that conceptual blending theorists evaluate the status of mental representations in their theory in relation to a set of proposals described by Arthur Markman in (Markman, 1999).

Gibbs's evaluation of conceptual blending theory with respect to Markman's criteria can be summarized as follows:

- (1) Conceptual blending theory seems to satisfactorily connect its models to a wide range of real world linguistic and cognitive phenomena, and specify processes that operate upon its mental representations (although this could be extended, e.g., accounting for temporal dynamics in blending).
- (2) It is not completely clear to Gibbs how, or whether, the grain size of mental representations affects blending processes.
- (3) Gibbs suggests that the mental spaces used in blending theory may be inadequate for accounting for a diversity of cognitive processes since various abilities (e.g., ranging from problem-solving to even aspects of perception) may rest on different representational bases such as "featural representations, structured representations, mental models, [and] image schematic representations."
- (4) Gibbs believes that conceptual blending theory should attend more to the differences between individual and average behavior, and that psychology in general tends to focus on average behavior across people while idiosyncratic data may be extremely important for modeling fine detail of human cognitive performance.
- (5) Gibbs suggests that conceptual blending theory should be more explicit about the relationship of individual cognitive processes to cognition as distributed socially between people, embodied in the



world, and as situated in particular contexts.

Cognitive linguists have offered responses to several of these criticisms. In (Coulson & Oakley, 2000), Coulson and Oakley believe that a mature theory should support falsifiable predictions and assert that researchers in conceptual blending should provide more explicit statements regarding grounding principles and more detailed specifications of the principles underlying blending analyses. Furthermore, Coulson and Oakley agree that post hoc analysis has played an important role in the development of the theory. However, they claim that such analysis often characterizes the early stages of scientific theories, that such analyses play an important role in building and refining models, and that the theory is quite compatible with other methods. Coulson and Oakley also respond to criticism that blending researchers employ *ad hoc* analysis, employing “temporary and improvisational procedures for dealing with specific instances.” Coulson and Oakley reject the validity of that criticism, asserting that it arises from a misunderstanding of the methodology of conceptual blending researchers. Specific novel cases of blending are used for analysis as described in **Subsection 2.2.2**, to mine for “golden events.” In other words, accounting for quite specific, at times seemingly exotic, cases reveals the power of the theory to account for a wide range of phenomena. However, as has been argued above in the discussion of cognitive linguistics methodology, (**Subsection 2.2.2**), this does not mean that the theory itself was developed in an improvisational and intuitive manner way. On the contrary, blending theorists emphasize the empirical nature of the theory.

Fauconnier responds to general critique of blending theory in (Fauconnier, 2000), and his arguments reveal a position regarding the distinguishing of linguistic “product” from “process.” Conceptual blending theory should not be seen as making claims for specific cognitive processes on the basis of linguistic data. On the contrary, empirical data from diverse areas of cognitive science has converged to suggest the general principles of conceptual blending. Evidence, not only from language but also from other types of discourse and media, has been found to fit the conceptual blending model. While results directly grounding conceptual blending theory in neurobiological processes is not developed, Fauconnier is clear that linguistic distributions are “just the tip of the iceberg.” (Fauconnier, 2000) The idea is that a model is necessary to explain the rich system of activations and associations triggered by language, and that a convergence of empirical evidence suggests conceptual blending as such a model.

Regarding Gibbs’s discussion of mental representations, the view taken in this dissertation is that grain size is indeed an important issue for future explication in conceptual blending theory (in accordance with by Coulson and Oakley). Conceptual spaces have been described as relatively small, temporary packets of meaning, however Turner has also described blending of narratives (Turner, 2003), which most researchers would describe as being at a different grain size than mental spaces in conventional metaphor for example. Also, it would be beneficial to clarify the range of cognitive processes meant to be encompassed in a blending framework as Gibbs suggests (following Markman).

Conceptual blending theory does, however, address differences between

average and individual performance and issues of situated and distributed cognition. Fauconnier and Turner suggest a set of constitutive principles, described above, that yield an astounding capacity for meaning construction on the basis of minimal linguistic information. However, Fauconnier and Turner's optimality principles are oriented toward capturing "average" blending performance. While issues of context and distribution between individuals and artifacts should be made even more precise within blending theory, these are the pressures which influence the selection of particular conceptual spaces. Within the particularities of a given context, optimality principles (constraints) as proposed by Fauconnier and Turner are claimed by Gibbs to be a ripe area for first introducing falsifiable hypotheses to blending theory. Coulson and Oakley also suggest that optimality principles are a response to the charge that blending theory is overly broad ("being too powerful, accounting for everything, and, hence, explaining nothing"). (Coulson & Oakley, 2000) The importance of optimality principles in all of these regards has also influenced the choice in the computational approach to blending described in **Chapter 3** of this dissertation to focus primarily upon implementing a set of computationally feasible optimality principles.

### **2.2.6 Computational Approaches to Analogy, Metaphor, and Conceptual Blending**

This section discusses several important systems that attempt to algorithmically implement aspects of analogy, metaphor, and conceptual blending. The computational analogy systems are described at a high level because, although

there is a very close connection between analogy and metaphor, the focus here is on computational approaches to conceptual blending. The computational analogy systems are discussed primarily in order to raise salient features that inform the problem of implementing conceptual blending systems, and to raise important distinctions in philosophical vantage point between the systems. A brief account of the structure-mapping approach to analogy is provided at the beginning of **Subsection 2.2.6.1** in order to provide background for the discussion of the computational systems. The seminal Structure Mapping Engine (SME) by Falkenheimer, Forbus, and Gentner is presented first as a milestone system designed to find analogical mappings between input spaces. (Forbus, 2001; Gentner, 1983) The Learning and Inference with Schemas and Analogies (LISA) system by Holyoak and Hummel is presented as a contrasting neural-network based approach to analogy, aimed at greater biological fidelity and addressing the issue of working-memory constraints in analogical thinking. (Holyoak & Hummel, 2001) The Sapper system represents another approach to analogy by Veale, O'Donoghue, and Keane in sympathy with the SME approach at a very general level (in the specifics there are notable differences and conflicting accounts of performance), but has been rationally reconstructed by Veale and O'Donoghue as a conceptual blending system. (Veale, 2000) Finally, the Divago system represents a traditional symbolic AI approach (though the system does innovatively employ a genetic algorithm to generate possible blends) to capture all of the fundamental aspects of conceptual blending in a computational system. (Pereira, 2004)

A system developed by Narayanan and Feldman geared toward investigating the neural foundations of metaphor, with applications to narrative inference, will be discussed separately. (Narayanan, 1999) It is discussed separately because, as an attempt to capture aspects of the image schematic foundations of conceptual metaphor, its aims are significantly different from the systems described above. It is more directly relevant to the issue of computational narrative.

The Alloy system, as one of the research contributions of this dissertation, is contrasted with these system at a high level in subsequent discussion in **Subsection 2.2.6.4** and is discussed in detail in **Chapter 3** after its formal and theoretical foundations are discussed in **Section 2.3**.

#### **2.2.6.1 Computational Approaches to Analogy**

Analogy, the ability to think “in terms of relational patterns,” is a phenomenon closely related to metaphor and conceptual blending. (Gentner, Holyoak, & Kokino, 2001) Of particular interest here is the influential **structure-mapping** theory of analogy proposed by Dedre Gentner. (Gentner, 1983) Like metaphor and conceptual blending theories, the structure-mapping theory of analogy also consists of mapping between concepts. Gentner describes fundamental aspects of structure-mapping theory as follows (with numbering added for additional clarity) (Gentner, Bowdle, Wolff, & Boronat, 2001):

- (1) Analogical mapping is the process of establishing *structural alignment* between two represented situations and then projecting inferences.

- (2) Structure-mapping theory assumes the existence of structured representations made up of objects and their properties, relations between objects, and higher-order relations between relations.
- (3) An alignment consists of an explicit set of correspondences between the representational elements of the two situations.
- (4) The alignment is determined according to *structural consistency* constraints: (1) one-to-one correspondence between the mapped elements in the base and the target, and (2) parallel connectivity, in which the arguments of corresponding predicates also correspond.
- (5) In addition, the selection of an alignment is guided by the *systematicity* principle: a system of relations connected by higher order constraining relations such as causal relations is preferred over one with an equal number of independent matches.
- (6) Systematicity also guides analogical inference: people do not import random facts that complete the system, but instead project inferences that complete the common system of relations.

The theories of analogical structure-mapping, metaphor, and conceptual blending overlap in the phenomena they describe (with conceptual blending being the most general of the three). Gentner has claimed that novel metaphors, those being processed in real time as opposed to relying upon convention, can be modeled using structure mapping. (Gentner, Bowdle et al., 2001) Fauconnier has suggested that research of conceptual blending is related to research of analogy in at least three ways (Fauconnier, 2001):

- (1) the function of some (but by no means all) conceptual integration networks is analogical,
- (2) conceptual integration networks whose function is not analogy still align two or more partial structures, typically by means of an analogical mappings, and
- (3) standard examples of analogy and metaphor often turn out to be cases of conceptual blending with analogical or metaphorical cross-space alignment.

Due to these interrelationships between these phenomena, computational approaches to analogy involving structural alignment are quite relevant to computational approaches to conceptual blending. The SME and LISA systems mentioned above involve structure mapping at a very general level, though the systems utilize different algorithms, have different underlying foundations, and have differing underlying values and philosophical bases. The following is a description of important distinguishing characteristics of these computational analogy systems in terms of:

- (1) functionality,
- (2) distinguishing characteristics,
- (3) research goals,
- (4) selected research results, and
- (5) underlying values<sup>7</sup>.

Similar descriptions of the Sapper and Divago systems follow in **Subsection 2.2.6.2**.

### **SME (Faulkenheimer, Forbus, and Gentner) (Forbus, 2001):**

**Functionality:** SME is based on the concepts of structure mapping and graph isomorphism. It takes two spaces as input, a base and a target, and computes a mapping or set of mappings. Mappings contain a set of correspondences that align particular items in the base and target. It also produces “candidate inferences” which

---

<sup>7</sup> Since not all of the authors are explicit about their underlying values, I have attributed implicit values to their systems based upon the authors’ descriptions of their systems and the research traditions in which they are based. This attribution of values is biased by the adherence to the cognitive linguistics paradigm here and any brief description of underlying values is necessarily simplified.

as statements about the base that are hypothesized to hold in the target by virtue of the correspondences.

**Distinguishing Characteristics:** SME is based upon empirical results in experimental psychology. It pioneered the computational approach to structure-mapping in analogy.

**Research Goals:** SME is used to generate mappings in order to facilitate new insights leading to new psychological experiments and results. It is also intended for use in AI systems to perform tasks associated with complex real-world problems (e.g., reasoning about international crises or evaluating battlefield courses of action).

**Research Results:** Results involving SME include support for the theory of systematic and structural consistency influencing interpretation of analogies and inferences made with them. Researchers using SME to investigate similarities between spaces also claim the following results: similarity comparisons use structure mapping, similarity based retrieval is surface driven, but reasoning is structurally driven.

**Underlying Values:** SME is not considered to be a cognitive modeling system, but it was created in order to capture crucial aspects of the systematicity of human analogy and inference based upon psychological results, and in turn to generate further insights to be tested on human subjects. In this regard, it is not a GOFAI system based in the



belief that symbolic approaches can lead to direct insight about human cognitive processes. However, another explicit goal is to use SME within large scale simulations, possibly to be integrated as the analogical reasoning component along with separate modules for vision, NLP, or other relevant tasks. In this regard, research using SME does not seem to be oriented toward integrating issues of distributed and embodied cognition into the modeling of analogy itself.

**LISA (Hummel and Holyoak) (Holyoak & Hummel, 2001):**

**Functionality:** LISA is based on the concept of structure-mapping, however it is implemented using a neural network model of analogy. Because its connections approach greatly distinguishes it from other systems considered in this section the functionality is described at a greater level of detail in the following discussion. The core of the system architecture is dynamically binding relations to their arguments in system of working memory (WM ) and encoding those bindings in a system of long-term memory (LTM). Predicates and objects are represented in WM as patterns of activation on a collection of **semantic units**. Predicates and objects fire in synchrony when they are bound, out of synchrony when they are not. Propositions (e.g., “love (Jim, Mary)”) are encoded in LTM by a hierarchy of **structure units** (a Proposition is at the top of the tree, Subpropositions lie at the next level (e.g., “Jim + lover” and “Mary + beloved”), and predicates and objects lie at the bottom level (e.g., “Jim” or “love”). Semantic units do not hold patterns of semantic content in a direct way, they just store content in LTM and generate corresponding patterns on the semantic units.

Analogs, divided into a **driver** and one or more **recipients** (the base and target spaces), are represented as separate sets of structure units that activate the same. Additionally, there are mappings between structure units of the same type in different analogs. Mapping is accomplished using guided pattern matching, as proposition units become active in the driver, they generate synchronized patterns of activation on the semantic units. Weights on mappings grow based on simultaneous activation of semantic units. At the conclusion of a simulation run, “corresponding structure units will have large positive weights on their mapping connections, and non-corresponding units will have strongly negative weights.”

**Distinguishing Characteristics:** The use of a connectionist model with some characteristics of symbolic processing is a unique feature of LISA. For Hummel and Holyoak, the heart of symbolic processing is the ability to represent relations independently of arguments; LISA has this capability. A second distinguishing characteristic of LISA is the concern for analogical mapping under memory constraints.

**Research Goals:** LISA was designed as an investigation of analogy in a biologically plausible symbol system. This means that there is a focus on psychological and (ultimately) biological fidelity, and knowledge representations that capture the symbolic nature of human cognition in a manner that roughly might be realized in the brain. Toward this end, it takes into account limitations of working memory, and

attempts fuller integration of the proposed major steps in analogical thinking (access, mapping, inference, and learning).

**Research Results:** Hummel and Holyoak see a major result of LISA as being a potential model of working memory in analogy. Secondly, LISA represents large gains in efficiency and plausibility over earlier related efforts at connectionist approaches to analogical structure-mapping.

**Underlying Values:** Hummel and Holyoak believe that a connectionist system with memory limitations is more faithful to human neurobiological structure and processing. They believe that symbolic systems based approaches are limited for cognitive modeling of analogy because they all “take symbolic, propositional representations as inputs, and perform complex symbolic operations to generate plausible sets of candidate mappings,” and approach that has been criticized for its lack of psychological plausibility, lack of concern for memory limitations, and limitations in its extensibility.

#### **2.2.6.2 Computational Approaches to Conceptual Blending**

The following are descriptions of the Sapper and Divago systems, presented in the same format as the analogy systems described above.

**Sapper (Veale, O'Donoghue, and Keane):**

**Functionality:** Sapper is a symbolic structure-mapping approach to analogy similar to SME, taking as input two or more input spaces, but the output is a structured representation of semantic relationships, which can be seen as input to further processes.

**Distinguishing Characteristics:** Sapper distinguishes itself by eschewing the notion of a workplace in which mapping is computed and assumes that mapping and integration processes are done in semantic memory. Sapper also uses a type of semantic network (called a **slipnet**) to describe a network that connects one set of relations to another set of relations onto which they can be mapped. Also, Sapper has been described as capable of conceptual blending operations beyond finding unidirectional analogical mappings.

**Research Goals:** Sapper attempts to provide a structure-mapping system that more robustly accounts for mappings based upon attribute similarity in addition to structural isomorphism. This is viewed as a weakness of the approach by SME researchers (see (Ferguson, Forbus, & Gentner, 1997)), however Veale et. al. consider this to be a strength of Sapper and believe that SME is inflexible with regard to attribute based (as opposed to structure based) analogies. (Veale & Keane, 1998) A key point in this controversy may be the relative status given to such analogies. The Sapper system

takes its approach because the goals motivating it include providing a model accounting for the following principles:

- (1) high-level analogies arise out of nascent, lower-level analogies automatically recognized by memory processes;
- (2) analogy is memory-situated inasmuch as it occurs *in situ* within the vast interconnected tapestry of long-term semantic memory, and may potentially draw upon knowledge any fragment; and
- (3) this memory-situatedness frequently makes analogy dependent on some form of attributive grounding to secure its analogical interpretations.

**Research Results:** Research results include increased efficiency on some data sets over previous systems, especially with regard to attributional analogy. Sapper has also been described by its authors as demonstrating the computational feasibility of conceptual blending.

**Underlying Values:** Sapper represents a symbolic AI approach to issues of analogy and conceptual blending. This is because of a focus on computational modeling as a demonstration of psychological plausibility, often on the basis of computational principles (though there is an appeal to psychological literature as well at times). Extensions to conceptual blending are proposed, if tentatively, on the basis of computational utility alone. For example, an additional intermediate space between input spaces and the blended space called **constructor** space, containing abstract structure for “constructing a generic space from inputs” and allowing “inputs to be structurally aligned and coherently fused in the blended space,” is proposed as a contribution to conceptual blending theory because it allows “computationalists to better articulate and parameterize their models of the blending process.”

### **Divago (Pereira):**

**Functionality:** Divago is an attempt to construct a computational system that captures all of the “fundamental aspects of conceptual blending.” (Pereira, 2004) Given a knowledge base, the system takes a pair of input spaces as input and constructs a blended space and the necessary mappings to the blended space. The knowledge base contains a set of concepts, each one defined according to several different kinds of representations (concept maps, rules, frames, integrity constraints, and instances). An overview of Divago’s functionality follows (the architecture is based on a system of modules whose names are introduced in boldface type):

- (1) Input knowledge (a pair of concepts) is either given by user or generated randomly.
- (2) The **Mapper** builds a structural alignment between them (a Sapper inspired approach is used).
- (3) The **Blender** produces a set of projections that define the set of all possible blends. This is the search space.
- (4) The **Factory** constructs possible blends based on a genetic algorithm that searches for the blend that best complies with the evaluation given by the **Constraint Module**.
- (5) Each blend goes to the **Elaboration Module** where it is subject to the application of domain or context dependent knowledge. The Elaboration Module applies elaboration based on internal-logic and rule-based elaboration. The rules are a part of the knowledge bases.

(6) The evaluation of a blend by the Constraint Module is built on an explicit implementation of the Fauconnier and Turner's optimality principles. The principles take into account knowledge that comes from the knowledge base (integrity constraints and frames), and accomplishment of a goal formulate as a query.

(7) After finding a satisfying solution or a specified number of iterations, the Factory stops the genetic algorithm and returns the best solution it has achieved. Sometimes there is an interpretation module that produces an interpretation of the new concept (like 2-D or 3-D images)

(8) Note that the Mapper and Elaboration Module are optional. The reasons for this are that it may be advantageous to hand-code the mappings for experimentation (since the output of the algorithm used could prove restrictive) and elaboration can obscure results of optimality constraints and mapping.

**Distinguishing Characteristics:** Divago is distinguished by explicitly representing and providing algorithms for a comprehensive set of the fundamental features of conceptual blending such as Fauconnier and Turner's optimality principles and the generation of emergent structure and content in the blended spaces. In contrast, some feature of blending in Sapper must be retrospectively interpreted as corresponding to aspects of conceptual blending theory, as opposed to explicitly having those aspects in its architecture.

**Research Goals:** Divago is intended as an attempt to provide an architecture for a system that can accomplish tasks that would be considered “creative” when accomplished by a human, such as the generation of novel concepts. This is an AI goal and Divago is not a cognitive modeling system in the sense of SME (providing results and bases for hypotheses for later investigation with human subjects) or LISA (striving for biological plausibility).

**Research Results:** Divago has produced output in sympathy with several examples of conceptual blending from (Fauconnier & Turner, 2002) and the blending literature.

**Underlying Values:** Divago is a symbolic AI system featuring the use of a genetic algorithm. The underlying value system relates strongly to those of GOF AI, however. While Sapper exhibits many GOF AI characteristics, the Sapper literature also exhibits a concern for issues of context and situatedness, at least philosophically. Divago takes the models of described in the blending literature and explicitly designs components corresponding to them without concern for psychological plausibility. At the same time, with its underlying concern for “computational creativity,” and its unmitigated use of an explicit symbolic knowledge base to facilitate generation of emergent structure, Divago seems to be an attempt to use traditional GOF AI approaches, along with more contemporary AI techniques to capture, in its entirety if possible, a humanly creative process that is embodied, distributed, and situated. The challenge of the system is whether or not an explicit, symbolic conceptual blending model will



facilitate for results that surpass, in terms of generative capacity and efficiency, previous GOFAI models of creativity.

### **2.2.6.3 A Computational Metaphor Theory Based Approach to Narrative**

The following discussion describes a research project carried out by Jerome Feldman and Srinivas Narayanan as an example of such work. This system is presented in a somewhat different spirit than other computational systems described here. It is of interest because it explicitly bridges the issue of computational approaches to metaphor with narrative representation. For this reason, the system described below has a special relevance to the issues of computational narrative that are central to this dissertation. Feldman and Narayanan explore the belief that metaphoric, and in some cases narrative, interpretation are grounded in image schematic structure. (Turner, 1996) Narayanan has implemented a computational model of metaphoric interpretation based on this belief. In particular, he examines the interpretation of simple causal narratives taken from newspaper articles. After pre-parsing the narratives, the system is able to generate inferences based upon the input.

The input to Narayanan's system consists of narratives such as the following (taken from a 1995 New York Times article):

In 1991, in response to World Bank pressure, India boldly set out on a path of liberalization. The government loosened its stranglehold on business, and removed obstacles to international trade. While great strides were made the first few years, the Government is currently stumbling in its efforts to implement the liberalization plan. (Narayanan)

In this passage, institutions described as actors and goals are described in spatial terms. Inferences can be made by considering conventionalized metaphors such as **Falling implies Failure**. The system required to model such interpretations is implemented as follows: The architecture consists of three components, the source domain, the target domain, and the metaphor map. They model the source space using a model of events called x-schemas. The formal computational model of the source domain is an extension of Stochastic Petri Nets, which are bipartite graphs containing ‘places’ and ‘transitions.’ In (Narayanan, 1999), Narayanan describes the function of this structure in the following manner:

Places hold *tokens* and represent predicates about the world or internal state. Transitions are the active component. When all of the places pointing to a transition contain an adequate number of tokens (usually 1) the transition is *enabled* and may *fire*, removing its input tokens and depositing a new set of tokens in its output places. The most relevant features of Petri nets for our purposes are their ability to model events and states in a distributed system and cleanly capture sequentiality, concurrency and event-based asynchronous control.

The main idea is that when a phrase involves a motion term people understand that phrase by performing a simulation of the event in the context set up by the phrase. Here the physical world is modeled using x-schema.

The target domain is modeled using belief networks with a temporal extension. Belief networks<sup>8</sup> are well-known structures and algorithms in artificial intelligence research used for reasoning with uncertainty and based on probability theory. (Russell & Norvig, 2002) The target domain represents a knowledge base over some subject matter, international economic policy in the example above.

---

<sup>8</sup> Also called Bayes’ Networks.

Metaphor maps connect x-schema to belief networks. The results of x-schema execution are mapped onto the belief networks as evidence. In this way information about the source, for example *falling*, is connected to information about the target, for example ‘failure of an economic liberalization plan.’ The account here omits many details of their system but is intended to capture the crucial elements of their representation.

Though a goal of Naryanan and Feldman was to provide some evidence for a neurological grounding of metaphor theory<sup>9</sup>, here the emphasis is not on the claims regarding human cognition but rather on the implementation of a cognitive linguistics inspired representation of metaphor and narrative. Though their goal is analysis and computationally producing suggestive empirical results, I would like to suggest that such work is also useful for synthesis, i.e. stories can be generated using computational accounts of relevant cognitive linguistic concepts such as metaphor. In this case, such a framework allows new stories to be dynamically created by applying a set of author specified target domains to a set of image schematic source domains. The stories generated would then have coherent structure according to commonplace metaphors but could range widely over a set of fictional target spaces. The narratives generated in this manner would be simple causal narratives that could be used within the framework of larger narrative structures. Though such a system is merely a thought experiment at this point, consideration of applications in this area is a

---

<sup>9</sup> This is controversial among some cognitive scientists because connectionist models are extremely simple models when contrasted to neurobiological structures that are idealized at best, and are so superficially related as to be unsuitable for neurobiological grounding at worst.

promising direction toward resolving the problem of creating interactive and generative computational narratives.

#### **2.2.6.4 Discussion of Computational Approaches to Cognition**

Considering the commitments of the cognitive linguistics enterprise, and the critique of formal accounts of meaning that the enterprise provides, deep questions are posed regarding whether, how, and when computational modeling is seen as a valid method for investigating issues of meaning construction such as in conceptual blending theory. Traditional GOFAI work in knowledge representation, expert systems, semantic networks, and other well-known projects has focused on complete and correct formal descriptions of domains to allow shared knowledge bases for making inferences. Though domains themselves may have encoded specific bodies of knowledge, within particular domains comprehensive global understanding was a goal. The results were systems with narrow expertise and the characteristic that “to solve a problem you almost have to know the answer already.” (Russell & Norvig, 2002) An early example of such a project was the blood infection diagnosis expert system MYCIN developed at Stanford University in the 1970’s by Ed Feigenbaum, Bruce Buchanan, and Edward Shortliffe. (Russell & Norvig, 2002) MYCIN produced successful diagnoses in a specific domain based upon rules (including certainty factors) acquired through extensive interviewing of experts. In contrast, dynamic, subjective meanings (both culturally entrenched and novel) and inferences, crucial

examples of conceptual blending, have not been traditional foci of inquiry in artificial intelligence research<sup>10</sup>.

Under this view, it is a welcome development that the systems above emphasize issues such as novel metaphor generation, emergent structure in conceptual blending, and dynamic and efficient implementations. It is also a welcome development that issues of biological plausibility are considered. However, it is important to exercise care in making claims about the relationship between computational modeling and human cognition. Strong critiques of the GOFAI enterprise have been posed in (Winograd & Flores, 1986) and (Agre, 1997) “on the grounds that its technical methods presuppose mistaken philosophies” or the necessity for critical technical practices aware of their historical and discursive foundations (and their limitations). These critiques contain the idea that words such as “intelligence,” “concept,” and “inference” can be subject to simplification and reformulation into algorithmic and data structural terms, which, on the basis of powerful implicit social metaphors of computation, obscure the limited status of computational results with regard to human cognition. Terry Winograd and Fernando Flores oppose a strand of cognitive science based on assumptions that they summarize as:

- (1) All cognitive systems are symbol systems. They achieve their intelligence by symbolizing external and internal situations and events, and by manipulating those symbols.
- (2) All cognitive systems share a basic underlying set of symbol manipulation processes.
- (3) A theory of cognition can be couched as a program in an appropriate formalism such that the program when run in the

---

<sup>10</sup> However, for computational narrative artworks and investigation of narrative human thought, such semantics are crucial areas of inquiry.

appropriate environment will produce the observed behavior.  
(Winograd & Flores, 1986)

Clearly, such assumptions contradict the views held within the cognitive linguistic enterprise for reasons sketched above. Realizing this behooves us computer scientists, who intend to contribute to the understanding of human semantics, to be explicit and cautious about the status of our implementations with respect to cognitive science results. The computational analogy, metaphor, and conceptual systems described above each provide valuable and extensive contributions, including efficient implementation with utility for a variety of practical and theoretical applications. This is especially the case when the claims made are in direct dialogue with cognitive science results as in the SME system (which also focuses on structural features of conceptual spaces). The hope here is that the claims and results of the Alloy system can be useful for clarifying and provoking dialogue about some of these issues and, more broadly, the role of computational modeling in cognitive linguistics. This discussion is elaborated and Alloy system is presented in **Section 2.3** and **Chapter 3** below.

## 2.3 Algebraic Semiotics

Algebraic semiotics is one of the primary theoretical tools utilized in the work described in this dissertation. It combines theories from semiotics and algebraic semantics, and has been developed in sympathy with work in mental spaces, metaphor, and conceptual blending theories from cognitive linguistics. The following

sections describe fundamental definitions and uses of algebraic semiotics, preceded by brief accounts of necessary concepts from the fields and theories that inform it for readers unfamiliar with the relevant disciplines.

### 2.3.1 Basic Semiotics Concepts

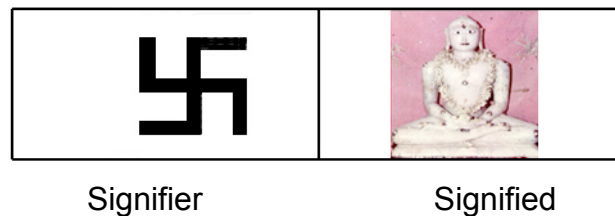
Semiotics is the study of signs; it is the investigation of how representations in the world convey meaning<sup>11</sup>. Though the semiotic model used in this dissertation is Joseph Goguen's algebraic semiotics, it is important to first present several foundational ideas from semiotics for readers unfamiliar with the field. The focus here is on two major insights from two of the founders of semiotics, Ferdinand Saussure and Charles Saunders Peirce. (Peirce, 1965; Saussure, 1965) Peirce emphasized (among other concerns) that the relation between a given token and the object it represents is not just a function, but a relation that depends on the situation in which the token is interpreted, while Saussure emphasized (among other concerns) that signs come in systems. Algebraic semiotics (described in **Subsection 2.3.2**) is a mathematical attempt to capture Saussure's insight, while the blending of semiotic systems captures and extends Peirce's insight. (Goguen & Harrell, 2007a) The original area of Saussure's theory was the field of linguistics, whereas Peirce's work was accomplished in the fields of philosophy and mathematical logic, but the idea of a sign is broadly conceived as any representation that conveys meaning and has been

---

<sup>11</sup> The cognitive linguistics work described in Section 2.2 also addresses issues of meaning that overlap with those in semiotics, but differs from the earlier semiotics approaches in its historical development, empirical basis, philosophical commitments, and concern for issues of embodied, distributed, and situated cognition.

extended to fields as diverse as literary studies, film studies, art theory, and cognitive science.

The central idea in any semiotic theory is the relationship between two terms, one term which represents and another term which is represented. (Barthes, 1972) In Saussure's terminology these are the "signifier" and the "signified" respectively. The combination of these two terms is a "sign." For example, for people of the Jain religion from India the swastika symbol represents the Tirthankara (person who has achieved enlightenment) Suparshvanath. ("Jainism Global Resource Center," 2006; Unknown, 2006) In **Figure 2.7** the right facing swastika acts as a signifier for the concept of the Tirthankara Suparshvanath (the image in the rectangle on the right represents the idea of the Tirthankara as opposed to the graphical image of the Tirthankara). (Unknown, 2006)



**Figure 2.7: The sign for the Tirthankara Suparshvananth**

In other contexts the swastika has (unfortunately) come to represent the Nazi political party of the twentieth century and its associated imagery such as Nazi atrocities, leaders, and soldiers. **Figure 2.8** depicts an alternate possibility for the swastika signifier.





Signifier                      Signified

**Figure 2.8: The sign for the Nazi party**

The combination of the signifier and the signified is termed a “sign.” The sign is the unification of a representation with its interpretation, it is how we experience the signifier as laden with the meaning of its signified. The sign in one system, however, can become the signifier in another system. For example, in on a website for a group called “Youth Against Racism in Europe” (“Youth Against Racism in Europe,” 2006) contains the sign in **Figure 2.9** depicting a swastika being tossed into a waste bin.

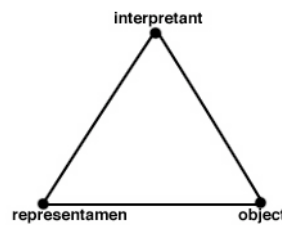


**Figure 2.9: A sign from the “Youth Against Racism in Europe” website**

Here the swastika is used not to represent the Nazi party of the 1930s and 1940s, but the more generalized concept racist fascism, neo-Nazism, and the reappropriated uses of the symbol by contemporary racists and fascists. The sign described in **Figure 2.8** is now a second level signifier, the signifier contains the meaning of the Nazi political party only at one level removed, it now signifies contemporary groups, policies, and attitudes that either sympathize with or echo those of Adolf Hitler’s Nazi party. If **Figure 2.9** was depicted in a context where the Jain interpretation of the symbol prevailed the interpretation could be quite different, the multi-layered meaning of the symbol and alternate meanings would be constructed – perhaps it could be used as a

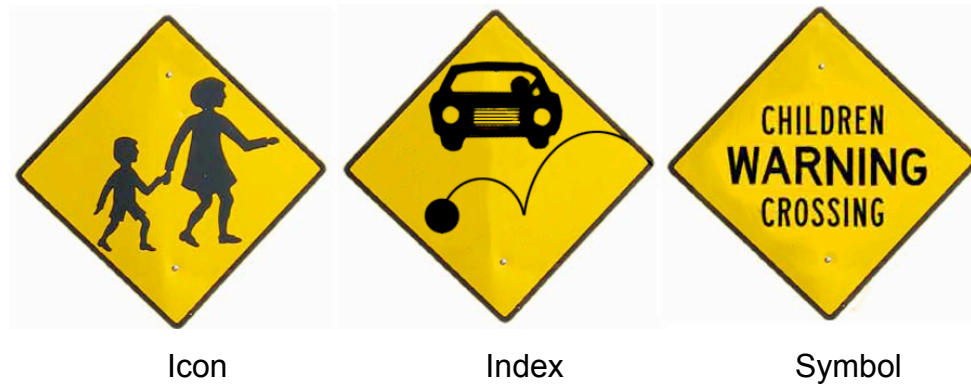
critique against recent Eurocentric interpretations of the ancient religious symbol. The core concept here, however, is that signs come in systems: signs become signifiers in new systems and are incorporated within other signifiers.

Peirce's semiotics expands the relationship between the signifier and signified to a triad as depicted in **Figure 2.10**. In his terminology the signifier is a "representamen," the signified is an "object," and the relation that holds between the two for some individual is the "interpretant." The interpretant reflects the idea that a particular observer in a particular situation may interpret the relationship between the representamen and object in an unlimited number of ways. Peirce referred to this notion of innumerable possible interpretants as "unlimited semiosis."



**Figure 2.10: Peirce's Semiotic Triad**

Peirce also classified signs by type, the basic triad of sign types being icons, indexes, and symbols. An icon is a sign in which the representamen holds similar qualities as the object that it represents. An index is a sign in which the representamen directly connects to the object via a physical or causal relationship. For example the presence of the representamen could indicate that there is usually an instance of the object. (Lechte, 1994) A symbol is a sign in which the relationship between the representamen and the object is attributable to arbitrary social convention, such as most signs used in speech and language.



**Figure 2.11: Three signs warning of the danger of a child being on the road**

For example, **Figure 2.11** depicts three road signs bearing representamens signifying the object of a child possibly crossing the road in the path of an oncoming vehicle.

The iconic sign depicts an actual child crossing the road in silhouette. This sign is iconic because it bears a clear similarity to the real image of a child crossing the road.

The fact that an older escort accompanies the child only adds to the signs iconicity, most commonly a child would not cross the road alone so the sign bears a strong similarity to a likely scenario. The indexical sign above<sup>12</sup> depicts a scenario in which a ball has bounced into the path of an oncoming vehicle. The scenario illustrated by this representamen exists in a type of causal relationship with the object. The scenario indicates a strong possibility that a child will soon cross the road in front of the vehicle to chase after the ball; the presence of the scenario indicates a strong possibility for the object. The symbolic sign above bears no physical resemblance to its object. The letters and words used in the English language sign gain their meaning only through conventional use. The core idea here in presenting Peirce's triadic sign, and his triad of basic sign types, is to emphasize his observations about how the relations in a given

---

<sup>12</sup> This is a fictitious road sign created only for this example, the iconic and symbolic signs are actual road signs used in the United States of America.

sign are necessarily interpreted in a particular context, by a particular individual.

Without this context the signs fail to convey meaning

### 2.3.2 Algebraic Semantics

Algebraic semantics has its origin in the mathematical foundations of abstract data type theory (Goguen, Thatcher, & Wagner, 1978; Goguen, Thatcher, Wagner, & Wright, 1975). This section introduces some basics of that theory using the BOBJ language (the content of this section can also be found in (Goguen & Harrell, 2007a)). Much more detail appears in the literature, e.g., (Goguen & Malcolm, 1996; Goguen et al., 1978) for algebraic semantics, (Goguen, Rosu, & Lin, 2002) for BOBJ, and (Goguen, Winkler, Meseguer, Futatsugi, & Jouannaud, 2000) for its ancestor OBJ3.<sup>13</sup> Modules in BOBJ are called **theories**, of which we use two kinds, **data theories**, and **loose theories**. Below is a simple data theory named MEDIUM which will be used in later examples to provide values for attributes of some objects in conceptual spaces.

```
Dth MEDIUM is sort Medium.
    Ops land water : -> Medium.
End
```

The keyword ‘dth’ indicates a data theory. The keyword ‘sort’ indicates a type declaration<sup>14</sup>. Next, the keyword ‘ops’ indicates a declaration for operations, in this case, two constants, ‘land’ and ‘water’, of the sort ‘Medium’, using the convention that constants are given as operations with no arguments; argument sorts come before the arrow, while the value sort comes after it. The sort and operation declarations

---

<sup>13</sup> “OBJ” refers to the language family, particular members of which include OBJ3 and BOBJ.

<sup>14</sup> The OBJ languages use the word “sort” to avoid the ambiguities of the word “type.”

together, possibly with some subsort declarations, are known as the **signature** of the theory. BOBJ imports the Booleans into every module by default, so that the two Boolean values and some standard Boolean operations are also included in MEDIUM.

The following is an example of a loose theory (this will be used in **Section 3.1** of this chapter to classify certain items and relations in some other theories that will be blended).

```
Th GENERIC is sorts Person Object.
  Pr MEDIUM.
  Op person : -> Person.
  Op object : -> Object.
  Op use : Person Object -> Bool.
  Op on  : Object Medium -> Bool.
  Eq use(person, object) = true.
Endth
```

This theory imports the MEDIUM theory, as indicated by ‘pr MEDIUM’ (‘pr’ is short for “protecting,” which is the usual mode of importation (Goguen, 1998)) and it then declares three new constants. The OBJ languages do not provide relations as such, but these can be represented by Boolean valued functions. This module declares two such relations, ‘use’ and ‘on’, using the keyword ‘op’; each has two arguments, the sorts of which are given before the arrow, while the value sort ‘Bool’ of Booleans comes after the arrow. Such a relation holds of its arguments if its value on those arguments is ‘true’, which can be expressed with an equation, as in the above theory. **Constructors** are operations that build new elements from their arguments; the theories above do not have any constructors, but examples of these will be presented later.

The models of a loose theory are all the structures (called **algebras**) that

provide a set for each sort symbol, and a function for each operation symbol in the signature, and that satisfy all the equations in the theory. For data theories, there is just one intended model up to isomorphism, consisting of the minimal (or “free”) set of terms generated over the signature. In the case of **MEDIUM**, this yields just the two constant terms, ‘land’ and ‘water’, a so-called enumerated type.

There is a basic duality between theories and models: A semiotic theory determines the class of models that satisfy it, which we may call its **model space**<sup>15</sup>; and a class of models has a unique (up to equivalence) most restrictive theory whose models include it.

### 2.3.3 Algebraic Semiotics Definitions and Key Concepts

Algebraic semiotics builds upon, and formalizes, some of the key concepts up Saussure and Peirce using techniques from algebraic semantics. Much of the content of this section on algebraic semiotics that follows can also be found in (Goguen & Harrell, 2007a) and the following comments concerning its philosophical orientation and subsequent OBJ based examples are due to Joseph Goguen and developed by Goguen and the author. Before discussing algebraic semiotics, it will 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

---

<sup>15</sup> This use of the word “space” differs from that in the conceptual spaces of cognitive linguistics, which are actually single models, rather than classes of models. Also, recall that in algebraic semiotics, a “conceptual space” is a theory, not a model.

mathematical thoughts of God.” Similarly, “situations” in the situation semantics of Barwise and Perry, which are similar to conceptual spaces (but more complexly structured) are considered to be actually existing, ideal Platonic entities. (Barwise & Perry, 1983) Somewhat less grandly, one might consider that conceptual spaces are somehow directly instantiated in the brain. However, the point of view here is that such formalisms are constructed by 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.

Whereas conceptual spaces, which consist of elements and relations among them, are good for studying meaning in natural language, they are not adequate for computational narrative. For example, conceptual spaces and conceptual blending can help us understand metaphors in poems, but more than elements and certain instances of relations among them is needed for an adequate treatment of structure, e.g., to describe how a particular meter combines with a specific rhyme scheme in a fixed poetic form; music raises similar issues, which again require an ability to handle structure. Conceptual spaces are good for talking about concepts about (e.g., how we talk about) things, but are awkward for talking about the structure of things.

Thus, to use blending as a basis for multimedia narrative, it is necessary to generalize conceptual spaces to take account of structure, which requires constructors and axioms; it also helps to have a hierarchical type system. Hence conceptual blending is distinguished from **structural blending**, also called **structural integration**, where the former is blending of conceptual spaces and the latter is

blending that in general involves non-trivial constructors.

### 2.3.3.1 Semiotic Spaces

Algebraic semiotics uses algebraic semantics 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. A **semiotic system** (also called a **semiotic theory** or **sign system**) (Goguen, 1998) consists of 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. Loose sorts classify the parts of signs, while data sorts classify 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 in the signature, some 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 an important role in determining these orderings. The **models** of a semiotic theory are just the models of its algebraic theory. Conceptual spaces are the special case where there are no operations except those representing constants and relations, and there is only one sort. Formally, a semiotic system ‘S’ consists of (Goguen, 1998):

- (1) A set of sorts for signs, not necessarily disjoint;
- (2) A partial ordering on S, called the subsort relation and denoted  $\leq$ ;
- (3) A set V of data sorts, for information about signs, such as colors, locations, and truth values;



- (4) A partial ordering of sorts by level, such that data sorts are lower than sign sorts, and such that there is a unique sort of maximal level, called top sort;
- (5) A set  $C_n$  of level  $n$  constructors used to build level  $n$  signs from signs at level  $n$  or less, and written  $r: s_1 \dots s_k d_1 \rightarrow s$ , indicating that its  $i$ th argument must have sort  $s_i$ , its  $j$ th parameter data sort  $d_j$ , and its result sort is  $s$ ; constants  $c: \rightarrow s$  are also allowed;
- (6) A priority (partial) ordering on each  $C_n$ ;
- (7) Some relations and functions on signs; and
- (8) A set of sentences (in the sense of logic), called axioms, that constrain the possible signs.

The following is a simple example of a semiotic theory for a space of structured objects, namely books. Because our purpose is to illustrate the concept of semiotic space rather than to precisely describe books, the theory is greatly simplified. The data theory for books just provides titles and page numbers:

```
dth BOOKDATA is
  pr QID *(sort Id to Title).
  Pr NAT *(sort Nat to PageNo).
End
```

Here ‘QID’ and ‘NAT’ are built-in modules for (quoted) identifiers and natural numbers, respectively, and the notation “ $M \text{ }*(\text{sort } A \text{ to } B)$ ” calls for renaming a sort  $A$  in module  $M$  to become  $B$  in a new module. Now we define books as lists of Chapters, where a ‘Chapter’ is a pair of a ‘Head’ and some ‘Content’, and a ‘Head’ is a pair of a ‘Title’ and a ‘PageNo’. It is convenient to use a ‘subsort’ declaration to say that Chapters are Books (i.e., one chapter books are allowed).

```
Th BOOK is sorts Book Chapter Head Content .
  subsort Chapter < Book .
  pr BOOKDATA .
  op <_,_> : Title PageNo -> Head .
  op <_,_> : Head Content -> Chapter .
  op _ _ : Chapter Book -> Book .
end
```

Here **Book** is the top sort, **Chapter** is the secondary sort, **Head** and **Content** are tertiary sorts, and **Title** and **PageNo** are fourth level sorts. The constructors are the two pairing operations, both of the form  $\langle \_, \_ \rangle$ , and the list forming operation,  $\_ \_$ ; the OBJ languages use the underbar character as a place holder, showing where the arguments of an operation should go. For example,  $\langle \text{'Semantics'}, 6 \rangle$  is a valid term of sort **Head**. Among the constituents of **Head**, **Title** has priority over **PageNo**, and among those for **Chapter**, **Head** has priority over **Content** (see (Goguen, 1999) for an explanation of this perhaps strange assertion). **Content** is deliberately left unspecified here; it could be anything in models.

The grammar for Labov narratives in **Section 2.2** of this chapter can also be described as a semiotic system. The top level sort is of course  $\langle \text{Narr} \rangle$ ; the second level sorts are  $\langle \text{Cls} \rangle$ ,  $\langle \text{Eval} \rangle$ ,  $\langle \text{Open} \rangle$ , and  $\langle \text{Coda} \rangle$ , while  $\langle \text{Ornt} \rangle$  and  $\langle \text{Abs} \rangle$  are third level sorts. It is a good exercise to write out further formal details of this semiotic system, including its priorities and constructors. It should not be thought that this semiotic system will be blended with some 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 structure (scene/shot/etc.) 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, it can be helpful to choose names that can help the reader’s intuition.

Using theories for semiotic systems is better than concrete model-based approaches 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 constrain the allowable models. For example, in formalizing the representation that produces indices from books, we might want to impose axioms on the target space of indices, e.g., requiring that indexed items must be phrases of 1, 2 or 3 words, and that the page total for indexed phrases must be not more than 2% of a book’s page total. Here is what the first of these axioms might look like in BOBJ syntax:

```
var I : IndexItem .
eq # I < 4 = true .
```

where the first line declares `I` to be a variable of sort `IndexItem`, where `# I` denotes the length of `I`, and where `<` is the usual “less than” relation on numbers, given as a `Bool` valued function.

There is a subtle point about how truth values are handled. We have written equations that explicitly give the value `true` for a relation on some constants (and

there can also be variables in such equations, so that they apply to a potentially infinite set of constants). If the given equations do not determine a truth value for some arguments, then the relation may be true for those arguments in some models and false in others. However, it is also possible, if that truth value is used in other equations, to set things up so that undefined values are treated as false; this corresponds to what is called the **closed world assumption**. See (Goguen, 1998) for some further details.

The reader may have noticed that the structures described by semiotic spaces, like those of 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 by Joseph Goguen in (Goguen, 2003). 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,” and being able to generate more striking and unusual blends by identifying sorts that are far apart, and also the discussion of type casting in **Subsection 2.3.4.1**). Another difference from cognitive linguistics is that the algebraic semiotics formulation of blending does not first construct a minimal image in the blend space, and then “project” it back to the target space, but instead, the entire result is built in the blend space, which is useful for the generative narrative applications discussed later in **Chapter 4**.

### 2.3.3.2 Semiotic Morphisms

Mappings between structures became increasingly important in twentieth century mathematics and its applications; examples include linear transformations (and their representations as matrices), continuous maps of spaces, differentiable and analytic functions, group homomorphisms, and much more. (Goguen & Harrell, 2007a) Mappings between sign systems are only now appearing in semiotics, as uniform representations for signs in a source space by signs in a target space; user interface design is an important application area for such mappings. (Goguen, 1998) Since sign systems are formalized sign as algebraic theories with additional structure, we should formalize semiotic morphisms as theory morphisms; however, these must be partial, because in general, not all of the sorts, constructors, etc. are preserved in the intended applications. For example, the semiotic morphism from the conceptual space for “king” into the blended space for the metaphor “The sun is a king” (most likely) omits the throne, court jester, queen, castle, and so on. In addition to the formal structure of algebraic theories, semiotic morphisms should also (partially) preserve the priorities and levels of the source space. The extent to which a morphism preserves the features of semiotic theories helps to determine its quality, as discussed in detail in papers including (Goguen, 1998, 1999, 2003; Goguen & Harrell, 2004).

Formally, a semiotic morphism ‘ $M: S_1 \rightarrow S_2$ ’, from sign system ‘ $S_1$ ’ to sign system ‘ $S_2$ ’ is defined as the following partial functions (Goguen, 1998):

- (1) Sorts of  $S_1 \rightarrow$  sorts of  $S_2$
- (2) Constructors of  $S_1 \rightarrow$  constructors of  $S_2$
- (3) Predicates and functions of  $S_1 \rightarrow$  predicates and functions of  $S_2$  such that:

- (1) if  $s \leq s'$  then  $M(s) \leq M(s')$
- (2) if  $c: s_1 \dots s_k$  is a constructor (or function) of  $S_1$ , then (if defined)  $M(c): M(s_1) \dots M(s_k) \rightarrow M(s)$  is a constructor (or function) of  $S_2$
- (3) if  $p: s_1 \dots s_k$  is a predicate of  $S_1$ , then (if defined)  $M(p): M(s_1) \dots M(s_k) \rightarrow M(s)$  is a predicate of  $S_2$
- (4)  $M$  is the identity on all sorts and operations for data in  $S_1$

The direction of a semiotic morphism can be interpreted the direction that models are mapped. Thus, if  $B$  is a semiotic system for books and  $T$  one for tables of contents, then books (which are models of  $B$ ) are mapped to their tables of contents, which are models of  $T$ . However, this map on models is determined by, and is dual to, the theory inclusion  $T \rightarrow B$ , which reflects the fact that the structure of tables of contents is a substructure of that of books. This is written in BOBJ as follows:

```

th TOC is sorts Toc Chapter Head .
  subsort Head < Toc .
  pr BOOKDATA .
  op <_,_> : Title pageNo -> Head .
  op _ _ : Head Toc -> Toc .
end

```

What are called morphisms in algebraic semiotics are called “views” in OBJ. Here is the inclusion view from the table of contents theory into the book theory:

```

view V from TOC to BOOK is
  sort Toc to Book .
end

```

where defaults (as discussed in the next subsection) are used to omit several obvious mappings, and where the sorts in operations in the shared subspace `BOOKDATA` are also preserved by a default convention. In addition, it is required that all data sorts and operations are preserved by morphisms, so that there are really two reasons for `Medium`, `land`, and `water` to be preserved.

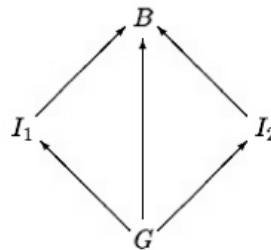
### 2.3.4 Algebraic Semiotic Account of Blending

Algebraic semiotics can be used to describe the models manipulated and theorized about by conceptual blending theorists in a mathematically precise way. It can be used to represent Fauconnier's mental spaces, but it also features additional structure originally useful for describing computational artifacts (user-interfaces) and broadly applicable to semiotic concerns in general. Such models compress complex cognitive processes to scales which humanly tractable and are intended to reveal cognitive regularities; certainly the full range of human cognition is omitted in service of developing a model to focus empirical investigation. Providing a formal notation for such models does not entail a belief that meaning and human thought are formal in nature. The main desideratum in using such a formal model is ensuring that it has the expressive capability for accurately and appropriately describing the structures theorized about by cognitive scientists. A formal notation may even have the advantage of allowing researchers to, as Joseph Goguen phrases it, make precise statements about imprecise concepts as opposed to imprecise statements about imprecise concepts! Secondly, it allows for implementation and algorithmic experimentation.

In order to be implemented computationally, aspects of blending need to be given a precise notation. Formalizing some notions from cognitive linguistics does not entail believing that formal structure alone can account for imaginative thought. On the contrary, it is hoped that a precise notation can aid in clear thinking about dynamic and contingent processes. The modest claim made here is that precise

notation can aid in empirical testing and clarity of discussion about these theories, and in implementing these ideas for artistic (and other) pursuits. A central assertion of this dissertation is that a computational model of some aspects of blending theory can be used to build models useful for concept generation within computational narratives. Algebraic semiotics and the Alloy algorithm (described in the next chapter) can be used for these purposes.

The simplest form of blend in algebraic semiotics is shown in **Figure 2.12**<sup>16</sup>, where  $I_1$  and  $I_2$  are called **input spaces**, and  $G$  is called a **generic space**.



**Figure 2.12: A Blend Diagram**

Elements of the generic space may be called generic sorts, generic constants, and generic relations.  $I_1$ ,  $I_2$ , and  $G$  together with the two morphisms  $I_1 \rightarrow G$  and  $I_2 \rightarrow G$  comprise an **input diagram**. Given an input diagram, a **blendoid** is 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 2.12** **weakly 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

---

<sup>16</sup> The form of this diagram is “upside down” from that used by Fauconnier and Turner, in that our arrows go up, with the generic  $G$  on the bottom, and the blend  $B$  on the 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 map  $G \rightarrow B$ .



morphisms are defined on it<sup>17</sup>. In general, all four spaces are 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. Criteria for this purpose will be called **optimality principles**, and a **blend** is then defined to be a blendoid that satisfies some given optimality principles to a significant degree. The blending algorithm in **Chapter 3** utilizes **structural** optimality principles, which are based only on the structure of blends, rather than their meaning. These include the degrees of commutativity, preservation of constants and axioms, and of type casting (these terms are defined in **Subsection 2.3.4.1**); there can still be many blends in this sense, as the example in the next subsection will make very clear. A more precise, but mathematically difficult and narrow, definition can be found in Appendix B of (Goguen, 1998), and as discussed in **Section 2.2** of this chapter, (Fauconnier & Turner, 2002) provides many powerful, but only humanly (as opposed to computationally), determinable principles.

Any diagram having the shape of **Figure 2.12** is referred to as a **diamond diagram**, and the composition of the two morphisms on its left is its **left morphism**, the composition of the two morphism on its right is its **right morphism**, to the middle upward morphism is its center morphism, to the triangle on its left is its **left triangle**, and the triangle on its right is its **right triangle**. A triangle is said to **commute** for a sort, constant, or relation iff that sort, constant, or relation is mapped to the same sort, constant, or relation in the blend space by the center morphism as by the left (or right)

---

<sup>17</sup> Strict commutativity, usually called just commutativity, means that the compositions are strictly equal, i.e., one morphism is defined on an element iff the other is, and then they are equal.

morphism (depending on the triangle), and similarly for relations.

### 2.3.4.1 An Example of Blending Using Algebraic Semiotics

This subsection considers blends of the concepts “boat” and “house,” using the notation of BOBJ, and following lines begun in (Goguen, 1998). First are theories for the conceptual spaces involved, noting that the generic theory GENERIC has already been given in **Subsection 2.3.2**. Here are theories for the two input spaces:

```

th HOUSE is sorts Person Object.
  Pr MEDIUM.
  Op resident : -> Person.
  Op house    : -> Object.
  Op live-in  : Person Object -> Bool.
  Op on       : Object Medium -> Bool.
  Eq live-in(resident, house) = true.
  Eq on(house, land) = true.
Endth

th BOAT is sorts Person Object .
  pr MEDIUM.
  Op passenger : -> Person.
  Op boat      : -> Object.
  Op ride      : Person Object -> Bool.
  Op on        : Object Medium -> Bool.
  Eq ride(passenger, boat) = true.
  Eq on(boat, water) = true.
Endth

```

The first mentioned sort of an OBJ module is called its **principal sort**; it plays a special role in default views, as discussed below. It is convenient to identify this sort with the top sort of a semiotic theory; a similar convention can be enlisted to express the rest of the level ordering, by listing the secondary sorts using a separate `sort` declaration, and then the tertiary sorts with another, etc. There does not seem to be

any convenient way to express priorities on constructors in OBJ, but that is less relevant in this example, since the only constructors are constants.

Here are the two morphisms from the generic space for blending the house and boat conceptual spaces:

```
view M1 from GENERIC to HOUSE is
  op person to resident.
  Op object to house.
  Op use to live-in.
endv

view M2 from GENERIC to BOAT is
  op person to passenger.
  Op object to boat.
  Op use to ride.
Endv
```

OBJ allows views to be simplified by omitting certain “obvious” mappings, including those of the forms “sort A to A” and “op a to a.” Using this convention, OBJ is able to deduce that M1 and M2 both map the sort `Object` in `GENERIC` to the sort `Object` in their target theories. Similarly for the operation `on`, which is not mentioned in either view. Moreover, everything in `MEDIUM` is automatically preserved, because `MEDIUM` is a shared subtheory. Principal sorts are also preserved unless this is explicitly overridden. The resulting abbreviated views are called **default views**. (Goguen et al., 2000) In fact, the views M1 and M2 above could be replaced by totally default views, with empty bodies. Default views make it tricky in OBJ to represent cases where there is something with the same name in both the source and the target space of a morphism, but we do not want this thing to be preserved. One solution is to define a subtheory of the source theory that does not include the unpreserved things, and then define the view from that subtheory. Fortunately, this

issue does not arise in the present example.

We say that constants  $c_1$  in input space  $I_1$  and  $c_2$  in  $I_2$  are **merged** or **identified** in a blend iff they map to the same constant in the blend space. In such cases, there is an ambiguity about what the merged constant should be called. A convenient convention for the case where  $c_1$  and  $c_2$  have the same sort is simply to merge the two names with a slash between them, as in `resident/passenger`; if the two names are the same, then we simply use the common name. The case where the sorts are different is more difficult and is discussed later, but even when the sorts are the same, one may still wonder which sort name should appear first in the merged name. Our view is that it does not matter, because the formal name is not a **structural** property; however, we try to choose an order that does not conflict with conventional usage. The same considerations apply to the names chosen for the blend space itself, and to merged relations.

The following are the theory and three morphisms (three because the center morphism must be included) for our first blend of the two given spaces and three given morphisms. Note that the equation for `on(house, land)` has not been preserved.

```

Th HOUSEBOAT is sorts Person Object.
  Pr MEDIUM.
  Op resident/passenger : -> Person.
  Op house/boat : -> Object.
  Op live-in/ride : Person Object -> Bool.
  Op on : Object Medium -> Bool.
    Eq live-in/ride(resident/passenger,
      house/boat) = true.
    Eq on(house/boat, water) = true.
Endth

view M3 from HOUSE to HOUSEBOAT is

```

```

        op resident to resident/passenger.
        Op house to house/boat.
        Op live-in to live-in/ride.
    Endv

view M4 from BOAT to HOUSEBOAT is
    op passenger to resident/passenger.
    Op boat to house/boat.
    Op ride to live-in/ride.
Endv

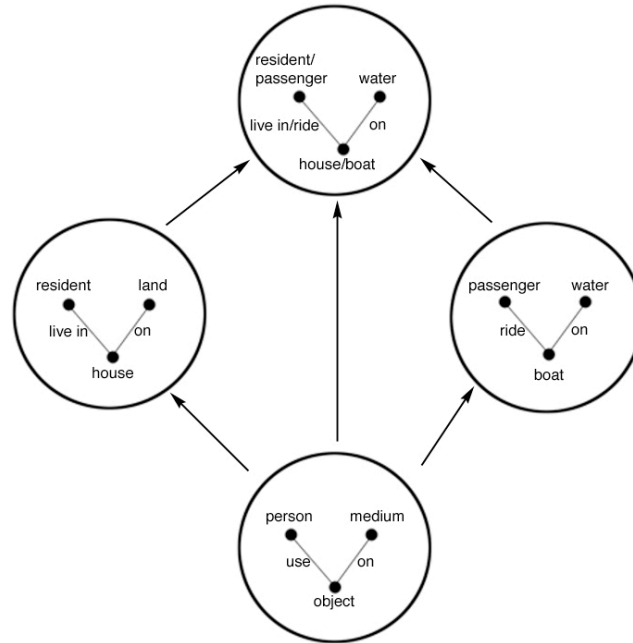
view C from GENERIC to HOUSEBOAT is
    op person to resident/passenger.
    Op object to house/boat.
    Op use to live-in/ride.
Endv

```

For this blend, the two triangles commute for all three sorts in the generic space (and the sort `Medium` is also preserved, because it is a shared data sort); similarly, the two generic constants `object` and `person` are preserved. Thus we have commutativity in the strong sense 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.

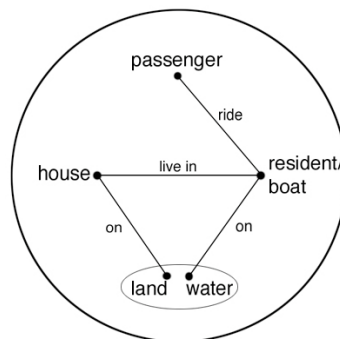
The reader may have noticed that a significant piece of information is lost in this blend, that `live-in` and `ride` each hold separately for `resident/passenger` and `boat/house`. However, in OBJ, by omitting `use` from the base space, we can avoid identifying the two relations, so that they can hold

separately in the blend space, if that is desired.



**Figure 2.13: The Houseboat Blend**

In contrast to the “houseboat” blend depicted in **Figure 2.13**, now let us consider a second blend, shown in **Figure 2.14** below, which in English is called a “boathouse.”



**Figure 2.14: The Boathouse Blend**

In it, the boat ends up in the house. Here are its space and three morphisms:

```
th BOATHOUSE is sorts Person Object.
  Pr MEDIUM.
  Op passenger : -> Person.
  Op resident/boat : -> Object.
  Op house : -> Object.
  Op live-in : Person Object -> Bool.
```

```

    Op ride : Person Object -> Bool.
    Op on : Object Medium -> Bool.
    Eq ride(passenger, resident/boat) = true.
    Eq live-in (r:Universal>Person
      (resident/boat), house) = true.
    Eq on(house, land) = true.
    Eq on(resident/boat, water) = true.
  Endth

view M3 from HOUSE to BOATHOUSE is
  op resident to
  r:Universal>Person(resident/boat).
Endv

view M4 from BOAT to BOATHOUSE is
  op boat to
  r:Universal>Person(resident/boat).
Endv

view C from GENERIC to BOATHOUSE is
  op person to resident/boat.
  Op object to house.
  Op use to live-in.
Endv

```

Notice that in the morphism M3, mapping `resident` to `boat` would not type check, whereas it does with `r:Universal>Person(boat)`, where `r:Universal>Person(boat)` is a **retract**, an instance of a family of built-in OBJ functions that change the sort of the term in its argument from that the first sort after the “:” to become the sort after the “>” (Goguen & Malcolm, 1996) in this case `boat`, `Universal` and `Person`, respectively, where `Universal` is a built-in OBJ sort such that every other sort is a subsort of it. Without this modification, which is called “type casting” in programming language theory, it would be impossible for the boat to live in the boathouse. This type change is a kind of metaphor, called **personification** in literary theory, in which an object (here, a boat) is considered a person.

For this blend, neither triangle commutes, because the generic element `object` is mapped to `boat` in the blend by the right morphism (the composition of morphisms `M2` and `M4`), and to `house` by the left morphism (the composition of `M1` with `M3`), but is not mapped to `boat/house` in the blend. Similarly, since `M4` preserves `passenger`, the central morphism cannot preserve the generic element `person`, and the same goes for the generic `use` operation. On the other hand, the generic relation `on` goes to the same place under all three maps.

There is a third blend which is similar to (in fact, symmetrical with) the above `BOATHOUSE` blend, in which a `house/passenger` ends up riding in the boat. (As noted in (Goguen, 1998), there are real examples of this blend, where a boat is used to transport prefabricated houses across some body of water, e.g., for a new housing development on a nearby island.) This blend has similar (in fact, symmetrical) commutativity properties to those of the `BOATHOUSE` blend. It is left as an exercise for the reader to write OBJ code for this blend.

There is a fourth blend, the meaning of which is less familiar than the first three, but the preservation and commutativity properties of which are very good, so that it is a very pure blend of its input, even though its physical existence is doubtful. This is an amphibious RV (recreational vehicle), a vehicle that you can live in, and that you can ride in on land and on water. Here all aspects of `HOUSE` and `BOAT` are realized, with no surprising transpositions, such as a house riding a boat. It is an interesting exercise to write OBJ code for this blend.

There is a fifth blend, the meaning of which is even less familiar. It is a livable boat for transporting livable boats. It was discovered by an early prototype of the



Alloy blending algorithm, and is rather counter-intuitive for humans. In this blend the axiom `ride(passenger/house/boat, passenger/house/boat)` is generated along with the axiom `live-in(resident, passenger/house/boat)`. The most recent version of Alloy implements optimality constraints for blending that preclude this counter-intuitive blend from being generated.

Finally, a sixth blend gives a boat used on land for a house; it arises from omitting the axioms that require a house/boat to be on water and a passenger to ride a house/boat.

It is encouraging that our intuitive sense of the relative purity of these blends, and the degree to which they seem “boat-like” and “house-like,” corresponds to principles such as degree of commutativity, and preservation of axioms in input spaces. This suggests (in a very preliminary way) that the principles for measuring the quality of blends in (Goguen, 1998) are reasonable, and reinforces the hope for good heuristics for finding high quality blends based only on structural properties of the conceptual space network.

The text of **Subsection 2.1.2** and **Section 2.3** from this chapter, in part, contains material that is a reprint of, or has been submitted for publication in: Goguen, J., & Harrell, D. F. (2007); Foundations for active multimedia narrative: Semiotic spaces and structural blending (in preparation); *Interaction Studies*. The dissertation author was a co-author of this paper.

## Chapter 3 Alloy

The Alloy system for conceptual blending is the result of research on a computational approach conceptual blending, including implementation techniques, experiments, and analysis of the results. This chapter describes the goals of the Alloy system and detailed descriptions each of its components using a comprehensive and detailed example. The primary goal of the research described is support for developing interactive and generative narratives. As a secondary goal, the work provides a formalization and framework for algorithmic experimentation for cognitive scientists. The primary elements of the system are (1) an algebraic semiotics based formalization (based on theory described in (Goguen, 1998)) of conceptual space networks (including conceptual spaces and mappings between them), (2) a data structure for representing these conceptual space networks, and (3) an algorithm for conceptual blending.

The Alloy blending algorithm provides support for development of interactive

and generative narratives in two ways. It provides a theoretical framework for how partial combinations of media elements can be used as a foundation for generativity based upon semantic content of those media elements. “Semantic content” here means that new content is generated based upon the structural relationships of formal axioms and how this structural relationship captures aspects of an author’s intended expressive meanings. Secondly, Alloy provides technology that can be used for generating such content. It allows for the blending of conceptual spaces, represented as algebraic semiotic spaces, and these blended spaces can be the basis for generated content in computational narratives. Structural optimality principles can be used to stylistically constrain the results (e.g., generating either common sense blends or exotic blends that map elements in surprising ways). Additionally, user input can be used to guide the selection of conceptual spaces to be integrated. Finally, the results can be used to drive selection and combination of media elements in text or other forms (such as images or audio files). **Chapter 4** describes the use of Alloy for generating text as used in computational poetry with the GRIOT system for implementing computational narratives.

Regarding the cognitive science contributions of the Alloy algorithm, the formalization used in Alloy is best thought of as a language to precisely represent some aspects of conceptual spaces as described in (Fauconnier & Turner, 2002), and semiotic spaces as described in (Goguen, 1998). A key aspect of conceptual blending theory, and cognitive linguistics in general, is that human imagination and language are not thought to be fundamentally formal or computational. (Fauconnier & Turner, 2002) Still, cognitive linguists express structures such as metaphorical mappings or

conceptual space networks in a variety of ways such as lists of mappings or conceptual blending diagrams. The work here attempts to show sensitivity to exactly how formalization can be useful to conceptual blending theory and postulates that the various forms of utility are: precise descriptions, algorithmic elaboration of such descriptions when it would be cumbersome for a human to do so, capturing structural regularities that seem to occur in common human thought processes, and applying cognitive science ideas to engineering tasks. The experiments described here are not intended as cognitive modeling; the conceptual spaces do not represent every aspect of concepts, but only those that are necessary to make the blends agree with our intuition. This is consistent with the definition of conceptual spaces as containing the minimal relevant information. (Grady et al., 1999) The motivation for these experiments is to improve the algorithm, the theory, and our intuitions.

These two goals are finally unified in an interesting way. Though no claim is being made that the blends that Alloy produces represent, or are generated in the same way as, real human concepts, the work may suggest some aspects of human conceptual blending. Secondly, when applied in the context of text (poetry) generation, though no claim is being made that the computer autonomously generates poetic concepts and language (or that such a goal is desirable), instead the work produces interesting results that trigger human conceptual blending processes in readers in somewhat predictable patterns. When mapped to output in natural language the blends generated in Alloy represent a type of linguistic anchor (akin to Hutchin's material anchors, but differing in specific ways per his objection to Fauconnier and Turner's broader use of the term). (Hutchins, 2005) In Fauconnier and Turner's

terminology this is an example of the “Eliza effect,” but an instance of the effect that is guided according to a set of authorial constraints intended to convey to users/readers to a restricted set of interpretations. Additionally, though Alloy does not involve elaboration (a central aspect to conceptual blending according to Fauconnier and Turner’s theory), the “linguistic anchors” produced can recruit additional content which also can trigger predictable patterns of elaboration by human readers. Of course, exploiting predictable patterns of interpretation and imagination is the natural realm of artists, the hope here is that this work can also provide a language, descriptions, and implementations of some specific ways that this process occurs. Such elements of this research thus have implications for expressive work in computing.

### 3.1 Knowledge Representation for Alloy

Conceptual space networks in Alloy are a faithful LISP implementation of algebraic semiotic conceptual space networks that is geared toward efficient algorithmic processing. The central elements are conceptual spaces, which have the following format (in a modified BNF notation, with italics denoting descriptions of atomic elements):

```

<conceptual-space> ::= “(conceptual space” conceptual-space-
    name <sorts-list> <data-sorts-list> <constants-list>
    <relations-list> <axioms-list> value “)”
<sorts-list> ::= “(“ <sort> {<sort>} “)”
<sort> ::= “(sort” sort-name conceptual-space-name value “)”
<data-sorts-list> ::= “(“ <data-sort> {<data-sort>} “)”

```

```

<data-sort> ::= "(data-sort" data-sort-name conceptual-space-name value ")"
<constants-list> ::= "(" <constant> {<constant>} ")"
<constant> ::= "(constant" constant name sort-name conceptual-space-name value ")"
<relations-list> ::= "(" <relation> {<relation>} ")"
<relation> ::= "(relation" relation-name <sort-name-list> conceptual-space-name value ")"
<sort-name-list> ::= sort-name {sort-name}
<axioms-list> ::= "(" <axiom> {<axiom>} ")"
<axiom> ::= "(axiom" axiom-name <constants-list> conceptual-space-name value ")"

```

As an example of this format, the “house” conceptual spaces discussed earlier is written in the LISP format described above as:

```

'(conceptual-space house
  ((sort person input-1 0)
   (sort object input-1 0))
  ((data-sort land input-1 0))
  ((constant house object input-1 0)
   (constant resident person input-1 0))
  ((relation live-in (person object) input-1
    0)
   (relation on (object datasort^medium)
    input-1 0))
  ((axiom live-in ((constant resident person
    input-1 0)
    (constant house object
    input-1 0)) input-1 0)
   (axiom on1 ((constant house object input-
    1 0)
    (constant land
    datasort^medium input-1 0))
    input-1 0)
  0)

```

The most obvious difference between this format and the BOBJ code earlier is the presence of ‘0’ values. These are used in the process of blending in order to assign measures of optimality of particular blends, ‘0’s are merely initial values. Also, the ‘input-1’ atom that appears multiple times is the name of the conceptual space, in

this case the name indicates that the house space is one input to the blending algorithm. The ‘*datasort<sup>^</sup>*’ atom is used to differentiate data-sorts from other sorts.

Semiotic morphisms are represented similarly to conceptual spaces with according to the following format:

```

<semiotic-morphism> ::= "(morphism" conceptual-space-name-1
conceptual-space-name-2 <sort-maps-list> <constant-maps-list>
<relation-maps-list> value ")"
<sort-maps-list> ::= <sort-map> {<sort-map>}
<sort-map> ::= "(mapping" <sort> <sort> ")"
<constant-maps-list> ::= <constant-map> {<constant-map>}
<constant-map> ::= "(mapping" <constant> <constant> ")"
<relation-maps-list> ::= <relation-map> {<relation-map>}
<relation-map> ::= "(mapping" <relation> <relation> ")"

```

Thus, semiotic morphism are represented in a very straightforward manner as pairs of elements from two conceptual spaces.

Finally, an input diagram consists of three conceptual spaces and two semiotic morphisms. The format for an input diagram follows:

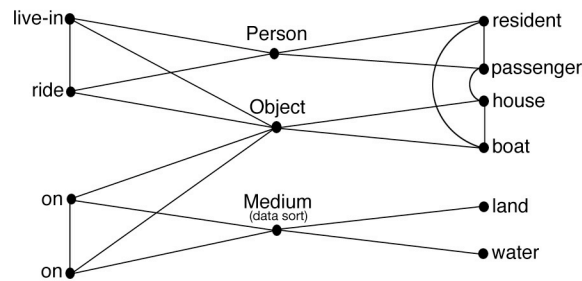
```

<input-diagram> ::= "(input-diagram" diagram-name <generic-
space> <conceptual-space> <conceptual-space> <semiotic-
morphism> <semiotic-morphism> ")"
<generic-space> ::= "(conceptual space" generic-space <sorts-
list> <data-sorts-list> <constants-list> <relations-list> <axioms-
list> value ")"

```

The only feature of note here is that the first conceptual space in the input diagram must be the generic space.

The blending algorithm convert input diagrams into a graph data structures called an input-graphs. An example of an input-graph for the “house” “boat” blend is depicted in **Figure 3.1** below. (Goguen & Harrell, 2007a)



**Figure 3.1: The House/Boat Input Graph**

The input-graph data structure has four kinds of node, for relations, data sorts, non-data sorts, and constants. These graphs have three columns, with relation nodes on the left, sort nodes in the center, and constant nodes on the right. The sort nodes represent sorts after any required non-data sort identifications have been made. An edge from a relation node to a sort node means that the relation is defined on that sort. An edge from a sort node to a constant node indicates that the constant has that sort. An edge from one constant or relation node to another indicates that the connected entities are from different input spaces. This representation makes it clear which axioms could potentially give values to a relation, by following edges from left to right, from relations to constants. It also allows visualization of type casting, as simply changing the sort node to which a constant node connects. Two relations can be identified iff they connect to the same sorts, and to each other.

### 3.2 The Alloy Blending Algorithm

This section describes the Alloy blending algorithm, its implementation, some experiments with it, and then discusses the results. These experiments are not



intended to produce comprehensive models of the human mind. Instead, the motivation is to improve the algorithm, the theory, and our intuitions about blending. When results differ from our intuitions (as should be expected in the early stages of such a project), it could be due to a bug in the algorithm, in the theory, or in our understanding. Thus, these experiments initiate a highly beneficial cycle of mutual improvements. A running program is also valuable, because even for relatively simple inputs, the number of blendoids is so large that it is difficult for a human to discover them all. In the houseboat example, the algorithm computes 48 primary blendoids (in which every possible axiom is preserved – this output can be found in **Appendix A**), and 736 if it also computes those that fail to preserve some axioms. An important conclusion is that efficient techniques for computing high quality blends are necessary for the theory to be useful for content generation and analysis.

The blending algorithm is programmed in LISP, and given an input diagram, it either computes one good blend, or else all blendoids (in the sense of **Subsection 2.3.4** over that diagram). The algorithm uses the graph data structure discussed above and depicted in **Figure 3.1** to efficiently compute all possible blends from a given input diagram.

We can avoid a great deal of computation by identifying relations and constants independently. Hence, we can generate the tree of ways to identify relations, and then combine it the tree of ways to identify constants. In the relation identification tree, the branches diverging from nodes arise from whether or not particular elements are identified. The constant identification tree is similar, except when constants have different sorts.

The algorithm is a depth first traversal over the binary trees describing the ways to identify relations and the ways to identify constants. Different elements from the same input space are never identified. Data sorts and data sort constants are never identified. Non-data sorts are identified iff required by being mapped to from a common sort in the base space. All 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. The blended constant is then cast to the appropriate sort in each axiom. There is a bijection between the blendoids that can be computed in the two cases, since type casting of the blended constant is necessary in one case iff it is unnecessary in the other. The number of primary blendoids is the number of leaves of the relation subtree multiplied by the number of leaves of the constant subtree. After all possible identifications have been computed, the algorithm concludes by enumerating the power set of axioms. This is necessary due to examples like sixth blend described in **Subsection 2.3.4.1** above, although it greatly increases the number of blendoids, which is  $2^A P$ , where  $P$  is the number of primary blendoids, and  $A$  is the number of axioms.

Updating axioms after some relations and constants are identified may result in duplicate axioms. For example, under the mappings:

```
live-in → live-in/ride
ride → live-in/ride
house → house/boat
boat → house/boat
resident → resident/passenger
passenger → resident/passenger
```

both axioms:

live-in(resident, house) = true  
ride (passenger, boat) = true

get updated to:

live-in/ride(resident/passenger, house/boat) = true

In such cases, the duplicate axioms are removed.

Experiments found that relatively few blendoids correspond to reasonable human concepts, thus motivating a search optimality principles, as discussed in **Subsection 2.3.4**, and carried forward in **Subsection 2.3.4.1**, with the degrees to which various properties are preserved. The current algorithm uses maximizing the degree of commutativity, minimizing the number of typecasts, and maximizing the number of preserved axioms and constants as its only optimality principles. The degree of commutativity is measured on a numerical scale as the ratio of actual to possible occurrences of commuting elements. The amount of type casting is also measured on a numerical scale, and is important for constants because the more of these a blendoid has, the more constants get unnatural sorts. Regarding axiom and constant preservation, each is also measured on a numerical scale, again as a ratio of actual to possible occurrences. Moreover, in the algorithm the optimality values are given weights, so that a single number is computed as a weighted sum. The final output consists of a list of output diagrams ordered by optimality of their blended spaces using the quicksort algorithm.

Thresholds can be set such that blendoids below threshold need not be considered. In the houseboat example, low values would be given to blendoids such

as a **passenger/house** that rides a **resident/boat** on water, and to a **passenger/boat** and **resident/boat** that do nothing. The highest value would go to the blend in which a **house/boat** is **ride/live-in** (please excuse the tense here) by a **resident/passenger**. As a default, partial ordering relations could be used, as in (Goguen, 1998).

In the case of generating a metaphorical blend, one input space is chosen as target, and attributes from the other input space are blocked, which can greatly reduce the search for good blends. (Grady et al., 1999) Another potentially useful principle is that metaphors do not usually call attention to incongruities among connected entities from the input spaces. In the longer term, tuning of optimality principles may be useful as a way to specify the type and quality of specific blends generated for purposes such as designating the style of output in computational narratives or other applications. (Goguen & Harrell, 2007b)

### **3.3 Summary Discussion of Alloy**

The Alloy system, intended as one of the results presented by this dissertation, implements a formal model of the constitutive principles of conceptual blending theory along with a structural subset of optimality principles. The algebraic semiotics formalization upon which it is based can also be used to precisely describe emergent structure in blended spaces. The goal motivating Alloy, however, is not cognitive modeling. Alloy is intended as an empirical tool for researchers to model their subjectively constructed conceptual integration networks and to experiment with the

limitations of partial composition and optimality constraints within the conceptual blending framework. This differs from GOF AI approaches because the subjectivity inherent in selecting an input diagram for integration or a knowledge based for elaboration is explicit and seen as problematic.

The Alloy algorithm is not an attempt to comprehensively capture the important aspects of conceptual blending theory. It is an attempt to capture some of the structural optimality principles that are amenable to formalization, to clarify aspects of conceptual blending theory that are *not* amenable to formalization and implementation, and to implement the partial compositionality of conceptual spaces using the rich mathematical framework of algebraic semiotics. Alloy is motivated by the realization of the importance of embodiment and context in cognition, and the inability of formal computational models to account for this. The conceptual blending systems described briefly above are explicit attempts to comprehensively implement the important aspects of blending theory, which distinguishes them from the intendedly modest and partial goals of Alloy. A goal of Alloy is to distinguish aspects of blending that are amenable to formalization and computation from those which are not. It is admittedly a difficult challenge to cleanly divide computational and formal issues from humanly subjective issues. It is not even always a useful or valid distinction. However, the commitments of the cognitive linguistics enterprise indicate that meaning is not formal and computational. The belief here is that the Alloy system represents a computational point of view that sees GOF AI and connectionist approaches to cognitive modeling as not being grounded in the philosophical

commitments of the cognitive linguistics enterprise, which may not be the case with some other computational analogy, metaphor, and blending systems.

The text of **Section 3.2** of this chapter, in part, contains material that is a reprint of, or has been submitted for publication in: Goguen, J., & Harrell, D. F. (2007); Foundations for active multimedia narrative: Semiotic spaces and structural blending (in preparation); *Interaction Studies*. The dissertation author was a co-author of this paper.

## Chapter 4 GRIOT

The GRIOT system is a platform for recombining and restructuring media elements according to principles of conceptual and structural blending. It is especially intended for recombining media elements according to a narrative structure, and has been used for implementing text-based and multimedia narrative poetry based (1) upon generating content using Alloy and (2) recombining content according to author defined narrative structures. This chapter describes the functionality, architecture, and application of GRIOT. **Section 4.1** presents a formal description of GRIOT's functionality, focusing on its use in text-based work, but with implications for multimedia. **Section 4.2** describes the GRIOT architecture, algorithm, and its application for authoring computational poetry, including the Event Structure Machine, a type of automaton GRIOT uses for structuring media elements or events into a narrative (or other) discourse structure. **Section 4.3** describes the four different levels of interaction that a human can have with GRIOT. **Section 4.4** describes

several examples of computational (“polymorphic”) poetry in depth and uses these examples to describe important theoretical aspects of this work.

## 4.1 Formalization of GRIOT’s Functionality: Narrative Construction as Structural Blending

This section, appearing first in (Goguen & Harrell, 2007b), develops an approach to text generation inspired by cognitive grammar and based on structural blending; it is illustrated using the Labov narrative syntax presented in **Subsection 2.1.2**. This material does not apply to the algorithm of **Chapter 3**, 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 the conversion is done) is:

$$\langle \text{Narr} \rangle \rightarrow \langle \text{Open} \rangle \langle \text{Cls} \rangle \langle \text{Eval} \rangle \langle \text{Coda} \rangle$$

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  $\langle \text{Narr} \rangle$  and saliency  $1^{18}$ ,

$$[\langle \text{Open} \rangle \langle \text{Cls} \rangle \langle \text{Eval} \rangle \langle \text{Coda} \rangle :: \langle \text{Narr} \rangle, 1]$$

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

---

<sup>18</sup> For such rules, our saliency is similar to entrenchment in the sense of (Langacker, 1999)); we assume saliency values are in the unit interval  $0 \leq v \leq 1$ , and that they follow the fuzzy logic of (Goguen, 1969).



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 GRIOT system, 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 **Chapter 3**, 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

$$[X : s'] \ \& \ [t(X) :: s, v] \ \& \ [t' :: s', v'] \Rightarrow [t(t') :: s, vv']$$

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  $\langle \text{Narr} \rangle$  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 the next section 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, but it seems more than adequate for the project of generating interesting computational narratives.

## 4.2 The GRIOT Architecture

This section describes technology for implementing interactive and generative narratives. The implemented GRIOT system is meant as a prototype for computational narratives involving user-feedback and generation of content via partial combination of axioms and media elements using the Alloy blending algorithm. Specifically, it is demonstrated here using the case of creating interactive poetry systems. GRIOT is the *general* platform for creating such systems, and the works generated with GRIOT must be distinguished from the platform itself. Though the longer term research goal is to enable a wide variety of narrative multimedia forms, text-based poetry presented a desirable initial case study for several reasons including:

- (1) issues of meaning construction are central,
- (2) poetry admits a wide range discourse forms (those used here often emphasize narrative structure and dense conceptual and metaphorical content),
- (3) poetry admits range of possibilities for interaction (including allowing for actions in a story world, but not limited to it),

- (4) text-based work provides a good foundation for a scaffolding multimedia research project (following the example of early text-based prototypes generated by the Oz project). (Reilly, 1996)

In the subsequent subsection it will be shown how multimedia functionality is being incrementally added to this work.

Joseph Goguen and the author have coined the phrases “polymorphic poems” and “polypoems” to describe the initial cases implemented using GRIOT. This work 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 mean that one cannot draw inferences about human poetry based on the successes and failures of various techniques. Based on the theories above, the GRIOT implementation has the following components (Harrell, 2005b):

- (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) The Alloy Conceptual Blending Algorithm: The blending algorithm generates new concepts and metaphors from input spaces selected from the theme domains.
- (3) Media Morphisms: These map conceptual blends to representations in a particular medium, in this case natural language.

- (4) Optimality Principles: GRIOT uses structural optimality principles inspired by those from conceptual blending theory. (Fauconnier & Turner, 2002) 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, and (3) the number of elements from the input spaces that are preserved in the blended space.
- (5) Narrative 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 called a “probabilistic bounded transition stack machine,” or more simply a “Narrative Structure Machine”, or “Event Structure Machine” in the general case<sup>19</sup>. (Goguen & Harrell, 2007b)

Using these components, the GRIOT system functions as follows (Harrell, 2005b):

- (1) A formatted polypoem is read in and the system is initialized. This polypoem consists of the theme domains and keywords, phrase templates, and a narrative structure, provided by a polypoem author.

---

<sup>19</sup> The Event Structure Machine grew out of a collaboration with Joseph Goguen on the performance *The Griot Sings Haibun*. He brilliantly suggested the clause format and specific updates to the narrative automaton in order to allow for bounded, nested narrative structures. This joint effort provided a valuable lesson on how computer science developments can coincide with, and be driven by, artistic expression.

- (2) If it is specified by the Event Structure Machine, the system waits for user input in the form of a keyword (later this could be a multimedia event).
- (3) A phrase, of the appropriate type, is selected from the list of phrases. A phrase is a line of text represented as a list such as “(Her dreams were (\* g-singular-noun) or (\* g-verb-clause) ).” The phrase type is decided by the narrative structure provided as a finite state automaton, or more generally a probabilistic pushdown automaton.
- (4) The phrase selected in (2) is checked for wildcards of the following types:
  - \*p = phrasal (used to select a particular phrase type in order to replace the wildcard with a subphrase)
  - \*g = grammatical (used to choose grammatical expression for a blend-space, this would be bolstered for richer media)
  - \*d = domain (used to select domain)
  - \*a = axiom (for finer grained conceptual space selection)

(additional types can be added for specific needs as well, for example an \*r type was added in the case of *The Griot Sings Haibun* to enable repeats)
- (5) A grammatical wildcard consists of a token to represent its grammatical type, and possibly additional tokens to indicate the domain that should be used to replace it. For each wildcard two

domains are selected to be blended. This selection is either random, specified by the wildcard itself, or according to user input provided in (1).

(6) Conceptual spaces are selected from the chosen domains as follows:

(6.1) Axioms are chosen from the first domain. An axiom is a relation represented as a list such as “(axiom “devours” ((constant “evil” “emotion” demons-space 0) (constant “hope” “emotion” demons-space 0))).” The “0's” are used in computing the blend optimality.

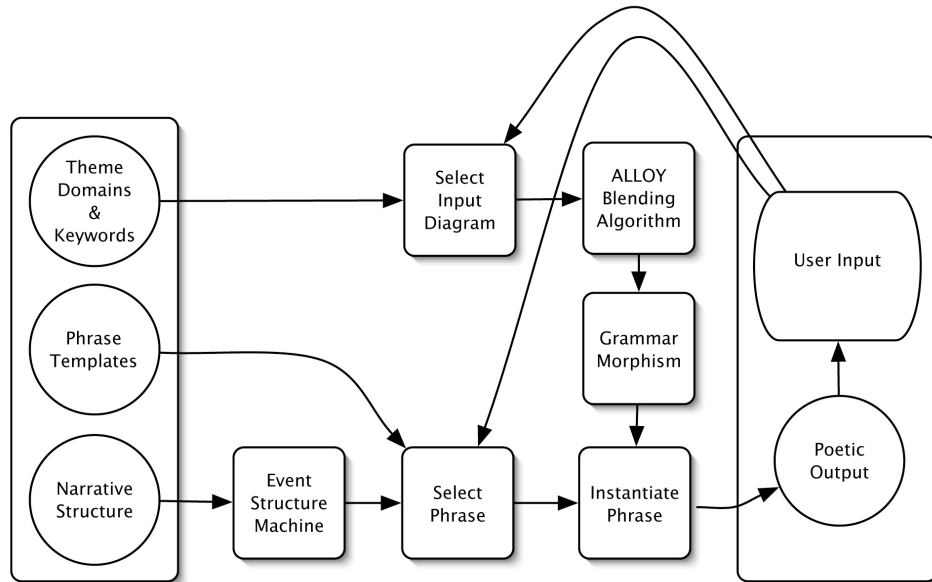
(6.2) A subdomain is formed from the second domain that consists only of axioms of sorts that match the chosen axioms; axioms are selected randomly from the subdomain.

(6.3) These spaces are used to create an input diagram (the generic space, two input spaces, and morphisms between the spaces).

(6.4) The input diagram is passed into the blending algorithm, which returns a conceptual blend and two morphisms to it.

(7) The wildcards are replaced using the returned conceptual blend. To replace a wildcard, a grammar morphism, implemented as a hash table of closures, is consulted. For each wildcard type it provides a mapping from a conceptual blend to a grammatical (later this could be another medium) form.

- (8) The phrase is output with all wildcards replaced, the phrase is said to be instantiated.
- (9) The system waits again for user input if it is specified by the Event Structure Machine, otherwise the system selects another phrase unless it has reached a finish state in the narrative structure.



**Figure 4.1: The GRIOT architecture**

The functionality of polymorphic poems authored with GRIOT can be demonstrated well using the first of several implemented cases entitled “The Girl with Skin of Haints and Seraphs,” which will be explored in more depth in **Subsection 4.4.1**. (Harrell, 2005a)



### 4.2.1 The Event Structure Machine

The initial implementation of an automaton to structure narrative clauses was relatively simple and was used to instantiate an adapted version of William Labov's empirical model of the narrative structure of personal experience from sociolinguistics discussed in **Section 2.1.2**. (Goguen, 2001; Labov, 1972) The format for specifying the automaton was designed in a way that is easy for a polypoem author to specify.

Subsequent projects have necessitated the development of a more powerful machine to structure clauses, in particular to enable hierarchically organized and nested narrative structures. Toward this end, Joseph Goguen and the author developed the "Event Structure Machine," technically we call it a probabilistic bounded transition stack machine, first described in (Goguen & Harrell, 2007b). The Event Structure Machine has the following format (in a modified extended BNF notation in which italicized phrases denote informal descriptions of atomic elements):

```

<Event Structure Machine> ::= "(structure" <clauses> ")"
<clauses> ::= <clause> {<clauses>}
<clause> ::= "(" <name> <number-pair> <subclause>
           <exit-to-clause> <read-flag> ")"
<name> ::= an atomic clause name
<number-pair> ::= "(" <minimum-number>
                 <maximum-number> ")"
<subclause> ::= "(" an atomic clause name ")" | "("
<exit-to-clause> ::= "(" an atomic clause name ")" | "("
<minimum-number> ::= a positive integer
<maximum-number> ::= a positive integer
<read-flag> ::= read | n

```

The most important structure in this format is the <clause>. The functioning of the event structure machine can be understood by examining the components of these “clauses.”

A clause consists of a name, pair of integers, subclause name, exit-to clause name, and a read-flag. A clause is to be interpreted as follows:

- (1) The <name> is a symbol used for referring to the clause type. This name can be anything and does not necessarily refer to specific clause types from various linguistic or narrative theories.
- (2) The <number-pair> consists of an integer indicating the minimum and maximum numbers of repetitions of the clause.
- (3) The <subclause> refers to the subsequent nested clause type to be selected.
- (4) The <exit-to-clause> refers to the subsequent clause type to be selected after all subclauses have been exhausted.
- (5) The <read-flag> determines whether or not user-input is to first be read and taken into account when instantiating a phrase template of the selected clause type.

For example, the polypoem *The Girl with Skin of Haints and Seraphs* was reimplemented in the Event Structure Machine format as follows (Harrell, 2005a, 2006):

```
(structure
  (ori (1 1) () narr read)
  (narr (3 5) eval coda read)
  (eval (0 1) () () n)
  (coda (1 1) () () read))
```

A possible poem output by such an automaton would have the following structure

(with clause names standing in for actual clauses):

```

ori
narr,
  eval
narr,
  eval
narr,
  eval
coda

```

In this example the “eval” clauses are nested under the “narr” clauses, and there are three repeats of “narr, eval” pairs. Such structures can easily be elaborated to define more complexly structured output, e.g., simply changing the event structure clause from “(ori (1 1) () narr read)” to “(ori (3 3) narr () read)” would result in the output consisting of three full repetitions of the output structure in the example above. The following is an example of real generated output, which incorporates blended concepts generated using Alloy (the LISP parentheses are left in so that clauses may be more easily distinguished):

```

> white
(her awareness begins when sunbather scaled-
  being envies melaninated and soul)
> demon
(she worked raising pain cool children of her
  own)
(death was better)
> evil
(her ears swam with bedrock dry-skin)
(and her spectre snow-queen feet danced)
> europe
(her ears swam with guillotine balm)
(she could laugh)
> angel
(she finally knew that a hate smugness woman
  would never be loved)

```

The meaning of the content of this poem is discussed in **Subsection 4.4.1** below.

### 4.3 Levels of Use

In order to express more clearly how human judgment, subjectivity, and interaction play a role in design and implementation with GRIOT, I have determined four levels of use of GRIOT encountered so far. These are:

Level 0) Coding GRIOT

Level 1) Designing a polypoem

Level 2) Providing input to a polypoem during execution

Level 3) Performing polypoem output

Level 0 refers to programming the platform. It includes originally implementing the GRIOT system and the ALLOY algorithm, but also refers to changes made to GRIOT as it iteratively develops to enable new types of polypoems. For example, reasonable changes a programmer may need to make to the GRIOT system (in LISP) include: introduction of new media morphisms, template variable types, changes to the Event Structure Machine, or updates to ALLOY. In the long term it may be desirable to allow some of these aspects to be changed by a polypoem author, however there is a trade-off as increases in the level of expressiveness for a polypoem author tend to also increase the level of programming expertise required by a polypoem author. This level also includes integrating GRIOT with a graphical user-interface, game, or another software application.

Level 1 denotes content creation using GRIOT. This is akin to authoring using Macromedia's Flash or Adobe's Photoshop. The difference here is that the author attempts to give the user some control over meaning, i.e. how concepts are represented and where user input must be taken into account in generation of new content. The polypoem author must input three components: 1) an ontology consisting of sets of axioms, 2) a template database consisting of text with embedded variable (wildcards), and 3) a particular narrative structure in the format required to instantiate the narrative automaton.

Level 2 denotes using a system, i.e. playing a game or reading a poem. This level complicates the notion of a "reader" since the reader/user may be required to interact with the system, for example entering keywords, selecting icons, or moving a character through a virtual world. Still, although a user may take a role in the construction of a particular poem, the user is not considered to be the "author" of the poem any more than the visitor of a building (who undoubtedly generates a particular experience of the building through her/his navigation of the space) is said to be the architect of the building. The line between user and author is not hard however, and a creative interactive work could blur this distinction. This is generally the most constrained level of participating in a GRIOT system. At the greatest degree of constriction the user is allowed no input and acts merely as a (mechanically) passive reader of a text.

Level 3 exists in the special case of a performance. As output is generated via user interaction with a polypoem, a human may be the vehicle for presenting this work. For example, in a polypoem and free jazz performance entitled *The Griot Sings*

*Haibun*, Joseph Goguen acted as the level 3 participant by reciting poetry output from a plasma screen in front of him (Goguen and Harrell 2005). This is not a trivial level, a great deal of interpretive nuance arises via a human's ability to adjust to the context of a situation. For example, the human voice afforded Goguen sensitive control for recitation. This level need not refer to vocal performance, however, it refers to any possible way in which a human can use the output of the user's interaction with the system to spur some performative act.

## 4.4 Examples of Interactive Polymorphic Poetry

This section presents several examples of interactive polymorphic poetry. Aside from presenting unique examples of generated content, each of these examples is used as a case study to illuminate further aspects of the GRIOT system. **Subsection 4.4.1** focuses on illustrating the GRIOT architecture and the use of the Labov model of narrative for structural blending. **Subsection 4.4.2** focuses on the use of a multimedia GUI with a polypoem, and the use of GRIOT as a platform to enable a different discourse structure than seen in **Subsection 4.4.1**. **Subsection 4.4.3** focuses on the notion of “double-scope stories” as described by Mark Turner in (Turner, 2003), and the relationship of such stories to computational narrative. Further examples of output from all three polypoems can be found in the **Appendices**.

#### 4.4.1 The Girl with Skin of Haints and Seraphs

*The Girl with Skin of Haints and Seraphs* was first implemented in a non-interactive form as the initial deployment of the Alloy algorithm for generative purposes within another system. It has been subsequently updated with each iteration of GRIOT and it provides a good example for tracing through the execution of an interactive polymorphic poem. As stated above, this polypoeem is a commentary on racial politics, the limitations of simplistic binary views of social identity, and the need for more contingent, dynamic models 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. This LISP program draws on a set of theme domains such as skin, angels, demons, Europe, and Africa, given as sets of axioms.

The following example and discussion illustrate the interactive nature of this polypoeem. It also recapitulates and extends the discussion of GRIOT architecture and formal description of text generation described above. After processing a user input keyword such as “Europe” entered at a “>” prompt, the first line could be:

```
her tale began when she was infected with white
female-itis
```

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:

```
her tale began when she was infected with
tribal-warrior spectre-itis
```

or

```
her tale began when she was infected with black
demon-itis
```

depending upon how the phrase template was instantiated.

The discussion of template instantiation in the text grammar of **Section 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 variable, here called a “wildcard,” 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 contents 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 as the user entered keywords determine one of the domains to be used in constructing 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 wildcards with English language mappings from conceptual blends produced by the algorithm of **Chapter 3**.

A sample poem generated by one interaction with the polypoem is given below (user input consists of terms entered at the “>” prompt, the other lines consist of content generated by the system):

```
> 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
```

The text grammar of **Section 4.1** 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 **Subsection 2.1.2** gives top level structure for poems. One poetic domain contains the template `(her tale began when (* g-verb-clause) )` in the LISP syntax of the implementation, where the inner parenthesis is a variable that gets instantiated with a verb phrase containing a past tense transitive verb, such as `(was infected with (* g-singular-noun) )` where again the inner parenthesized phrase is a variable that gets instantiated with a conceptual blend produced by the algorithm of **Chapter 3**. 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 issue for 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 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 must take precedence over dogmatic Turing test reductionism.

A related point is raised by Espen Aarseth's analysis of text generation systems, which takes relationships among programmer, system, and reader as a basis for critical analysis. Aarseth's point is that readers' authorial models affect their interpretations of works, causing the approaches of traditional literary criticism to fail when computers are putative authors. (Aarseth, 1997) 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."

Narrative construction such as described above provides a nice example of structural blending. In particular, *The Girl with Skin of Haints and Seraphs* illustrates structural blending using the formalization of Labov's model of narrative from **Subsection 2.1.2**. We first define a "narrative space" for the rules in **Subsection 2.1.2**, seen as constructors by reversing their direction and expanding the \*s; in this context, such constructors are conventionally called "templates." The narrative space also needs additional rules to supply clauses to instantiate the arguments of the Labov constructors. Some arguments would be blended with elements from other spaces to provide particular persons (e.g., a protagonist), places, objects, etc.; this cross-space sharing is indicated by shared generic constants from a generic space. All these spaces would vary from one narrative to another; it is a major task of the artist to choose them appropriately. The Labov structure is not the only possibility here, but it seems likely to be more familiar and recognizable first case that allows for a wide range of generated texts.

It is intriguing that the implementation of these ideas closely resembles what a good natural language generation (NLG) programmer would likely do anyway, the main difference being that this deeper understanding of the principles involved supports a clearer system architecture (as shown in **Figure 4.1**). In this sense, it is a rational reconstruction of existing practice. However, we also expect that new possibilities will emerge as we gain more experience building active (interactive and generative) media systems. For example, “optimality” principles for narrative coherence, and more generally for intersentential coherence, need further investigation. Other issues include blending constructors to create new hybrid templates (which in effect would be a novel view of grammar as process), and combining the engines for conceptual and structural blending. Such research problems seem especially relevant for generating new content that seamlessly blends multiple media.

In addition to Labov’s narrative of personal experience, the GRIOT constructed polypoem *The Girl with Skin of Haints and Seraphs* invokes attributes of other culturally specific narrative forms. Dynamic improvisation and call and response structures are familiar aspects of Pan-African narrative forms as diverse as the African Brazilian martial art and dance Capoeira Angola, Charles Mingus’s calling out of the segregationist Governor of Arkansas in “Fables of Faubus” from (Mingus, 1960), the penetratingly satirical fiction of Ishmael Reed, and hip-hop freestyle rhyming. Written prose poetry, and its more recent descendant flash fiction (“short short” stories that encapsulate full narrative arcs within extremely abbreviated word counts), traditionally have not incorporated these techniques. (Lehman, 2003; Thomas,

Thomas, & Hakuza, 1992) On-the-fly improvisation has not been incorporated for the simple reason that the nature of medium of printed text is not dynamically reconfigurable. Computational media have dynamic information structure and feedback loops built into the nature of the medium. The output of *The Girl with Skin of Haints and Seraphs* combines this type of prose poetry, dynamically reconfigurable and founded in African and African American vernacular traditions of signification. (Gates Jr., 1988)

#### 4.4.2 The Griot Sings Haibun

Another case is the polypoem *The Griot Sings Haibun*, developed by the author and Joseph Goguen and based on a poem entitled *November Qualia* by Joseph Goguen. (Goguen, 2005b; Goguen & Harrell, 2005) It extends the framework used in GRIOT 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). The graphical user interface (GUI) was programmed using the Processing development tool and a knowledge base (text-file) output by GRIOT. *The Griot Sings Haibun* polypoem was used with a graphical interface (see **Figures 4.2** and **4.3**) 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. (Goguen & Harrell, 2005) This work provides an example of using the GRIOT architecture to implement a polypoem with a specific, predefined style and discourse structure in mind. The output successfully replicated Goguen’s style (in

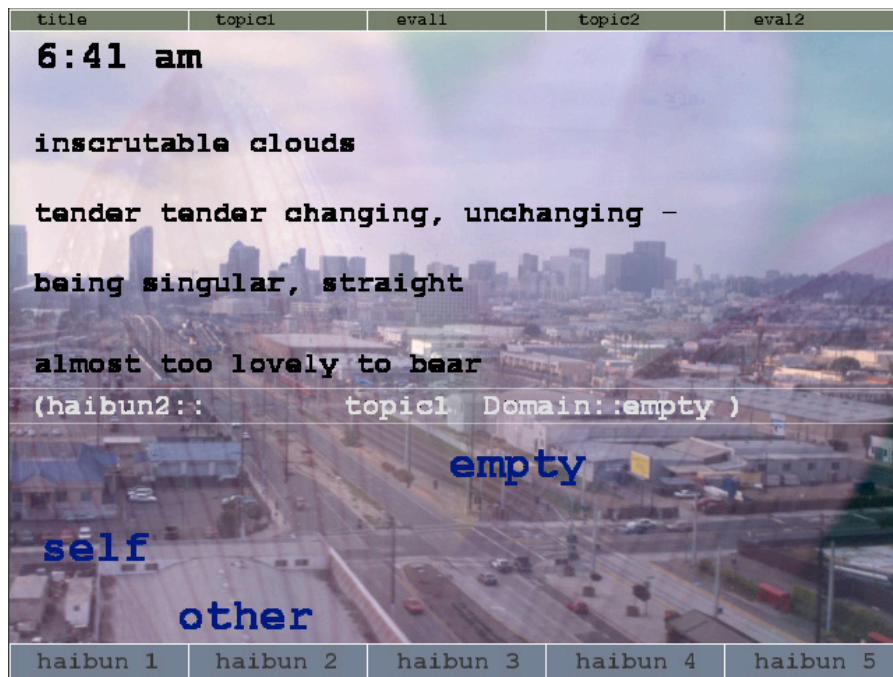
Goguen’s opinion), though it sacrifices nuance and specificity of meaning for variability.



**Figure 4.2:** A screenshot from *The Griot Sings Haibun*

During the performance, the GUI was projected onto a large screen behind the performers for the audience to see. The GUI was mirrored on a plasma screen facing the performers so that the musicians and narrator (Goguen) could see. A snapshot of one such image seen by the performers and audience is shown in **Figure 4.2**. The author acted a system performer, improvisationally generating text output based on what the musicians played. The musicians could also respond improvisationally to the text visible on the plasma screen. In this sense the performance was a collective improvisation. Using the GUI for *The Griot Sings Haibun*, the system performer selects the desired clause type using buttons arranged in a row at the top of the screen. The user selects a clause by clicking on one of the keywords (e.g., “self,” “empty,” or

“other”) on the bottom one-third of the screen. This selects the use of particular theme domains, just as entering keywords did in the polypoem *The Girl with Skin of Haints and Seraphs*. At various times, clauses of only particular types would appear on the screen and would be regenerated on-the-fly. Thus, during performance the discourse structure was much more dynamic and variable than in the pure LISP interface. Several examples of haibun poetry were implemented, and buttons along the bottom of the screen allowed the performer to shift from one haibun polypoem to another. This also shifted between background images composed by the author<sup>20</sup>. **Figure 4.3** depicts a snapshot of the screen featuring a different haibun polypoem.



**Figure 4.3:** A screenshot from *The Griot Sings Haibun*

<sup>20</sup> Some of the backgrounds were created using photographs taken by Joseph Goguen as raw images.

### 4.4.3 Walking Blues Changes Undersea

*Walking Blues Changes Undersea* is a polypoem that provides an example of what Mark Turner terms a “double-scope story.” **Subsection 4.4.3.1** extends the cognitive linguistics discussion in **Section 2.2** toward this end, and **Subsection 4.4.3.2** describes the theory’s influence on the development and output of this polypoem.

#### 4.4.3.1 Narrative Imagining

Narrative imagining combines a host of cognitive operations (including sequence recognition and construction, categorization of objects and events, projection of image schemas, and more) and the metaphor and conceptual blend generation and interpretation discussed above play a central role. (Turner, 1996) Turner describes stories as “complex dynamic integrations of objects, events, and actors.” Under this view, stories can be described using conceptual blending theory. Blends that integrate input spaces from different, even clashing, organizing frames are considered “double-scope blends.” (Fauconnier & Turner, 2002) When these input spaces represent stories the blended results are “double-scope stories.” (Turner, 2003)

In (Turner, 2003) there are examples of several double-scope stories featuring an astounding succession of blending operations in which characters magically, metaphorically transform from one entity to another. One of his examples is the song “O, Magali” from Frederic Mistral’s *Mireille*, which contains a sequence where a suitor pursues the object of his affection only to have her transform into a different



entity to escape him. He transforms to capture her, she transforms to escape again, and so on in a spiraling competitive conversation:

- If you become a fish, I will become a fisherman.
- Well then I will become a bird and fly away.
- Then I will become a hunter and hunt you.
- Then I will become a flowering herb in the wild.
- Then I will become water and sprinkle you.
- Then I will become a cloud and float away to America.
- Then I will become the sea breeze to carry you.

...

Turner's account of "O, Magali" and similar stories is especially useful here for two reasons: (1) the works are structured by the pattern of an abstract story: the rhetorical structure of a narrated competitive conversation between two people, and (2) it provides a theoretical model to describe blending processes in stories involving transforming characters as in (Turner, 2004). *Walking Blues Changes Undersea* parallels these two aspects of Turner's analysis as discussed below in **Subsection 4.4.3.2**. Turner's account also considers how such work can possibly achieve meaningful impact upon users as they project these metaphorical narratives onto "the stories we live in," our real life experiences. **Subsection 4.4.4** uses *Walking Blues Changes Undersea* to illustrate one method one might use to aesthetically analyze and evaluate polymorphic poetry.

#### **4.4.3.2 *Walking Blues Changes Undersea* Functionality and Structure**

*Walking Blues Changes Undersea* is a polypoem that tells the double-scope story of an individual traveling through everyday locations, while performing

everyday actions in a dynamic, transitory world of unstable social identity. It evokes an unsteady world increasingly blended with oceanic fantasy as the world sinks beneath the waves (recalling the Atlantis myth). The everyday world initially consists of banal events in everyday locations: waking up in bed, taking a shower, eating at the breakfast table, working, eating lunch, eating again, at home, and back to bed to sleep. The user selects location specific actions such as ‘sleep,’ ‘scrub,’ ‘munch,’ ‘procrastinate,’ ‘watch-tv,’ ‘exercise,’ ‘slumber,’ and more. Each of these actions is associated (the exact associations are hidden from the user) with a particular emotional disposition such as lazy, aggressive, apathetic, or peaceful. The output generated in response to the user input incorporates blends generated from mental spaces selected from ontologies corresponding to these dispositions and to the locations. Additionally, over the course of the narrative the descriptions of the locations and the protagonist (which can be considered the user’s player character) are also blended with spaces selected from ontologies describing undersea themes. Finally, each execution tells a completely different, but similarly themed, tale.

The structure of the *Walking Blues Changes Undersea* polypoems is as follows:

```
(structure
  (orient (1 1) () intro n)
  (intro (1 1) () location-1 n)
  (location-1 (1 1) action-1 location-2 n)
  (action-1 (1 1) transform-response () read)
  (location-2 (1 1) action-2 location-3 n)
  (action-2 (1 1) transform-response () read)
  (location-3 (1 1) action-3 location-4 n)
  (action-3 (1 1) transform-response () read)
  (location-4 (1 1) action-4 location-5 n)
  (action-4 (1 1) transform-response () read)
  (location-5 (1 1) action-5 location-6 n))
```

```

        (action-5 (1 1) transform-response () read)
        (location-6 (1 1) action-6 location-7 n)
        (action-6 (1 1) transform-response () read)
        (location-7 (1 1) action-7 sleep n)
        (action-7 (1 1) transform-response () read)
        (sleep (1 1) () coda n)
        (coda (1 2) () () n)
        (transform-response (0 1) () () n)
    )

```

Given theme domains, keywords, and phrase templates as described above the polypoem generates output such as<sup>21</sup>:

```

my world was so small and heavy,
rooms for waking, bathing, consuming,
    sweating, sunning, devouring, sleeping, and
    waking
my first movement of the day awaits, I
> stretch
and encounter fishermen, soft grogginess
a whiff of sea passes through me
reciting a pop song like a mantra while
    washing, I decide to
> scrub
daily tidepool quiet cleaning
an Atlantean aroma
I need to manufacture energy to confront the
    day, I need vicious and cool, so first I
> consume
on toward my job
the air shimmers a bit
always imagining resident, fish at my desk, I
    must
> procrastinate
gulls and feeble working is not bad
it's a water, simple ever-changing days
the day's break is here, I need to
> eat
full, satisfied, satisfied
colors seem a bit duller today
placating bureaucrats on and on, I must
> work
the building fills with angler-fish

```

---

<sup>21</sup> User input again follows the ">" prompt and, unlike earlier polypoems created with GRIOT, is meant to be read as a part of the text.

compassion as I repeat yesterday's tasks  
 an Atlantean aroma  
 after laboring I am in my room to  
 > **watch-tv**  
 my defeated ineffectual life

*Walking Blues Changes Undersea* could also be implemented using a GUI in a manner similar to *The Griot Sings Haibun*. **Figures 4.4 and 4.5** depicts concept art for such an implementation in which users click on buttons containing keywords to generate text, and the user's avatar (center screen) is also altered based upon the theme domains selected. Further screenshots and sample output can be found in **Appendix D**.



**Figure 4.4:** GUI design screenshot for *Walking Blues Changes Undersea*



**Figure 4.5: A GUI design screenshot for *Walking Blues Changes Undersea***

#### **4.4.4 Remarks on Aesthetic Analysis of Computational Poetry**

Evaluative analysis of artistic work cannot be carried out using methods arising from engineering disciplines based on values such as efficiency, feasibility, or generalizability. Analysis requires a framework suited for subjectively assessing cultural artifacts. One direction in developing an evaluative and interpretive framework is to base analyses in the blending metaphor theory discussed above in **Section 2.2.3**. Aside from its role as a cognitive theory, Turner and Lakoff have applied metaphor theory to literary criticism. (Lakoff & Turner, 1989) They assert that despite “an infinitude of potential metaphors,” there is a small set of metaphors with special status called “basic metaphors” discovered through systematic analysis of large bodies of linguistic data and claimed to be a part of the common conceptual apparatus shared by members of a culture. Analysis can be performed by examining the roles of such metaphors in poetry. Additionally, although the output discussed here is

termed “poetry,” polypoems are a new cultural form that recalls some structural aspects of narrative prose poetry, omits others, and includes computational characteristics such as polymorphic structure, procedural generation, formal knowledge representation, user feedback loops. The fact that literary criticism based in metaphor theory finds its analyses on accounts of cognitive semantics, as opposed to examination of structural and rhetorical aspects of traditional poetry, also makes it suitable for analysis of polypoems.

Informal analysis of *Walking Blues Changes Undersea* reveals that the tale invokes several basic metaphors. The story of travel through a day is an example of the “Life is a Journey” basic metaphor and is an instance of the “A Lifetime is a Day” basic metaphor. In the sample output above, the first line, “my world was so small and heavy,” invokes the “Life is a Burden” metaphor, which among other things maps the notion of “weight” onto life elements. This is illustrated in the different instances of output both from above and from **Appendix D** where “the indolent body atmosphere,” “the sleeping-beauty, lazy atmosphere” and “the cave heavenly atmosphere” are all described as “a little heavy.” The concept of weight is also invoked in the line “my head had been rock and my heart black lead” from the *Walking Blues Changes Undersea* output example 2 in **Appendix D**. The story of a world sinking below the waves is an example of a conceptual blend between the basic “Down is Bad” metaphor and the mundane world the protagonist inhabits. The example of output above ends with the phrase “my defeated, ineffectual life,” which reflects the “Sad is Down” metaphor when interpreted in the context of the sinking world. Similarly, the

output instances in **Appendix D** invoke the “Unconscious is Down” metaphor when interpreted in the context of the sinking world as seen in excerpts such as:

a rectangular cushion awaits  
 I love you, good night  
 crisp sheets fade to boring warmth  
 and  
 just like before, the day is done  
 the tale of my every day  
 good night.

In (Lakoff & Turner, 1989) Lakoff and Turner describe three general approaches that poets have traditionally taken toward employing basic metaphors:

- (1) versifying them in automatic ways resulting in trite verse,
- (2) deploying them masterfully through combination, extension, and realizing them in striking imagery, and
- (3) deploying them in unusual ways or destabilizing them by revealing their inadequacies for making sense of lived experience in the real world.

A preliminary suggestion for deeper aesthetic analysis of this work is to carefully document the invocation of basic metaphors and relating them to the criteria above. Secondly, the output should be analyzed for creation of expressive and relevant novel metaphors. Thirdly, the output should be analyzed for grammaticality and coherence, taking into account perceived and actual authorial intention and when these issues are to be considered “technical details” as opposed to important aesthetic concerns.

Methods for analyzing double-scope stories in static text can also be applied to polypoems. Following the rhetorical discourse structure of a conversation between a level 2 user and the polypoem that I designed (as a level 1 user), the structure of the output is slightly reminiscent of the structure of “O Magali” discussed in **Section 4.4.3.1** and it can be analyzed similarly. (Turner, 2003) There is a succession of blends involving actions (user input), events (transformation of the protagonist and locations), and objects (specific items described in locations). Although the Alloy algorithm is not considered to capture what humans do when we blend concepts, there is a clear mapping between the Turner inspired account of the conceptual blending that occurs when we participate in the polypoem and the actual output generated when the polypoem is processed and executed using GRIOT. This result is encouraging because it suggests that utilizing the framework of conceptual blending theory is a promising direction for generating interactive double-scope stories with consistently structured output in which user interaction drives the generation of new content for each instance of output.



The text of this chapter, in part, contains material that is a reprint of, or has been submitted for publication in: Goguen, J., & Harrell, D. F. (2007); *Style as a Choice of Blending Principles* (submitted); In S. Argamon, K. Burns & S. Dubnov (Eds.), *The Structure of Style: Algorithmic Approaches to Understanding Manner and Meaning*; Berlin: Springer; Harrell, D. F. (2005); *Shades of Computational Evocation and Meaning: The GRIOT System and Improvisational Poetry Generation*; Paper presented at the 6th Digital Arts and Culture Conference, Copenhagen, Denmark; and Harrell, D. F. (2006). *Walking Blues Changes Undersea: Imaginative Narrative in Interactive Poetry Generation with the GRIOT System*; Paper presented at the AAAI 2006 Workshop in Computational Aesthetics: Artificial Intelligence Approaches to Happiness and Beauty, Boston, MA. The dissertation author was either the sole author or a co-author of these papers.

## Chapter 5 Conclusion

I was the shadow of the waxwing slain  
By the false azure in the windowpane;  
I was the smudge of ashen fluff – and I  
Lived on, flew on, in the reflected sky.

– Vladimir Nabokov, from *Pale Fire* (Nabokov, 1962)

He is not himself. Only a character in a story someone else was writing. Why he is so sure of this would be another story, and if he paused to ruminate about other stories he'd never make it up the hill. So he leaves everything behind.

– John Edgar Wideman, from *The Cattle Killing* (Wideman, 1996)

Two worlds divided by an azure sheet of glass, each inscrutable to the denizens of the other, alien, strange to each other, and irreconcilable. Computer science and expressive culture have sat mostly at odds, exchanging misunderstanding stares. A few little birds have chipped away, however, and noticed that on either side of the division lies a reflection of the other. In part, this dissertation is aimed toward entering these reflected worlds. There is a risk, as for Nabokov's poor waxwing, to be

slain by false possibility. But there is possibility, again as for Nabokov's birdie, to soar into uncharted imaginative realms.

For me, it is the issue of narrative that pushes me to leave disciplinary constraints behind. The possibility to create illusory stories and meanings, whether immersive story worlds or experimental, improvisational, and metaphorically rich texts. I reflect on the processes of telling stories, of inhabiting stories, of being constrained and enabled by narrative possibilities and the striking range of human imagination for creating such structures. I have chosen to explore stories as told through computational media, and, like Wideman's character, I have no time to reflect on other stories that might be told. I am dedicated toward composing this new one.

## **5.1 Recapitulation of Results**

For clarity, it is important to recap specifically what has been accomplished technically so far with the approach taken in this dissertation. At the core of the work is the generation and analysis of content using partial composition of sign systems (represented using algebraic semiotics). This work contributes to cognitive science by implementing a formal model useful for experimentation with the structural aspects of conceptual blending. A second technical accomplishment is implementing the combination of templates (pre-structured content) according to formalized narrative structures. The strength of this approach is that conceptual blending (the generative component) and structural blending (the media composition component) can be accounted for using the same theoretical underpinning (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., within multimedia artwork or computer games).

Some specific technical contributions of this work are (recapped earlier in (Goguen & Harrell, 2007b)):

- (1) data structures for sign systems and semiotic morphisms (mappings between sign systems),
- (2) an algorithm for conceptual blending that is efficient and exhaustive,
- (3) three formal 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 (this arranges templates into a particular discourse structure),
- (7) an interpreter that reads user input and outputs generated content,
- (8) and a scripting format to allow an author to create “improvisational” media.

On the basis of these technical contributions, several examples of polymorphic poetry have been implemented as described in **Section 4.4** above. These works have been presented and performed internationally and have been published in peer-reviewed journals, computer science conferences, and as book chapters. These polymorphic

poems represent early steps toward a narrative art form that privileges meaning and the possibility for evocative emotional and intellectual statements as exist in media such as books or film.

The technical, cognitive, and expressive contributions of this work comprise a new theoretical approach. Scientific approaches to imagination, computational approaches to narrative structure and meaning representation, and expressive approaches to cultural analysis and narrative creation are unified toward a common goal. Here, I have developed a theory that articulates a set of powerfully expressive narrative possibilities, possibilities centered upon meaningfully generating and interacting with narrative content and reconfiguring stories improvisationally in the computational medium. Toward this end, it was necessary to base the generative and interactive capabilities of computational narrative upon a robust model of meaning and its construction, and to consider the limits of computation as a means toward that end.

My integration of cognitive semantics models with computational techniques that are sensitive to issues of human context and subjectivity comprise both theoretical and technical contributions. Cognitive science methods and results allowed me to consider human meanings and how they are creatively constructed, in particular I explored meaning construction within narrative imagining. I recognized the positive value (and deep challenges) in allowing a human author to generate subjective media assets that are not opaque to algorithmic manipulation and reconfiguration. This recognition led to my use of Joseph Goguen's algebraic semiotics approach with its sensitivity to exactly which structural components can be manipulated computationally, and which should be directly manipulated by a human

author/programmer/operator. My developments in allowing authors to develop computationally manipulable, but authorially subjective, media assets is a theoretical and technical contribution in several ways. I combine bottom-up methods (basic building blocks of meaning – conceptual spaces – are used as bases for narrative construction without privileging one particular cultural or literary theoretic model of narrative) and top-down methods (authors are provided with facilities to implement particular cultural narrative models and to construct structured knowledge bases of both media and conceptual content). I believe that, though the results so far have been modest and implemented in predominantly text-only or relatively simple multimedia formats, the theory and technology created heralds a new approach to computational narrative that admits both richer and more immersive media technologies and richer and more expressive accounts of human narratives, creative imagining, and even aesthetic style.

## **5.2 Toward Dimensions of Aesthetic Style**

The issue of aesthetic style is an important one because well developed conventions of style, and traditions of stylistic innovation and experimentation, are hallmarks of a mature medium. The implications of this work for issues of style presented in this section have been described in (Goguen & Harrell, 2007b). In that paper, Goguen and the author pose the following questions: So can the technical contribution of this dissertation 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?

I believe so. Though inspired technically by some of the approaches above, this approach includes being very forthright about the limitations of computational techniques and introduces human judgment, subjectivity, and social context at appropriate points in the design, development, and output processes. Thus, for me:

- (1) Style is understood in the interpretation of sign systems via dialogic interaction with media.
- (2) Style is determined by the particular executions of human concepts in media.
- (3) Style is created by developing sign systems and artifacts for both of the above.
- (4) Style exists in the context of interpreting sign systems.
- (5) Style exists in the context of use of artifacts.
- (6) 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, this work systematically examines 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 this work pays close heed to the ways that humans encode knowledge that are not amenable to computational analysis or manipulation, and explicitly require human input and judgment in design processes. The approach here

is a combination of formal methods, awareness of their limitations, and strategies to get around them.

The GRIOT 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*, of which a nice recounting can be found in (Turner, 1996) ), 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 the broad sense of that phrase used here).

This gives rise to twelve 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 these are counted as parameters, then the



result is eighteen parameters. This is not meant to be a reductive account, of course it is possible to cut this cake more finely or more coarsely to get different numbers and to later find other parameters that are also important for style. Every parameter can be considered a principle that controls blending and are interesting to consider in this regard. It is clear that all principles must be carefully tuned to achieve a reasonable approximation to an existing style (such is in approximating Joseph Goguen's haibun style in *The Griot Sings Haibun*), but it is also clear that the results are unlikely to be extremely close to the nuanced expressive voicing of a human author directly creating expressive work without the intermediary of computationally amenable representation. What is gained is a new potential authorial voice that takes algorithmic manipulation and generation of meaning, and the interactive participation of the user as central to a new aesthetic palette.

### 5.3 Directions for Future Work

This dissertation concludes with a brief sketch of promising future directions to explore that have been suggested by the results and approach described above. An important and fundamental observation is that media elements of even the lowest level of granularity encode some attributes of meaning. Of course, not all of the aspects of meaning imposed by these attributes can, or should, be formalized. In order to make these semantic attributes amenable to algorithmic manipulation, the media elements must be annotated in some manner. A future research problem is to investigate principles for media annotation. My intuition is that annotating media elements with

semiotic space descriptions may be useful, though blending of the semiotic spaces does not provide technical means, or an implicit theoretical framework, for how media elements should be combined. The constructors necessary for such blending can be imported from media theory, e.g., from film theory's rich body of conventions for blending cinematic structures, or from work such as Scott McCloud's *Understanding Comics*, which describes principles for combining image and text within comics. (McCloud, 1993)

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 static. 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. Many contemporary highly narrative computer games follow the same basic model that was first implemented in interactive fiction systems in the early 1970s. This is either a testament to the strength of the model or to the need for a theory and technology base to support new genres of interactive narrative games. This gaming model has several key characteristics: locations and characters with state information, puzzles involving figuring out specific sequences of actions, and open ended exploration with a linear solution (though in contemporary games locations are no longer always subdivided into discrete spaces, and user actions occurring within locations are often derived from alternate action oriented gaming models).

One, possibly surprising, result of this research is that a combination of

conceptual and structural blending can produce interesting poetry. Another is that both large grain structure and syntax can be handled by blending in ways that are close to, but somewhat 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. A third result is that it is easy to extend the approach to interaction, to media other than text, and to forms other than narrative. Aside from the development of further optimality principles in line with Fauconnier and Turner's, and enabling further capacity for creation of emergent structure through blending, these results relate to another hypothesis to explore in future work. The optimization principles proposed in (Fauconnier & Turner, 2002), though good for common sense blends like "house boat," may not work well for some types of creative poetry, where it seems that some kind of "*disoptimization*" principles are more appropriate, at least for language with widely varying, surprising, and unconventional metaphors and conceptual constructions. For example, Joseph Goguen and the author have performed an analysis of poetry by Pablo Neruda and encountered blends (quite intensified and novel metaphors) that exhibit characteristics that seem to be highly "*disoptimal*" by the standards of the principles in (Fauconnier & Turner, 2002). (Goguen & Harrell, 2007b; Neruda, 1947) Though this observation is tentative and preliminary, I believe this phenomenon is a ripe area for future investigation.

My hope is that the theoretical framework introduced here, which integrates imagination, computation, and expression, will be a sensitive enough tool for one day implementing poetic narrative experiences that can truly invoke and reflect upon the human condition. Social power relationships and conventional categories that

constrain individuals to narrow, haunting boxes; the weight of daily life, the dynamic, contingent nature of social identity, tender joys, and continual loss; if one day this work can lead toward evocation of these and other unknown, sensitive, and human, meanings, then the value of my modest contributions shall be considered redeemed.

The text of **Section 5.1** and **Section 5.2** of this chapter, in part, contains material that is a reprint of, or has been submitted for publication in Goguen, J., & Harrell, D. F. (2007); Style as a Choice of Blending Principles (submitted); In S. Argamon, K. Burns & S. Dubnov (Eds.), *The Structure of Style: Algorithmic Approaches to Understanding Manner and Meaning*; Berlin: Springer. The dissertation author was a co-author of this paper.

# Appendices

## Appendix A: Alloy Output for the Houseboat Blend

Given an input diagram composed of the spaces and mappings described in **Section 2.3**, and in a format similar to that described in **Section 3.1**, the 48 computed primary houseboat blends in order of optimality (optimality values are indicated in square brackets) are:

```
#<blend [0]:  Constants: ((house/boat : Object)
(resident/passenger : Person))
              Sorts: ((person : Person) (object : Object))
              Datasorts: ((medium : Medium))
              Relations: ((live-in/ride (person object))
                          (on1/on2 (object datasort^medium))
                          )
              Axioms: ((axiom live-in/ride ((constant
resident/passenger person blended-space 0) (constant
house/boat object blended-space 0)) blended-space 0)
(axiom on1/on2 ((constant house/boat object blended-space
0) (constant land datasort^medium input-1 0)) blended-
space 0) (axiom on1/on2 ((constant house/boat object
blended-space 0) (constant water datasort^medium input-2
0)) blended-space 0))>

#<blend [200]:  Constants: ((house/boat : Object)
(resident : Person) (passenger : Person))
              Sorts: ((person : Person) (object : Object))
              Datasorts: ((medium : Medium))
              Relations: ((live-in/ride (person object))
                          (on1/on2 (object datasort^medium))
                          )
              Axioms: ((axiom live-in/ride ((constant
resident person input-1 0) (constant house/boat object
blended-space 0)) blended-space 0) (axiom live-in/ride
((constant passenger person input-2 0) (constant
house/boat object blended-space 0)) blended-space 0)
(axiom on1/on2 ((constant house/boat object blended-space
0) (constant land datasort^medium input-1 0)) blended-
space 0) (axiom on1/on2 ((constant house/boat object
blended-space 0) (constant water datasort^medium input-2
0)) blended-space 0))>

#<blend [200]:  Constants: ((house : Object) (boat :
Object) (resident/passenger : Person))
              Sorts: ((person : Person) (object : Object))
```

```

      Datasorts: ((medium : Medium))
      Relations: ((live-in/ride (person object))
                  (on1/on2 (object datasort^medium))
                  )
      Axioms: ((axiom live-in/ride ((constant
resident/passenger person blended-space 0) (constant
house object input-1 0)) blended-space 0) (axiom live-
in/ride ((constant resident/passenger person blended-
space 0) (constant boat object input-2 0)) blended-space
0) (axiom on1/on2 ((constant house object input-1 0)
(constant land datasort^medium input-1 0)) blended-space
0) (axiom on1/on2 ((constant boat object input-2 0)
(constant water datasort^medium input-2 0)) blended-space
0)))>

```

```

#<blend [200]:  Constants: ((house/boat : Object)
(resident/passenger : Person))
      Sorts: ((person : Person) (object : Object))
      Datasorts: ((medium : Medium))
      Relations: ((live-in/ride (person object))
                  (on1 (object datasort^medium))
                  (on2 (object datasort^medium))
                  )
      Axioms: ((axiom live-in/ride ((constant
resident/passenger person blended-space 0) (constant
house/boat object blended-space 0)) blended-space 0)
(axiom on1 ((constant house/boat object blended-space 0)
(constant land datasort^medium input-1 0)) blended-space
0) (axiom on2 ((constant house/boat object blended-space
0) (constant water datasort^medium input-2 0)) blended-
space 0)))>

```

```

#<blend [200]:  Constants: ((house/boat : Object)
(resident/passenger : Person))
      Sorts: ((person : Person) (object : Object))
      Datasorts: ((medium : Medium))
      Relations: ((live-in (person object))
                  (ride (person object))
                  (on1/on2 (object datasort^medium))
                  )
      Axioms: ((axiom live-in ((constant
resident/passenger person blended-space 0) (constant
house/boat object blended-space 0)) blended-space 0)
(axiom ride ((constant resident/passenger person blended-
space 0) (constant house/boat object blended-space 0))
blended-space 0) (axiom on1/on2 ((constant house/boat
object blended-space 0) (constant land datasort^medium
input-1 0)) blended-space 0) (axiom on1/on2 ((constant
house/boat object blended-space 0) (constant water
datasort^medium input-2 0)) blended-space 0)))>

```

```

#<blend [400]:  Constants: ((house : Object) (boat :
Object) (resident : Person) (passenger : Person))
      Sorts: ((person : Person) (object : Object))
      Datasorts: ((medium : Medium))
      Relations: ((live-in/ride (person object))
                  (on1/on2 (object datasort^medium))

```

```

    )
    Axioms: ((axiom live-in/ride ((constant
resident person input-1 0) (constant house object input-1
0)) blended-space 0) (axiom live-in/ride ((constant
passenger person input-2 0) (constant boat object input-2
0)) blended-space 0) (axiom on1/on2 ((constant house
object input-1 0) (constant land datasort^medium input-1
0)) blended-space 0) (axiom on1/on2 ((constant boat
object input-2 0) (constant water datasort^medium input-2
0)) blended-space 0))>

```

```

#<blend [400]: Constants: ((house/boat : Object)
(resident : Person) (passenger : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in/ride (person object))
(on1 (object datasort^medium))
(on2 (object datasort^medium))
)
Axioms: ((axiom live-in/ride ((constant
resident person input-1 0) (constant house/boat object
blended-space 0)) blended-space 0) (axiom live-in/ride
((constant passenger person input-2 0) (constant
house/boat object blended-space 0)) blended-space 0)
(axiom on1 ((constant house/boat object blended-space 0)
(constant land datasort^medium input-1 0)) blended-space
0) (axiom on2 ((constant house/boat object blended-space
0) (constant water datasort^medium input-2 0)) blended-
space 0))>

```

```

#<blend [400]: Constants: ((house : Object) (boat :
Object) (resident/passenger : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in/ride (person object))
(on1 (object datasort^medium))
(on2 (object datasort^medium))
)
Axioms: ((axiom live-in/ride ((constant
resident/passenger person blended-space 0) (constant
house object input-1 0)) blended-space 0) (axiom live-
in/ride ((constant resident/passenger person blended-
space 0) (constant boat object input-2 0)) blended-space
0) (axiom on1 ((constant house object input-1 0)
(constant land datasort^medium input-1 0)) input-1 0)
(axiom on2 ((constant boat object input-2 0) (constant
water datasort^medium input-2 0)) input-2 0))>

```

```

#<blend [400]: Constants: ((house/boat : Object)
(resident : Person) (passenger : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
(ride (person object))
(on1/on2 (object datasort^medium))
)

```

```

      Axioms: ((axiom live-in ((constant resident
person input-1 0) (constant house/boat object blended-
space 0)) blended-space 0) (axiom ride ((constant
passenger person input-2 0) (constant house/boat object
blended-space 0)) blended-space 0) (axiom on1/on2
((constant house/boat object blended-space 0) (constant
land datasort^medium input-1 0)) blended-space 0) (axiom
on1/on2 ((constant house/boat object blended-space 0)
(constant water datasort^medium input-2 0)) blended-space
0))>

```

```

#<blend [400]:  Constants: ((house : Object) (boat :
Object) (resident/passenger : Person))
      Sorts: ((person : Person) (object : Object))
      Datasorts: ((medium : Medium))
      Relations: ((live-in (person object))
                  (ride (person object))
                  (on1/on2 (object datasort^medium))
                  )
      Axioms: ((axiom live-in ((constant
resident/passenger person blended-space 0) (constant
house object input-1 0)) blended-space 0) (axiom ride
((constant resident/passenger person blended-space 0)
(constant boat object input-2 0)) blended-space 0) (axiom
on1/on2 ((constant house object input-1 0) (constant land
datasort^medium input-1 0)) blended-space 0) (axiom
on1/on2 ((constant boat object input-2 0) (constant water
datasort^medium input-2 0)) blended-space 0))>

```

```

#<blend [400]:  Constants: ((house/boat : Object)
(resident/passenger : Person))
      Sorts: ((person : Person) (object : Object))
      Datasorts: ((medium : Medium))
      Relations: ((live-in (person object))
                  (ride (person object))
                  (on1 (object datasort^medium))
                  (on2 (object datasort^medium))
                  )
      Axioms: ((axiom live-in ((constant
resident/passenger person blended-space 0) (constant
house/boat object blended-space 0)) blended-space 0)
(axiom ride ((constant resident/passenger person blended-
space 0) (constant house/boat object blended-space 0))
blended-space 0) (axiom on1 ((constant house/boat object
blended-space 0) (constant land datasort^medium input-1
0)) blended-space 0) (axiom on2 ((constant house/boat
object blended-space 0) (constant water datasort^medium
input-2 0)) blended-space 0))>

```

```

#<blend [401]:  Constants: ((house/passenger : Person)
(boat : Object) (resident : Person))
      Sorts: ((person : Person) (object : Object))
      Datasorts: ((medium : Medium))
      Relations: ((live-in/ride (person object))
                  (on1/on2 (object datasort^medium))
                  )

```



```

        Axioms:      ((axiom live-in/ride ((constant
house/passenger person blended-space 0) (constant boat
object input-2 0)) blended-space 0) (axiom on1/on2
((constant house/passenger object blended-space 0)
(constant land datasort^medium input-1 0)) blended-space
0) (axiom on1/on2 ((constant boat object input-2 0)
(constant water datasort^medium input-2 0)) blended-space
0))>

```

```

#<blend [401]:  Constants: ((house/passenger : Object)
(boat : Object) (resident : Person))
        Sorts: ((person : Person) (object : Object))
        Datasorts: ((medium : Medium))
        Relations: ((live-in/ride (person object))
(on1/on2 (object datasort^medium))
)
        Axioms:      ((axiom live-in/ride ((constant
resident person input-1 0) (constant house/passenger
object blended-space 0)) blended-space 0) (axiom on1/on2
((constant house/passenger object blended-space 0)
(constant land datasort^medium input-1 0)) blended-space
0) (axiom on1/on2 ((constant boat object input-2 0)
(constant water datasort^medium input-2 0)) blended-space
0))>

```

```

#<blend [401]:  Constants: ((house : Object)
(resident/boat : Object) (passenger : Person))
        Sorts: ((person : Person) (object : Object))
        Datasorts: ((medium : Medium))
        Relations: ((live-in/ride (person object))
(on1/on2 (object datasort^medium))
)
        Axioms:      ((axiom live-in/ride ((constant
passenger person input-2 0) (constant resident/boat
object blended-space 0)) blended-space 0) (axiom on1/on2
((constant house object input-1 0) (constant land
datasort^medium input-1 0)) blended-space 0) (axiom
on1/on2 ((constant resident/boat object blended-space 0)
(constant water datasort^medium input-2 0)) blended-space
0))>

```

```

#<blend [401]:  Constants: ((house : Object)
(resident/boat : Person) (passenger : Person))
        Sorts: ((person : Person) (object : Object))
        Datasorts: ((medium : Medium))
        Relations: ((live-in/ride (person object))
(on1/on2 (object datasort^medium))
)
        Axioms:      ((axiom live-in/ride ((constant
resident/boat person blended-space 0) (constant house
object input-1 0)) blended-space 0) (axiom on1/on2
((constant house object input-1 0) (constant land
datasort^medium input-1 0)) blended-space 0))>

```

```

#<blend [402]:  Constants: ((house/passenger : Person)
(resident/boat : Object))
        Sorts: ((person : Person) (object : Object))

```

```

        Datasorts: ((medium : Medium))
        Relations: ((live-in/ride (person object))
                    (on1/on2 (object datasort^medium))
                    )
        Axioms: ((axiom live-in/ride ((constant
house/passenger person blended-space 0) (constant
resident/boat object blended-space 0)) blended-space 0)
(axiom on1/on2 ((constant resident/boat object blended-
space 0) (constant water datasort^medium input-2 0))
blended-space 0))>

#<blend [402]:  Constants: ((house/passenger : Person)
(resident/boat : Person))
        Sorts: ((person : Person) (object : Object))
        Datasorts: ((medium : Medium))
        Relations: ((live-in/ride (person object))
                    (on1/on2 (object datasort^medium))
                    )
        Axioms:  (>)

#<blend [402]:  Constants: ((house/passenger : Object)
(resident/boat : Object))
        Sorts: ((person : Person) (object : Object))
        Datasorts: ((medium : Medium))
        Relations: ((live-in/ride (person object))
                    (on1/on2 (object datasort^medium))
                    )
        Axioms: ((axiom on1/on2 ((constant
house/passenger object blended-space 0) (constant land
datasort^medium input-1 0)) blended-space 0) (axiom
on1/on2 ((constant resident/boat object blended-space 0)
(constant water datasort^medium input-2 0)) blended-space
0))>

#<blend [402]:  Constants: ((house/passenger : Object)
(resident/boat : Person))
        Sorts: ((person : Person) (object : Object))
        Datasorts: ((medium : Medium))
        Relations: ((live-in/ride (person object))
                    (on1/on2 (object datasort^medium))
                    )
        Axioms: ((axiom live-in/ride ((constant
resident/boat person blended-space 0) (constant
house/passenger object blended-space 0)) blended-space 0)
(axiom on1/on2 ((constant house/passenger object blended-
space 0) (constant land datasort^medium input-1 0))
blended-space 0))>

#<blend [600]:  Constants: ((house : Object) (boat :
Object) (resident : Person) (passenger : Person))
        Sorts: ((person : Person) (object : Object))
        Datasorts: ((medium : Medium))
        Relations: ((live-in/ride (person object))
                    (on1 (object datasort^medium))
                    (on2 (object datasort^medium))
                    )

```

```

Axioms: ((axiom live-in/ride ((constant
resident person input-1 0) (constant house object input-1
0)) blended-space 0) (axiom live-in/ride ((constant
passenger person input-2 0) (constant boat object input-2
0)) blended-space 0) (axiom on1 ((constant house object
input-1 0) (constant land datasort^medium input-1 0))
input-1 0) (axiom on2 ((constant boat object input-2 0)
(constant water datasort^medium input-2 0)) input-2 0))>

```

```

#<blend [600]: Constants: ((house : Object) (boat :
Object) (resident : Person) (passenger : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
(ride (person object))
(on1/on2 (object datasort^medium))
)

```

```

Axioms: ((axiom live-in ((constant resident
person input-1 0) (constant house object input-1 0))
input-1 0) (axiom ride ((constant passenger person input-
2 0) (constant boat object input-2 0)) input-2 0) (axiom
on1/on2 ((constant house object input-1 0) (constant land
datasort^medium input-1 0)) blended-space 0) (axiom
on1/on2 ((constant boat object input-2 0) (constant water
datasort^medium input-2 0)) blended-space 0))>

```

```

#<blend [600]: Constants: ((house/boat : Object)
(resident : Person) (passenger : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
(ride (person object))
(on1 (object datasort^medium))
(on2 (object datasort^medium))
)

```

```

Axioms: ((axiom live-in ((constant resident
person input-1 0) (constant house/boat object blended-
space 0)) blended-space 0) (axiom ride ((constant
passenger person input-2 0) (constant house/boat object
blended-space 0)) blended-space 0) (axiom on1 ((constant
house/boat object blended-space 0) (constant land
datasort^medium input-1 0)) blended-space 0) (axiom on2
((constant house/boat object blended-space 0) (constant
water datasort^medium input-2 0)) blended-space 0))>

```

```

#<blend [600]: Constants: ((house : Object) (boat :
Object) (resident/passenger : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
(ride (person object))
(on1 (object datasort^medium))
(on2 (object datasort^medium))
)
Axioms: ((axiom live-in ((constant
resident/passenger person blended-space 0) (constant
house object input-1 0)) blended-space 0) (axiom ride

```

```
((constant resident/passenger person blended-space 0)
 (constant boat object input-2 0)) blended-space 0) (axiom
on1 ((constant house object input-1 0) (constant land
datasort^medium input-1 0)) input-1 0) (axiom on2
((constant boat object input-2 0) (constant water
datasort^medium input-2 0)) input-2 0))>
```

```
#<blend [601]:  Constants: ((house/passenger : Person)
(boat : Object) (resident : Person))
      Sorts: ((person : Person) (object : Object))
      Datasorts: ((medium : Medium))
      Relations: ((live-in/ride (person object))
                  (on1 (object datasort^medium))
                  (on2 (object datasort^medium))
                  )
      Axioms: ((axiom live-in/ride ((constant
house/passenger person blended-space 0) (constant boat
object input-2 0)) blended-space 0) (axiom on2 ((constant
boat object input-2 0) (constant water datasort^medium
input-2 0)) input-2 0))>
```

```
#<blend [601]:  Constants: ((house/passenger : Object)
(boat : Object) (resident : Person))
      Sorts: ((person : Person) (object : Object))
      Datasorts: ((medium : Medium))
      Relations: ((live-in/ride (person object))
                  (on1 (object datasort^medium))
                  (on2 (object datasort^medium))
                  )
      Axioms: ((axiom live-in/ride ((constant
resident person input-1 0) (constant house/passenger
object blended-space 0)) blended-space 0) (axiom on1
((constant house/passenger object blended-space 0)
 (constant land datasort^medium input-1 0)) blended-space
0) (axiom on2 ((constant boat object input-2 0) (constant
water datasort^medium input-2 0)) input-2 0))>
```

```
#<blend [601]:  Constants: ((house : Object)
(resident/boat : Object) (passenger : Person))
      Sorts: ((person : Person) (object : Object))
      Datasorts: ((medium : Medium))
      Relations: ((live-in/ride (person object))
                  (on1 (object datasort^medium))
                  (on2 (object datasort^medium))
                  )
      Axioms: ((axiom live-in/ride ((constant
passenger person input-2 0) (constant resident/boat
object blended-space 0)) blended-space 0) (axiom on1
((constant house object input-1 0) (constant land
datasort^medium input-1 0)) input-1 0) (axiom on2
((constant resident/boat object blended-space 0)
 (constant water datasort^medium input-2 0)) blended-space
0))>
```

```
#<blend [601]:  Constants: ((house : Object)
(resident/boat : Person) (passenger : Person))
      Sorts: ((person : Person) (object : Object))
```

```

Datasorts: ((medium : Medium))
Relations: ((live-in/ride (person object))
            (on1 (object datasort^medium))
            (on2 (object datasort^medium))
            )
Axioms: ((axiom live-in/ride ((constant
resident/boat person blended-space 0) (constant house
object input-1 0)) blended-space 0) (axiom on1 ((constant
house object input-1 0) (constant land datasort^medium
input-1 0)) input-1 0))>

```

```

#<blend [601]: Constants: ((house/passenger : Person)
(boat : Object) (resident : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
            (ride (person object))
            (on1/on2 (object datasort^medium))
            )
Axioms: ((axiom ride ((constant
house/passenger person blended-space 0) (constant boat
object input-2 0)) blended-space 0) (axiom on1/on2
((constant boat object input-2 0) (constant water
datasort^medium input-2 0)) blended-space 0))>

```

```

#<blend [601]: Constants: ((house/passenger : Object)
(boat : Object) (resident : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
            (ride (person object))
            (on1/on2 (object datasort^medium))
            )
Axioms: ((axiom live-in ((constant resident
person input-1 0) (constant house/passenger object
blended-space 0)) blended-space 0) (axiom on1/on2
((constant house/passenger object blended-space 0)
(constant land datasort^medium input-1 0)) blended-space
0) (axiom on1/on2 ((constant boat object input-2 0)
(constant water datasort^medium input-2 0)) blended-space
0))>

```

```

#<blend [601]: Constants: ((house : Object)
(resident/boat : Object) (passenger : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
            (ride (person object))
            (on1/on2 (object datasort^medium))
            )
Axioms: ((axiom ride ((constant passenger
person input-2 0) (constant resident/boat object blended-
space 0)) blended-space 0) (axiom on1/on2 ((constant
house object input-1 0) (constant land datasort^medium
input-1 0)) blended-space 0) (axiom on1/on2 ((constant
resident/boat object blended-space 0) (constant water
datasort^medium input-2 0)) blended-space 0))>

```

```
#<blend [601]:  Constants: ((house : Object)
(resident/boat : Person) (passenger : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
(ride (person object))
(on1/on2 (object datasort^medium))
)
Axioms: ((axiom live-in ((constant
resident/boat person blended-space 0) (constant house
object input-1 0)) blended-space 0) (axiom on1/on2
((constant house object input-1 0) (constant land
datasort^medium input-1 0)) blended-space 0))>
```

```
#<blend [602]:  Constants: ((house/passenger : Person)
(resident/boat : Object))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in/ride (person object))
(on1 (object datasort^medium))
(on2 (object datasort^medium))
)
Axioms: ((axiom live-in/ride ((constant
house/passenger person blended-space 0) (constant
resident/boat object blended-space 0)) blended-space 0)
(axiom on2 ((constant resident/boat object blended-space
0) (constant water datasort^medium input-2 0)) blended-
space 0))>
```

```
#<blend [602]:  Constants: ((house/passenger : Person)
(resident/boat : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in/ride (person object))
(on1 (object datasort^medium))
(on2 (object datasort^medium))
)
Axioms:  (>
```

```
#<blend [602]:  Constants: ((house/passenger : Object)
(resident/boat : Object))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in/ride (person object))
(on1 (object datasort^medium))
(on2 (object datasort^medium))
)
Axioms: ((axiom on1 ((constant
house/passenger object blended-space 0) (constant land
datasort^medium input-1 0)) blended-space 0) (axiom on2
((constant resident/boat object blended-space 0)
(constant water datasort^medium input-2 0)) blended-space
0))>
```

```
#<blend [602]:  Constants: ((house/passenger : Object)
(resident/boat : Person))
```

```

Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in/ride (person object))
            (on1 (object datasort^medium))
            (on2 (object datasort^medium))
            )
Axioms: ((axiom live-in/ride ((constant
resident/boat person blended-space 0) (constant
house/passenger object blended-space 0)) blended-space 0)
(axiom on1 ((constant house/passenger object blended-
space 0) (constant land datasort^medium input-1 0))
blended-space 0))>

```

```

#<blend [602]: Constants: ((house/passenger : Person)
(resident/boat : Object))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
            (ride (person object))
            (on1/on2 (object datasort^medium))
            )
Axioms: ((axiom ride ((constant
house/passenger person blended-space 0) (constant
resident/boat object blended-space 0)) blended-space 0)
axiom on1/on2 ((constant resident/boat object blended-
space 0) (constant water datasort^medium input-2 0))
blended-space 0))>

```

```

#<blend [602]: Constants: ((house/passenger : Person)
(resident/boat : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
            (ride (person object))
            (on1/on2 (object datasort^medium))
            )
Axioms: ()>

```

```

#<blend [602]: Constants: ((house/passenger : Object)
(resident/boat : Object))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
            (ride (person object))
            (on1/on2 (object datasort^medium))
            )
Axioms: ((axiom on1/on2 ((constant
house/passenger object blended-space 0) (constant land
datasort^medium input-1 0)) blended-space 0) (axiom
on1/on2 ((constant resident/boat object blended-space 0)
(constant water datasort^medium input-2 0)) blended-space
0))>

```

```

#<blend [602]: Constants: ((house/passenger : Object)
(resident/boat : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))

```

```

Relations: ((live-in (person object))
            (ride (person object))
            (on1/on2 (object datasort^medium))
            )
Axioms: ((axiom live-in ((constant
resident/boat person blended-space 0) (constant
house/passenger object blended-space 0)) blended-space 0)
(axiom on1/on2 ((constant house/passenger object blended-
space 0) (constant land datasort^medium input-1 0))
blended-space 0))>

```

```

#<blend [800]: Constants: ((house : Object) (boat :
Object) (resident : Person) (passenger : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
            (ride (person object))
            (on1 (object datasort^medium))
            (on2 (object datasort^medium))
            )
Axioms: ((axiom live-in ((constant resident
person input-1 0) (constant house object input-1 0))
input-1 0) (axiom ride ((constant passenger person input-
2 0) (constant boat object input-2 0)) input-2 0) (axiom
on1 ((constant house object input-1 0) (constant land
datasort^medium input-1 0)) input-1 0) (axiom on2
((constant boat object input-2 0) (constant water
datasort^medium input-2 0)) input-2 0))>
)

```

```

#<blend [801]: Constants: ((house/passenger : Person)
(boat : Object) (resident : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
            (ride (person object))
            (on1 (object datasort^medium))
            (on2 (object datasort^medium))
            )
Axioms: ((axiom ride ((constant
house/passenger person blended-space 0) (constant boat
object input-2 0)) blended-space 0) (axiom on2 ((constant
boat object input-2 0) (constant water datasort^medium
input-2 0)) input-2 0))>

```

```

#<blend [801]: Constants: ((house/passenger : Object)
(boat : Object) (resident : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
            (ride (person object))
            (on1 (object datasort^medium))
            (on2 (object datasort^medium))
            )
Axioms: ((axiom live-in ((constant resident
person input-1 0) (constant house/passenger object
blended-space 0)) blended-space 0) (axiom on1 ((constant

```



```
house/passenger object blended-space 0) (constant land
datasort^medium input-1 0)) blended-space 0) (axiom on2
((constant boat object input-2 0) (constant water
datasort^medium input-2 0)) input-2 0))>
```

```
#<blend [801]:  Constants: ((house : Object)
(resident/boat : Object) (passenger : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
(ride (person object))
(on1 (object datasort^medium))
(on2 (object datasort^medium))
)
Axioms: ((axiom ride ((constant passenger
person input-2 0) (constant resident/boat object blended-
space 0)) blended-space 0) (axiom on1 ((constant house
object input-1 0) (constant land datasort^medium input-1
0)) input-1 0) (axiom on2 ((constant resident/boat object
blended-space 0) (constant water datasort^medium input-2
0)) blended-space 0))>
```

```
#<blend [801]:  Constants: ((house : Object)
(resident/boat : Person) (passenger : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
(ride (person object))
(on1 (object datasort^medium))
(on2 (object datasort^medium))
)
Axioms: ((axiom live-in ((constant
resident/boat person blended-space 0) (constant house
object input-1 0)) blended-space 0) (axiom on1 ((constant
house object input-1 0) (constant land datasort^medium
input-1 0)) input-1 0))>
```

```
#<blend [802]:  Constants: ((house/passenger : Person)
(resident/boat : Object))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
(ride (person object))
(on1 (object datasort^medium))
(on2 (object datasort^medium))
)
Axioms: ((axiom ride ((constant
house/passenger person blended-space 0) (constant
resident/boat object blended-space 0)) blended-space 0)
(axiom on2 ((constant resident/boat object blended-space
0) (constant water datasort^medium input-2 0)) blended-
space 0))>
```

```
#<blend [802]:  Constants: ((house/passenger : Person)
(resident/boat : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
```

```

Relations: ((live-in (person object))
            (ride (person object))
            (on1 (object datasort^medium))
            (on2 (object datasort^medium))
            )
Axioms:    (>)

#<blend [802]:  Constants: ((house/passenger : Object)
(resident/boat : Object))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
            (ride (person object))
            (on1 (object datasort^medium))
            (on2 (object datasort^medium))
            )
Axioms:    ((axiom on1 ((constant
house/passenger object blended-space 0) (constant land
datasort^medium input-1 0)) blended-space 0) (axiom on2
((constant resident/boat object blended-space 0)
(constant water datasort^medium input-2 0)) blended-space
0))>

#<blend [802]:  Constants: ((house/passenger : Object)
(resident/boat : Person))
Sorts: ((person : Person) (object : Object))
Datasorts: ((medium : Medium))
Relations: ((live-in (person object))
            (ride (person object))
            (on1 (object datasort^medium))
            (on2 (object datasort^medium))
            )
Axioms:    ((axiom live-in ((constant
resident/boat person blended-space 0) (constant
house/passenger object blended-space 0)) blended-space 0)
(axiom on1 ((constant house/passenger object blended-
space 0) (constant land datasort^medium input-1 0))
blended-space 0))>

```

## Appendix B: The Girl with Skin of Haints and Seraphs

Three samples of output from *The Girl with Skin of Haints and Seraphs* follow (Harrell, 2005a):

- (1) The first sample poem and a detailed description of its generation follows. User input is differentiated by being preceded by a ‘>’ prompt. The system output is

italicized. My commentary on how some of the content of each line of text is generated follows the system output. LISP code for an axiom describing subjective information about the domain follows my commentary.

*her arrival onto this earth was marked when first-born and charcoal-girl  
transforms to impoverished-elder or charcoal-woman  
she worked raising snow-queen original-lady children of her own  
the young lady would prevail  
a caress across her skin scares up demon black  
her failure was ignoring her wings and original-lady nature  
and she felt glad  
as she grew older she saw entitlement defiance wrinkles upon her face  
ebony-wood-like brimstone defines fetish bedrock,  
the sign that let her know she finally really alive*

#### **>Africa**

*her arrival onto this earth was marked when first-born and charcoal-girl  
transforms to impoverished-elder or charcoal-woman*

The concepts of first born people, the impoverished elder, and charcoal skin are selected from the 'Africa' domain in this opening clause. The LISP axiom selected for blending is:

```
(axiom "is-now"
  ((constant "first-born" "person" afrika-space)
   (constant "impoverished-elder" "person" afrika-space)))
```

#### **>Europe**

*she worked raising snow-queen original-lady children of her own*

The concept of the snow queen is selected from the 'Europe' domain in this narrative clause. The LISP axiom selected for blending is:

```
(axiom "wears"
  ((constant "snow-queen" "person" europe-space)
   (constant "wintery-skin" "object" europe-space)))
```

**>demon**

*the young lady would prevail*

The 'Demon' domain is selected, but not used in this evaluative clause.

**>demon**

*a caress across her skin scares up demon black*

The concept of a demon is selected from the 'Demon' domain in this narrative clause.

The LISP axiom selected for blending is:

```
(axiom "stokes"
  ((constant "demon" "person" demons-space)
   (constant "hate" "emotion" demons-space)))
```

**>angel**

*her failure was ignoring her wings and original-lady nature*

The concept of wings is selected from the 'Angel' domain in this evaluative clause.

The LISP axiom selected for blending is:

```
(axiom "covered-with"
  ((constant "wings" "person" angels-space)
   (constant "feathers" "object" angels-space)))
```

**>black**

*and she felt glad*

The 'Africa' domain is selected, but not used in this evaluative clause.

**>white**

*as she grew older she saw entitlement defiance wrinkles upon her face*

The concept of entitlement is selected from the 'Europe' domain for use in this narrative clause. The LISP axiom selected for blending is:

```
(axiom "experiences"
  ((constant "european" "person" europe-space)
   (constant "entitlement" "sensation" europe-space)))
```

**>juju**

*ebony-wood-like brimstone defines fetish bedrock, the sign that let her know she finally really alive*

The concept of an ebony wood fetish is selected from the ‘Africa’ domain in this closing clause. The LISP axiom selected for blending is:

```
(axiom "constructs"
  ((constant "ebony-wood" "object" afrika-space)
   (constant "fetish" "object" afrika-space))
```

(2) The following is output produced when user input selects the use of the ‘Europe’ domain for constructing conceptual spaces for blending.

**>Europe**

*her tale began when she was infected with white female-itis*

**>Rome**

*she worked raising bullet, spiked-tail children of her own*

**>Norway**

*in the shadows*

**>Greece**

*when she was no longer a child peasant, august-being marks streaked her thighs*

**>Europe**

*her barbarian, impoverished-elder spirit would live on*

(3) A poem with generated content derived from blending concepts from the ‘Skin’ domain with concepts selected by the system.

**>skin**

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

**>skin**

*she peeped out shame, hate*

**>skin**

*finally she fell from a cloud and skin and black drenched days were left behind*

## Appendix C: The Griot Sings Haibun

This appendix presents sample output, code, poetry, and screenshots related to the polypoem performance *The Griot Sings Haibun*. Joseph Goguen described the poems as follows (Goguen, 2005a):

Each example has five sections, an introduction followed by four short haiku-like poems, each representing a particular moment of consciousness of the polypoem author. The high level structure of this polypoem is a variant of haibun, which is most famously used in Basho's great *Oku no Hosomichi*.

The following are two examples of output such as generated in a performance of *The Griot Sings Haibun*:

### *November Qualia I*

qualia are moments of luminous world,  
     empty, suffering, compassion  
 mind body snapshots  
 neither arising, departing, or dwelling  
 gone beyond  
 gone far beyond

6:41 am  
 mind cloud ocean  
 unmoved moving trees  
 connecting blue high, blue emptiness flesh  
 forever being sky  
 timeless, perfected tender self

6:53 am  
 the pipes:-  
 connecting, pipes of compassion  
 vivid bolted  
 shining, empty  
 neither atomic nor not  
 always connecting  
 timeless, perfected  
 beyond being beyond

7:26 pm  
 save us from fear, wanting  
 addict & moonlight, wanting & timeless forgotten  
 bright grasping fear  
 vultures of mind, forever circling  
 embracing transient form  
 ecstatic revulsion, wanting

1:47 am  
 translucent flesh, tender, inscrutable  
 unending mother ocean  
 biomass being, vital & vivid  
 all connected, empty, void  
 burning mind & self  
 ecstatic tender burning void

*November Qualia 2*

qualia are moonlight moments of luminous form,  
 moments of emptiness & consciousness  
 fleeting, luminous, empty  
 qualia are not objective platonic truths  
 gone, no loss, no gain

6:41 am  
 clear moments of empty cloud masses  
 mind cloud ocean  
 luminous static canopy  
 motionless moving  
 past unending & empty blue mind road  
 almost too lovely to bear

6:53 am  
 pipes, vivid dust red  
 connected form & heart dust  
 vivid bolted  
 their shining  
 dependent, empty, no void  
 silent red  
 always connecting  
 still more than stillness

7:26 pm  
 save us from desire, tender

fixed, forgotten  
 bright hunger, concept billboard  
 neon, void, lost  
 hanging vapor & desire, other  
 beyond form & emptiness

1:47 am  
 zooplankton inscrutable  
 writhing, mutated, infinite  
 light, dots, ever  
 swimming, flesh empty luminous  
 infinite empty perception  
 living illusion & moving, completed

The code used to produce this output was authored by Joseph Goguen using GRIOT, a sample of code that produced one section of the poem follows (in an “easy input” format that is simpler than the original LISP version):

```
(setf poem '(
;; *** novq1.grt ***
;; *** 25 oct 05 ***
;; *** 1st poem of haibun 4 poem structure ***
(discourse-structure
  (title topic1)
  (topic1 eval1)
  (eval1 topic2)
  (topic2 topic2 eval2)
  (eval2 eval2) )

(templates

(title
  ("6:41" am)
  ("6:41" am)
  (clear early morning)
  (grey morning early)
  (white-yellow cloud morning)
)

(topic1
  (time is a traveller too)
  (clear (* g-singular-noun) (* g-singular-noun) cloud
masses)
  ((* g-singular-noun) vivid solid ",")
  (inscrutable (* g-singular-noun) clouds)
  (inscrutable clouds)
  ((* g-singular-noun) compassion clouds)
  ((* g-singular-noun) cloud ocean)
  ((* g-singular-noun) shaped clouds)
  ((* g-singular-noun d-other) are ",")
```



```

)

(eval1
  (motionlessly moving)
  (shining ",")
  (unmoved moving trees)
  (luminous static canopy ",")
  (glowing (* g-singular-noun d-other) ",")
  (frozen protecting (* g-singular-noun))
)

(topic2
  ((* g-singular-noun d-other) singular ", " straight)
  (past unending (* g-singular-noun d-other) road "-")
  (forever (* g-singular-noun d-other) sky "-")
  ((* g-singular-noun d-other) changing ", " unchanging "-")
)
  (blue highway blue (* g-singular-noun d-other) ",")
  (connecting (* g-singular-noun d-other) ",")
)

(eval2
  (almost too lovely to bear)
  (almost too lovely to bear)
  (infinite perfect painful stillness)
  (beyond being gone ", " gone beyond being)
  (no (* g-singular-noun) d-self)
  (gone ", " gone ", " gone way beyond)
  (timeless ", " perfected (* g-singular-noun d-other))
)
)

```

The original neo-haibun from which the qualia of the polypoem are derived is *November Qualia* by Joseph Goguen . The original *November Qualia* follows (Goguen, 2005b):

### *November Qualia*

Four moments: driving, office stairwell, supermarket, sleep. Author is Buddhist, but everyone has similar small moments, usually unnoticed, unremembered. Timeless, frozen, neither atomic nor not, they connect, inner-outer, past-present-future, percept-concept, body-mind, & more.

6:41 am

Clear leaf cloud masses  
     motionlessly moving  
 past the static gray road -  
     almost too lovely to bear.

6:53 am

Pipe dust red vivid bolted,  
     shining,  
 is, not more not less,  
     timeless, perfected.

7:26 pm

Save us from shopping, wanting  
     vultures  
 in markets of flesh, hung  
     with gaudy christmas banners.

1:47 am

Unbounded plankton sea,  
     swimming - light - dots -  
 meshing connected, no  
     body no  
 self.

The following are seven screenshots from *The Griot Sings Haibun*.

title	topic1	eval1	topic2	eval2
november qualia - four poem haibun La Jolla, 21-22 November 2004 qualia are moments of heart and mind such emptiness, luminous moments always exist everyone has compassion percept qualia moments, usually unnoticed, unremembered (haibun1::           eval2   Domain::other       ) empty self other				
haibun 1	haibun 2	haibun 3	haibun 4	haibun 5

Figure A.1: Seven screenshots from *The Griot Sings Haibun*

title	topic1	eval1	topic2	eval2
grey morning early				
inscrutable empty awareness clouds				
motionlessly moving				
connecting tender other,				
beyond being gone, gone beyond being				
(haibun2::	eval2	Domain::self	)	
self			empty	
other				
haibun 1	haibun 2	haibun 3	haibun 4	haibun 5

title	topic1	eval1	topic2	eval2
6:53 am				
flow of compassion, grasping				
their redness,				
emptiness is emptiness				
timeless, perfected, red dust				
(haibun3::	eval2	Domain::other	)	
self			empty	
other			dogen	
haibun 1	haibun 2	haibun 3	haibun 4	haibun 5

Figure A.1: Seven screenshots from *The Griot Sings Haibun* (continued)

title	topic1	eval1	topic2	eval2
7:26 pm				
save us from mind billboard				
bright shopping and revulsion				
in markets of wanting consumption				
ecstatic hungry wanting				
(haibun4::	eval2	Domain::consume	)	
	empty			
self				
other			consume	
haibun 1	haibun 2	haibun 3	haibun 4	haibun 5

title	topic1	eval1	topic2	eval2
indoor evening market				
save us from suffering shopping				
vultures,				
in markets of emptiness and suffering				
extinguish shopper				
(haibun4::	eval2	Domain::consume	)	
	empty			
self				
other			consume	
haibun 1	haibun 2	haibun 3	haibun 4	haibun 5

Figure A.1: Seven screenshots from *The Griot Sings Haibun* (continued)



title	topic1	eval1	topic2	eval2
late night,sleep				
people biomass				
limitless transparent fear				
forever luminous emptiness interconnected				
ecstatic zooplankton				
(haibun5::	eval2	Domain::biomorphic )		
self		empty		
		biomorphic		
other				
haibun 1	haibun 2	haibun 3	haibun 4	haibun 5

title	topic1	eval1	topic2	eval2
1:47 am				
translucent and tender vivid vital				
fertile ocean inquisitive movement				
living being birth cells				
no self, no illusion and movement				
(haibun5::	eval2	Domain::biomorphic )		
self		empty		
		biomorphic		
other				
haibun 1	haibun 2	haibun 3	haibun 4	haibun 5

Figure A.1: Seven screenshots from *The Griot Sings Haibun* (continued)

## Appendix D: Walking Blues Changes Undersea

The following are two samples from an early version of *Walking Blues*

*Changes Undersea* output. In each case the user selected keywords all serve to select theme domains highlighting a particular emotional disposition.

1) Lazy disposition:

no gills, no webbing between digits,  
 it wouldn't be a watery grave, but a salt water  
 life  
 the day's initial action : I  
 -> **sleep**  
 and feel like a lazy sleeping-beauty  
 the indolent body atmosphere is a little heavy  
 the day cannot begin without being clean, I need  
 to  
 -> **soak**  
 I think playful contented thoughts, then of the  
 breakfast table  
 you will recall the importance of hearty  
 breakfast cuisine to  
 -> **munch**  
 feeling tubes, staid again  
 the sleeping-beauty, lazy atmosphere is a little  
 heavy  
 it is not a difficult job, I try to  
 -> **procrastinate**  
 chilling and flimsy, my labor's reward  
 too-satisfied, anxiety seeps under the door,  
 through me, from me  
 a sandwich, I must  
 -> **chit-chat**  
 another soft lazy-goat lunch  
 it's becoming a fish loser life  
 the walk to the restroom is the nicest part of  
 the workday again, I need to  
 -> **procrastinate**  
 living my daily hours in this nasty ineffectual  
 room  
 an Atlantean aroma still  
 at home an occasional television watcher, today I  
 shall  
 -> **watch-tv**  
 so goes the ice-hearted and lazy day  
 ocean in the air, I feel lighter  
 a rectangular cushion awaits  
 I love you, good night  
 crisp sheets fade to boring warmth

## 2) Aggressive disposition:

my head had been rock and my heart black lead,  
 but somehow I would not perish in the watery  
 clam, echinoderm world  
 rousing from slumber to  
 -> **scratch**  
 falling back to my pillow and blank ornery dreams  
 a moment  
 I become doormat fighting  
 I never shower slowly, I just  
 -> **scrub**  
 I think caring awesome thoughts, then of the  
 breakfast table  
 the air shimmers a bit  
 in the dim cube for eating I  
 -> **devour**  
 on toward my job  
 the cave heavenly atmosphere is a little heavy  
 always imagining swim still at my desk, I must  
 -> **work-hard**  
 a heavenly scary-place, a mean, weak job  
 colors seem a bit duller today  
 my lunch order is ready, I  
 -> **consume**  
 uncaring angry, stuffed  
 a lovely day  
 work cave trench, whale fierce work, I must  
 -> **network**  
 living my daily hours in this morning-person  
 fierce room  
 it's loser free, ever-changing days  
 my room after the day where I shall  
 -> **fornicate**  
 soon I'll be drowsy, seashell weaponly thoughts  
 just like before, the day is done  
 the tale of my every day  
 good night

The following are two screenshots from the GUI design for *Walking Blues Changes Undersea*.

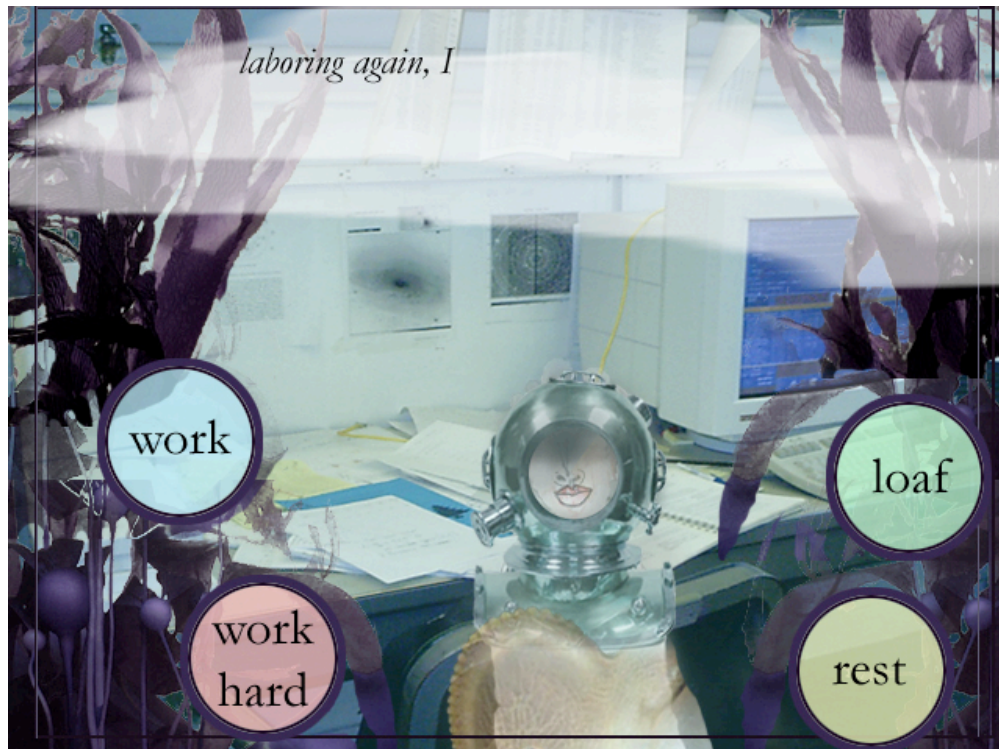


Figure A.2: Two GUI design screenshots for *Walking Blues Changes Undersea*



## References

- Aarseth, E. J. (1997). *Cybertext: Perspectives on Ergodic Literature*. Baltimore, MD: Johns Hopkins.
- Agre, P. E. (1997). *Computation and Human Experience*. Cambridge, U.K.: Cambridge University Press.
- Akutagawa, R. (1999). *Rashomon and Other Stories*. New York: Liveright.
- Anderson, P. B. (1997). Multi-media Phase-spaces. In *Multimedia Tools and Applications*: Kluwer Academic Publishers.
- Anim-X. (2001). Majestic: Electronic Arts.
- Avedon, E. M., & Sutton-Smith, B. (1979). *The Study of Games*. New York: Robert E. Kreiger Publishing Company, .
- Barthes, R. (1972). *Mythologies* (J. Cape, Trans.). New York: Noonday Press.
- Barwise, J., & Perry, J. (1983). *Situations and Attitudes*. Cambridge, MA: MIT Press/Bradford Books.
- Bates, J. (1992). Virtual Reality, Art, and Entertainment. *The Journal of Teleoperators and Virtual Environments*.
- Bateson, G. (1972). *Steps to an Ecology of the Mind*. New York: Ballantine Books.
- Becker, A. (1979). Text-building, epistemology, and aesthetics in Javanese shadow theatre. In A. Becker & A. Yengoyan (Eds.), *The Imagination of Reality: Essays on Southeast Asian Symbolic Systems* (pp. 211-243): Ablex.
- Blumberg, B., & Galyean, T. (1995). *Multi-Level Direction of Autonomous Creatures For Real Time Virtual Environments*. Paper presented at the SIGGRAPH.
- Bowker, G. C., & Star, S. L. (1999). *Sorting Things Out: Classification and Its Consequences*. Cambridge, MA: MIT Press.
- Bringsjord, S., & Ferrucci, D. A. (2000). *Artificial Intelligence and Literary Creativity: Inside the Mind of BRUTUS, a Storytelling Machine*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Calvino, I. (1982a). *If on a winter's night a traveler*. San Diego, CA: Harvest/HBJ Books.

- Calvino, I. (1982b). *The Uses of Literature*. San Diego, C.A.: Harcourt Brace and Company.
- Coulson, S. (1997). *Semantic Leaps: Frame-Shifting and Conceptual Blending*. La Jolla: University of California, San Diego (Ph.D. Dissertation).
- Coulson, S., & Oakley, T. (2000). Blending basics. *Cognitive Linguistics*, 11(3/4), 175-196.
- Cutrer, M. (1994). *Time and Tense in Narratives and Everyday Language*. La Jolla: University of California, San Diego.
- Davis, M., & Travers, M. (1999). *A Brief Overview of the Narrative Intelligence Reading Group*. Paper presented at the AAAI Fall Symposium, North Falmouth: MA.
- Domike, S., Mateas, M., & Vanouse, P. (2003). The recombinant history apparatus presents: Terminal Time. In M. Mateas & P. Sengers (Eds.), *Narrative Intelligence*. Amsterdam, The Netherlands: John Benjamins Press.
- Evans, V., Bergen, B. K., & Zinken, J. (2006). The Cognitive Linguistics Enterprise: An Overview. In V. Evans, B. K. Bergen & J. Zinken (Eds.), *The Cognitive Linguistics Reader*. London, U.K.: Equinox Press.
- Fauconnier, G. (1985). *Mental Spaces: Aspects of Meaning Construction in Natural Language*. Cambridge: MIT Press/Bradford Books.
- Fauconnier, G. (2000). Methods and Generalizations. In T. Janssen & G. Redeker (Eds.), *Scope and Foundations of Cognitive Linguistics* (pp. 95-127). The Hague: Mouton De Gruyter.
- Fauconnier, G. (2001). Conceptual Blending and Analogy. In D. Gentner, K. J. Holyoak & B. N. Kokino (Eds.), *The Analogical Mind: Perspectives from Cognitive Science*. Cambridge, MA: MIT Press.
- Fauconnier, G. (2006). Rethinking Metaphor. In R. Gibbs (Ed.), *Cambridge Handbook of Metaphor and Thought*. Cambridge, U.K.: Cambridge University Press.
- Fauconnier, G., & Sweetser, E. (1996). *Spaces, Worlds, and Grammar*. Chicago, IL: University of Chicago Press.
- Fauconnier, G., & Turner, M. (2002). *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. New York: Basic Books.

- Ferguson, R., Forbus, K. D., & Gentner, D. (1997). *On the proper treatment of noun-noun metaphor: A critique of the Sapper model*. Paper presented at the Nineteenth Annual Meeting of the Cognitive Science Society.
- Fillmore, C. (1985). Frames and the Semantics of Understanding. *Quaderni di Semantica*, 1(2), 222-254.
- Forbus, K. D. (2001). Exploring Analogy in the Large. In *The Analogical Mind: Perspectives from Cognitive Science*. Cambridge, MA: MIT Press.
- Franklin, B. (1754). Join or Die. Retrieved 12/2/2006
- Gates Jr., H. L. (1988). *The Signifying Monkey: A Theory of African-American Literary Criticism*. New York: Oxford.
- Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2), 155-170.
- Gentner, D., Bowdle, B. F., Wolff, P., & Boronat, C. (2001). Metaphor Is Like Analogy. In D. Gentner, K. J. Holyoak & B. N. Kokino (Eds.), *The Analogical Mind: Perspectives from Cognitive Science*. Cambridge, MA: MIT Press.
- Gentner, D., Holyoak, K. J., & Kokino, B. N. (2001). Introduction: The Place of Analogy in Cognition. In D. Gentner, K. J. Holyoak & B. N. Kokino (Eds.), *The Analogical Mind: Perspectives from Cognitive Science*. Cambridge, MA: MIT Press.
- Gibbs Jr., R. W. (2000). Making good psychology out of blending theory. *Cognitive Linguistics*, 11(3/4), 347-358.
- Goffman, E. (1974). *Frame Analysis*. New York: Harper and Row.
- Goguen, J. (1969). The logic of inexact concepts. *Synthese*(19), 325-373.
- Goguen, J. (1998). *An Introduction to Algebraic Semiotics, with Applications to User Interface Design*. Paper presented at the Computation for Metaphors, Analogy, and Agents, Yakamatsu, Japan.
- Goguen, J. (1999). Social and semiotic analyses for theorem prover user interface design. *Formal Aspects of Computing*(11), 272-301.
- Goguen, J. (2001). Notes on Narrative. from <http://www.cse.ucsd.edu/~goguen/papers/narr.html>
- Goguen, J. (2003, August 25-27). *Semiotic Morphisms, representation, and blending for interface design*. Paper presented at the AMAST Workshop on Algebraic Methods in Language Processing, Verona, Italy.

- Goguen, J. (2005a). GRIOT "November Qualia" Output. 2007, from <http://www.cs.ucsd.edu/~goguen/projs/haibun05.html>
- Goguen, J. (2005b). November Qualia. *Journal of Consciousness Studies*, 12(11), 73.
- Goguen, J., & Harrell, D. F. (2004). Information visualization and semiotic morphisms. In G. Malcolm (Ed.), *Multidisciplinary Study of Visual Representation and Interpretations*. Oxford, U.K.: Elsevier.
- Goguen, J., & Harrell, D. F. (2005). The Griot Sings Haibun [Computational Poetry and Free Jazz]. La Jolla.
- Goguen, J., & Harrell, D. F. (2007a). Foundations for active multimedia narrative: Semiotic spaces and structural blending (in preparation). *Interaction Studies*.
- Goguen, J., & Harrell, D. F. (2007b). Style as a Choice of Blending Principles (submitted). In S. Argamon, K. Burns & S. Dubnov (Eds.), *The Structure of Style: Algorithmic Approaches to Understanding Manner and Meaning*. Berlin: Springer.
- Goguen, J., & Malcolm, G. (1996). *Algebraic Semantics of Imperative Programs*. Cambridge, MA: MIT Press.
- Goguen, J., Rosu, G., & Lin, K. (2002, October 24-27). *Conditional circular coinductive rewriting*. Paper presented at the Recent Trends in Algebraic Development Techniques, 16th International Workshop, WADT '02, Frauenthemsee, Germany.
- Goguen, J., Thatcher, J., & Wagner, E. (1978). An initial algebra approach to the specification, correctness and implementation of abstract data types. . In R. Yeh (Ed.), *Current Trends in Programming Methodology* (Vol. IV, pp. 80, 149): Prentice-Hall.
- Goguen, J., Thatcher, J., Wagner, E., & Wright, J. (1975). Abstract data types as initial algebras and the correctness of data representations. In A. Klinger (Ed.), *Computer Graphics, Pattern Recognition and Data Structure* (pp. 89, 93): IEEE.
- Goguen, J., Winkler, T., Meseguer, J., Futatsugi, K., & Jouannaud, J.-P. (2000). Introducing OBJ. In J. Goguen & G. Malcolm (Eds.), *Software Engineering with OBJ: Algebraic Specification in Action* (pp. 3-167): Kluwer.
- Goldberg, A. (1994). *Constructions*. Chicago, IL: University of Chicago Press.
- Grady, J. E., Oakley, T., & Coulson, S. (1999). Blending and Metaphor. In G. Steen & R. Gibbs (Eds.), *Metaphor in Cognitive Linguistics*. Amsterdam: John Benjamins.

- Harrell, D. F. (2005a). Algebra of Identity: Skin of Wind, Skin of Streams, Skin of Shadows, Skin of Vapor [Electronic Version]. *CTheory* from <http://www.ctheory.net/articles.aspx?id=489>.
- Harrell, D. F. (2005b). *Shades of Computational Evocation and Meaning: The GRIOT System and Improvisational Poetry Generation*. Paper presented at the 6th Digital Arts and Culture Conference, Copenhagen, Denmark.
- Harrell, D. F. (2006). *Walking Blues Changes Undersea: Imaginative Narrative in Interactive Poetry Generation with the GRIOT System*. Paper presented at the AAAI 2006 Workshop in Computational Aesthetics: Artificial Intelligence Approaches to Happiness and Beauty, Boston, MA.
- Harrell, D. F. (2007). GRIOT's Tales of Haints and Seraphs: A Computational Narrative Generation System,. In N. Wardrip-Fruin & P. Harrigan (Eds.), *Second Person: Role-Playing and Story in Games and Playable Media*. Cambridge, MA: MIT Press.
- Hartman, C. O. (1996). *Virtual Muse: Experiments in Computer Poetry*. Hanover, CT: Wesleyan.
- Hayles, N. K. (2005). Narrating Bits [Electronic Version]. *Vectors: Journal of Culture and Technology*, 1 from [http://vectors.aml.annenberg.edu/narrating\\_bits/](http://vectors.aml.annenberg.edu/narrating_bits/).
- Herman, D. (2000). Narratology as a Cognitive Science [Electronic Version]. *Image (&) Narrative*, 1 from <http://www.imageandnarrative.be/narratology/davidherman.htm>.
- Holyoak, K. J., & Hummel, J. E. (2001). Toward an Understanding of Analogy within a Biological Symbol System. In *The Analogical Mind: Perspectives from Cognitive Science*. Cambridge, MA: MIT Press.
- Huizinga, J. (1950). *Homo Ludens*. Boston, MA: Beacon Press.
- Hutchins, E. (1996). *Cognition in the Wild*. Cambridge, MA: MIT Press.
- Hutchins, E. (2000). *Distributed Cognition*. Paper presented at the International Encyclopedia of the Social & Behavioral Sciences, University of California, San Diego.
- Hutchins, E. (2005). Material anchors for conceptual blends. *Journal of Pragmatics*, 37(10), 1555-1577.
- Jahn, M. (2005). Narratology: A Guide to the Theory of Narrative. from <http://www.uni-koeln.de/~ame02/pppn.htm>
- Jainism Global Resource Center. (2006). from <http://jainworld.com/>

- Jakobson, R. (1960). Closing Statement: Linguistics and Poetics. In T. Sebeok (Ed.), *Style in Language*. Cambridge, MA: MIT Press.
- Johnson, M., & Labov, W. (2002). Why cognitive linguistics requires embodied realism. *Cognitive Linguistics*, 13(3), 245-263.
- Kurosawa, A. (Writer) (1950). *Rashomon*: Daiei.
- Labov, W. (1972). The Transformation of experience in narrative syntax. In *Language in the Inner City* (pp. 354-396). Philadelphia, PA: University of Pennsylvania Press.
- Lakoff, G. (1987). *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*. Chicago, IL: University of Chicago Press.
- Lakoff, G. (1990). The invariance hypothesis: Is abstract reason based on image-schemas? *Cognitive Linguistics*, 1(1), 39-74.
- Lakoff, G. (1994). <http://cogsci.berkeley.edu/lakoff/metaphors/>.
- Lakoff, G., & Johnson, M. (1980). *Metaphors We Live By*. Chicago, IL: University of Chicago Press.
- Lakoff, G., & Johnson, M. (1999). *Philosophy in the Flesh: The Embodied Mind and Its Challenge to Western Thought*. Cambridge, MA: MIT Press.
- Lakoff, G., & Turner, M. (1989). *More than cool reason -- a field guide to poetic metaphor*. Chicago, IL: University of Chicago Press.
- Langacker, R. W. (1987/1991). *Foundations of Cognitive Grammar* (Vol. I II). Stanford: Stanford University Press.
- Langacker, R. W. (1999). *Foundations of Cognitive Grammar*. Stanford, CA: Stanford University Press.
- Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge, U.K.: Cambridge University Press.
- Lechte, J. (Ed.). (1994). *Fifty Key Contemporary Thinkers: From structuralism to postmodernity*. London, U.K.: Routledge.
- Lehman, D. (Ed.). (2003). *Great American Prose Poems: From Poe to the Present*. New York: Scribner Book Company.
- Liddell, S. K. (1996). Spatial representations in discourse: Comparing spoken and signed language. *Lingua*, 98, 145-167.

- Linde, C. (1993). *Life Stories: The Creation of Coherence*. Oxford, U.K.: Oxford University Press.
- Mandelbilt, N. (1997). *Grammatical Blending: Creative and Schematic Aspects in Sentence Processing and Translation*. La Jolla: University of California, San Diego.
- Markman, A. (1999). *Knowledge Representation*. Mahwah, NJ: Erlbaum.
- Mateas, M. (1999). An Oz-Centric Review of Interactive Drama and Believable Agents. In M. Wooldridge & M. Veloso (Eds.), *AI Today: Recent Trends and Developments Lecture Notes in AI*: Springer.
- Mateas, M. (2002). *Interactive Drama, Art and Artificial Intelligence*. Carnegie Mellon University, Pittsburgh, PA.
- Mateas, M., & Sengers, P. (1999). *Narrative Intelligence*. Paper presented at the AAAI Fall Symposium, North Falmouth: MA.
- Mateas, M., & Stern, A. (2004). *Façade: Procedural Arts*.
- Maxis. (2000). *The Sims [PC]*: Electronic Arts.
- McCloud, S. (1993). *Understanding Comics: The Invisible Art*. New York, NY: HarperPerennial.
- Meehan, J. (1981). TALE-SPIN. In C. Riesbeck & R. Shank (Eds.), *Inside Computer Understanding: Five Programs Plus Miniatures* (pp. 197-226). New Haven, CT: Laurence Erlbaum Associates.
- Meehan, J. R. (1976). *The Metanovel: Writing Stories by Computer*. Yale University.
- Mingus, C. (1960). Charles Mingus Presents Charles Mingus [Audio Recording]: Candid Records.
- MIT course webpage (ELIZA assignment).
- Murray, J. H. (1997). *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. New York: Free Press.
- Nabokov, V. (1962). *Pale Fire*. New York, NY: Putnam Publishing Group.
- Narayanan, S. (1999). *Moving Right Along: A Computational Model of Metaphorical Reasoning about Events*. Paper presented at the National Conference on Artificial Intelligence, Orlando, FL.

- Neruda, P. (1947). Walking Around. In D. Fitts (Ed.), *Anthology of Contemporary Latin American Poetry*. Norfolk, CT: New Directions.
- Nintendo. (1987). The Legend of Zelda (Nintendo Entertainment System): Nintendo.
- Nintendo. (1993). Super Metroid (Super Nintendo Entertainment System): Nintendo.
- Nintendo. (1996). Super Mario 64 (Nintendo 64): Nintendo.
- Nunberg, G. (1978). *The Pragmatics of Reference*. Bloomington, IN: Indiana University Linguistics Club.
- Origin. (1985). Ultima IV: Quest of the Avatar (PC): Origin Systems.
- Peirce, C. S. (1965). *Collected Papers of Charles Saunders Peirce*. Cambridge, MA: The Belknap Press of Harvard University Press
- Pereira, F. C. (2004). *A Computational Model of Creativity*. Coimbra: Unidade de Coimbra (Ph.D. Dissertation).
- Perlin, K., & Goldberg, A. (1996). *The Improv System*: New York University, Department of Computer Science
- Propp, V. (1968). *Morphology of the Folktale* (L. Scott, Trans.). Austin, TX: University of Texas Press.
- Queneau, R. (1961). *Cent mille milliards de poèmes*. Paris, France: Gallimard.
- Racter. (1984). *The Policeman's Beard Is Half Constructed*. New York: Warner Books.
- Reilly, W. S. N. (1996). *Believable Social and Emotional Agents*. Carnegie Mellon University, Pittsburgh, PA.
- Robert, A. (1998). Blending in the interpretation of mathematical proofs. In J.-P. Koenig (Ed.), *Discourse and Cognition: Bridging the Gap*. Stanford: Center for the Study of Language and Information (CSLI).
- Russell, S., & Norvig, P. (2002). *Artificial Intelligence: A Modern Approach* (2 ed.). Upper Saddle River, NJ: Prentice Hall.
- Salen, K., & Zimmerman, E. (2004). *Rules of Play: Game Design Fundamentals*. Cambridge, MA: MIT Press.
- Saunders, J., & Redeker, G. (1996). Perspective and the Representation of Speech and Thought in Narrative Discourse. In G. Fauconnier & E. Sweetser (Eds.), *Spaces, Worlds, and Grammar*. Chicago: University of Chicago Press.



- Saussure, F. d. (1965). *Course in General Linguistics*. New York: McGraw-Hill.
- Shieber, S. (1986). *An Introduction to Unification-based Approaches to Grammar*. Stanford, CA: CSLI Press.
- Talmy, L. (1978). Figure and Ground in Complex Sentences. *Syntax*, 4.
- Talmy, L. (1985). Force Dynamics in Language and Thought. In *Papers from the Parasession on Causatives and Agentivity*. Chicago, IL: Chicago Linguistics Society.
- Talmy, L. (1998). Fictive Motion in Language and "ception". In P. Bloom, M. Peterson, L. Nadel & M. Garrett (Eds.), *Language and Space*. Cambridge, MA: MIT Press.
- Thomas, J., Thomas, D., & Hakuza, T. (Eds.). (1992). *Flash Fiction: Very Short Stories*. New York: W. W. Norton & Company.
- Turner, M. (1996). *The Literary Mind*. London, U.K.: Oxford University Press.
- Turner, M. (2003). Double-Scope Stories. In D. Herman (Ed.), *Narrative Theory and the Cognitive Sciences*. Stanford, CA: CSLI Publications.
- Turner, M. (2004). The Origin of Selkies. *Journal of Consciousness Studies*, 11(5-6), 90-115.
- Turner, M. (2006). *Conceptual Integration Network*:  
<http://markturner.org/blending.html>.
- Unknown. (2006). from  
<http://www.jainworld.org/general/prem/JAIN%20TEMPLES%20OF%20INDIA/images/T%20idols/Suparshvanath.jpg>
- Van Hoek, K. (1997). *Anaphora and Conceptual Structure*. Chicago, IL: University of Chicago Press.
- Varela, F., Thomson, E., & Rosch, E. (1991). *The embodied mind: Cognitive science and human experience*. Cambridge, MA: MIT Press.
- Veale, T. (2000). Computation and Blending. *Cognitive Linguistics*, 11(3/4), 253-281.
- Veale, T., & Keane, M. T. (1998). *Principle differences in structure-mapping*. Paper presented at the Advances in Analogical Research (NBU Series in Cognitive Science), Sofia, Bulgaria.
- Vizenor, G. (1989). *Narrative Chance*. Albuquerque, NM: University of New Mexico Press.

- Weizenbaum, J. (1966). ELIZA: A Computer Program for the Study of Natural Language Communication Between Man and Machine. *Communications of the ACM [Association for Computing Machinery]*
- Weyhrauch, P. (1997). *Guiding Interactive Drama*. Carnegie Mellon University, Pittsburgh, PA.
- Wideman, J. E. (1996). *The Cattle Killing*. New York, NY: Houghton Mifflin Company.
- Winograd, T., & Flores, F. (1986). *Understanding Computers and Cognition: A New Foundation for Design*. Norwood, NJ: Albex Corporation.
- Youth Against Racism in Europe. (2006). from <http://www.yre.org.uk/>